18

Advances in Simulation of Planetary Wheeled Mobile Robots

Liang Ding, Haibo Gao, Zongquan Deng and Weihua Li
State Key Laboratory of Robotics and System,
Harbin Institute of Technology
P. R. China

1. Introduction

Ever since the Sojourner rover of the United States landed on Mars in 1997 (Jet Propulsion Laboratory [JPL], a), there has been an upsurge in the exploration of planets using wheeled mobile robots (WMRs or rovers). The twin rovers that followed, Spirit and Opportunity, have endured many years of activity on Mars and have made many significant discoveries (JPL, b). Several other new missions are in progress to explore Mars (Volpe, 2005; Van et al., 2008) and the Moon (Neal, 2009) using planetary rovers that are expected to traverse more challenging terrain with scientific objectives such as searching for evidence of life and investigating the origin of the solar system.

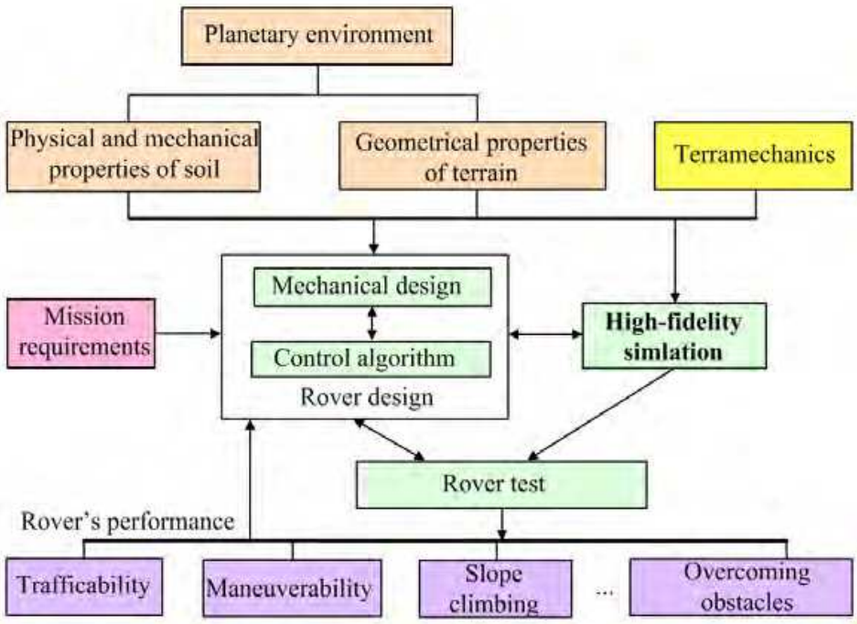
The present planet exploration rovers are advanced WMRs that show excellent performance and have integrated the cutting-edge technologies of many fields, and overcoming new frontier issues that are specific to planetary rovers has promoted the development of terrestrial WMRs.

Simulation technology plays an important role in both the research and development (R&D) and exploration phases of planetary WMRs (Ding et al., 2008). During the R&D phase of a WMR, a simulation system can be used for mechanical design (e.g., performance analysis and optimization), control algorithm verification, and performance testing, and during the exploration phase, the simulation system can be used to support three-dimensional (3D) predictive displays for successive teleoperation (such as in the case of a lunar rover) or to validate command sequences for supervised teleoperation (such as in the case of a Mars rover). As compared to conventional simulation systems used for WMRs, the simulation system for planetary rovers is characterized by high fidelity, high speed, and comprehensiveness.

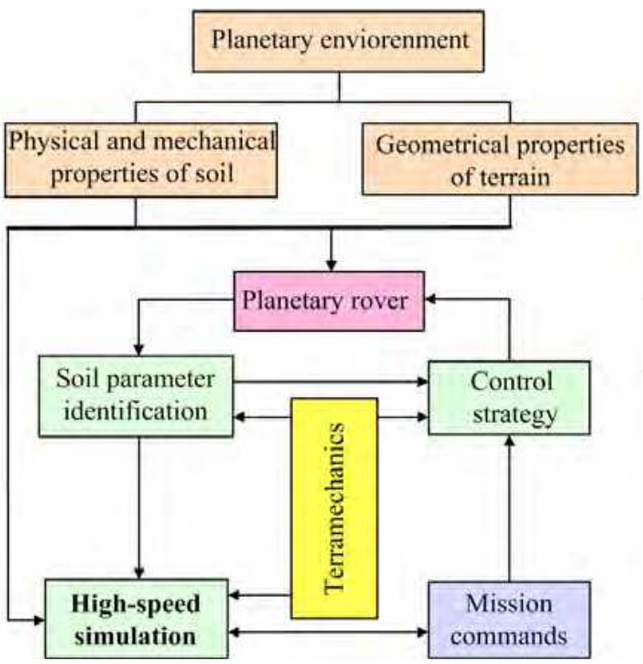
The recent development and cutting-edge technologies of simulation systems for planetary rovers are summarized in this article to extend their application to conventional WMRs. The significance of simulation for planetary WMRs is discussed in Section 2. An overview of the simulation technology for planetary WMRs is presented in Section 3. Section 4 introduces key theories (models of terramechanics, dynamics, and terrain geometry) for developing a simulation system for planetary rovers. In Section 5, the research results for the simulation of planetary rovers at the State Key Laboratory of Robotics and System (SKLRS) of China, including simulation methods, systems, and verification results, are presented to provide examples of different simulation methods (based on the commercial dynamics simulation software, general simulation software, and real-time simulation software) in detail for different applications.

2. Importance of simulation for planetary rovers

Virtual simulation can guarantee successful of WMRs for exploring the planets as it plays important roles in both the R&D phase and the exploration phase of the rovers, which are shown in Fig. 1. It is of great importance to planetary rovers in three aspects.



(a) R&D phase



(b) Exploration phase

Fig. 1. Roles of simulation for planetary rovers

2.1 Platform for rover design, evaluation, testing, and control

During the R&D phase of a rover, virtual simulation can be used for configuration design, evaluation and optimization, mobility performance analysis, and control strategy research before a rover is manufactured. For example, before the Mars Pathfinder probe carrying the Sojourner rover was launched, with the help of virtual simulation technology, researchers at JPL predicted that it would turn over while landing because of the interaction between the breaking rocket and Martian wind. The technical program was revised to solve the problem and ensured successful soft landing on Mars (Chen et al., 2005). To support the study on Mars rovers, the United States developed a comprehensive simulation system ROAMS. Europe is committed to the development of the simulation tools RCET, RPET, and RCAST for the evaluation and optimization of Mars rovers. Ye, the general director and general designer of the Chinese Chang'e-1 satellite, advocated numerical simulation technology to study the mobility performance of lunar rovers and simulate floating lunar dust (Ye & Xiao, 2006). Researchers from the Beijing Institute of Control Engineering discussed the technical scheme for researching lunar rovers and pointed out that a virtual simulation system having the abilities of modeling, dynamics simulation, and control simulation should be developed to test the configuration and dynamics parameters of a rover, as well as the control algorithms (Liang et al., 2005).

2.2 Supporting 3D predictive display for successive teleoperation

Exploration activities and the moving path of a rover are subject to the great uncertainty in our knowledge of a planet's surface environment. Therefore, even the most advanced rovers at present, Spirit and Opportunity, do not employ a completely autonomous control strategy but are teleoperated by scientists on the Earth. The time delay in transmission between the Earth and Moon is about several seconds owing to the long distance and limited bandwidth, which makes continuous closed-loop control of a rover unstable. This problem can be solved effectively by successive remote operation using a 3D predictive display. This has been studied by researchers at Japan's Meiji University, China's Jilin University, and elsewhere (Lei et al., 2004; Kuroda et al., 2003; Kunii et al., 2001). Ground computers construct a virtual simulation environment according to the rover status and imagery information of the terrain, through which the rover will move after a time delay. Operators control the rover in a virtual environment, and the same commands are sent to the real rover after error compensation. The real rover repeats the motion of the virtual rover after a while. If the timing of the sent commands is regulated well, the rover can move successively without stopping.

2.3 Validating command sequence for supervised teleoperation

As the time delay in transmission from the Earth to Mars is tens of minutes, supervised teleoperation is adopted instead of nonfeasible continuous teleoperation to control Mars rovers. Researchers at JPL learned an important lesson from the Mars Pathfinder mission that a fast, accurate, and powerful tool for driving the rover is necessary (Maxwell et al., 2005). Thus, the rover sequencing and visualization program (RSVP) has been developed to drive rovers (Wright et al., 2006). It is a suite consisting of two main components: RoSE (rover sequence editor) and Hyperdrive. Hyperdrive is an immersive 3D simulation of the rover, and its environment enables operators to construct detailed rover motions and verify their safety. It uses the state information of a rover to analyze and review the current state, identify any anomalous issues, review previously commanded activities, and verify that the

commanded activities have been completed. Scientists define science activities and plan a rover's path with 3D terrain models built according to images obtained by stereo cameras. Then, command sequences are constructed. Sequence validation is done by simulation, and the verified sequence is transmitted to the real rover for the exploration of a solar day. RSVP is used to generate the final command sequences for the mission and plan and validate all mobility and manipulation activities.

The supervised teleoperation method has its disadvantages. For instance, the rover must stop to wait for command sequences and it should be very intelligent. However, it also has obvious advantages. The science activities are defined systematically, and the operators can easily control the rover. Therefore, the two teleoperation modes are used according to the terrain and mission. No matter which is used, virtual simulation is indispensable.

3. Overview of current simulation technology for planetary WMRs

3.1 General simulation tools used in research of planetary WMRs

Various general simulation tools are used by researchers to investigate the kinematics, dynamics, control, and visualization of rovers. They are summarized in Table 1 (Ding et al., 2008). General simulation tools are used by researchers to realize different research purposes, but they do not meet the requirements of rover-based exploration missions well because of their limitations.

3.2 Virtual simulation system developed for planetary WMRs

To support the R&D of WMRs for related planetary exploration missions, customized simulation tools or tools with different simulation functions have been developed.

3.2.1 ROAMS, WITS, and RSVP developed by JPL, United States

The Mars Technology Program, in conjunction with the Mars Science Laboratory Mission, has funded three complementary infrastructure elements: ROAMS, the web interface for tele-science (WITS), and CLARAty (Volpe, 2003).

ROAMS is a comprehensive physics-based simulator for planetary-exploration roving vehicles. It includes a mechanical subsystem, electrical subsystem, internal and external sensors, on-board resources, on-board control software, a terrain environment, and terrain/vehicle interactions. ROAMS can be used for stand-alone simulation, closed-loop simulation with on-board software, and operator-in-the-loop simulation (Yen et al., 1999). It includes rover kinematics, dynamics, navigation, locomotion, and visualization. ROAMS provides simulation services for off-line analysis, and acts as a virtual rover platform for CLARAty control software (Fig. 2), which is reusable rover software architecture being developed in collaboration by JPL, NASA Ames, Carnegie Mellon University and other institutions. A goal of the CLARAty development is to provide open architecture for component algorithm developers to develop and integrate their capabilities into.

RSVP, which is capable of kinematics simulation, was developed for rover teleoperation as mentioned above. WITS provides collaborative downlink data visualization and uplink activity planning for multiple Mars lander and rover missions. For the 2003 Mars Exploration Rover mission, WITS was the primary science operations tool for downlink data visualization and uplink science activity planning. It can build scripted sequences and execute them on a 3D simulated rover, which can be controlled interactively (Backes et al., 2004).

Simulation contents	Explanation	Purpose	Examples of simulation tools
Forward kinematics	Calculates the position and yaw angle of a rover according to feedback information of pitch and roll angles and the positions of motors and joints	Determines the position and orientation of a rover (dead reckoning method)	Matlab, VC++
Inverse kinematics	Calculates the altitude, pitch, and rock angles of the rover, as well as the positions of motors and joints given the yaw angle, horizontal position of a rover, and the path on known terrain	Analyzes the stability and traversability of a rover while following a certain path	
Dynamics	Used in research on wheel–soil interaction mechanics, multi-body dynamics, and methods for solving differential equations with high efficiency	Analyzes the dynamics performance such as vibratility and the ability to overcome obstacles; the basis for designing control strategy	ADAMS, DADS, Vortex
Control strategy	Used in research on control strategies of a rover's locomotion, path planning, path following, and intelligent navigation	Develops a program for rover control	Matlab
Visualization	Builds a visual virtual simulation environment according to topography and surface features of the planet and the rover's configuration	Provides a visualization platform for virtual simulation	Vega, 3D Max, MultiGen Creator

Table 1. Rover simulation and tools

3.2.2 RCET, RPET, and RCAST developed in Europe

Mars rover chassis evaluation tools (RCET) have been developed in Europe to support the design of planetary rovers (Michaud et al., 2006). RCET, which was developed jointly by Contraves Space, Ecole Polytechnique Fédérale de Lausanne, the German Aerospace Center (DLR) and others, accurately predicts the rover performances of the locomotion subsystem. It consists of a two-dimensional (2D) rover simulator that uses a tractive prediction module to compute the wheel/ground interaction. 3D simulation can also be performed with the help of RoverGen.

The rover performance evaluation tool (RPET) consists of the rover mobility performance evaluation tool (RMPET) and mobility synthesis (MobSyn). RMPET computes mobility

performance parameters such as drawbar pull, motion resistances, soil thrust, slippage, and sinkage for a mobility system selected by the user for the evaluation of particular terrain (Patel et al., 2004).

RCAST was developed to characterize and optimize the ExoMars rover mobility in support of the evaluation of locomotion subsystem designs before RCET was available (Fig. 3). It uses the AESCO soft soil tire model (AS²TM) software package for terramechanics (Bauer et al., 2005).

At present, DLR is responsible for modeling, simulating and testing the entire mobility behavior of the rover within the ExoMars mission preparation phases. The commercial software tool Simpack, which includes contact modeling based on polygonal contact modeling, is used for simulation (Schäfer et al., 2010). The terramechanics for wheel–soil contact dynamics modeling and simulation and its experimental validation for the ExoMars rover has been introduced (Schäfer et al., 2010).

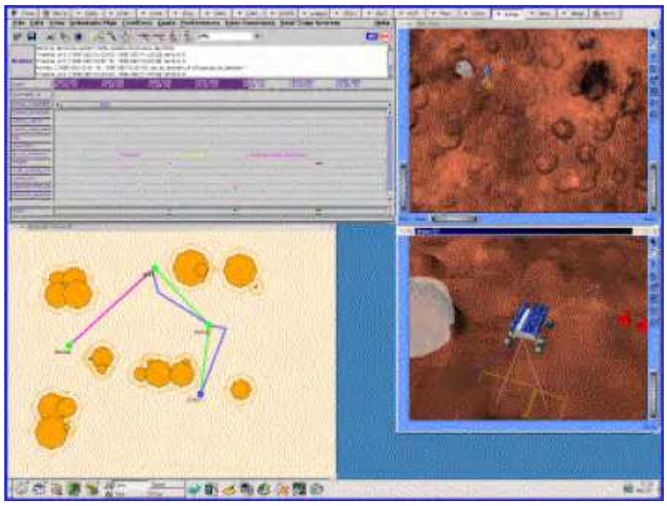


Fig. 2. Simulation with ROAMS and CLARAty

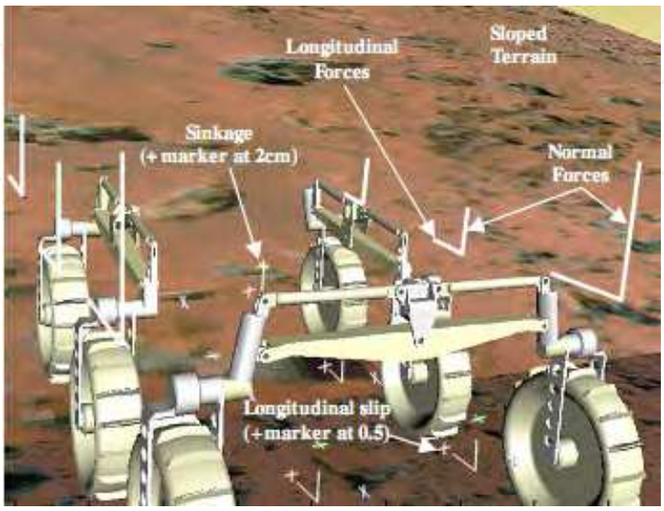


Fig. 3. ExoMars simulation with RCAST

3.2.3 SpaceDyn developed by Tohoku University, Japan

Space Robotics Laboratory (SRL) of Tohoku University developed the dynamics simulation toolbox SpaceDyn with Matlab. SpaceDyn has been successfully used for ETS-VII robot-arm simulation and touchdown of the Hayabusa spacecraft on Itokawa. SpaceDyn is also used to simulate the motion dynamics of a rover with a slip-based traction model (Yoshida & Hamano, 2002) and steering characteristics of a rover on loose soil based on terramechanics (Ishigami & Yoshida, 2005). A path-planning method taking into account wheel-slip dynamics of a planetary exploration rover was developed to generate several paths for a rover moving over rough terrain. Dynamics simulation was carried out by controlling the rover to follow the candidate paths for evaluation, as shown in Fig. 4 (Ishigami et al., 2007).

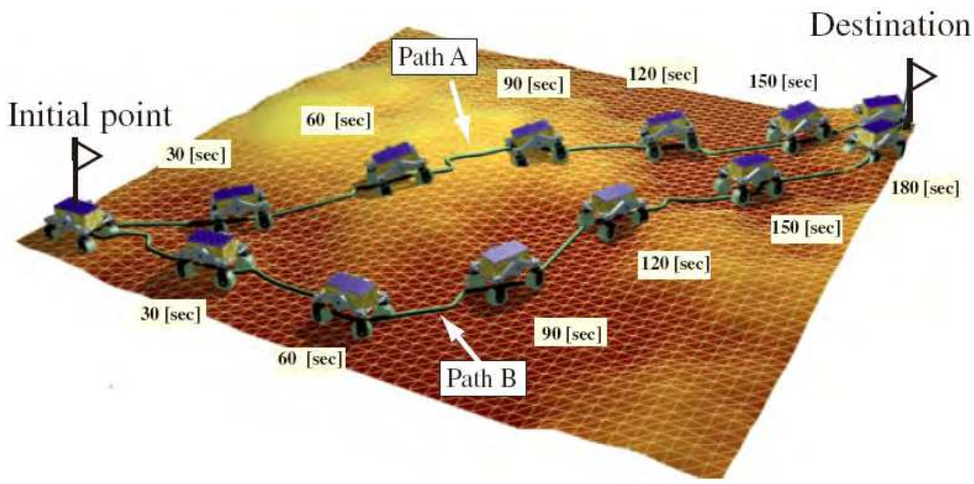


Fig. 4. Simulation of path planning and evaluation

3.2.4 Simulation platform for China's lunar rover

Researchers at the State Key Laboratory of Robotics and System (SKLRS) of the Harbin Institute of Technology developed a simulation platform entitled rover simulation based on terramechanics and dynamics (RoSTDyn) (Li et al., 2012) under a contract with the Chinese Academy of Space Technology to evaluate the performance of the Chang'e lunar rover and assist with its teleoperation.

Yang et al. of Shanghai Jiaotong University presented the framework and key technologies of a virtual simulation environment for a lunar rover (Yang et al., 2008). A fractional Brownian motion technique and statistical properties were used to generate the lunar surface. The multi-body dynamics and complex interactions with soft ground were integrated in the environment.

Researchers at Tsinghua University investigated a test and simulation platform for lunar rovers (Luo & Sun, 2002). The platform provides modules for creating the topography of the terrain and an environmental components editor. The virtual lunar environment can be constructed with the terrain modules built in advance. COM technology was used to support distributed simulation.

4. Key theories for development of simulation system for planetary WMRs

Planetary exploration missions require comprehensive simulation systems that have the abilities of modeling, kinematics, dynamics, control, and visualization, and have the characteristics of high speed and high fidelity.

Figure 5 shows the architecture of a comprehensive virtual simulation system for the high-fidelity/high-speed simulation of rovers. It comprises the RoSTDyn and interactive virtual planetary rover environment (IVPRE) systems. RoSTDyn is a comprehensive simulation system similar to ROAMS. Control commands of three levels received from itself, IVPRE, or other control software can be accepted by RoSTDyn; i.e. the goals, paths, and motor's position. Users can control the virtual rover interactively with IVPRE, which constructs a virtual lunar environment with terrain components or images from the real rover. Digital evaluation model (DEM) terrain data are then generated for RoSTDyn. It can also calculate the mechanics parameters of the soil for RoSTDyn. This system can be further developed for successive lunar rover teleoperation based on 3D predictive display. Key technologies of the simulation system include generalized dynamics modeling, wheel–soil interaction terramechanics models, and deformable rough-terrain geometry modeling.

4.1 Generalized recursive dynamics modeling

4.1.1 Recursive kinematics and Jacobian matrices

If $\mathbf{a}=[a_1 \quad a_2 \quad a_3]^{\mathrm{T}}$, $\mathbf{b}=[b_1 \quad b_2 \quad b_3]^{\mathrm{T}}$, and we define $\tilde{\mathbf{a}}=\begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$, then $\mathbf{a}\times\mathbf{b}=\tilde{\mathbf{a}}\mathbf{b}$ and $\mathbf{b}\times\mathbf{a}=-\tilde{\mathbf{a}}\mathbf{b}=\tilde{\mathbf{a}}^{T}\mathbf{b}$.

Let $\mathbf{q}=[q_1 \quad q_2 \quad \cdots \quad q_{n_v}]^{\mathrm{T}}$ denote joint variables, where n_v is the number of joints. The WMRs are articulated multi-body systems with a moving base and n_w end-points (wheels). Let $\mathbf{q}_s=[q_l \quad q_m \quad q_n \quad \cdots \quad q_s]^{\mathrm{T}}$ denote a branch from the rover body to a wheel and n_s denote the number of elements in $\mathbf{q}_s$. Replace the joint number l, m, n, …, s of the branch with 1, 2, 3,…, n_s, as shown in Fig. 1, which also shows the inertial coordinate $\{\Sigma_l\}$ and the coordinates $\{\Sigma_i\}$ related to link i ($i = l, m, n, \ldots, s$) and related vectors, where $\boldsymbol{p}_i$ is the position vector of joint i; $\boldsymbol{r}_i$ is the position vector of the centroid of link i; $\boldsymbol{c}_{ij}$ is the link vector from link i to joint j; $\boldsymbol{l}_{ij} = \boldsymbol{p}_j - \boldsymbol{p}_i$ is the link vector from joint i to joint j; and $\boldsymbol{l}_{ie}$ is the vector from joint i to end-point e.

The position vector of end-point $\boldsymbol{p}_e$ is

$$\mathbf{p}_e=\mathbf{r}_0+\mathbf{c}_{01}+\sum_{i=1}^{n_s-1}\mathbf{l}_{i(i+1)}+\mathbf{l}_{n_s e} \tag{1}$$

The derivative of Eq. (1) is

$$\begin{aligned}\mathbf{v}_e&=\mathbf{v}_0+\boldsymbol{\omega}_0\times(\mathbf{p}_e-\mathbf{r}_0)+\sum_{i=1}^{n_s}\mathbf{A}_i\,{}^{i}\mathbf{Z}_i\times(\mathbf{p}_e-\mathbf{p}_i)\dot{q}_i \\ &=\begin{bmatrix}\mathbf{J}_{BTe} & \mathbf{J}_{MTe}\end{bmatrix}\begin{bmatrix}\begin{pmatrix}\mathbf{v}_0^{\mathrm{T}} & \boldsymbol{\omega}_0^{\mathrm{T}}\end{pmatrix} & \dot{\mathbf{q}}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\end{aligned} \tag{2}$$

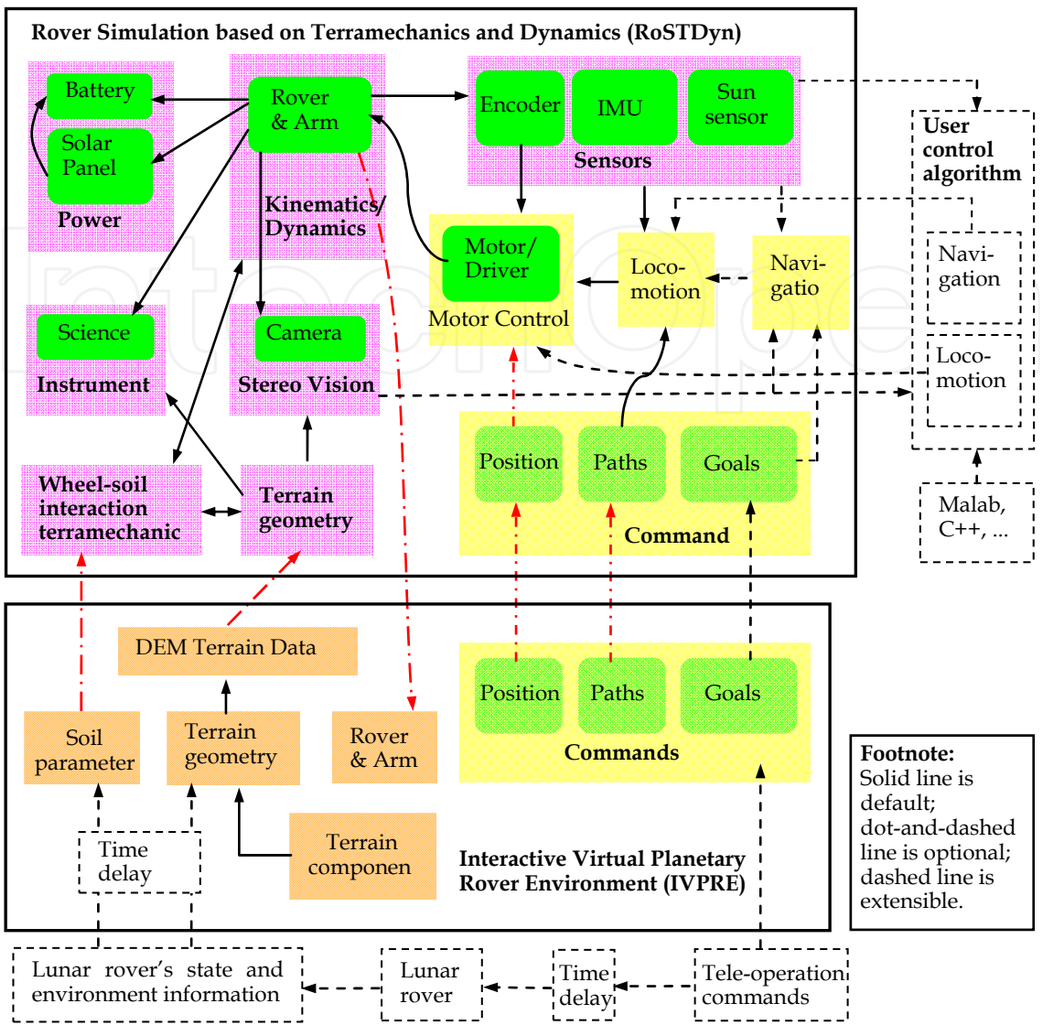


Fig. 5. Architecture of planetary rover's comprehensive simulation system

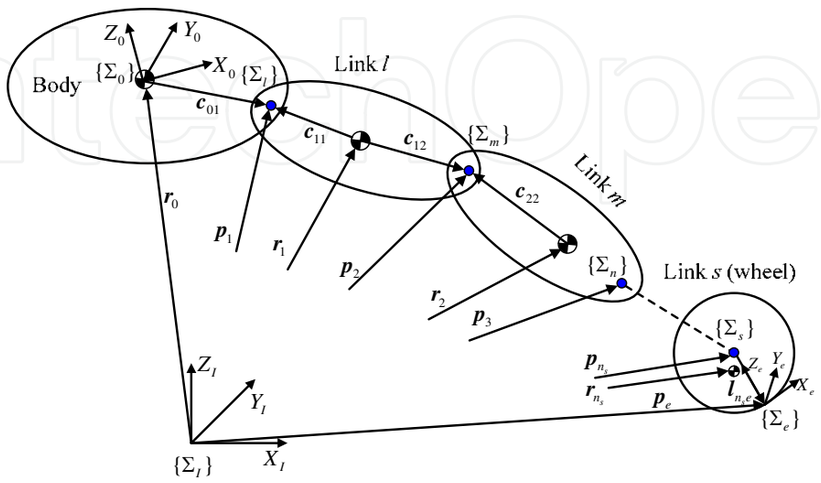


Fig. 6. Coordinates and vectors from rover body to wheel

where $\mathbf{J}_{MTe} = [\mathbf{L}_{s1}\mathbf{A}_1\,{}^1\mathbf{Z}_1 \times \mathbf{P}_{1e} \quad \mathbf{L}_{s2}\mathbf{A}_2\,{}^2\mathbf{Z}_2 \times \mathbf{P}_{2e} \quad \cdots \quad \mathbf{L}_{sn_v}\mathbf{A}_{n_v}\,{}^{n_r}\mathbf{Z}_{n_v} \times \mathbf{P}_{n_v e}]$ is a 3 × n_v matrix, $\mathbf{A}_i = {}^I\mathbf{A}_i$ is the transformation matrix from $\{\Sigma_i\}$ to $\{\Sigma_I\}$, ${}^i\mathbf{Z}_i = [0 \quad 0 \quad 1]^{\mathrm{T}}$ because the z axis is set to coincide with the joint displacement axis, $\mathbf{L}_{ij}$ is an element of matrix $\mathbf{L}_{n_v \times n_v}$ that indicates whether link j is on the access road from link 0 to link i ($\mathbf{L}_{ij}$ =1) or not ($\mathbf{L}_{ij}$ = 0), and $\mathbf{P}_{ie}$ is the vector from the origin of $\{\Sigma_i\}$ to the end-point. $\mathbf{J}_{BTe} = [\mathbf{E} \quad \tilde{\mathbf{P}}_{er0}^T]$ is a 3 × 6 matrix, where $\mathbf{P}_{er0} = \mathbf{p}_e - \mathbf{r}_0$.

The angular velocity of the end-point is

$$\boldsymbol{\omega}_e = \boldsymbol{\omega}_0 + \sum_{i=1}^{n_v} \mathbf{A}_i\,{}^i\mathbf{Z}_i \dot{q}_i = [\mathbf{J}_{BRe} \quad \mathbf{J}_{MRe}] \left[\begin{pmatrix} \mathbf{v}_0^{\mathrm{T}} & \boldsymbol{\omega}_0^{\mathrm{T}} \end{pmatrix} \quad \dot{\mathbf{q}}^{\mathrm{T}} \right]^{\mathrm{T}}, \tag{3}$$

where $\mathbf{J}_{MRe} = [\mathbf{L}_{s1}\mathbf{A}_1\,{}^1\mathbf{Z}_1 \quad \mathbf{L}_{s2}\mathbf{A}_2\,{}^2\mathbf{Z}_2 \quad \cdots \quad \mathbf{L}_{sn_v}\mathbf{A}_{n_v}\,{}^{n_r}\mathbf{Z}_{n_v}]$ is a 3 × n_s matrix and $\mathbf{J}_{BRe} = [0 \quad \mathbf{E}]$ is a 3 × 6 matrix.

Let $\mathbf{J}_e = [\mathbf{J}_{Be} \quad \mathbf{J}_{Me}] = \begin{bmatrix} \mathbf{J}_{BTe} & \mathbf{J}_{MTe} \\ \mathbf{J}_{BRe} & \mathbf{J}_{MRe} \end{bmatrix}$ be a 6 × (6 + n_v) Jacobian matrix for mapping generalized velocities to the end-points; let $\boldsymbol{\Phi} = \left[\begin{pmatrix} \mathbf{v}_0^{\mathrm{T}} & \boldsymbol{\omega}_0^{\mathrm{T}} \end{pmatrix} \quad \dot{\mathbf{q}}^{\mathrm{T}} \right]^{\mathrm{T}}$ be a vector with (6 + n_v) elements, which are the linear velocities and angular velocities of the body, and joint velocities. Let $\dot{\mathbf{X}}_{ae}$ and $\mathbf{J}_{ae}$ denote the velocities of all the wheel–soil interaction points and the corresponding Jacobian matrix:

$$\dot{\mathbf{X}}_{ae} = \begin{bmatrix} \mathbf{v}_e(1) \\ \boldsymbol{\omega}_e(1) \\ \vdots \\ \mathbf{v}_e(n_w) \\ \boldsymbol{\omega}_e(n_w) \end{bmatrix}, \quad \mathbf{J}_{ae} = \begin{bmatrix} \mathbf{J}_e(1) \\ \mathbf{J}_e(2) \\ \vdots \\ \mathbf{J}_e(n_w) \end{bmatrix},$$

which are a $6n_w \times 1$ vector and $6n_w \times (n_v + 6)$ matrix, respectively. We thus obtain

$$\dot{\mathbf{X}}_{ae} = \mathbf{J}_{ae}\dot{\boldsymbol{\Phi}}. \tag{4}$$

The same method is used to deduce the Jacobian matrix by mapping the velocities from the generalized coordinates to the link centroid:

$$\dot{\mathbf{X}}_a = \mathbf{J}_a\dot{\boldsymbol{\Phi}}, \tag{5}$$

where $\dot{\mathbf{X}}_a$ ($6n_v \times 1$) is the velocity vector of all centroids, and $\boldsymbol{J}_a$ ($6n_v \times (n_v + 6)$) is the Jacobian matrix. In Eq. (5),

$$\dot{\mathbf{X}}_a = \begin{bmatrix} \mathbf{v}_1 \\ \boldsymbol{\omega}_1 \\ \vdots \\ \mathbf{v}_{n_v} \\ \boldsymbol{\omega}_{n_v} \end{bmatrix}, \quad \mathbf{J}_a = \begin{bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \\ \vdots \\ \mathbf{J}_{n_v} \end{bmatrix}, \quad \mathbf{J}_i = [\mathbf{J}_{Bi} \quad \mathbf{J}_{Mi}] = \begin{bmatrix} \mathbf{J}_{BTi} & \mathbf{J}_{MTi} \\ \mathbf{J}_{BRi} & \mathbf{J}_{MRi} \end{bmatrix},$$

where $\boldsymbol{J}_i$ is a 6 × (6 + n_v) matrix. $\mathbf{J}_{BTi} = \begin{bmatrix} \mathbf{E} & \tilde{\mathbf{r}}_{0i}^{\mathrm{T}} \end{bmatrix}$ and $\mathbf{J}_{BRi} = \begin{bmatrix} 0 & \mathbf{E} \end{bmatrix}$ are both 3 × 6 matrices, and

$$\mathbf{J}_{MRi} = \begin{bmatrix} \mathbf{L}_{i1}{}^{1}\mathbf{Z}_1 & \mathbf{L}_{i2}{}^{2}\mathbf{Z}_2 & \cdots & \mathbf{L}_{in_v}{}^{n_v}\mathbf{Z}_{n_v} \end{bmatrix} \text{ and}$$

$$\mathbf{J}_{MTi} = \begin{bmatrix} \mathbf{L}_{i1}{}^{1}\mathbf{Z}_1 \times (\mathbf{r}_i - \mathbf{p}_1) & \mathbf{L}_{i2}{}^{2}\mathbf{Z}_2 \times (\mathbf{r}_i - \mathbf{p}_2) & \ldots & \mathbf{L}_{in_v}{}^{n_v}\mathbf{Z}_{n_v} \times (\mathbf{r}_i - \mathbf{p}_{n_v}) \end{bmatrix}$$

are both $3 \times n_v$ matrices.

4.1.2 Generalized dynamics model

Substituting Eq. (5) into the kinetic energy equation gives

$$T = \frac{1}{2}\sum_{i=0}^{n_v}(\boldsymbol{\omega}_i^{\mathrm{T}}\mathbf{I}_i\boldsymbol{\omega}_i + m_i\mathbf{v}_i^{\mathrm{T}}\mathbf{v}_i) = \frac{1}{2}\dot{\boldsymbol{\Phi}}^{\mathrm{T}}\mathbf{H}_{sys}\dot{\boldsymbol{\Phi}}\,, \tag{6}$$

where $\mathbf{H}_{sys}$ is the $(n_v + 6) \times (n_v + 6)$ system generalized inertia matrix (Yoshida, 2000):

$$\mathbf{H}_{sys} = \begin{bmatrix} M_a(\mathbf{E})_{3\times3} & M_a(\tilde{\mathbf{r}}_{0g}^{\mathrm{T}})_{3\times3} & (\mathbf{J}_{Tg})_{3\times n_v} \\ M_a(\tilde{\mathbf{r}}_{0g})_{3\times3} & (\mathbf{H}_{\omega})_{3\times3} & (\mathbf{H}_{\omega\phi})_{3\times n_v} \\ (\mathbf{J}_{Tg}^{\mathrm{T}})_{n_v\times3} & (\mathbf{H}_{\omega\phi}^{\mathrm{T}})_{n_v\times3} & (\mathbf{H}_{\phi})_{n_v\times n_v} \end{bmatrix}. \tag{7}$$

In Eq. (7), M_a is the overall mass of the robot,

$$\mathbf{r}_{0g} = \mathbf{r}_g - \mathbf{r}_0\,,\ \mathbf{H}_{\omega} = \mathbf{I}_0 + \sum_{i=1}^{n_v}(\mathbf{I}_i + m_i\tilde{\mathbf{r}}_{0i}\tilde{\mathbf{r}}_{0i}^{\mathrm{T}}) = \mathbf{I}_0 + \sum_{i=1}^{n_v}(\mathbf{I}_i + m_i\tilde{\mathbf{r}}_{0i}^{\mathrm{T}}\tilde{\mathbf{r}}_{0i})\,,\ \mathbf{J}_{Tg} = \sum_{i=0}^{n_v} m_i\mathbf{J}_{MTi}\,,$$

$$\mathbf{H}_{\phi} = \sum_{i=1}^{n_v}(\mathbf{J}_{MRi}^{\mathrm{T}}\mathbf{I}_i\mathbf{J}_{MRi} + m_i\mathbf{J}_{MTi}^{\mathrm{T}}\mathbf{J}_{MTi})\,, \text{ and } \mathbf{H}_{\omega\phi} = \sum_{i=1}^{n_v}(\mathbf{I}_i\mathbf{J}_{MRi} + m_i\tilde{\mathbf{r}}_{oi}\mathbf{J}_{MTi})\,.$$

According to the Lagrange function,

$$\mathbf{F}_{sys} = \mathbf{H}_{sys}(\boldsymbol{\Phi})\ddot{\boldsymbol{\Phi}}_{sys} + \mathbf{C}(\boldsymbol{\Phi}, \dot{\boldsymbol{\Phi}})\dot{\boldsymbol{\Phi}} + \mathbf{f}(\dot{\boldsymbol{\Phi}}) + \mathbf{G}(\boldsymbol{\Phi})\,, \tag{8}$$

where $\boldsymbol{C}$ is an $(n_v + 6) \times (n_v + 6)$ stiffness matrix describing the Coriolis and centripetal effects, which are proportional to $\dot{q}_i^2$ and $\dot{q}_i\dot{q}_j$, respectively; $\boldsymbol{f}$ is an $(n_v + 6) \times 1$ matrix that describes viscous and Coulomb friction (typically negligible in a rigid-body dynamics system); $\boldsymbol{G}$ is an $(n_v + 6) \times 1$ gyroscopic vector reflecting gravity loading; and $\boldsymbol{F}_{\mathrm{sys}}$ is the vector of generalized forces:

$$\mathbf{F}_{sys} = \mathbf{N} + \mathbf{J}_{ae}{}^{\mathrm{T}}\mathbf{N}_{ae}\,. \tag{9}$$

In Eq. (9), $\boldsymbol{N}$ is an $(n_v + 6) \times 1$ matrix including the forces ($\mathbf{F}_0$) and moments ($\mathbf{M}_0$) acting on the body, and those acting on the joints ($\boldsymbol{\tau} = [\tau_1 \quad \tau_2 \quad \cdots \quad \tau_{n_v}]^{\mathrm{T}}$); $\mathbf{N}_{ae}$ is a $6n_w \times 1$ vector including the external forces ($\mathbf{F}_e$) and moments ($\mathbf{M}_e$) from the soil that act on the wheel:

$$\mathbf{N}=\begin{bmatrix}\mathbf{F}_0\\ \mathbf{M}_0\\ \boldsymbol{\tau}\end{bmatrix},\ \mathbf{N}_{ae}=[\mathbf{F}_e^{\mathrm{T}}(1)\ \ \mathbf{M}_e^{\mathrm{T}}(1)\ \ \cdots\ \ \mathbf{F}_e^{\mathrm{T}}(n_w)\ \ \mathbf{M}_e^{\mathrm{T}}(n_w)]^{\mathrm{T}}.$$

The dynamics equation of a WMR including the wheel–soil interaction terramechanics is

$$\mathbf{H}_{sys}(\boldsymbol{\Phi})\ddot{\boldsymbol{\Phi}}_{sys}+\mathbf{C}(\boldsymbol{\Phi},\dot{\boldsymbol{\Phi}})\dot{\boldsymbol{\Phi}}+\mathbf{f}(\dot{\boldsymbol{\Phi}})+\mathbf{G}(\boldsymbol{\Phi})-\mathbf{N}-\mathbf{J}_{ae}{}^{\mathrm{T}}\mathbf{N}_{ae}=0. \tag{10}$$

Let $\mathbf{C}(\boldsymbol{\Phi},\dot{\boldsymbol{\Phi}})\dot{\boldsymbol{\Phi}}+\mathbf{f}(\dot{\boldsymbol{\Phi}})+\mathbf{G}(\boldsymbol{\Phi})=\mathbf{D}$. The generalized accelerations can then be calculated as

$$\ddot{\boldsymbol{\Phi}}_{sys}=\mathbf{H}_{sys}^{-1}(\mathbf{N}+\mathbf{J}_{ae}{}^{\mathrm{T}}\mathbf{N}_{ae}-\mathbf{D}). \tag{11}$$

The recursive Newton–Euler method is used to deduce an equation equivalent to Eq. (10) to calculate the unknown ***D***.
The Newton–Euler equations are

$$\begin{cases}\mathbf{F}_i=m_i\dot{\mathbf{v}}_i\\ \mathbf{N}_i=\mathbf{I}_i\dot{\boldsymbol{\omega}}_i+\boldsymbol{\omega}_i\times\mathbf{I}_i\boldsymbol{\omega}_i\end{cases}. \tag{12}$$

According to D'Alembert's principle, the effect of $\mathbf{f}_i$ and $\mathbf{m}_i$ on link *i* through joint *i* is

$$\begin{cases}\mathbf{m}_i=\mathbf{M}_i+\sum_{j=i+1}^{n}\mathbf{S}_{ij}(\mathbf{l}_{ij}\times\mathbf{f}_j+\mathbf{m}_j)-\mathbf{S}_{ii}\times\\ \quad[\lambda_P(i)\mathbf{A}_i{}^i\mathbf{Z}_i q_i-\mathbf{c}_{ii}]\times(\mathbf{F}_i-m_i\mathbf{g})-\mathbf{S}_{ei}(\mathbf{l}_{ie}\times\mathbf{F}_{ei}+\mathbf{M}_{ei}),\\ \mathbf{f}_i=\mathbf{F}_i-m_i\mathbf{g}+\sum_{j=i+1}^{n}\mathbf{S}_{ij}\mathbf{f}_j-\mathbf{S}_{ei}\mathbf{F}_{ei}\end{cases} \tag{13}$$

where $\lambda_P(i)$ is 1 for a prismatic joint and zero for a rotational joint, ***S*** is the incidence matrix to find the upper connection of a link, and S_{ei} indicates whether *i* is an end-point. The generalized force/moment of link *i* is

$$\tau_i=\begin{cases}\mathbf{m}_i^{\mathrm{T}}\mathbf{A}_i{}^i\mathbf{Z}_i & \text{(Rotational joint)}\\ \mathbf{f}_i^{\mathrm{T}}\mathbf{A}_i{}^i\mathbf{Z}_i & \text{(Prismatic joint)}\end{cases}. \tag{14}$$

The forces and moments that act on the body are

$$\begin{cases}\mathbf{F}_0=\sum_{j=1}^{n}\mathbf{S}_{0j}\mathbf{f}_j+m_0(\dot{\mathbf{v}}_0-\mathbf{g})\\ \mathbf{M}_0=\sum_{j=1}^{n}\mathbf{S}_{0j}(\mathbf{c}_{0j}\times\mathbf{f}_j+\mathbf{m}_j)+\mathbf{I}_0\dot{\boldsymbol{\omega}}_0+\boldsymbol{\omega}_0\times\mathbf{I}_0\boldsymbol{\omega}_0\end{cases}, \tag{15}$$

where S_{0j} is a flag vector that indicates whether *j* has a connection with the body.
Following Eq. (10), let the accelerations of all the generalized coordinates and the external forces/moments be zero; it is then possible to obtain ***D*** with Eqs. (14) and (15).

4.2 Wheel–soil interaction terramechanics models

The soil applies three forces and three moments to each wheel, as shown in Fig. 4. The normal force F_N can sustain the wheel. The cohesion and shearing of the soil can generate a resistance moment M_R and a tractive force; the resistance force is caused by the wheel sinking into the soil; the composition of the tractive and resistance forces is called the drawbar pull F_{DP}, which is the effective force of driving a wheel. As a wheel steers or when there is a slip angle, there is a side force F_S, a steering resistance moment M_S, and an overturning moment M_O acting on the wheel.

4.2.1 Driving model

Figure 7 is a diagram of the lugged wheel–soil interaction mechanics, where z is wheel sinkage; θ_1 is the entrance angle at which the wheel begins to contact the soil; θ_2 is the exit angle at which the wheel loses contact with the soil; θ_m is the the angle of maximum stress; θ_1' is the angle at which the soil starts to deform; W is the vertical load of the wheel; DP is the resistance force acting on the wheel; T is the driving torque of the motor; r is the wheel radius; h is the height of the lugs; v is the vehicle velocity; and ω is the angular velocity of the wheel. The soil interacts with the wheel in the form of continuous normal stress σ and shearing stress τ, which can be integrated to calculate the interaction mechanics. To improve the simulation speed, a simplified closed-form formula (Ding et al., 2009a) is adopted and improved considering the effect of the normal force:

$$\begin{cases} F_{DP}=[\dfrac{M_R(A^2+B^2)}{r_s AC}-\dfrac{BF_N}{A}](1+c_{P1}+c_{P2}s)(1+c_{P3}\dfrac{\overline{W}-F_N}{\overline{W}}) \\ F_N = rbA\sigma_m + r_s bB\tau_m \\ M_R=\dfrac{r_s^2 CD[bc+[1+c_M(\overline{W}-F_N)/\overline{W}]F_N\tan\varphi/(rA)]}{1+r_s BD\tan\varphi/(rA)} \end{cases} . \quad (16)$$

In Eq. (16), s is the slip ratio defined by Ding et al. (2009); c_{P1} and c_{P2} are adopted to reflect the influence of the slip ratio on the drawbar pull, and θ_m can thus be simplified as half of θ_1; c_{P3} and c_M are parameters that compensate for the effect of the normal force; $\overline{W}$ is the average normal force of the wheels; and $\sigma_m = K_s r^N(\cos\theta_m - \cos\theta_1)^N$, $C=(\theta_1-\theta_2)/2$,

$$A=(\cos\theta_m-\cos\theta_2)/(\theta_m-\theta_2)+(\cos\theta_m-\cos\theta_1)/(\theta_1-\theta_m),$$

$$B=(\sin\theta_m-\sin\theta_2)/(\theta_m-\theta_2)+(\sin\theta_m-\sin\theta_1)/(\theta_1-\theta_m), \text{ and}$$

$$\tau_m=(c+\sigma_m\tan\phi)\times (1-\exp\{-r_s[(\theta_1'-\theta_m)-(1-s)(\sin\theta_1'-\sin\theta_m)]/k\}) .$$

The newly introduced parameters are
$\theta_1'=\text{acos}[(r-z)/R_j]$, $K_s=k_c/b+k_\varphi$, $N=n_0+n_1 s$, and $\theta_2\approx 0$.

The radius R_j is a value between r and $r+h$ that compensates for the lug effect (Ding et al., 2009b). The soil parameters in the equations are k_c, the cohesive modulus; k_φ, the frictional modulus; N, an improved soil sinkage exponent; c, the cohesion of the soil; φ, the internal

friction angle; and k, the shearing deformation modulus. n_0 and n_1 are coefficients for calculating N, which are important when predicting the slip-sinkage of wheels.

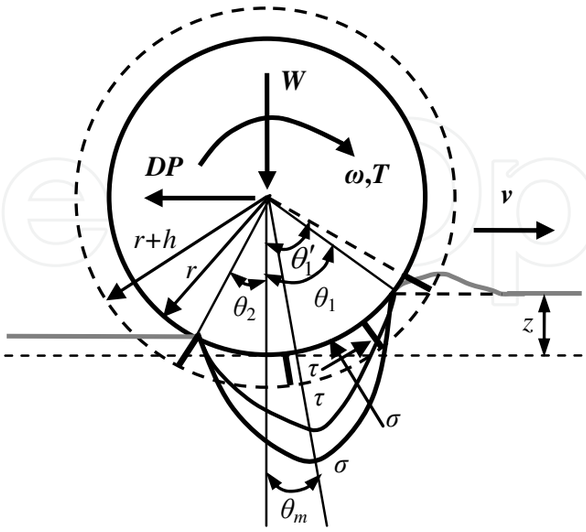


Fig. 7. Lugged wheel–soil interaction mechanics

4.2.2 Steering model

The model for calculating the side force F_S is (Ishigami & Yoshida, 2005)

$$F_S = rb\int_{\theta_2}^{\theta_1} \tau_y(\theta)d\theta + \int_{\theta_2}^{\theta_1} R_b(r - h(\theta)\cos\theta)d\theta \,, \tag{17}$$

$$\tau_y(\theta) = [c + \sigma(\theta)]\{1 - \exp[-r(1-s)(\theta_1 - \theta)\tan\beta / k_y]\} \,, \tag{18}$$

$$R_b = \cot X_c + \tan(X_c + \phi)\left\{hc + \frac{1}{2}\rho h^2\left[(\cot X_c + \frac{\cot X_c{}^2}{\cot\phi}\right]\right\}, \tag{19}$$

where $X_c = \pi / 4 - \varphi / 2$; k_y is the shearing deformation modulus in the y direction; β is the skid angle; and h is the wheel height in the soil.
The overturning moment is approximated by

$$M_O \approx F_S r \,. \tag{20}$$

The steering resistance moment M_s is considered to be zero, and the motion of steering is simulated employing the kinematics method, as the steering torque has little effect on the motion of the entire rover, and the model is still under development.

4.3 Deformable rough-terrain geometry modeling (Ding, 2009)

4.3.1 Calculation of contact area

For simplicity, the literature often assumes that wheel–soil interaction occurs at a single point, which may result in large errors when the robot moves over deformable rough

terrain, and even in simulation failure because of abrupt changes in wheel sinkage and other forces. Calculating the interaction area of a wheel moving on soft soil is important for high-fidelity simulation, which is employed to predict and transform the interaction mechanics. Figure 8 shows the interaction area of a wheel moving on rough terrain. The known parameters are the position of a wheel's center W, (x_w, y_w, z_w); the yaw angle of a wheel, φ_w; and the DEM of the terrain. The interaction area is simplified as an inclined plane determined by points P_1, P_2, and P_3, the normal vector of which is

$$\mathbf{z_e} = \begin{bmatrix} A_t \\ B_t \\ C_t \end{bmatrix} = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{bmatrix} \times \begin{bmatrix} x_3 - x_1 \\ y_3 - y_1 \\ z_3 - z_1 \end{bmatrix}. \tag{21}$$

The equation of the inclined plane $P_1P_2P_3$ is therefore

$$A_t(x - x_1) + B_t(y - y_1) + C_t(z - z_1) = 0 \tag{22}$$

P, the foot of the perpendicular line drawn from point w to plane $P_1P_2P_3$, is located on the line $(x - x_w) / A_t = (y - y_w) / B_t = (z - z_w) / C_t$. The coordinates of point E can be solved by substituting the line equation into Eq. (22). The length of wP is thus deduced:

$$\overline{wP} = \frac{\left| A_t(x_w - x_1) + B_t(y_w - y_1) + C_t(z_w - z_1) \right|}{\sqrt{A_t^2 + B_t^2 + C_t^2}}. \tag{23}$$

The wheel sinkage is then determined as

$$z = \overline{Pe} = r - \overline{wP}. \tag{24}$$

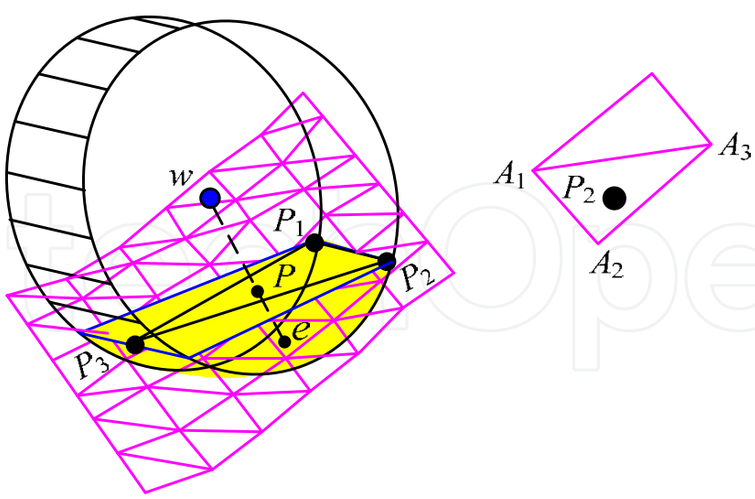


Fig. 8. Interaction area of wheel moving on deformable rough terrain

Point P_2 is used to illustrate how to obtain the coordinates of points P_1, P_2, and P_3. A wheel moving on a random plane can be decomposed into climbing up/down a slope at an angle θ_{cl} and traversing a slope with an inclination angle θ_{cr}, as shown in Fig. 4. The x and y coordinates of point P_2 are then

$$\begin{cases} x_{P2} = x_w + r\cos\theta_{cr} \\ y_{P2} = y_w + r\sin\theta_1\cos\theta_{cl} \end{cases}. \tag{25}$$

The coordinates of points A_1, A_2, and A_3 are easy to find by referring to the DEM. z_{P2} can then be determined using the same method as that for calculating point E.

4.3.2 Terminal force transformation matrix

Figure 9 shows the forces and moments of the soil that act on a wheel. $\{\Sigma_e\}$ and $\{\Sigma_w\}$ are coordinate systems with the same orientation and different origins, at the end-point and wheel center, respectively.

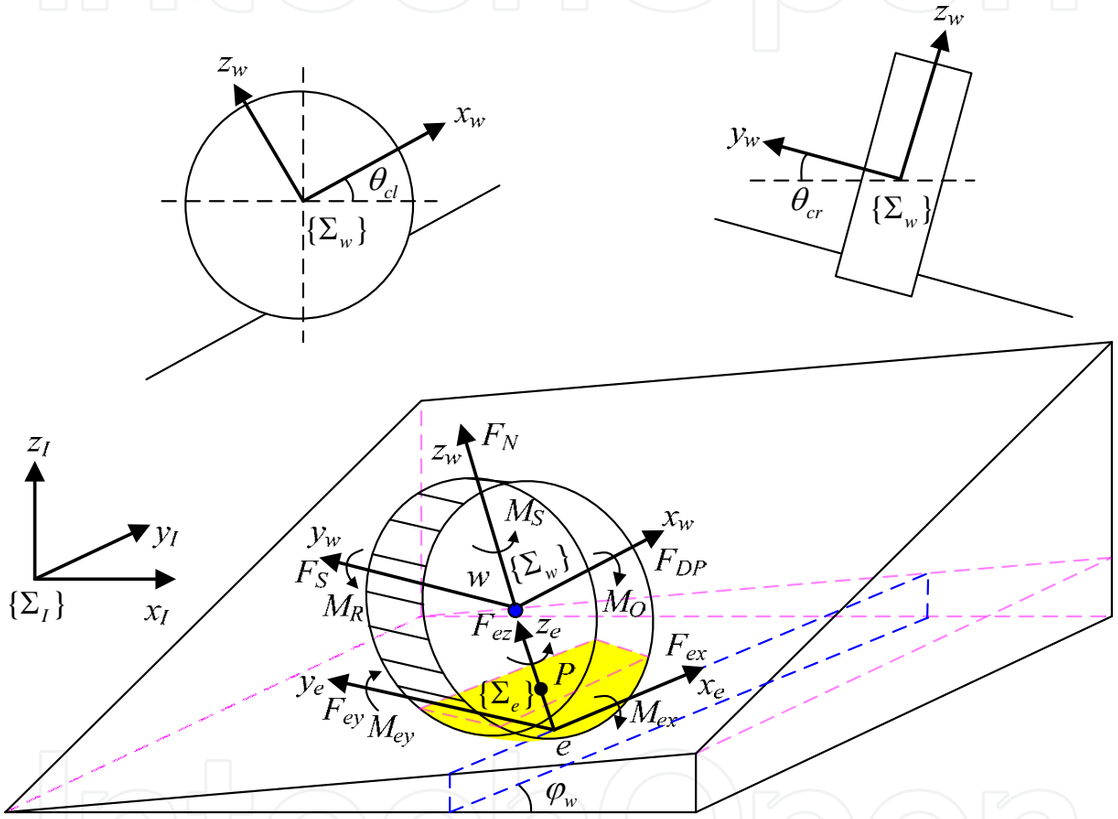


Fig. 9. Force analysis of wheel moving on random slope

x_e is the line of intersection of the wheel–soil interaction plane and the plane with an included angle of φ_w to the x axis: $x\tan\varphi_w - y + D' = 0$. It is deduced that

$$\mathbf{x_e} = \{C_t, C_t\tan\varphi_w, -A_t - B_t\tan\varphi_w\}. \tag{26}$$

The vector direction of y_e is then

$$\mathbf{y}_e = \mathbf{z}_e \times \mathbf{x}_e = \begin{bmatrix} -A_tB_t - (B_t^2 + C_t^2)\tan\varphi_w \\ C_t^2 + A_t(A_t + B_t\tan\varphi_w) \\ A_tC_t\tan\varphi_w - B_tC_t \end{bmatrix}. \tag{27}$$

θ_{cl} (θ_{cr}) is the angle between $\boldsymbol{x}_e$ ($\boldsymbol{y}_e$) and the horizontal plane, which can be calculated as

$$\begin{cases} \theta_{cl} = \arcsin[(-A_t - B_t \tan\varphi_w) / X_1] \\ \theta_{cr} = \arcsin[C_t (A_t \tan\varphi_w - B_t) / X_2] \end{cases}, \tag{28}$$

where $X_1 = \sqrt{C_t^2 (1 + \tan^2 \varphi_w) + (A_t + B_t \tan\varphi_w)^2}$,

$X_2 = \sqrt{X_3 [A_t^2 + C_t^2 + 2A_t B_t \tan\varphi_w + (B_t^2 + C_t^2)\tan^2 \varphi_w]}$, and $X_3 = \sqrt{A_t^2 + B_t^2 + C_t^2}$.

According to $\boldsymbol{x}_e$, $\boldsymbol{y}_e$, and $\boldsymbol{z}_e$, the transformation matrix from $\{\Sigma_e\}$ to $\{\Sigma_l\}$ is

$$\mathbf{A}_e = \begin{bmatrix} \dfrac{C_t}{X_1} & \dfrac{-A_t B_t - (B_t^2 + C_t^2)\tan\varphi_w}{X_2} & \dfrac{A_t}{X_3} \\ \dfrac{C_t \tan\varphi_w}{X_1} & \dfrac{C_t^2 + A_t^2 + A_t B_t \tan\varphi_w}{X_2} & \dfrac{B_t}{X_3} \\ \dfrac{-A_t - B_t \tan\varphi_w}{X_1} & \dfrac{A_t C_t \tan\varphi_w - B_t C_t}{X_2} & \dfrac{C_t}{X_3} \end{bmatrix}. \tag{29}$$

The external forces and torques that act at the wheel–soil interaction point are

$$\begin{cases} {}^e\mathbf{F}_e = {}^w\mathbf{F}_e = [F_{DP} \quad F_S \quad F_N]^{\mathrm{T}} \\ {}^e\mathbf{M}_e = [M_O - rF_S \quad -M_R + rF_{DP} \quad M_S]^{\mathrm{T}} \end{cases}. \tag{30}$$

The equivalent forces and moments that act on the wheel in the inertial coordinates $\{\Sigma_l\}$ are

$$\begin{cases} \mathbf{F}_e = \mathbf{A}_e \, {}^e\mathbf{F}_e \\ \mathbf{M}_e = \mathbf{A}_e \, {}^e\mathbf{M}_e \end{cases}. \tag{31}$$

5. Research on simulation for planetary WMRs at SKLRS

Simulation technology has played an important role in the research of planetary rovers at SKLRS for more than 10 years. Different simulation methods are used for different purposes. In collaboration with Tohoku University, Japan, the Matlab toolbox entitled SpaceDyn is used to develop a simulation system for control algorithm verification, by integrating high-fidelity terramechanics models. To realize real-time simulation for hardware-in-loop testing and successive teleoperation, Vortex, a real-time simulation platform, is adopted for the development of RoSTDyn with C++ language.

5.1 ADAMS-based simulation for planetary WMRs

Commercial dynamics software ADAMS is adopted in the mechanical design of rovers for purposes such as performance analysis, optimization design, and control algorithm verification. The Contact model, Tire model, and self-developed terramechanics model are used to predict wheel–soil interaction mechanics.

5.1.1 Simulation with Contact model

Using the Contact model provided by ADAMS software is an easy way to realize dynamics simulation of a wheeled rover. The wheel–soil interaction is considered as solid-to-solid contact. Different frictional coefficients and dumping ratios can be set by the users according to their understanding of the soil properties. The structured terrain can be constructed with ADAMS or imported from computer-aided design software such as Pro/E. This approach is simple and reflects the characteristics of the mechanism, at least at the kinematics level, although the fidelity of wheel–soil interaction mechanics is poor, and it is thus used during the initial phase of rover design. For instance, Tao et al. designed a six-wheeled robot with good locomotion performance on rough terrain, and they simulated different locomotion modes with ADAMS to verify the concept of its mechanism, as shown in Fig. 10 (Tao et al., 2006).

Fig. 10. Simulation for WMR with ADAMS and Contact model

5.1.2 Comprehensive simulation using Tire model

The structure of a comprehensive virtual simulation system for lunar rovers is shown in Fig. 11 (Gao et al., 2011). The simulation system integrates the software of ADAMS, Pro/E, 3DS Max, and Matlab to realize the functions of rover vehicle modeling, terrain modeling, kinematics/dynamics analysis including the mechanics of wheel–soil interaction, and control, respectively.

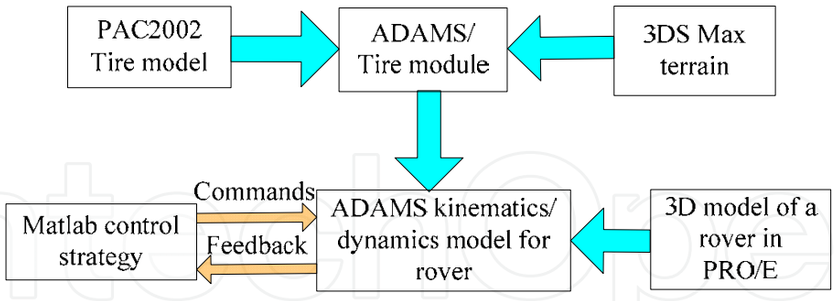


Fig. 11. Diagram of comprehensive virtual simulation system for lunar rovers

The 3D model of a lunar rover constructed using Pro/E can be imported into ADAMS through an ADAMS/Pro connection module; the rough terrain model of the lunar surface produced with 3DS Max can be imported into an ADAMS/Tire module through an interface; the PAC2002 Tire model is called by the ADAMS/Tire module for prediction of the terramechanics of the wheel–soil interaction. By configuring the simulation parameters for the rover model, the terrain, and Tire model in ADAMS, a virtual rover is constructed whose kinematics and dynamics can be solved by ADAMS/Solver. Figure 12 presents an example.

In directing a virtual rover to follow a planned path while compensating for slipping on deformable rough terrain, the required control strategy can be realized with Matlab. A virtual rover created with ADAMS software can be controlled through the interface between ADAMS/Control and Matlab/Simulink. ADAMS/Control provides an interactive environment for establishing and demonstrating an S-function "controlled object," which can be controlled with the Simulink toolbox of Matlab. In each integration step, ADAMS/Control is called as a subprogram by Simulink. The control instructions generated by Simulink are then directed to the corresponding mechanisms (such as the driving motors of the wheels) of lunar rovers in ADAMS through ADAMS/Control. The rover's motion is then calculated by ADAMS/Solver on the basis of dynamics simulation, and the related information is fed back to Simulink through ADAMS/Control. The entire transmission process is automatic and transparent.

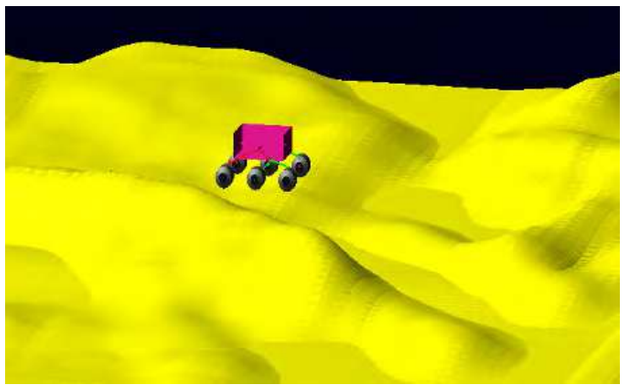


Fig. 12. Dynamics simulation of lunar rover with ADAMS

ADAMS/Tire is a module for predicting the interactive forces and moments between wheels and roads. Several types of tire models are provided by ADAMS, including MF-Tyre, UA, Ftire, SWIFT, PAC89, and PAC2002. Each tire model has its own advantages and disadvantages. After careful analysis and comparison, the PAC2002 model was finally chosen for calculation of the mechanics of the wheel–soil interaction of a lunar rover. This model is applicable to the interaction between a wheel and 3D roads or obstacles.

The PAC2002 model uses trigonometric functions to fit the experimental data and thus predict the forces and moments that the soil exerts upon a wheel, including the drawbar pull F_x, lateral force F_y, sustaining force F_z, overturning moment M_x, resistance torque M_y, and aligning torque M_z. The general expression of the function, which is called the magic formula, is

$$Y(X) = D\cos\left(C\arctan\left\{BX - E[BX - \arctan(BX)]\right\}\right), \tag{32}$$

where $Y(X)$ is a force or moment, the independent variable X reflects the effect of the slip angle or longitudinal slip ratio of a wheel for an actual situation, parameters B, C, and D are determined by the wheel's vertical load and camber angle, while E is the curvature factor.

The unknown parameters can be determined by a data fitting method based on experimental results.

5.1.3 Simulation using self-developed terramechanics model

The simulation fidelity is improved to some extent using the Tire model instead of the Contact model. However, phenomenon such as of severe slip-sinkage and the lug effect still cannot be reflected well. ADAMS provides a method for users to embed the newly developed terramechanics models with higher precision (Jiao, 2009).

The GFORCE command is used to define the wheel–soil interaction mechanics that consists of three mutually orthogonal translational force components and three orthogonal torque components. As the force and torque expressions are lengthy and complex, the GFOSUB evaluation subroutine is used to compute wheel–soil interaction mechanics applied by a GFORCE statement with FORTRAN language. The wheel–soil interaction mechanics program is compiled and linked to generate an object file (*.obj). The Create Custom Solver command is then used to generate a dynamic link library (*.dll) file and library file (*.lib). The general force, GFORCE, which is applied to the center of the wheel, is set as shown in Fig. 13 to call the wheel–soil interaction function through the *.dll file.

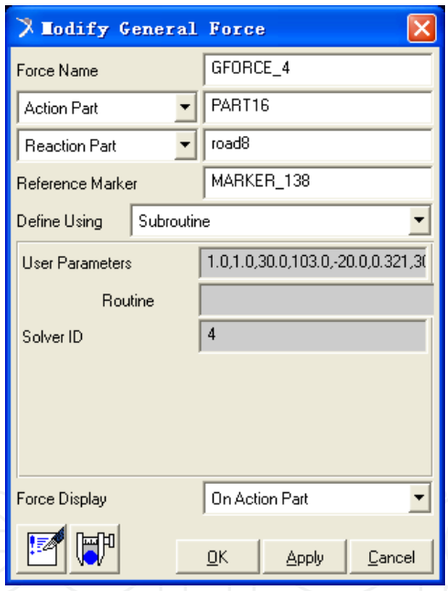


Fig. 13. GFORCE Subroutine

5.2 Matlab-based high-fidelity simulation platform for planetary WMRs (Ding, 2010a)

5.2.1 Implementation of simulation platform

A numerical simulation program based on a Matlab toolbox called SpaceDyn was developed (Yoshida 2000). The principle diagram is shown in Fig. 14. Given the DEM terrain, soil parameters, and rover model parameters, the program calculates the wheel–soil interaction area, predicts the external forces that act on the wheel, calculates the accelerations of the generalized coordinates on the basis of the dynamics model, and then integrates them to obtain their velocities and positions on the basis of kinematics equations.

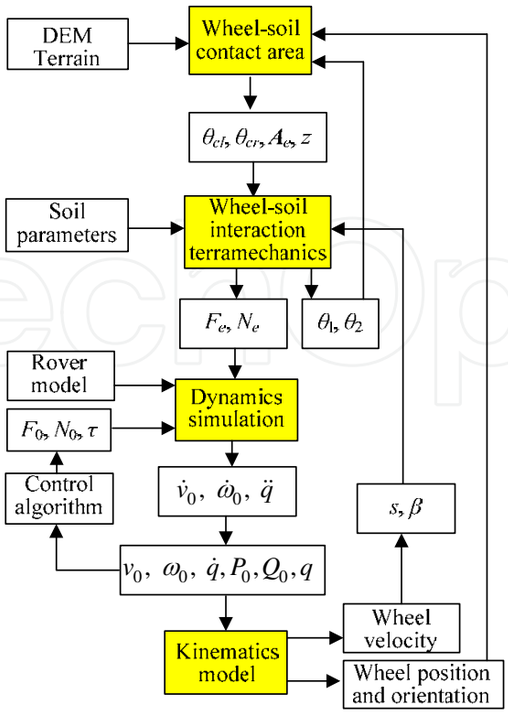


Fig. 14. Principle diagram of dynamics simulation

5.2.2 Experimental validation

El-Dorado II, a four-wheeled mobile robot developed at Space Robotics Laboratory of Tohoku University in Japan was used to validate the simulation. The robot uses four force/torque (F/T) sensors to measure the wheel–soil interaction terramechanics. A visual odometry system was developed to measure the position of the rover body and the slip ratio of the wheels. The entrance angles used for calculating the wheel sinkage were measured with an angle meter. Two groups of experiments were performed. In group 1, resistance forces were applied to the rover with counterweights from 0 to 60 N in intervals of 10 N, to generate different slip ratios. In group 2, the rover was controlled to climb slopes ranging from 0° to 15° in intervals of 3° (Fig. 15).

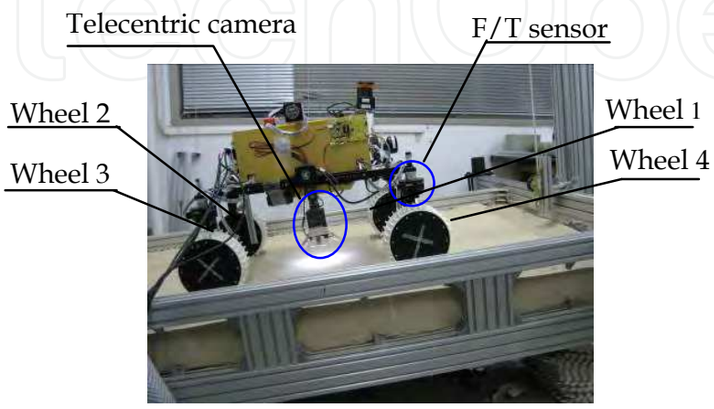


Fig. 15. Slope-climbing experiment using El-Dorado II robot

The parameters of Toyoura soft sand are identified from the experimental data: K_s = 1796 Kpa/m^N, c = 24.5 Pa, φ = 35.75°, and K = 10.45 mm. k_y is 19 mm. When the robot climbs a slope, the remaining parameters are n_0 = 0.66, n_1 = 0.72, c_{P1} = –0.379, c_{P2} = 0.616, c_{P3} = –0.448, and C_M = 0.214; on flat terrain, the parameters are n_0 = 0.63, n_1 = 0.72, c_{P1} = –0.276, c_{P2} = 0.633, c_{P3} = –0.304, and C_M = 0.354.

Comparisons of the simulation and experimental results are shown in Figs. 16 and 17. Not only can the motion of the robot be predicted with high fidelity, as indicated by the slip ratio, so too can the drawbar pull, moment of resistance, the normal force, and wheel sinkage.

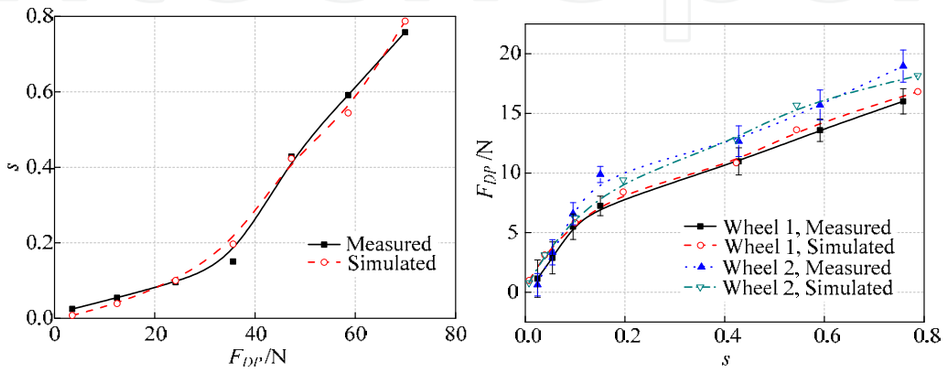


Fig. 16. Simulation and experimental results for robot moving on flat terrain

5.2.3 Simulation for deformable rough terrain

The robot was controlled to move from (0.5 m, 0.5 m) to (5 m, 5 m) on the randomly generated rough terrain shown in Fig. 17, with an initial yaw angle of 45°. While moving, the robot deviates from the scheduled path because of the inclination angle of the terrain. Figure 18 shows the slope angles that wheel number 4 traverses, the RPY (roll, pitch, and yaw) angles of the body and q_1 and q_2 joint angles (q_1 = –q_2), the slip ratios, and normal forces. The simulation platform was used to verify the slip-ratio-coordinated control (Ding, 2010b) and path-following strategy (Ding, 2009c), as shown in Fig. 19.

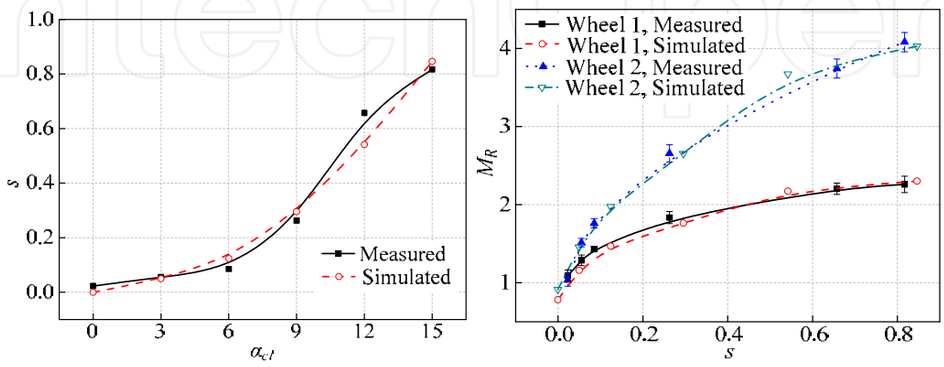
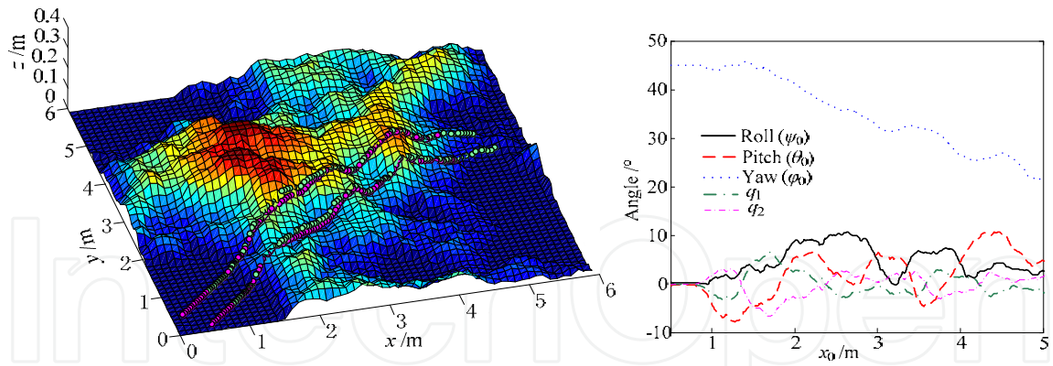


Fig. 17. Simulation and experimental results for robot climbing slope



(a) Rough terrain and wheel trajectories (b) RPY and q1 and q2 angles

Fig. 18. Simulation results for El-Dorado II moving on deformable rough terrain

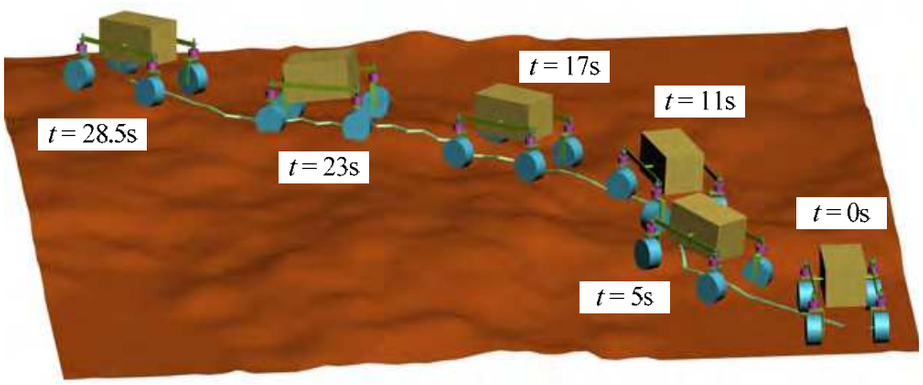


Fig. 19. Path-following simulation

5.3 RoSTDyn: Vortex-based high-fidelity and real-time simulation platform

5.3.1 Structure of RoSTDyn

RoSTDyn is composed of five modules as shown in Fig. 20 (Li et al., 2012). The module of the planetary rover model is used to create the simulation object-rover model, which is composed of a physical model and scenic model. The physical model is the real model that is used in the collision detection and dynamics calculation; the scenic model does not participate in any calculation, and it is driven according to the message transferred from the physical model, so it can be created vividly with 3D modeling software such as 3DS Max or Creator.

The controlling module focuses on realizing the interaction between the user and the simulation platform. This module is mainly used to control the movable joints of the rover and thus control, for example, the speed of the driving wheel, the turning degree of a wheel, and the posture of the solar panels and mast.

The module of the terrain model is used to simulate the terrain. It also includes a physical model and scene model. For the terrain, the physical model is a file of the node information including X, Y, and Z; the scenic model is generated by the 3D modeling software on the basis of node information.

The contact-area computing module is used to compute the parameters for the contact area between the wheel and terrain. These parameters are the precondition of the interaction force module, and this module will be introduced in detail in part III.

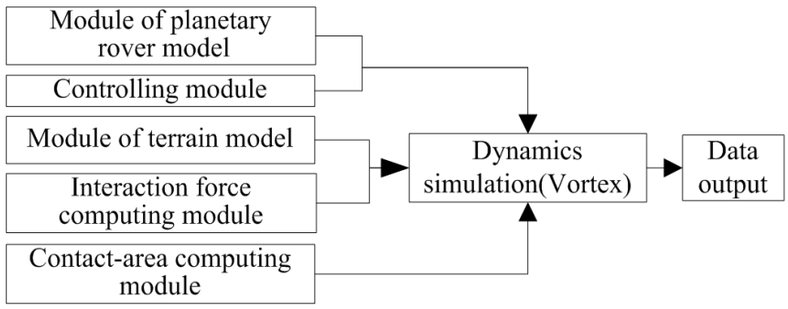


Fig. 20. Structure of RoSTDyn

The interaction force computing module is used to compute the interactive force between the wheel and terrain. These forces act on the rover model, and drive it to move and turn. The five modules are integrated in Vortex. Each module only needs to do its own work and transfer necessary information to other modules; in this way, the five modules work together in the simulation. Snapshots of rover simulation with RoSTDyn are shown in Fig. 21.

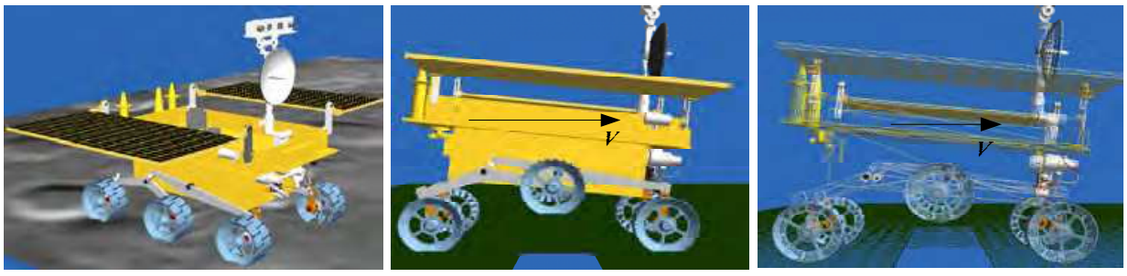


Fig. 21. Snapshots of rover simulation with RoSTDyn

5.3.2 Testing of real-time properties

To test the simulation speed, an experiment is designed. The rover's velocity is set to a constant value. At the moment that the simulation starts, a stopwatch is used to record the physical time, and the simulation time is exported by the program. At the same time, to test the calculation speed, another test is designed while the display of the scene is closed. Table 2 lists the time that the rover passes over three types of terrain. The display is opened for the first three and closed for the last three.

Terrain (display)	Simulation time t_s (s)	Physical time t_p (s)	t_s:t_p
Flat (Yes)	24.33	30.45	0.799:1
Slope (Yes)	34.50	42.91	0.804:1
Random (Yes)	43.50	53.76	0.809:1
Flat (No)	41.25	41.07	1:1
Slope (No)	38.50	38.43	1:1
Random (No)	45.00	44.82	1:1

Table 2. Time that the rover passes over three types of terrain

Here, because the rate is 80 Hz, the display wastes much time, and the time ratio is about 0.8; however, if the display is closed, the ratio reaches 1.0. Therefore, the calculation of

RoSTDyn is done in real time. To demonstrate this, the rate is multiplied by a factor of 10, and it is found that the time ratio decreases only to 0.8. This shows that it is still possible to increase the simulation speed of RoSTDyn.

6. Conclusions and future work

Virtual simulation can guarantee the successful exploration of planets by WMRs as it plays important roles in both the R&D phase and the exploration phase of the rovers. Customized simulation tools or tools with different simulation functions have been developed to support the R&D of planetary rovers, because general simulation tools do not meet the requirements of rover-based exploration missions well. The key technologies for developing high-fidelity comprehensive simulation systems include recursive dynamics, wheel–soil interaction terramechanics, and deformable rough terrain modeling. At SKLRS, different methods have been employed for rover simulation. ADAMS software is adopted along with the Contact model, Tire model, and self-developed GFORCE model during the design phase of rovers. Matlab is used to develop a high-fidelity simulation platform by embedding the terramechanics model and rough terrain model into support control strategy simulation. Vortex is used to develop RoSTDyn to realize real-time simulation and thus support the teleoperation of rovers.

In the future, successive teleoperation of planetary rovers using a 3D predictive display will be explored on the basis of the high-fidelity/real-time simulation platform. A faster-than-real-time simulation system will be developed to support the supervised control of rovers.

7. Acknowledgments

This work was supported by the National Natural Science Foundation of China (50975059/61005080), the Postdoctoral Foundation of China (20100480994), the Foundation of the Chinese State Key Laboratory of Robotics and System (grant No. SKLRS200801A02), the Key Natural Science Foundation of Heilongjiang Province in China (ZJG0709), the Postdoctoral Foundation of Heilongjiang Province, and the “111” Project (B07018).

8. References

Backes, P. G., Noms, J. S., & Powell, M. W. (2004). Multi-mission activity planning for Mars lander and rover missions. *IEEE Aerospace Conference Proceedings*, pp. 877-886, ISBN: 0-7803-8155-6, Big Sky, Montana, USA, March 2004

Bauer, R., Leung, W., & Barfoot, T. (2005). Development of a dynamic simulation tool for the exomars rover. *Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Munich, Germany, September 2005

Chen, L., Zhang, Y., & Ren, W. (2005). *Dynamic Analysis of Mechanical System and ADAMS Application*, Tsinghua University Press, ISBN 7302100969, Beijing, China

Ding, L., Gao, H., Deng, Z., et al. (2008). Design of comprehensive high-fidelity/high-speed virtual simulation system for lunar rover. *Proceedings of 2008 IEEE Int. Conf. on Robotics, Automation and Mechatronics*, pp. 1118-1123, ISBN 978-1-4244-1675-2, Chengdu, China, September 2008

Ding, L., Yoshida, K., Nagatani K., et al. (2009a). Parameter identification for planetary soil based on a decoupled analytical wheel–soil interaction terramechanics model.

Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4122-4127, ISBN 978-1-4244-3803-7, St. Louis, MO, USA, October 2009

Ding, L., Gao H., Deng Z., et al. (2009b). Slip ratio for lugged wheel of planetary rover in deformable soil: definition and estimation. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3343-3348, ISBN 978-1-4244-3803-7, St. Louis, MO, USA, October 2009

Ding, L. (2009c). *Wheel–soil Interaction Terramechanics for Lunar/planetary Exploration Rovers: Modeling and Application*. Doctoral Thesis of Harbin Institute of Technology (December, 2009), Harbin, China

Ding, L., Nagatani, K., Sato, K., et al. (2010a). Terramechanics-based high-fidelity dynamics simulation for wheeled mobile robot on deformable rough terrain. *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 4922-4927, ISBN: 978-1-4244-5038-12010, Anchorage, Alaska, USA, May 2010

Ding, L., Gao, H., Deng, Z., & Liu, Z. (2010b). Slip-ratio-coordinated control of planetary exploration robots traversing over deformable rough terrain. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4958-4963, ISBN: 978-1-4244-6674-0, Taipei, Taiwan, China, October 2010

Gao, H., Deng, Z., Ding, L., & Wang M. (2011). Virtual simulation system with path following control for lunar rovers moving on rough terrain. *Chinese Journal of Mechanical Engineering*, Vol. 24, No. 5, (May 2011), pp. 1-10, ISSN 0577-6686

Ishigami, G. & Yoshida, K. (2005). Steering characteristics of an exploration rover on loose soil based on all-wheel dynamics model. *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pp. 2041-2046, ISBN 0-7803-8912-3, Roman, Italy, August 2005

Ishigami, G., Nagatani, K., & Yoshida, K. (2007). Path planning for planetary exploration rovers and its evaluation based on wheel slip dynamics. *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2361-2366, ISBN 1-4244-0601-3, Roman, Italy, April 2007

Jet Propulsion Laboratory. (a). http://marsrovers.jpl.nasa.gov/home/index.html

Jet Propulsion Laboratory. (b). http://marsprogram.jpl.nasa.gov/MPF/rover/sojourner.html

Jiao, Z. (2009). *Dynamics Modelling and (ADAMS) Simulation for Lunar Rover Based on Terramechanics*. Master Dissertation of Harbin Institute of Technology (July 2009), Harbin, China.

Kuroda, Y., Kawanishi, M., & Matsukuma, M. (2003). Tele-operating system for continuous operation of lunar rover. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2565-2570, ISBN 0-7803-7860-1, Las Vegas, Nevada, USA, October 2003

Kunii, Y., Suhara, M., Kuroda, Y., & Kubota, T. (2001). Command data compensation for real-time tele-driving system on lunar rover: Micro-5. *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, pp. 1394-1399, ISBN 0-7803-6576-3, Seoul, Korea, May 2001

Lei, Z. (2004). *3D Predictive Display and Force Feedback Control of Tele-operation with Large Time Delay*. Dissertation of Master degree, Jilin University, Changchun, Jilin

Li, W., Gao, H., Ding, L., et al. RoSTDyn: A High-fidelity Real-time Simulation Platform for Planetary Rovers. 2012 IEEE International Conference on Robotics and Automation, submitted

Liang, B., Wang, W., & Wang, C. (2005). Primary concept on developing lunar rover. *Journal of Space International*, Vol. 2, (February, 2003), pp. 22-25, ISSN 009-2366

Luo, X., Ye, J., & Sun, Z. (2002). Research on the lunar rover simulation platform. *Journal of System Simulation*, Vol. 14, No. 9, (September 2002), pp. 1235-1238, ISSN 1004-731X

Maxwell, S., Cooper, B., Hartman, F., et al. (2005). The best of both worlds: integrating textual and visual command interfaces for mars rover operations. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pp. 1384-1388, ISBN 0-7803-9298-1, Waikoloa, HI, USA, October 2005

Michaud, S., Richter, L., Patel, N., et al. (2004). RCET: Rover Chassis Evaluation Tools. *Proceedings of the 8th ESA Workshop on Advanced Space Technologies for Robotics and Automation*, ESTEC, Noordwijk, Netherlands, November 2004

Neal, C. R. (2009). The Moon 35 years after Apollo: what's left to learn? *Chemie der Erde Geochemistry*, Vol. 69, No. 1, (February 2009), pp. 3-43, ISSN 0009-2819

Patel, N., Ellery, A., Allouis, E., et al. (2004). Rover Mobility Performance Evaluation Tool (RMPET): a systematic tool for rover chassis evaluation via application of Bekker theory. *Proceedings of the 8th ESA Workshop on Advanced Space Technologies for Robotics and Automation*, pp. 251-258, ESTEC, Noordwijk, Netherlands, November 2004

Schäfer, B. & Krenn, R. (2010). Multibody and contact dynamics: from satellite capturing to planetary rover terrainability. *Proceedings of the 1st ESA Workshop on Multibody Dynamics for Space Applications*, ESTEC, Noordwijk, Netherlands, February 2010

Schäfer, B., Gibbesch, A., Krenn, R., & Rebele, B. (2010). Planetary rover mobility simulation on soft and uneven terrain. *Journal of Vehicle System Dynamics*, Vol. 48, No. 1, (January, 2010), pp. 149-169, ISSN: 0042-3114

Tao J., Deng Z., Fang H., et al. (2006). Development of a wheeled robotic rover in rough terrains. *Proceedings of the 6th World Congress on Intelligent Control and Automation*, pp. 9272-9276, ISBN: 1-4244-0332-4, Dalian, China, June 2006

Van winnendael, M., Baglioni, P., Elfving, A., et al. (2008). The ExoMars Rover—Overview of phase B1 results. *Proceedings of 9th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Hollywood, Los Angles, California, USA, February 2008

Volpe, R. (2003). Rover functional autonomy development for the Mars Mobile Science Laboratory. *Proceedings of IEEE Aerospace Conference,* pp. 643–652, ISBN 0-7803-7651-X, Big Sky, Montana, USA, March 2003

Volpe, R. (2005). Rover technology development and mission infusion beyond MER. *Proceedings of IEEE Aerospace Conference*, pp. 971-981, ISBN 0-7803-8870-4, Big Sky, Montana, USA, March 2005

Wright, J., Hartman, F., Cooper, B., et al. (2006). Driving on Mars with RSVP: building safe and effective command sequences. *IEEE Robotics and Automation Magazine*, Vol. 13, No. 2, (March 2006), pp. 37-45, ISSN 070-9932

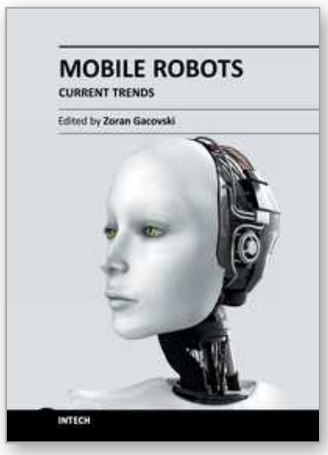
Yang, Y., Bao, J., Jin, Y., & Cheng, Y. (2008). A virtual simulation environment for lunar rover: framework and key technologies. *International Journal of Advanced Robotic Systems*, Vol. 5, No. 2, (June 2008), pp. 201-208, ISSN 1729-8806

Ye, P. & Xiao, F. (2006). Environment problem for lunar exploration engineering. *Spacecraft Environment Engineering*, Vol. 23, No. 1, (January 2006), pp. 1-11, ISSN 1673-1379

Yen, J., Jain, A., & Balaram, J. (1999). ROAMS: Rover Analysis Modeling and Simulation Software. *Proceedings of International Symposium on Artifcial Intelligence, Robotics and Automation in Space*, pp. 249-254, ESTEC, Noordwijk, the Netherlands, June 1999

Yoshida, K. (2000). The SpaceDyn: a MATLAB toolbox for space and mobile robots. *Journal of Robotics and Mechatronics*, Vol. 12, No. 4, (August, 2000), pp. 411–416, ISSN 0915-3934

Yoshida, K. & Hamano, H. (2002). Motion dynamics of a rover with slip-based traction model. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3155-3160, ISBN 0-7803-7272-7, Washington, D.C. USA, May 2002



Mobile Robots - Current Trends

Edited by Dr. Zoran Gacovski

ISBN 978-953-307-716-1
Hard cover, 402 pages
Publisher InTech
Published online 26, October, 2011
Published in print edition October, 2011

This book consists of 18 chapters divided in four sections: Robots for Educational Purposes, Health-Care and Medical Robots, Hardware - State of the Art, and Localization and Navigation. In the first section, there are four chapters covering autonomous mobile robot Emmy III, KCLBOT - mobile nonholonomic robot, and general overview of educational mobile robots. In the second section, the following themes are covered: walking support robots, control system for wheelchairs, leg-wheel mechanism as a mobile platform, micro mobile robot for abdominal use, and the influence of the robot size in the psychological treatment. In the third section, there are chapters about I2C bus system, vertical displacement service robots, quadruped robots - kinematics and dynamics model and Epi.q (hybrid) robots. Finally, in the last section, the following topics are covered: skid-steered vehicles, robotic exploration (new place recognition), omnidirectional mobile robots, ball-wheel mobile robots, and planetary wheeled mobile robots.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen