Full paper

Dynamic Simulation-Based Action Planner for a Reconfigurable Hybrid Leg–Wheel Planetary Exploration Rover

Eric Rohmer[a,*], **Giulio Reina**[b] **and Kazuya Yoshida**[a]

[a] Department of Aerospace Engineering, Tohoku University, Aoba 6-6-01, 980-8579 Sendai, Japan
[b] Department of Engineering for Innovation, University of Salento, Via Arnesano, 73100 Lecce, Italy

Received 17 September 2009; accepted 5 February 2010

Abstract
In this paper, an action planning algorithm is presented for a reconfigurable hybrid leg–wheel mobile robot. Hybrid leg–wheel robots have recently receiving growing interest from the space community to explore planets, as they offer a solution to improve speed and mobility on uneven terrain. One critical issue connected with them is the study of an appropriate strategy to define when to use one over the other locomotion mode, depending on the soil properties and topology. Although this step is crucial to reach the full hybrid mechanism's potential, little attention has been devoted to this topic. Given an elevation map of the environment, we developed an action planner that selects the appropriate locomotion mode along an optimal path toward a point of scientific interest. This tool is helpful for the space mission team to decide the next move of the robot during the exploration. First, a candidate path is generated based on topology and specifications' criteria functions. Then, switching actions are defined along this path based on the robot's performance in each motion mode. Finally, the path is rated based on the energy profile evaluated using a dynamic simulator. The proposed approach is applied to a concept prototype of a reconfigurable hybrid wheel–leg robot for planetary exploration through extensive simulations and real experiments.

Keywords
Action planning, reconfigurable robot, hybrid leg–wheel mechanism, planetary exploration

1. Introduction

Planetary exploration to collect scientific data in order to increase our knowledge of the universe or prior to an establishment requires efficient robot surface mobility. The surface of the Moon or a planet such as Mars is covered with fine-grained soil,

[*] To whom correspondence should be addressed. E-mail: eric@astro.mech.tohoku.ac.jp

DOI:10.1163/016918610X501499

sparse rocks and boulders that make this kind of environment highly challenging for autonomous rovers. Therefore, efficient and versatile locomotion systems and appropriate path planning are necessary to improve the degree of mobility, ensuring at the same time the safety of the robot. The future of lunar exploration, as an example, greatly depends on successfully prospecting for ice-water. Some local concentrations are expected in the Moon's polar craters that are in permanently shaded areas. For their exploration, one needs to take extra care of mobility as well as energy consumption as the terrain is expected to be rough and no sunlight is available to charge the batteries of the rover. Although most of the commonly used technologies to move on the surface of a planet use wheeled mobile robots, the space exploration community has been considering alternatives to improve robot mobility. One solution is to consider legged mechanisms as in Refs [1, 2]; however, as walking is a very energy-consuming locomotion mode, it is not appropriate for exploration. The ideal approach is, then, to use hybrid a wheel–leg mechanism (e.g., Refs [3–7]) in order to gain maximum terrain adaptivity with minimum power consumption. In this case, the robot drives on smooth terrain and walks when the soil conditions require it or uses boths locomotion modes at the same time, as in Ref. [6]. However, in order to fully reach the hybrid mechanism potential, it is necessary to establish an approach to optimize the course of action that a robot has to undertake to use one or the other locomotion mode, depending on the soil conditions and topology. A large body of research has been devoted to path/motion planning issues for leg or wheeled robots individually [8, 9]. Despite such intensive research, little attention has been devoted to defining an optimal path considering the switch of locomotion mode for hybrid mechanisms.

Here, we propose a solution that fills this gap for reconfigurable hybrid leg–wheel mechanisms. The action planner is composed of four steps: (i) path planning to generate a candidate path, (ii) action planning to generate a script of motion mode switching along the path, (iii) dynamic simulation of the robot's behavior along the path and (iv) path evaluation based on the dynamic simulation results. In order to validate this approach, it is integrated within a fully functional platform (hardware and software) that deals with the mobility and energy consumption constraints of the exploration of highly challenging terrains, such as those expected on the shaded lunar polar craters. The platform is composed of a prototype hybrid wheel–leg robot—LEON (Lunar Exploration Omnidirectional Netbot)—and a physics engine-based simulator/telerobotic platform—ERode. While LEON's mechanical design fits the requirement of mobility on highly challenging terrain, ERode offers a tool to simulate its behavior in terms of mobility and energy consumption.

This paper is structured as follows. Section 2 presents LEON—the prototype that embodies a novel concept of a reconfigurable hybrid leg–wheel mechanism and that will serve as an experimental test bed to validate the proposed action planner. Section 3 describes the models used to control LEON and to simulate its behavior within the simulator ERode. Next, Section 4 introduces the action planning algorithm, its criteria and the path evaluation method that provides the supervisor with the least energy-consuming yet feasible path toward a point of scientific interest (POSI). Section 5 describes real experiments and simulations to validate the proposed framework. Relevant conclusions are drawn in Section 6.

2. Reconfigurable Hybrid Wheel–Leg Robot LEON

Planetary rovers are expected to move either on deformable, soft terrain or rocky terrain. Legged robots are known to be efficient on uneven terrain, but they are energy consuming, challenging to control and have low speed. On the other hand, wheeled robots are faster and more energy efficient, but they need to move on smoother terrains. In order to overcome these drawbacks, preserving the advantages of both locomotion systems, hybrid leg–wheel robots represent a promising solution on highly challenging terrain as expected, for example, in the lunar polar craters.

Previous research on hybrid leg–wheel systems has proposed solutions either with small wheels at the leg's end tip [4, 7] or owning both locomotion systems, using them simultaneously (or alternatively) [6]. However, neither of those approaches is well suited to lunar exploration. End-tip wheels are usually reduced in size in order to ensure precise leg contact on the ground or have retraction ability, but they prevent mobility on soft terrain. On the other hand, keeping both independent mechanisms leads to actuator redundancy and bulkiness. NASA's Athlete [5] uses large end-tip wheels that can deal with soft soil, but the robot's size is rather large. None of those solutions offers a compact, legged robot with very large wheels. In contrast to these approaches, we propose LEON's novel design [10]. LEON is a hexapod that can fold two of its limbs to transform them into wheels. In that way, LEON turns into a large-wheeled robot, with a very compact size, as shown in Fig. 1. Its three-dimensional (3-D) structure is formed by a central body with a hexagonal shape and six limbs symmetrically distributed around the body. This allows a near omnidirectional motion and the ability to manipulate objects at the same time. The legs equipped with the necessary tools can also be used for simple manipulation tasks in cooperation with other similar robots, for sampling return, sensor positioning and surface processing like digging, scratching or piercing the soil even when in two-wheeled mode.

We recognize that LEON's idea is still a concept, and more research and development is necessary. However, we believe this architecture is feasible and has high potential. Figure 1b shows the first prototype embodying this design that we will use as an experimental test bed for experimental validation purposes in this work.

2.1. *Reconfigurable Hybrid Leg–Wheel Mechanisms*

In order to allow a walking and driving motion, we consider the use of four actuators: driving, hip/steer, knee and ankle per hybrid leg. In general, two types of arrangements along the kinematic chain from the body can be defined: drive, hips/steer, knee and ankle or hips, drive, knee and ankle (Fig. 2), defined as Type 1 and Type 2. While Type 2 allows only a differential drive-wheel control, Type 1 also

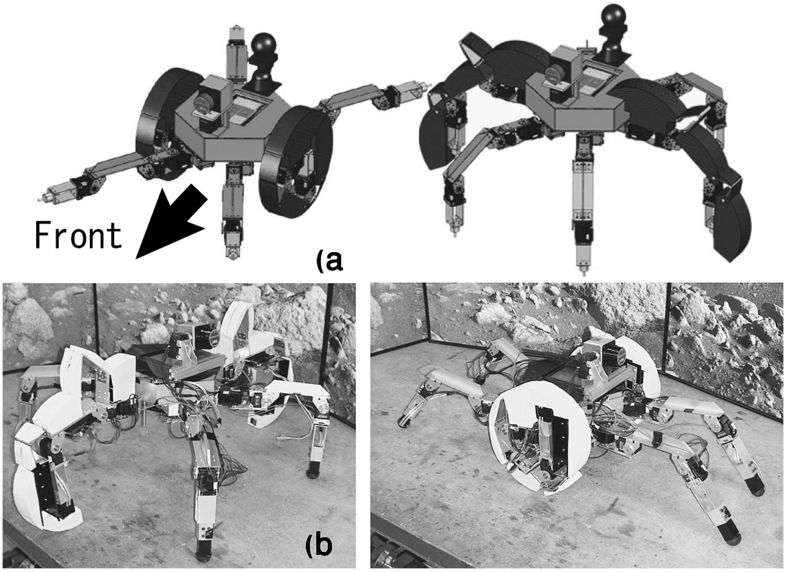


Figure 1. (a) CAD models of LEON and (b) current implementation.

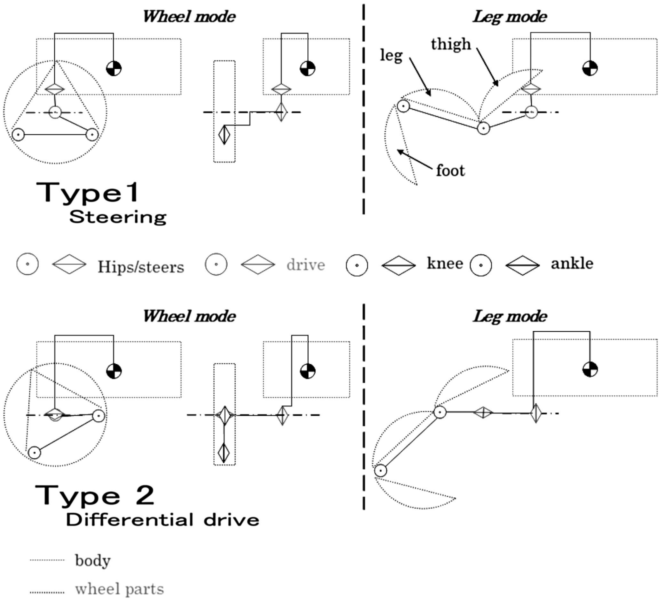


Figure 2. Kinematic arrangement of the actuators: Type 1 and Type 2.

ensures steering-wheel control of the robot's heading. For the design of LEON, we chose Type 2. This choice is motivated by two main reasons. (i) LEON's symmetry to the center of its body allows only the use of two wheels with a differential drive control. Hence, Type 1 steering ability is not required. (ii) As only two out of the six legs are hybrid, the choice of Type 2 over Type 1 allows us to keep the homogeneity of the design of the six legs, as the drive actuator is optionally linked to the body and not a part of the kinematic chain of the leg.

An additional issue that is addressed in the wheel–leg mechanism design is the wheel shape closing mechanism. If the actuators are not located on the perimeter of the wheel, when reconfiguring to leg, parts of the wheels on separate limbs collide as seen in Fig. 3. As a consequence, the wheel requires 'holes' in its shape to prevent this collision, to allow deployment the leg and to have an appropriate leg workspace. Several mechanism to close the holes in the wheel shape were considered and a solution based on foldable parts of wheels using a set of hinges was selected as seen in Fig. 3. This mechanism reduces the leg workspace, but offers a solid and complete wheel nearly everywhere along the transformed leg.

2.2. Transformation of Locomotion Mode

LEON is designed to transform from a walking mode to a wheel mode, where the locomotion system changes from legs to two large wheels. The hybrid legs fold in on themselves, curling dynamically to create a wheel. The hybrid wheel–leg's exterior have a rubber and foam shell, which when the legs are folded forms a round wheel capable of supporting LEON on rugged terrain. For a safe transition from six-legged walking mode to wheel mode the criteria of space and terrain bumpiness all need to be satisfied (as detailed later in Section 4.2). To initiate the switching phase between the respective modes, an area clear of obstacles is required for the transformation. The switching phase begins by creating a stable base and then the robot lifts its main body up away from the ground with the four non-hybrid legs (Fig. 4a). The legs are spread at a broad angle to increase the stability of the trans-

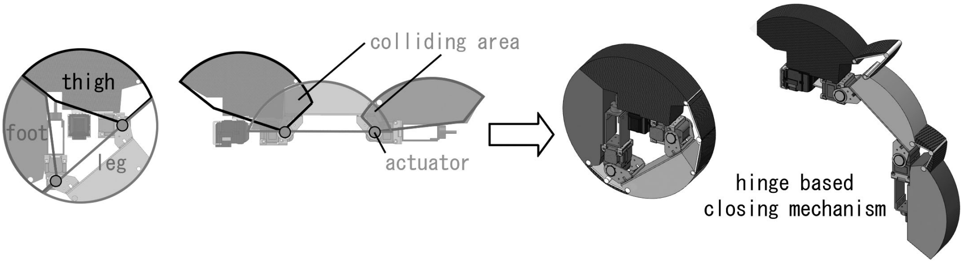


Figure 3. Wheel shape overlapping issue and hinge-based wheel-closing mechanism.

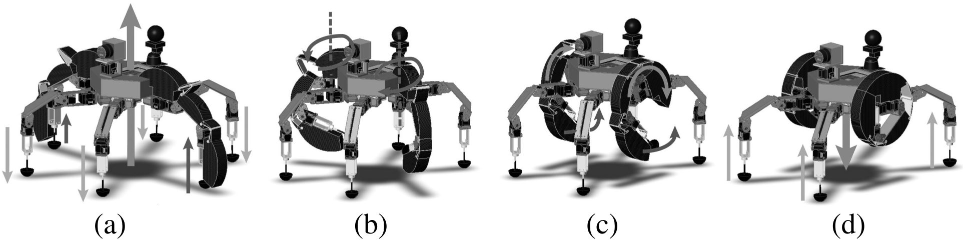


Figure 4. Four phases of LEON's transformation.

formation, whilst not interfering with the hybrid legs. The hybrid legs then lift off the ground and swivel into LEON's main body (Fig. 4b). This allows the hybrid legs to begin to rotate. As the drive actuators begins to rotate, the hybrid legs fold in on themselves, beginning to form the wheel (Fig. 4c). If the leg was simply to fold in and form a wheel, small pieces of debris could get caught in the mechanisms and reduce functionality. Finally, once the wheels are formed, the body is lowered to the surface by the four non-hybrid legs (Fig. 4d). The hybrid wheels are then able to take the load applied by LEON's weight and locomotion can start. In the case of transforming from wheel to leg mode, the reverse process is used.

3. Dynamic Simulator ERode

In order to simulate the dynamic behavior of LEON, we developed the physics-based simulator ERode. ERode is a simulation environment based on the ODE (Open Dynamics Engine) library [11], which allows easy creation of a virtual world, its visualization and runs real-time interactive simulations. It also features convenient functions to control the appearance for realistic or scientific rendering of the values of dynamic parameters (gravity, terrain properties, etc.). Specifically, the dynamic model comprises two modules—the vehicle dynamics model and the leg/wheel–soil interaction model. Both models are described in detail in the remainder of this section.

3.1. Robot Dynamics Model

The robot was modeled according to our hybrid prototype LEON (see Fig. 5). LEON's overall weight is about 6 kg with a wheelbase of 0.2 m and a wheel radius of 0.11 m in wheeled configuration, and a hexagonal convex hull, enclosing the contact points, with a major and minor axis length of 0.25 and 0.2 m, respectively, during legged locomotion. The height of the robot from the ground is equal to the wheel radius during wheeled mode and 0.15 m for the legged configuration.

Given a path defined over a digital elevation map augmented with a set of instructions to switch from one locomotion system to the other, the dynamic behavior

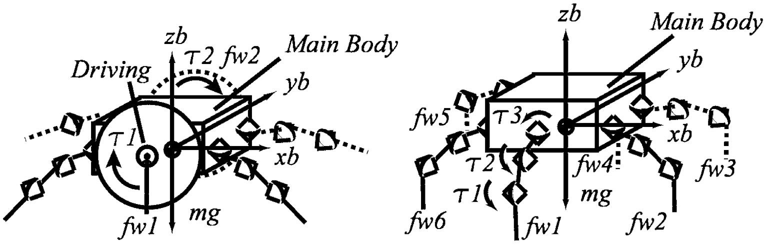


Figure 5. LEON's dynamic models.

of the robot is numerically obtained by successively solving the following motion equation:

$$H\begin{bmatrix}\dot{v}_0\\ \dot{\omega}_0\\ \ddot{q}\end{bmatrix}+C+G=\begin{bmatrix}F_0\\ N_0\\ \tau\end{bmatrix}+J^{\mathrm{T}}\begin{bmatrix}F_{\mathrm{e}}\\ N_{\mathrm{e}}\end{bmatrix}, \tag{1}$$

where H represents the inertia matrix of the robot, C is the velocity-dependent term, G is the gravity term, v_0 and ω_0 are the translational and angular velocity of the main body, q is the joint angle vector, F_0 and N_0 are the external forces and moments acting at the centroid of the rover, τ is the vector of the torques acting at each joint of the rover, J is the Jacobian matrix, $F_{\mathrm{e}} = [f_{\mathrm{w}1}, f_{\mathrm{w}2}, \ldots, f_{\mathrm{w}m}]$ is the vector of forces applied at the m contact points by the external environment, N_{e} is the counterpart of F_{e} for the moments. Note that each external contact force $f_{\mathrm{w}i}$ is derived by the leg/wheel–soil contact model described later. Equation (1) is a general equation and can be applied to a vehicle with any configuration.

The virtual robot can be considered dynamically equivalent to the real LEON prototype. Specific parameters of the robot kinematics and dynamics were experimentally determined to match the behavior of the simulated model with the real system as shown later in Section 5.

3.2. Wheel–Soil Contact Model

During wheeled locomotion, the drive wheel contact forces can be decomposed into a longitudinal component F_x, usually referred to as the drawbar pull, a lateral component F_y and a vertical component F_z. Based on terramechanics theory [12, 13], the contact force components components can be obtained as:

$$F_x = rb\int_{\theta_{\mathrm{r}}}^{\theta_{\mathrm{f}}}(\tau_x(\theta)\cos\theta - \sigma(\theta)\sin\theta)\,\mathrm{d}\theta \tag{2}$$

$$F_y = \int_{\theta_{\mathrm{r}}}^{\theta_{\mathrm{f}}}(r\cdot b\cdot\tau_y(\theta) + R_{\mathrm{b}}\cdot(r - h(\theta)\cos\theta))\,\mathrm{d}\theta \tag{3}$$

$$F_z = rb\int_{\theta_{\mathrm{r}}}^{\theta_{\mathrm{f}}}(\tau_x(\theta)\sin\theta + \sigma(\theta)\cos\theta)\,\mathrm{d}\theta, \tag{4}$$

where b represents the width of the wheel, r the wheel radius, $\sigma(\theta)$ the normal stress beneath the wheel, and $\tau_x(\theta)$ and $\tau_y(\theta)$ the shear stress along the longitudinal and lateral direction of the wheel. The contact region of the wheel on loose soil is determined by the entry angle θ_{f} and the exit angle θ_{r}. In addition, R_{b} is modeled as a reaction resistance generated by the bulldozing phenomenon on a side face of the wheel and it is expressed as a function of the wheel sinkage h.

3.3. Foot–Soil Contact Model

Similarly, in legged configuration the contact force components on each flat foot can be expressed as:

$$F_x = l_y \int_0^{l_x} \tau_x(x)\,\mathrm{d}x \tag{5}$$

$$F_y = l_x \int_0^{l_y} \tau_y(y)\,\mathrm{d}y \tag{6}$$

$$F_z = l_y \int_0^{l_x} \sigma(x)\,\mathrm{d}x, \tag{7}$$

where l_x and l_y are the longitudinal and lateral length of each foot. It should be noted that the leg/wheel–soil interaction model is replaced by a classical Coulomb friction model when the area is rocky, i.e., the terrain roughness index is large, as explained later in Section 4.1.

4. Action Planning

In this section, the proposed action planner is presented in detail. Given a digital map of the environment surrounding the robot and a point of interest, our goal is to define the optimal path toward the desired destination. We define as optimal the path that minimizes the energy consumption while ensuring at the same time the safety of the hybrid leg–wheel robot. One should note that hybrid leg–wheel robots are unique in that they feature alternative locomotion modalities. Therefore, their motion planning should also include the choice of the locomotion mode that best fits to a given portion of the path. In general, we will refer to action planning for hybrid leg–wheel robots rather than simply path planning. The various stages of the proposed approach are summarized in the flowchart in Fig. 6. During Steps 1 and 2,

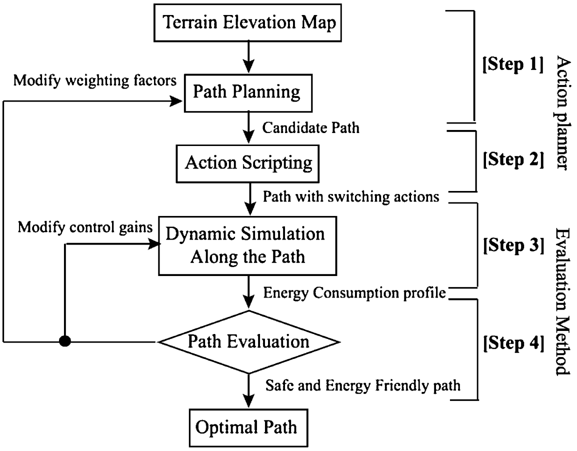


Figure 6. Flowchart of the action planning algorithm and evaluation method.

a candidate action is proposed, i.e., a path with an indication of the locomotion systems to be used along its different parts. In Steps 3 and 4, the dynamic behavior of the robot is checked along the proposed path using ERode and the candidate action is evaluated. The final output is the optimal action for the robot, that the mission manager and team selected in order to be safe, and/or less energy consuming, and/or faster depending on their needs. The path enhanced with the action script will be uploaded to the robot.

Specifically, in the first step, the path planning problem is addressed as an extended version of a search of the minimal path. A candidate path is obtained using Dijkstra's algorithm [15]. In the second step, a set of switching conditions of the locomotion system is assigned to the candidate path. When in wheel mode, the action planner seeks for a soil condition requiring a leg mode motion; when in leg mode, the planner follows several rules described in the next section to avoid getting back in wheel mode too often and to verify that the switching condition is possible. The third step is the dynamic simulation carried out to define the energy consumption profile. The virtual robot is controlled to follow the candidate path file augmented with the set of actions to be taken. The simulator switches wheel and leg model, and control thems accordingly to the script defined in the previous step. In the fourth step, the candidate path is evaluated based on the result of the dynamic simulation. The evaluation criteria are the total energy consumption of the robot to reach the final destination, the accuracy to follow the path and the elapsed time/total travel distance from the initial point to the target destination. We will now detail the first two stages.

4.1. Path Planning Algorithm

It is assumed that we have a perfect, i.e., without uncertainties, knowledge of the terrain map, represented by a digital elevation map (DEM). The DEM is defined as a series of elevations at a grid's node n_i in (x_i, y_i, z_i), where i is the node's index. Each node is, then, assigned with criteria indexes to build a cost map. An objective function can be defined based on those indexes and minimized to generate the candidate optimal path to a goal.

The objective function to find a candidate path is composed of three criteria indexes:

- *Terrain roughness index.* This index defines traversability over uneven terrain. The planner avoids defining the path over too rough zones that the robot would not be able to overcome even in leg mode. The terrain roughness index B_i is given as a standard deviation of the terrain elevation over a projection region of the robot R_i [16]:

$$B_i = \sqrt{\frac{1}{n}\sum_{R_i}(z(R_i) - \bar{z}(R_i))^2}. \tag{8}$$

As shown in Fig. 7, the projection region R_i includes the set of terrain elevation points inside the region surrounded by the end tips of the legs spread in

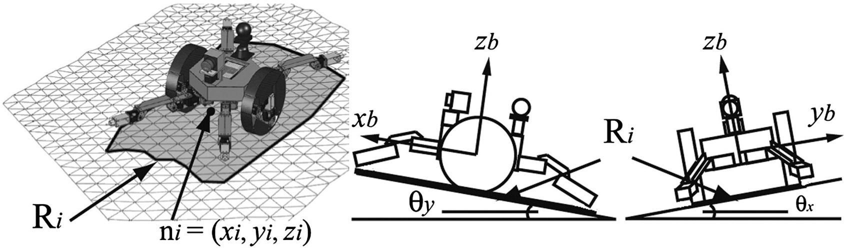


Figure 7. Projection region on the terrain and inclination angles in wheel mode.

wheel motion mode. In (8), n represents the node's number inside the region and $\bar{Z}(R_i)$ denotes an average elevation in R_i. The rougher the terrain is, the larger B_i becomes.

- *Path length index*. This index aims to define the shortest path from an initial point to a destination. The path length index L_i between adjacent nodes is calculated by:

$$L_i = |n_i - n_j| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}. \tag{9}$$

If the nodes n_i are not adjacent to the nodes n_j, L_i gets a large value.

- *Terrain inclination index*. When the robot is climbing up a hill or traversing a slope of a crater, the risks of slippage or tipping over increase. The index of terrain inclination aims to mitigate such risks. The terrain inclination angles are divided into two axis related to the robot body coordinates described in Fig. 7. An inclination angle around the x-axis of the robot coordinates is denoted by θ_x, while one on the y-axis is θ_y. The indexes Θ_{xi} and Θ_{yi} associated with each terrain inclination are, respectively, determined by the average inclination at the region R_i:

$$\Theta_{xi} = \bar{\theta}_x(R_i) \tag{10}$$

$$\Theta_{yi} = \bar{\theta}_y(R_i). \tag{11}$$

The above criteria indexes are weighted in order to define the cost function $C(p)$ to be minimized, that generates a candidate path p:

$$C(\mathbf{p}) = \sum_{i=\mathbf{p}} (W_B N_B B_i + W_L N_L L_i + W_{\theta_x} N_{\theta_x} \Theta_{xi} + W_{\theta_y} N_{\theta_y} \Theta_{yi}), \tag{12}$$

where W_B, W_L, W_{θ_x} and W_{θ_y} are the weighting factors to give specific priorities between the terrain roughness, path length and terrain inclinations. Note that W_{θ_x} or W_{θ_y}, respectively, take large enough values when the index Θ_{xi} or Θ_{yi} exceeds threshold angles $\theta_{x_{\max}}$ and $\theta_{y_{\max}}$. N_B, N_L, N_{θ_x} and N_{θ_y} are constants to normalize the corresponding indexes and eliminate the dimensions. The path p consists of a series of neighboring nodes, $p = \{n_{\text{start}}, \dots, n_i, \dots, n_{\text{goal}}\}$. In principle, a small index means that the robot is less affected by a given criterion. For example, the smaller W_B is, the less the planner will try to avoid rocks and boulders. Therefore, the smaller the sum of the weighted indexes is, the optimal, as less hazardous the path is, and supposing that the objective function is a hypothetical distance function, the path planning problem can be seen as a shortest path search. Considering that the minimum objective function derives the 'shortest' path p_{s}, the following equation can be defined:

$$\min C(p) = C(p_{\mathrm{s}}). \tag{13}$$

We are using a Dijkstra algorithm to derive p_{s}.

4.2. Action Scripting Algorithm

In this section the action scripting algorithm is presented along with its criteria to define the switching points along a given path. Initially, the robot is folded in the lander, then it transforms and starts in wheel mode. Alternatively, leg mode is chosen if the environment surrounding the lander is too rough. During the analysis of the candidate path, the algorithm tends to keep the wheel mode as long as possible, but it will look for a switching space as soon as the terrain condition is too rough for the wheels. The robot will return to wheel mode based on three switching condition indexes: terrain roughness, continuity, and safe switching space. However, the change will be planned only if the three criteria are satisfied at the same node:

- *Terrain roughness criterion* μ. The action planner seeks the soil conditions at each node of the trajectory along the candidate path, using (8). When the roughness condition is too high, the robot needs to switch or keep leg motion: $\forall B_i, i \in P = \{1, \ldots, m\}, B_i \geqslant \mu \Rightarrow$ leg mode, where m is the total number of nodes on the path, i its current node and P the set of all the nodes along the path.

- *Continuity criterion* λ. This index allows the planner to verify that switching from leg mode to wheel mode is worth the time and energy spent in the transformation. λ avoids too frequent transformations along the path: $\forall B_i, i \in S = \{j, \ldots, m\}, \exists l, B_l \geqslant \mu$, $(l < \lambda \vee l = m \Rightarrow$ leg mode$) \vee (l \geqslant \lambda \Rightarrow \mathit{wheelmode})$. l is either the index of the first node, since i requiring a switching to leg mode, or the last node m, and j the index of the last switching node from wheel to leg. λ can be seen as the minimal distance between j and the next switch from leg to wheel mode.

- *Safe switching space criterion* Δ. A safe region around the current robot position is necessary for a proper switching manoeuver, especially for reconfigurable hybrid leg–wheel robots like LEON. This criterion aims to fulfill this requirement by verifying that when a locomotion switch is planned at node i, $B(\Delta_i) \leqslant B_{\mathrm{transfo}}$ where $B(\Delta_i)$ is the roughness over the surface Δ at node i, and B_{transfo} is the highest acceptable roughness for a safe transformation. Figure 8 better clarifies this concept. LEON is at the given node m of a squared-grid with a pitch of $p = 0.16$ m. In the vicinity of m, each node inside the

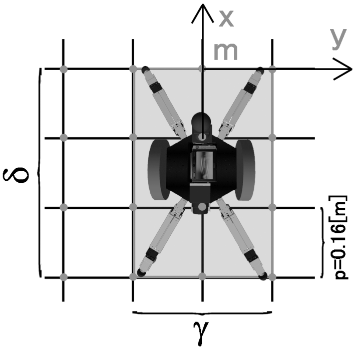


Figure 8. Safe switching area.

square shaded (green) surface must ensure low bumpiness. The safe switching area can be defined as $\Delta = \delta \times \gamma$, expressed in node surface units with $\delta = 3$ node surfaces along the x direction and $\gamma = 2$ node surfaces along y. If $B(\Delta_m) > B_{\text{transfo}}$, there is no space for safe transformation at node m.

In order to better explain the idea behind the action script algorithm, let L and T be two binary spaces representing, respectively, the so-called locomotion and transformation space along a given path P. For each node, space L is set 0 when the path is addressed in wheel mode and 1 when in leg mode. Similarly, space T assumes value 1 if a transformation is deemed necessary and value 0 otherwise. Initially, the algorithm checks along the path P where the robot needs to be in leg mode, according to the terrain roughness criterion. For each node i along P, if $P(i) \geqslant \mu$, $L(i) = 1$, else $L(i) = 0$.

To satisfy the safe switching space criterion on space T, all the nodes where a transformation can occur are set to 1: for each node i along P, if $P(\Delta_i) \geqslant B_{\text{transfo}}$, $T(i) = 1$, else $T(i) = 0$. Afterwards, the algorithm verifies that for each leg mode areas, there is a prior and a posterior transformation zone. If not, the leg zone will be extended until this condition is not satisfied. To do so, the scripting algorithm checks along $P(i)$ from $i = 0$ to $i = \textit{endnode}$ for rising and falling edges on space L. If a rising edge is found on $L(i)$, it searches backwards from $j = 0$ to $j = i$, until $T(i - j) = 1$. $T(i - j)$ is the closest transformation allowed spot prior to i along P. Then $L((i - j) \cdots i)$ are set 1. If a falling edge on $L(i)$ is detected, the algorithm checks forward on from $k = 0$ to $k = \textit{endnode}$ until $T(i + k) = 1$. $T(k)$ is the closest spot to i that allows the switch from leg to wheels. Then, $L(i \cdots (i + k))$ are set 1. Next, the validity of the continuity criteria is verified to avoid too frequent transformations. If the number of nodes between a falling edge and the next rising edge is less or equal than λ, the nodes at this interval are set 1 in space L. Finally, the algorithm detects along L each edge and records the associated node number into the action files that defines where to transform along P.

Figure 9 is a typical output of the planner along a 20-node path, where the values for the switching criteria are $\mu = 0.3$ m, $B_{\text{transfo}} = 0.3$ m, $\lambda = 3$ nodes and

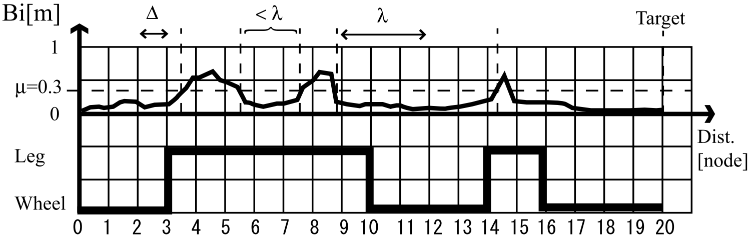


Figure 9. Example of generated profile.

$\Delta = (1, 1)$ node surface. The y-axis of the plot shows the terrain roughness index B_i along the given candidate path. The algorithm keeps the robot in wheel mode until node 3 where safe switching is possible and where legged locomotion is deemed necessary based on terrain unevenness. Nodes 6 and 7 would offer a more gentle ground, but legged mode is still retained since the continuity criteria is not met ($\lambda \leqslant l$). Wheeled locomotion is indeed preferred along the longer stretch comprised between nodes 10 and 14. A second locomotion switch is required when roughness increases again at nodes 14–16. Finally, the robot can reach the target in wheeled mode. The resulting action file records nodes 3, 10, 14 and 16 as required transformation spots.

5. System Validation

In this section, we first describe preliminary results aimed to assess the performance of the prototype LEON. These tests were useful to experimentally tune the simulator ERode to properly reflect the real behavior of LEON and to identify the parameters defined in the action planner. Then, the use of ERode allowed us to analyze the action planner through extensive simulations. In all experiments, the ground truth position of LEON was estimated by a stereo camera ceiling-mounted over the test bed. In the simulations, LEON's parameters were defined based on the actuator's specification as well as the CAD model dimensions and inertial parameters. The environment was modeled based on the 3-D data provided by a laser range finder mounted on the top of the scene (Fig. 10).

5.1. Performance on a Slope

In this set of experiments, LEON was driven in leg mode on an increased slope to estimate the maximum inclination it can handle without slippage on soft and hard soil. The simulation is run under the same conditions to verify the synchronization between the real world and the virtual one. In Fig. 11, the experimental setup and the simulation for the slope experiment are shown. When the limit of a servo's torque is reached (2.9 N·m), it shuts down and the motion cannot continue. Figure 12a and 12b shows the simulated and measured maximum torque recorded during a motion on a slope. We can observe that the experimental and simulated results show similar torques under the same slope constraints. Hence, ERode pro-

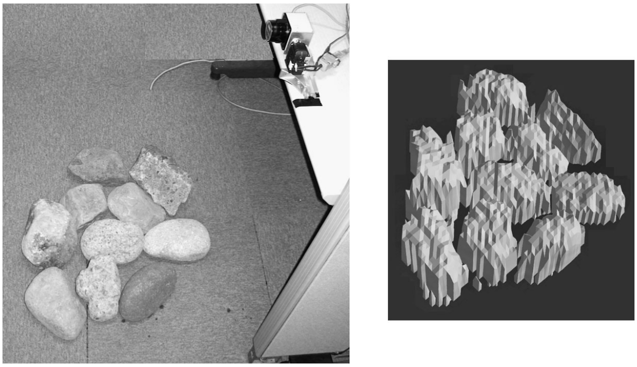


Figure 10. The 3-D reconstruction of the terrain from laser range finder data.

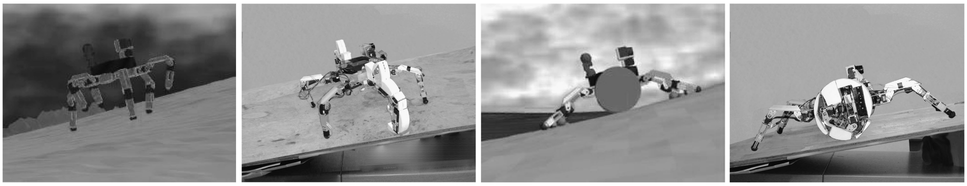


Figure 11. Experimental and simulated slope climbing.

vides an accurate simulation of LEON. (ii) We can define the terrain inclination indexes maxima $\theta_{x_{\max}}$ and $\theta_{y_{\max}}$ of (10) and (11) (see Section 4.1 for more details). From Fig. 12 we can see that the maximum slope LEON can walk on under Earth gravity is $\theta_y = 44°$ under both soft and hard soil conditions. In wheel mode, the maximum torque each of the drive servos provides is 6.4 N·m. Slippage occurs before the torque limit is reached in both cases of high grip (on rock) in Fig. 12c and low grip (on sand) in Fig. 12d with friction coefficients of 0.7 and 0.3, respectively. The maximum slope LEON can climb is then about $\theta_y = 22°$ on sand and $\theta_y = 35°$ on rock. The lowest angle is considered in the planner, for obvious safe reasons, so $\theta_y = 22°$.

An additional experiment was conducted over the slope to define the side tipping-over angle in order to define θ_x. In leg mode, the robot is symmetric to its center and no tipping over occurred before slippage. On the other hand, tipping over in wheel mode was empirically established at $\theta_x = 32°$.

5.2. Performance of LEON on Rough Terrain

We evaluate LEON's performance over rough terrain experimentally, using the index of bumpiness B of (8) to quantify the terrain roughness (see Fig. 13). The same experimental setup shown in Fig. 10 provides a DEM for ERode. The rougher the terrain is, the larger B becomes.

Rocks of different shape and size are spread along the path of LEON. The experiment is considered successful if LEON, traveling straight, can overcome the 1.2 m

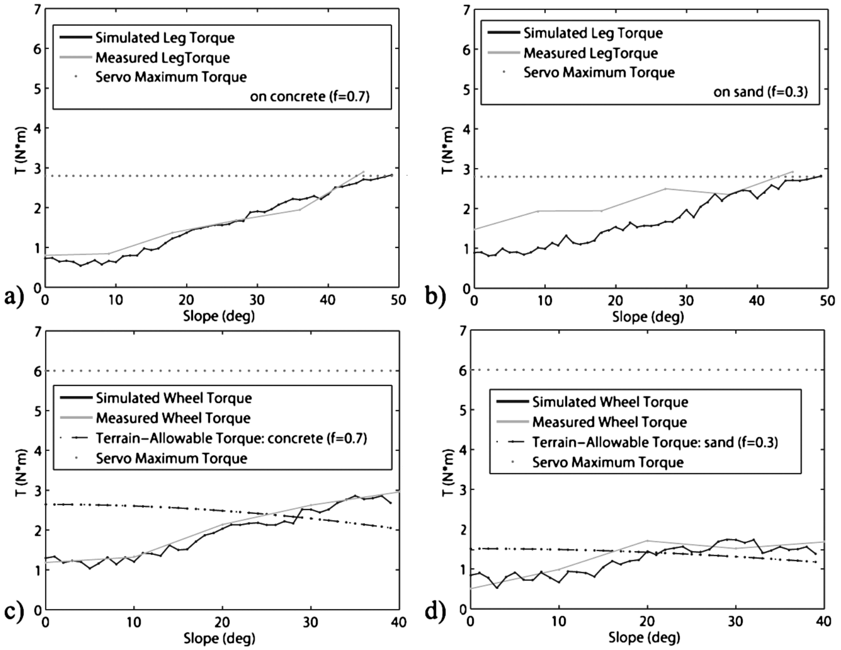
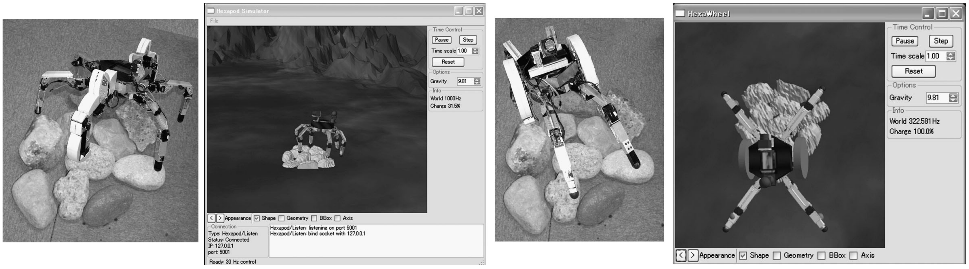


Figure 12. Performance of motion on a slope in leg and wheel mode.



Time scale 1.00
Gravity 9.81
World 1000Hz
Charge 31.5%
☑ Shape □ Geometry □ BBox □ Axis
Type: Hexapod/Listen
Status: Connected
IP: 127.0.0.1
port: 5001

```
Hexapod/Listen: listening on port 5001
Hexapod/Listen: bind socket with 127.0.0.1
```

Ready: 30 Hz control

Time scale 1.00
Gravity 9.81
World 322.581 Hz
Charge 100.0%
☑ Shape □ Geometry □ BBox □ Axis

Figure 13. Experimental and simulation over a rough terrain.

goal ahead of it. We used three sets of terrain with an increased roughness for this experiment: low roughness $B < 0.04$ m, middle roughness $0.04 < B < 0.06$ m, and high roughness $B > 0.06$ m. For each terrain, stones of about the same caliber were homogeneously distributed on the surface to be scanned. The first set of stones has a diameter of 0.04 m with $B = 0.03$ m. The second one has an average of size of 0.092 m with $B = 0.08$ m. The last set of stones has an average of 0.13 m and $B = 0.1$ m.

Figure 14a and 14c shows the results of the experiment in leg and wheel mode, respectively. Figure 14b and 14d demonstrates the efficiency of the simulator by comparing the simulated results to the experimental results. The simulation is based on the DEM, so the dynamic of the reconstructed stones is not taken into account.

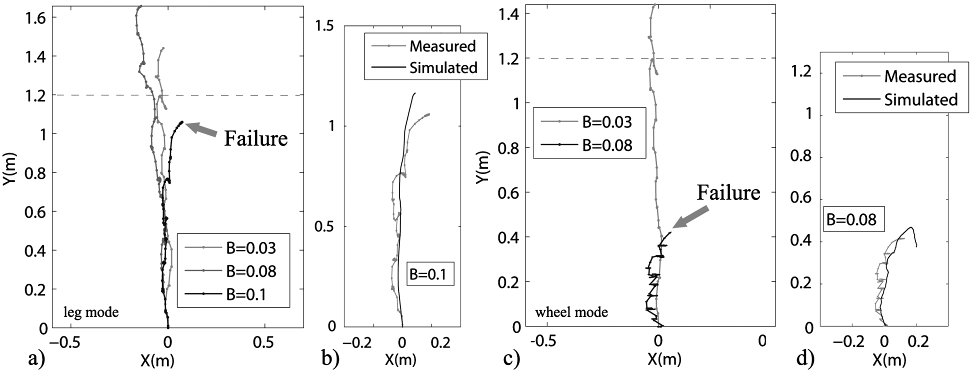


Figure 14. Measured and simulated leg and wheel motions on rough soil.

The difference between the simulated and experimental results can be explained when considering that, as LEON drives over stones in the virtual environment, the stones might be moved due to the traction force the wheels or the weight of the robot in leg mode applied on them.

As expected, LEON leg mode is more able to cross rough terrains than wheel mode. The roughest terrain LEON could cross in leg mode is for $B = 0.08$ m, whereas it failed at $B = 0.1$ m. In wheel mode, LEON's simulation could not overcome the rough section with a roughness index $B = 0.08$ m. Finally, we can quantify the terrain roughness criteria μ that, if over passed on a given node, tells the action script to switch to leg mode. We can define $\mu = 0.03$ m to be the triggering roughness to have the action scripting algorithm constraining LEON to walk.

5.3. Case Study

Here, results of a case study obtained using the proposed action planner are presented. Since LEON's position estimation system is not fully developed, we rely on the ERode simulator. The testing environment was given in the form of a 8 m × 8 m square DEM with a grid of 50 × 50 equally spaced terrain nodes, as shown in Fig. 15. Two candidate paths generated by the action planner along with the correspondent actions are depicted as large colored dots overlapped over the grid nodes. They were obtained using different values of the weighting factors, as summarized in Table 1. Path B is shown by bright, yellow dots in Fig. 15, whereas Path A is shown by dark, red dots for the portion where LEON planned motion is in wheeled mode and by green dots where it is planned to walk. Details of the length and inclination of the two paths are also collected in Table 1. The weight indexes of Path A are equal, and the result is the shortest traveling distance between the start and destination, but at a cost of traversal of a rocky area, along which LEON is set by the planner to switch into leg mode. The choice for Path B was connected with the search for the safest path, along which the robot was able to traverse using only its wheeled locomotion mode. To generate Path B, the roughness index was set very high in comparison to the length index.

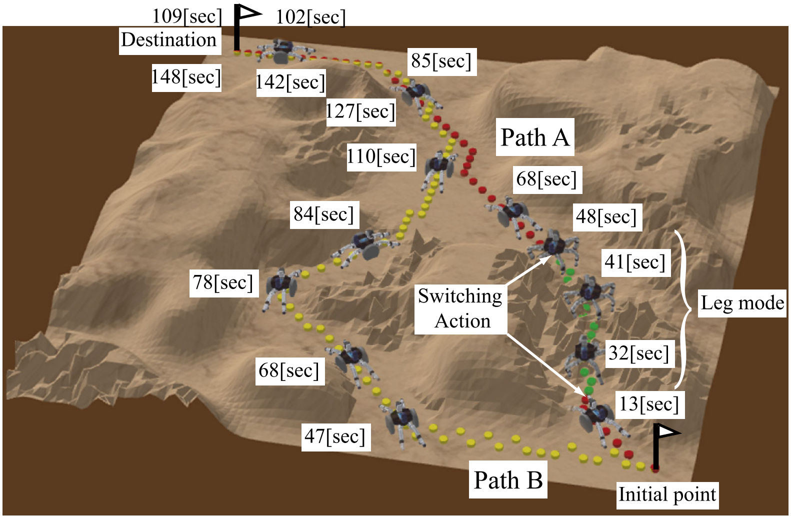


Figure 15. Illustration of the path evaluation on ERode. This figure is published in colour in the online version of this journal, that can be accessed *via* http://www.brill.nl/ar

Table 1.
Simulation setup and results

Path	W_B	W_D	W_{θ_x}	W_{θ_y}	Total length (m)	Elapsed time (s)	U (N·m)
A	0.1	0.1	0.1	0.1	11.65	109	20.9
B	0.8	0.01	0.2	0.2	16.23	148	35.4

In order to compare Path A with Path B in terms of travel safety, energy efficiency, total driving distance and travel time from the starting point to the final destination, the dynamic behavior of LEON was simulated along each path using the simulator ERode. Specifically, a global power consumption parameter U is defined as $U = \sum_{t_s}^{t_f} \Delta\bar{\Gamma}_{t_i}$, where t_s and t_f are the starting and final time along the path, and $\Delta\bar{\Gamma}t_i$ is the average energy consumption during a time iteration. $\Delta\bar{\Gamma}t_i$ is defined as $\Delta\bar{\Gamma}_{t_i} = \frac{1}{n}\sum_{j=1}^{n} \tau_j$, where τ_j is the torque at actuator j and n the robot's degree of freedom ($n = 20$: 18 leg actuators and two wheel motors). Since torque is roughly proportional to the electrical current in DC motors, we regard U as representative of the robot's energy consumption along the path.

The time profile of the energy consumption parameter U is shown in Fig. 16, and the resulting total traveling time and energy consumption are collected in Table 1. Path B is a more energy-consuming path, even if Path A had to deal with a walking mode sequence. The torque profile shows LEON in leg mode from time $t \approx 21$ to $t \approx 41$ in Path A. In addition, the energy profile for Path A is pretty constant, proving that the terrain was relatively flat. On the other hand, for Path B, we clearly see

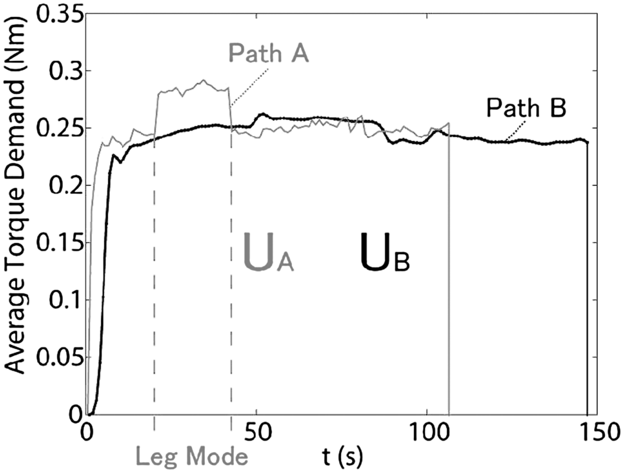


Figure 16. Energy consumption along Path A and Path B.

an increase of the average energy consumption over the time from $t \approx 47$. The elevation map confirms that a slope is the reason. Afterwards, the tendency is inverted at $t \approx 84$ with an average torque lower than on a flat ground, due to the presence of a descending slope. In conclusion, Path A is the most appropriate choice as it consumes less energy and is faster than Path B to reach the POSI, and it is chosen as the optimal path. This also proves the advantage of using reconfigurable robots, which are able to adapt their locomotion system to the terrain.

6. Conclusions and Future Work

In this paper, we addressed an action planner and its evaluation method for lunar/planetary exploration reconfigurable hybrid leg–wheel robots, considering the planning of switching locomotion mode. The proposed technique is composed of four steps: (i) path planning to generate a candidate path, (ii) action planning to generate a script of switching of motion mode along the path, (iii) dynamic simulation in which the robot drives along the path and (iv) path evaluation based on the dynamic simulation results.

The action planner was validated within a fully functional platform comprising a reconfigurable hybrid leg–wheel prototype and a dynamic simulator. LEON was an embodiment of a new concept of a reconfigurable robot that folds two of its six legs to transform them into wheels. ERode accurately simulated the robot's behavior in highly challenging scenarios. Some preliminary results are presented about the performance of the action planner and the robot, showing the feasibility of the proposed approach. More investigation may be conducted on the implementation of different soil characteristics in the planner, to have more accurate simulation results, as well as an automatic choice of the weighting factors for the cost function, based on the soil condition. For an onboard implementation of the planner the path searching method shall be faster as well as an *in situ* measurement system of the soil characteristics will be required to be implemented.

References

1. B. Kennedy, H. Aghazarian, Y. Cheng, M. Garrett, G. Hickey, T. L. Huntsberger, L. Magnone, C. Mahoney, A. Meyer and J. Knight, LEMUR: legged excursion mechanical utility rover, *Autonomous Robots* **11**, 201–205 (2001).
2. T. Takubo, T. Arai, K. Inoue, H. Ochi, T. Konishi, T. Tsurutani, Y. Hayashibara and E. Koyanaga, Integrated limb mechanism robot ASTERISK, *J. Robotics Mechatoron.* **18**, 203–214 (2006).
3. T. Oomichi and T. Ibe, Development of vehicles with legs and wheels, *Adv. Robotics* **1**, 343–356 (1986).
4. G. Endo and S. Hirose, Leg–wheel hybrid walking vehicle (Roller–Walker), *Adv. Robotics* **13**, 241–242 (1999).
5. K. Hauser, T. Bretl, J.-C. Latombe and B. Wilcox, Motion planning for a six-legged lunar robot, in: *Proc. Workshop on Algorithmic Foundations of Robotics*, New York, pp. 1–6 (2006).
6. S. Nakajima, T. Takahashi and E. Nakano, Motion control technique for practical use of a leg–wheel robot on unknown outdoor rough terrains, in: *Proc. Int. Conf. on Intelligent Robots and Systems*, Sendai, vol. 2, pp. 1353–1358 (2004).
7. A. Rovetta, E. C. Paul, D. Fregonara, D. Salvarini, X. Ding and Z. Wang, NOROS, a colony of robots, with a hybrid system of locomotion, in: *Proc. 9th ESA Workshop on Advanced Space Technology for Robotics and Automation*, Noordwijk, pp. 1–6 (2006).
8. A. De Luca, G. Oriolo and C. Samson, *Robot Motion Planning and Control.* Springer, London (1998).
9. T. Kubota, Y. Kuroda, Y. Kunii and T. Yoshimitsu, Path planning for newly developed microrover, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Seoul, pp. 3710–3715 (2001).
10. E. Rohmer, G. Reina and K. Yoshida, A novel teleoperated hybrid wheel–limbed hexapod for the exploration of lunar challenging terrains, in: *Proc. Int. Symp. on Space Technology and Sciences*, Hamamatsu, pp. 3902–3907 (2008).
11. R. Smith, Open Dynamics Engine, www.ode.org (2007).
12. G. Ishigami and K. Yoshida, Steering characteristics of an exploration rover on loose soil based on all-wheel dynamics model, in: *Proc. Int. Conf. on Intelligent Robots and Systems*, Barcelona, pp. 2041–2046 (2005).
13. J. Y. Wong, *Theory of Ground Vehicles.* Wiley, New York, NY (1978).
14. G. Ishigami, K. Nagatani and K. Yoshida, Path planning for planetary exploration rovers and its evaluation based on wheel slip dynamics, in: *Proc. Int. Conf. on Robotics and Automation*, Rome, pp. 2361–2366 (2007).
15. S. LaValle, *Planning Algorithms.* Cambridge University Press, Cambridge (2006).
16. K. Iagnemma and S. Dubowsky, *Mobile Robots on Rough Terrain (Springer Tracts in Adv. Robotics 12).* Springer, Berlin (2004).

About the Authors

Eric Rohmer received his Master degree at Ecole Superieure d'Informatique et Applications de Lorraine, in 2000, in Nancy, France. After working in industry as a Mechanical Engineer, he was awarded a Doctoral scholarship to study in Japan by the Japanese Ministry of Education. He received his Research Doctorate degree in Computer Science in the Advance Robotics Laboratory of Tohoku University, Sendai, Japan, in 2005. He then worked as a Research Fellow in the Space Robotics Laboratory of Tohoku University, where was awarded, in 2006, a Japanese Society for Promotion of Science (JSPS) fellowship for a 2-year Postdoctoral research. Currently, he is working as a Research Fellow in the Human–Robot Informatics Laboratory of Tohoku University. His research interests include planetary exploration and rescue robots, mobility on rough terrain, and reconfigurable mobile robots and their control.

Giulio Reina received the Laurea degree and the Research Doctorate degree from the Politecnico of Bari, Italy, in 2000 and 2004, respectively, both in Mechanical Engineering. From 2002 to 2003, he worked at the University of Michigan Mobile Robotics Laboratory as a Visiting Scholar. In 2007, he was awarded a Japanese Society for Promotion of Science (JSPS) fellowship for a 1-year research at the Space Robotics Laboratory of Tohoku University, Sendai, Japan. Currently, he is an Assistant Professor in Applied Mechanics with the Department of Engineering for Innovation of the University of Salento, Lecce, Italy. His research interests include planetary rovers, mobility and localization on rough terrain, computer vision applied to robotics, and agricultural robotics.

Kazuya Yoshida received the BE and MS degrees in Mechanical Engineering Science from Tokyo Institute of Technology, Japan, in 1984 and 1986, respectively. He received the DE from Tokyo Institute of Technology, in 1990. He served as a Research Associate at Tokyo Institute of Technology, from 1986 to 1994, and a Visiting Scientist at Massachusetts Institute of Technology, USA, in 1994. From 1995 to 2003, he was appointed Associate Professor of Tohoku University, Japan and, since 2003, as a Professor in the Department of Aerospace Engineering, Tohoku University. He also serves as a Visiting Lecturer of the International Space University, since 1998. His research activities include dynamics and control issues of variety types of space robots ranging from free-flying robots to planetary exploration rovers. Recently, his activities have been extended to terrestrial applications of space technology, such as telerobotics for disaster mitigation missions.