

Research Article

Fault-Tolerant Control Strategy for Steering Failures in Wheeled Planetary Rovers

Alexandre Carvalho Leite,¹ Bernd Schäfer,² and Marcelo Lopes de Oliveira e Souza³

¹*Institute of System Dynamics and Control, German Aerospace Center (DLR), 82205 Weßling, Germany*

²*Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82205 Weßling, Germany*

³*Space Mechanics and Control Division, National Institute for Space Research (INPE), 12227-010 São José dos Campos, SP, Brazil*

Correspondence should be addressed to Alexandre Carvalho Leite, alexandre.carvalholeite@dlr.de

Received 1 June 2012; Accepted 27 November 2012

Academic Editor: Kazuya Yoshida

Copyright © 2012 Alexandre Carvalho Leite et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Fault-tolerant control design of wheeled planetary rovers is described. This paper covers all steps of the design process, from modeling/simulation to experimentation. A simplified contact model is used with a multibody simulation model and tuned to fit the experimental data. The nominal mode controller is designed to be stable and has its parameters optimized to improve tracking performance and cope with physical boundaries and actuator saturations. This controller was implemented in the real rover and validated experimentally. An impact analysis defines the repertory of faults to be handled. Failures in steering joints are chosen as fault modes; they combined six fault modes and a total of 63 possible configurations of these faults. The fault-tolerant controller is designed as a two-step procedure to provide alternative steering and reuse the nominal controller in a way that resembles a crab-like driving mode. Three fault modes are injected (one, two, and three failed steering joints) in the real rover to evaluate the response of the nonreconfigured and reconfigured control systems in face of these faults. The experimental results justify our proposed fault-tolerant controller very satisfactorily. Additional concluding comments and an outlook summarize the lessons learned during the whole design process and foresee the next steps of the research.

1. Introduction

Like all engineering systems, planetary exploration rovers (PERs) are also subjected to faults. Every component is subjected to anomalous behavior (fault) or permanent/intermittent complete nonfunctioning (failure). From the top level (PER's point of view), component faults or failures can be seen as internal disturbances in the nominal behavior of the rover and then called rover faults. The peculiarity with PERs is that maintenance cannot be employed; parts cannot be replaced or repaired. This characteristic is common in space systems and typically approached by control reconfiguration; the Ørsted Satellite is a nice application example of the fault-tolerant control on a space vehicle [1]. Reconfiguration schemes used to achieve fault tolerance in dynamic systems were recently well documented in textbooks [2–4] having the major part of the results applicable to linear systems. Zhang and Jiang [5] classify a sufficiently

representative sample of what has been done in the field of the fault-tolerant control.

As practice has been shown in the case of the Spirit rover [7], one of the twin of Mars Exploration Rovers of NASA, faults happen either because of unknown factors or environmental interactions. In [7] the authors reported ad hoc operation strategies to tolerate failures in motors and driving limitations due to performance degradation imposed by the fault mode. A premise on which this work is based is exactly performance degradation in face of the system fault modes. A fault-tolerant control strategy is just an attempt to use alternative signal paths to make the degraded performance as close as possible to the nominal performance.

A planetary exploration rover is a class of wheeled mobile robots devoted to drive in rough terrains in the presence of obstacles and sandy environment with slopes. Fault detection and diagnosis (FDD) and fault-tolerant control (FTC) topics

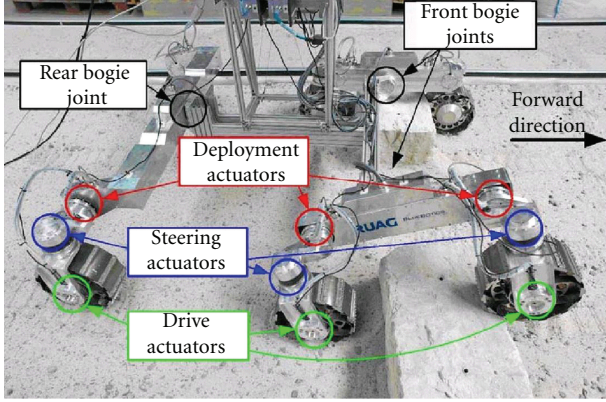


FIGURE 1: ExoMars rover in DLR's PEL (Planetary Exploration Laboratory) [6].

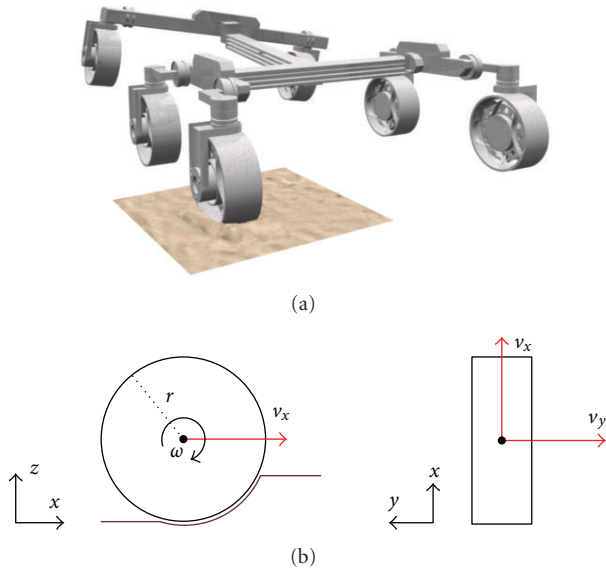


FIGURE 2: MBS model and contact model illustrated (a). Velocity vectors diagram in the coordinate system of a wheel (b).

for such vehicles were discussed in a survey; see [8]. However, their approach is strongly oriented to sensor fusion techniques and faults in sensors, not actuator faults like those experienced in [7]. Sensors, actuators, controller, and communication faults were treated in [9] while designing and testing a fault-tolerant controller for a wheeled mobile robot dealing with agricultural environments. It should be mentioned that they considered a kind of fault which is also a focus in our fault repertory: stalled steering joint. In this paper indeed, a deeper analysis and handling is provided.

Our application case is the ExoMars rover (see Figure 1). All modeling, simulations, impact analysis, control design, and experiments aimed to design and implement a functional fault-tolerant controller for the ExoMars rover. Note that we focus on control; fault detection and diagnosis are not addressed here. First, a simple contact model is described in Section 2. This is the basis to design the control

system for the nominal operating mode in Section 3. The impact analysis of steering faults is treated in Section 4 based on terramechanics modeling. The output of Section 4 is the fault repertory and an insight about the complexity of the problem to be treated. Thus, Sections 5 and 6 effectively describe the fault-tolerant control strategy and its experimental validation. Conclusions and outlook are given in the last section as we summarize the lessons learned during the whole design process and foresee the next steps of the research.

2. Modeling, Simulation, and Experimental Validation of the ExoMars Rover

Most of the effort to model PERs is normally concentrated in wheel-soil contact modeling; see [10–12]. However, our testbed [6] is filled with a kind of sand which does not allow extreme sinkage depths during simple path-following maneuvers on a plane. In this case, a simpler contact model can be used to simulate the maneuvers of the ExoMars rover driving on a plane. One advantage of this model is computation time; it is faster than a detailed contact model because complex numerical procedures are not necessary to compute all contact forces involved and deformed terrain interaction. The disadvantage is that it requires several tests, simulations, and optimization to tune the model parameters accordingly. Figure 2(a) illustrates the suspension of the ExoMars rover modeled as a multibody simulation (MBS) model with a wheel-soil contact representing vehicle-environment interaction. The software package Dymola was used to construct the MBS model and implement the simplified contact model. Figure 2(b), is a diagram with the coordinate system and velocity vectors in the wheel necessary to define the independent variables of the simplified contact model.

Slip ratio and slip angle are the main independent variables used in our simplified contact model. These variables are defined based on the velocity vectors of Figure 2(b). Slip ratio s is a function of longitudinal velocity v_x , wheel radius r , and angular velocity ω . Slip angle β is a function of lateral velocity v_y and longitudinal velocity. They are defined as follows:

$$s = \begin{cases} \frac{r\omega - v_x}{r\omega}, & |r\omega| > |v_x|, \\ \frac{r\omega - v_x}{v_x}, & |r\omega| < |v_x|, \end{cases} \quad (1)$$

$$\beta = \arctan \frac{v_y}{v_x}.$$

Forces in longitudinal direction F_{LO} , lateral direction F_{LA} , and reaction torque about the rotating axis τ_y are sufficient to describe the wheel-soil interaction:

$$F_{LO} = \left[(a_0 + za_1) (1 - e^{-|s|c_1}) \right] \text{sign}(s) - c_2 v_x,$$

$$F_{LA} = \left[(a_0 + za_1) (1 - e^{-|\beta|d_1}) \right] \text{sign}(\beta), \quad (2)$$

$$\tau_y = \left[b_2 (a_0 + za_1) (1 - e^{-|s|b_1}) \right] \text{sign}(s).$$

The independent variables are s , β , and z (sinkage depth). The parameters to be estimated are $a_0, a_1, b_1, b_2, c_1, c_2$, and d_1 . Experiments were performed with the ExoMars rover driving straight ahead and several arc lengths under the Ackermann steering configuration. The maneuvers were repeated setting new nominal angular velocities to the wheels; a total of 26 experiments were performed. The agreement between simulation and all experiments is aimed by changing the parameters' set to reduce error between the simulated and measured displacement variables, that is, heading angle and translational displacement in the plane. Reaction torque can be measured and is used as a constraint in the estimation process to avoid pure signal modeling. Sinkage was modeled as dependent on slip: $z = 0.01|s|$. Note that this simplified model is based on the well-known "magic formula," commonly used in the automotive industry [13]. In the present case, the amplitude of the contact forces is not only dependent on the nominal wheel load (expressed by parameter a_0), but also on the penetration of the wheel into the soil (sinkage). It introduces a coupling between amplitude and transient behavior in slip domain. The force in the lateral direction and the reaction torque are just environmental reactions produced during the movement of the vehicle; these quantities prevent side slip in cross-hill drives and provide needed driving torque, respectively. The longitudinal force propels the vehicle constantly balanced by the motion resistance term $c_2 v_x$. These force and torque models are sufficient to simulate a vehicle driving on a plane without obstacles, since there is no strong variation of the nominal normal force action on each wheel.

Experience showed that this model is suitable when simulation results are compared with the tracks generated by the rover. Hence, an in-house developed optimization tool (multiobjective parameter synthesis—MOPS) from DLR was used to solve the optimization problem stated as

$$\begin{aligned} & \underset{\mathbf{p}}{\text{minimize}} && \sqrt{\mathbf{x}_e^T \mathbf{x}_e + \mathbf{y}_e^T \mathbf{y}_e + \psi_e^T \psi_e} \\ & \text{subject to} && \omega_s = \omega_m, \\ & && \delta_s = \delta_m, \\ & && \tau_m^{\text{MIN}} \leq \tau_s \leq \tau_m^{\text{MAX}}, \end{aligned} \quad (3)$$

where \mathbf{x}_e , \mathbf{y}_e , and ψ_e are error vectors between simulated and measured longitudinal, lateral, and heading angle displacements, respectively. The constraints guarantee that all simulated driving velocities ω_s and steering angles δ_s will be the same as the measured ones (ω_m and δ_m). The smallest and higher torque measurements of each experiment are used as upper and lower bounds to constrain the vector of simulated reaction torques τ_s in a feasible range. Figure 3 shows correlations between tests and simulations of two maneuvers after parameter estimation considering all 26 experiments.

The tuned model is considered sufficiently robust for our purposes and is used to verify the proposed control system before implementation in the real hardware. The dynamic

model here is used for verification and validation of the fault-tolerant control, while just kinematics is used to design the actual control system.

3. Control of the Nominal System

We use the tuned simulation model in the design process of the nominal path-following controller. No definition of fault modes is necessary in this phase because all components are considered to work perfectly. It is also assumed that navigation by waypoints is a reasonable way to pass reference trajectory to a PER. This means that trajectories abruptly changing in a range less than 10 m are not reasonable for a vehicle like the ExoMars rover, designed to rotate its driving wheels not faster than 3 RPM. Hence, we consider a straight path Γ in the plane of motion as illustrated in Figure 4.

The theoretic results of this chapter were mostly extracted from [14, 15]; the part regarding general path-following control can also be found in [16]. The authors of this paper just introduced the assumption of a straight path, performed optimization to find the control parameters, and tested experimentally the controller in a six-wheeled rover.

3.1. Kinematic Rover Model and Path-Following Control. Adopting the unicycle case, the Frénet frame representation in Figure 4 is reduced to the following equations:

$$\begin{aligned} \dot{h} &= v_0 \cos \theta_e, \\ \dot{l}_e &= v_0 \sin \theta_e, \\ \dot{\theta}_e &= \omega_0, \end{aligned} \quad (4)$$

where v_0 is the nonzero longitudinal velocity of the vehicle, $\theta_e = \theta_0 - \theta_h$ is the attitude of the vehicle with respect to the path, and ω_0 is the angular velocity of the vehicle. The straight line Γ is formed by the waypoints w_1 and w_2 to make the path where the inclined abscissa h at the point P_h is obtained by orthogonal projection of P on Γ . The objective of the path-following controller is to force $l_e \rightarrow 0$ and $\theta_e \rightarrow 0$ driving and steering the wheels. But note that the kinematic model has just v_0 and ω_0 as input variables. Thus, a higher level control system is designed to meet the path following objectives. This is possible by first constructing the Lyapunov function:

$$V = \frac{1}{2} (l_e^2 + \theta_e^2). \quad (5)$$

The correspondent derivative is

$$\begin{aligned} \dot{V} &= \dot{\theta}_e \theta_e + \dot{l}_e l_e \\ &= \theta_e \underbrace{\omega_0}_{\text{control}} + l_e v_0 \sin \theta_e. \end{aligned} \quad (6)$$

A suitable control input ω_0 should guarantee $\dot{V} < 0$, which can be accomplished by a constant nonzero velocity v_0 and

$$\omega_0 = - \left(k_1 |v_0| \theta_e + k_2 |v_0| \dot{\theta}_e + k_3 v_0 l_e \sin \theta_e \frac{1 + k_2 |v_0|}{\theta_e} \right). \quad (7)$$

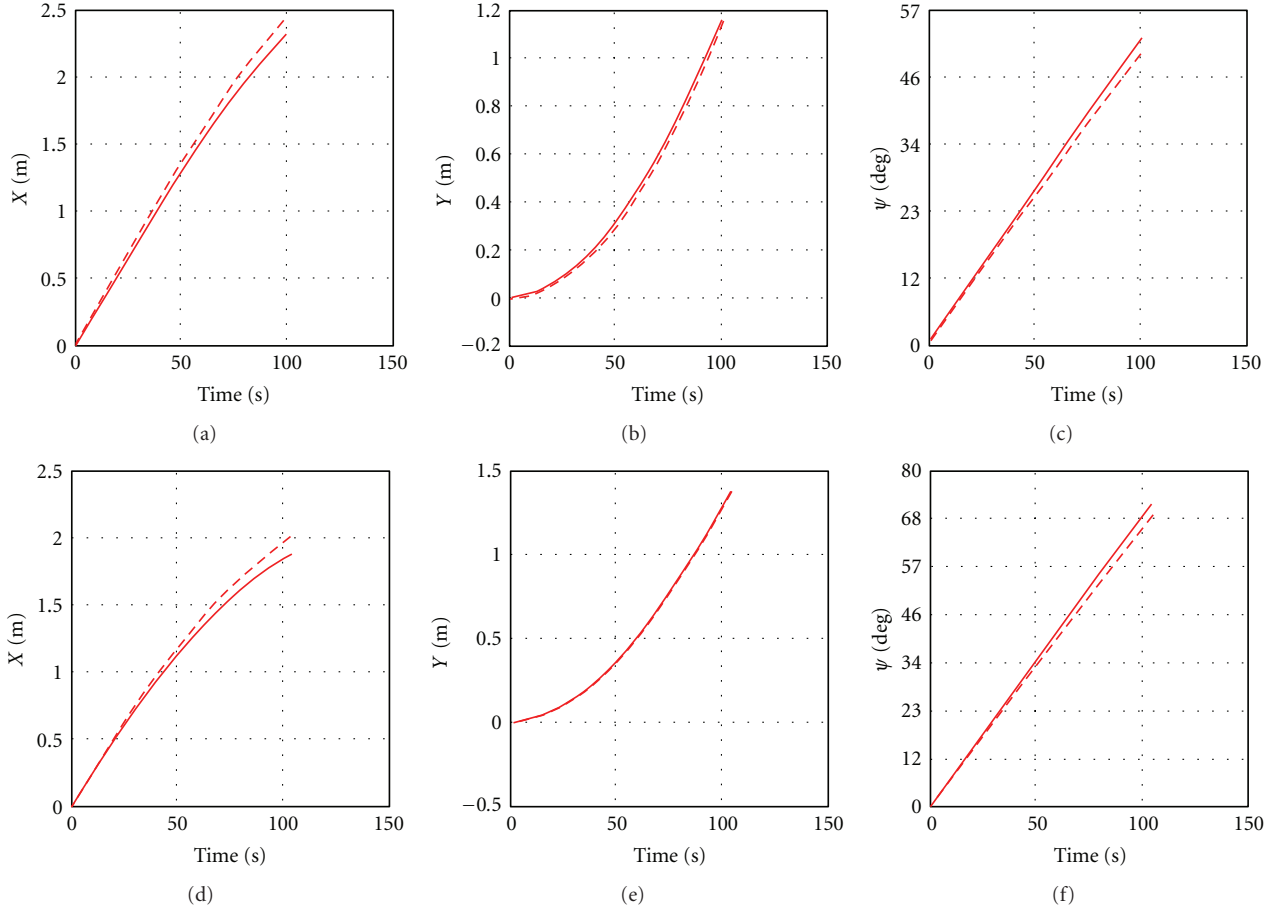


FIGURE 3: Experiment (dashed line) and simulation (solid line) correlation in two steering maneuvers: test 1 (top) and test 2 (bottom).

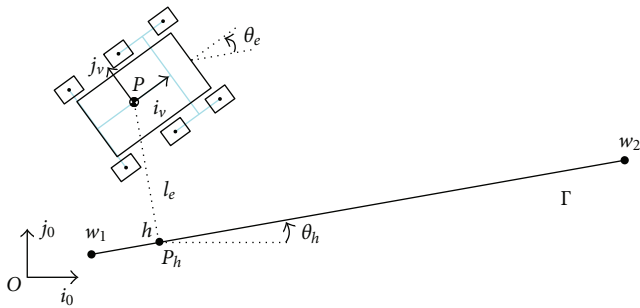


FIGURE 4: ExoMars rover and given trajectory represented in the Frénet frame.

Three constants were introduced in this control law; they must obey the following conditions in order to stabilize the system: $k_1 > 0$, $k_2 > -1/|v_0|$, and $k_3 > 1$.

In [14] nonholonomic constraints were imposed to deal with slip in the wheels and with useful results obtained to control a nonholonomic system. We use the same approach here, the constraints are

$$\begin{aligned} \dot{x}_0 \sin \theta_0 - \dot{y}_0 \cos \theta_0 &= 0, \\ \dot{x}_i \sin \phi_i - \dot{y}_i \cos \phi_i &= 0. \end{aligned} \quad (8)$$

Subscript indices $i \in \{1, 2, 3, 4, 5, 6\}$ are labeling the wheels and 0 the vehicle chassis; x_i , y_i , x_0 , and y_0 are the respective absolute position variables in the frame $i_0 j_0$; each angle ϕ_i constraining the wheels individually is a function of wheel steering angle and chassis orientation $\phi_i = \theta_0 + \delta_i$. Consequently, steering angles δ_i are computed as

$$\delta_i = \arctan \frac{\dot{y}_i}{\dot{x}_i} - \theta_h. \quad (9)$$

Driving velocities are obtained from the pure rolling constraint $\omega_i = v_i/r$ and relation $v_i = \dot{x}_i/\cos \phi_i = \dot{y}_i/\sin \phi_i$ as

$$\omega_i = \begin{cases} \frac{\dot{x}_i}{r \cos \phi_i} & \left(\theta_h \leq \frac{\pi}{4} \right), \\ \frac{\dot{y}_i}{r \sin \phi_i} & \left(\theta_h \geq \frac{\pi}{4} \right). \end{cases} \quad (10)$$

Individual wheel displacements in the frame $i_0 j_0$ can take geometric constraints into account

$$\begin{aligned} \begin{bmatrix} x_i \\ y_i \end{bmatrix} &= \begin{bmatrix} x_d \\ y_d \end{bmatrix} + \begin{bmatrix} \cos \theta_h & -\sin \theta_h \\ \sin \theta_h & \cos \theta_h \end{bmatrix} \overline{PP}_i \\ &= \begin{bmatrix} x_d \\ y_d \end{bmatrix} + \begin{bmatrix} X_d(\theta_h) \\ Y_d(\theta_h) \end{bmatrix}, \end{aligned} \quad (11)$$

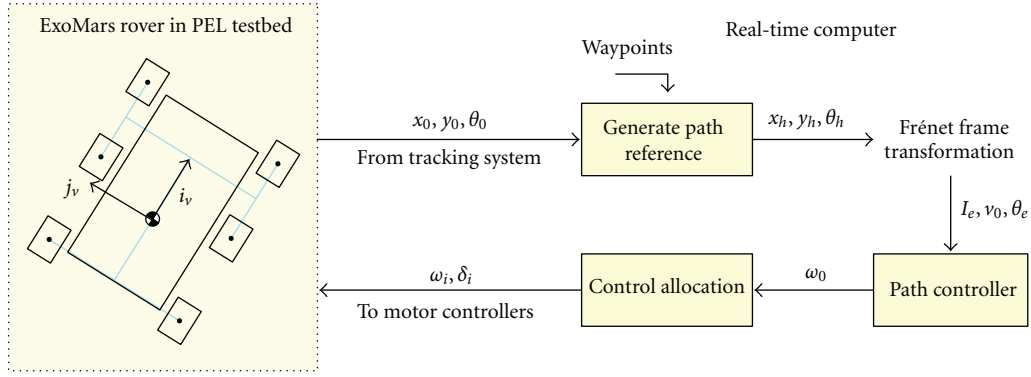


FIGURE 5: Block diagram of the path-following control in nominal mode.

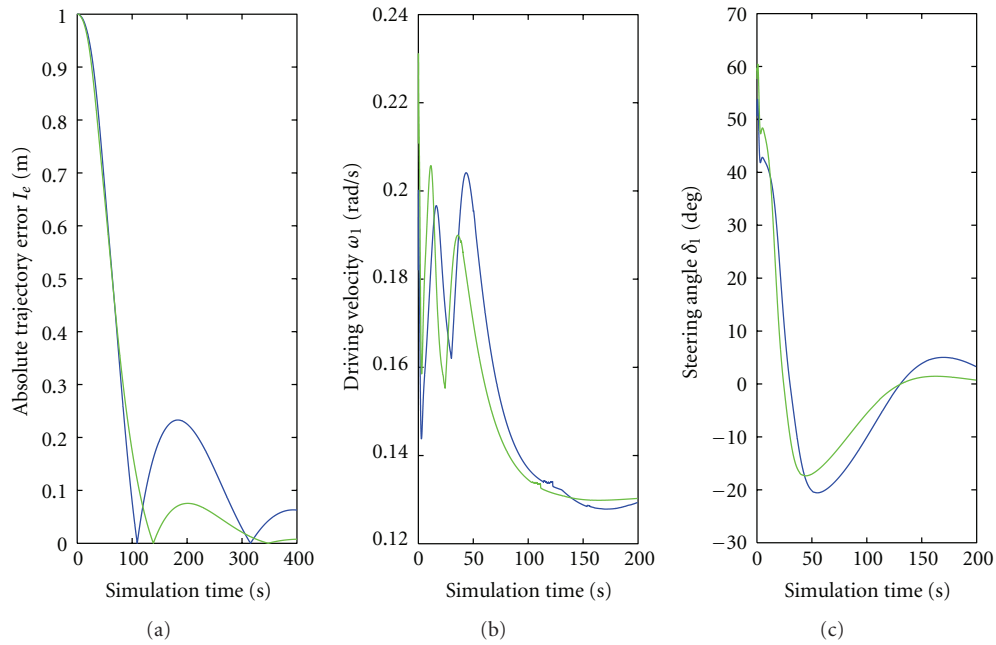


FIGURE 6: Tracking performance (a), angular velocity of the front-left wheel (b), and steering position of the front-left wheel (c).

where \overline{PP}_i is the vector from the center of mass of the rover to one of the wheels with the equivalent index. Both control signals can be revised to include the geometric constraint relationship differentiated with respect to time:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix} + \begin{bmatrix} \dot{X}_{di}(\theta_h) \\ \dot{Y}_{di}(\theta_h) \end{bmatrix} = \begin{bmatrix} v_0 \cos \phi_i \\ v_0 \sin \phi_i \end{bmatrix} + \begin{bmatrix} \dot{X}_{di}(\theta_h) \\ \dot{Y}_{di}(\theta_h) \end{bmatrix}. \quad (12)$$

This leads to the revised steering angle and driving velocity signals:

$$\delta_i = \arctan \frac{v_0 \sin \phi_i - \dot{Y}_{di}(\theta_h)}{v_0 \cos \phi_i - \dot{X}_{di}(\theta_h)} - \theta_h, \quad (13)$$

$$\omega_i = \begin{cases} \frac{v_0 \cos \phi_i - \dot{X}_{di}(\theta_h)}{r \cos \phi_i} & \left(\theta_h \leq \frac{\pi}{4} \right), \\ \frac{v_0 \sin \phi_i - \dot{Y}_{di}(\theta_h)}{r \sin \phi_i} & \left(\theta_h \geq \frac{\pi}{4} \right). \end{cases} \quad (14)$$

Note that these are commanded position and velocities to each pair of steering/driving joint, references of the low-level position and velocity controllers embedded in the hardware of the vehicle. These signals are the function of current angular position of the chassis, the desired velocity v_0 , and the orientation angle θ_h . Since the rolling constraint cannot be fulfilled in soft soil, [14] still suggests the use of $\hat{\omega}_i = \omega_i / [1 - (s_{\text{ref}} - s_i)]$ to reduce slippage s_i in each wheel to a certain amount s_{ref} ; this is also used here. The block diagram of the control system for path following in the nominal mode is that of Figure 5.

3.2. Synthesis of the Controller Parameters. The control law was synthesized to assure stability, but performance can be improved by choosing the parameters k_1 , k_2 , and k_3 . At this stage, we use the simplified contact model integrated in the MBS model to simulate the vehicle dynamics interacting with the path-following controller. Several simulations were

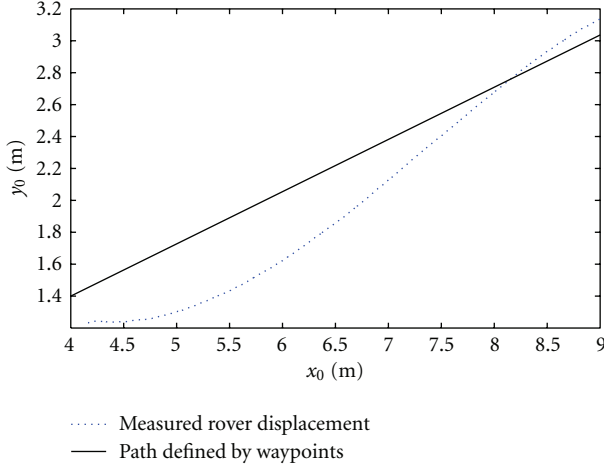


FIGURE 7: Experimental result of the path following control in nominal mode, vehicle starts in position and orientation indicated in the left-bottom part of the graph.

carried out during the numerical optimization process defined in MOPS to minimize the tracking errors (in translational displacement and orientation) constrained to the saturations of the actuators and to the lower bounds of the control law parameters. The control parameter values found and used to implement in the plant are $k_1 = 1.7 \text{ m}^{-1}$, $k_2 = -3.5 \text{ s/m}$, and $k_3 = 1.7 \text{ rad/m}^2$. The parameter k_2 has its lower bound limited by the value of v_0 , in this case chosen as $v_0 = 0.015 \text{ m/s}$. Figure 6 shows a comparison between the behavior of the controlled rover with an initial set of controller parameters (blue curves) and those values obtained after model tuning with MOPS (green curves). The wheels are limited to a maximal angular velocity $\omega_i = 0.26 \text{ rad/s}$ and confined to the steering range $-90^\circ < \delta_i < 90^\circ$; these limitations do not allow the tracking performance to be further improved as we see after a search through the global optimization algorithms in MOPS.

3.3. Experimental Results. Figure 7 shows the experimental results of the implemented path-following controller. It converges to the desired path inside the limits of the testbed, note the wheels and chassis steering to refine the pointing and tracking in order to reduce the steady-state error.

4. Selected Steering Failures in the Locomotion Subsystem

A PER has its locomotion properties (i.e., terrainability, trafficability, and maneuverability) substantially deteriorated when the prime movers do not behave as expected anymore. In the case of the Spirit rover [7], heating probably caused reflow of lubricant into the teeth of the gears. The faulty motor was no longer used because it has begun to draw too much current as compared with the other wheels; this fault vanished just after 4 months of infrequent driving and diurnal temperature cycles which allowed a lubricant to redistribute through the drivetrain and command the motor

normally again. Terrain interaction also stalled a motor because of a rock jammed between wheel and the housing of the steering actuator; it required tricky maneuvers to get rid of the rock. These faults deteriorated the performance of the Spirit rover like unsuccessful achievement of waypoints, induced heading error, and downhill slipping while trying to reach a goal. In other words, actuator faults reduce the capability of a PER to surmount obstacles, drive in uneven/steep terrain, and overcome environmental resistance to the translational and rotational movements inherent to commanded maneuvers. The vehicle can be equipped in different configurations with steering and driving motors; they can be mounted both in each wheel or suitably arranged according to the rover mass budget and kinematic structural properties. Thus, the arrangement of the actuators and the concept of the vehicle's suspension have to be taken into account during the choice of a fault repertory.

The ExoMars rover is a six-wheeled rover with independent steering and driving capabilities in each wheel; that is, 12 actuators can fail. The equivalent number of fault modes can be indefinitely increased as long as we assume additional faults in the actuator sets (gearboxes and electrical motors). In order to simplify modeling and focus on faulty scenarios which will certainly occur in some stage of the PER's useful life, motor failures are considered. These failures at the component level can be seen as faults at the PER's level. Once a driving motor failure happens, the functioning driving motors still provide traction as the nonfunctioning motor becomes a source of motion resistance. But the produced motion resistance is not sufficient to rotate the failed driving or steering joints due to the high gear ratio present in these joints. The most significant term of the motion resistance in this case is the bulldozing force; this resistance force is a result of displacement of the soil in front and also at the side of the wheel when it slips sideways with the slip angle β :

$$F_b = F_F(z) \cos \beta + F_L(z) \sin \beta. \quad (15)$$

Since the wheel is considered as a cylinder, the bulldozing resistance term F_F can be obtained in a closed form as in [17]. But the term regarding the lateral soil displacement depends on the shape of the sides of the wheel. To compute F_L , we represent the lateral shape of the wheel as a triangle mesh submerged on a pressure vector field (Figure 8(c)) defined according to the earth passive pressure vector σ_p acting on each triangle like $\Delta T_a T_b T_c$ in Figure 8(c); see the circular plate in Figure 8(b) spiked with circular holes as the lateral design of the wheel in Figure 8(a).

Both F_F and F_L are monotonically increasing functions of the sinkage depth z . This dependence is directly related to the area sunk into the soil. The detailed computation procedure of F_L can be found in [18]. Changing profiles of F_F and F_L are shown in Figure 9(a); the corresponding bulldozing resistance force in the vehicle's travelling direction is shown in Figure 9(b) as a function of slip angle.

This force model is used to evaluate the impact of failures in different joints (steering or driving) on the locomotion capability of the ExoMars Rover. In the case of failure in a driving motor, the bulldozing resistance can be minimized if

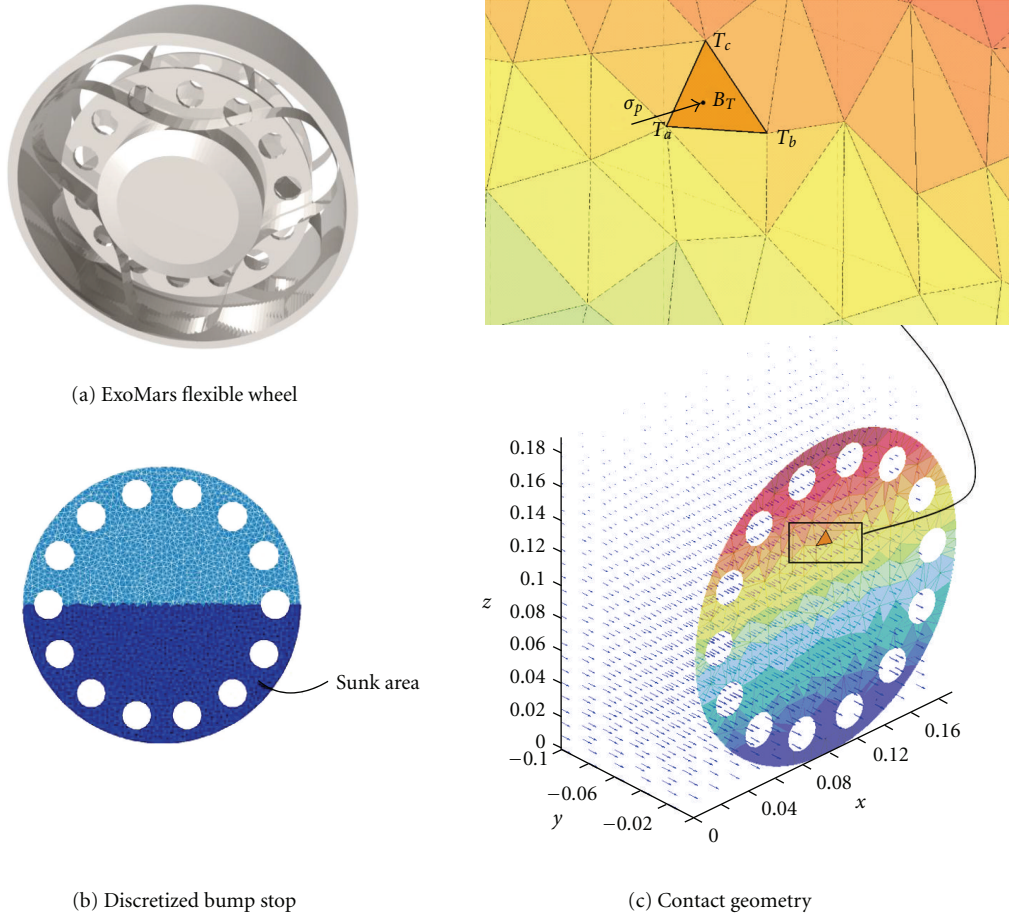


FIGURE 8: Lateral bulldozing.

the faulty wheel is forced to steer in order to point in the same direction of the velocity vector of the vehicle. This strategy can be used until the remaining working wheels are no longer able to provide sufficient traction to pull one or more wheels dragging soil. The same reasoning is not possible when failure in a steering motor occurs; in this case the available traction can still be prejudiced if the wheel attached to the nonfunctioning steering joint keeps rotating. Multiple failures, that is, more than one motionless steering joint, impose different constraints to the motion of the vehicle as they stuck in different angular positions. Figure 10 shows four different situations involving failures in the steering joints of the ExoMars rover; these examples illustrate that failures in steering motors are a complex situation when compared with failures in driving motors. This type of fault was chosen just as illustrative examples; other complex situations may arise involving a varied distribution and amplitude.

Figure 10(a) shows a diagram with the ExoMars rover traveling in the direction of the velocity vector v_0 ; the respective driving motor rotates to produce translational movement in the direction v_x resulting in slip angle β . The bulldozing resistance has increasing values for higher values of slip angle and affects the traction and heading angle of the entire rover. In Figures 10(b) and 10(c), there are

two configurations which can generate equivalent bulldozing resistances in the v_0 direction, but symmetrically opposed resistances during steering maneuvers. Figure 10(d) shows two failed wheels overloading motion resistance in one of the sides of the rover; this situation can disturb traction more than all failed steering motors stuck in the same angular position. In this case, there is no driving direction where the bulldozing resistance could be satisfactorily reduced.

Immobile steering joints can disturb driving and steering maneuvers in very different ways as dissimilar failure situations happen. Number of failures, their distribution, and amplitude (angular positions at which they stuck) assort an infinity of fault modes of the ExoMars rover. Due to the severity of this kind of fault, it is assumed as the main fault to be approached in this work. Nevertheless, it is a worst case at the component level, because faults at the component level could reduce available torque or velocity or change dynamic behavior without causing the complete loss of functionality (failure) of the joint. Thus, failure in all steering joints freely distributed through the wheels and allowing the full steering range compose the fault repertory summarized in Table 1.

Figure 10 shows 4 situations out of the 63 shown in Table 1. The amplitude in the last column of Table 1 is individually applicable to each single nonfunctioning steering joint of a given configuration. All fault modes

TABLE 1: Fault repertory summarized.

Index of fault mode	Faulty steering joint(s)	Faulty configurations	Amplitude
F1	Single joint	6	Full range ($\pm 90^\circ$)
F2	Two joints	15	Full range ($\pm 90^\circ$)
F3	Three joints	20	Full range ($\pm 90^\circ$)
F4	Four joints	15	Full range ($\pm 90^\circ$)
F5	Five joints	6	Full range ($\pm 90^\circ$)
F6	Six joints	1	Full range ($\pm 90^\circ$)

in Table 1 can be handled by the reconfiguration strategy proposed in this paper. Note that each fault has not only different configurations and amplitudes, but the amplitudes for each configuration are related to the individual faulty wheels as well. The number of faulty configurations is very important to conceive the proper reconfiguration strategy: a single fault mode has very different handling possibilities and causes distinct impacts in the degraded path-following maneuver depending on the distribution of the failures (faulty configuration). This demands a sufficiently general control reconfiguration approach.

5. Control Reconfiguration Strategy

Multiple steering joints allow several configurations to maneuver a rover and handle undesired situations like faults, obstacles, and high sinkage. These configurations are normally commanded by some operator, but our idea is to use them to maneuver a failed rover automatically. In the case of the ExoMars rover, the five most used configurations are Ackermann's steering, crab mode, point turn, skid steering, and turning about one wheel (i.e., the location of a wheel determines an instantaneous center of rotation). Automatic generation of these maneuvers is possible if we design a controller able to generate a point (a virtual hinge) in the plane about which the vehicle should turn. Henceforth this will be called the virtual hinge approach (VHA).

The VHA has to provide steering without overload the structure in excess. It becomes clear when multiple steering motors fail and the remaining functioning steering motors try to steer with a small turning radius; in this case it forces the failed wheels to sink into the soil. It overloads the structure and makes the vehicle "fight" against its own structure instead of performing a steering maneuver. A bottom-up reasoning is employed by observing Table 1 and drawing plausible strategies to handle each situation with the available steering capability. The simplest fault mode is a single failed steering joint (mode F1 in Table 1). It is favorable for the rover under F1 to drive straight ahead in the direction of the failed steering joint and use the driving capability of the failed wheel to push the rover. This is the crab mode, sufficient to drive straight ahead but not to follow a predefined path. Thus, we conceived a three-step procedure as follows.

- (1) Turn the vehicle about the failed wheel until the desired orientation and the longitudinal axis of failed wheel coincide.

- (2) Shift the coordinate system and zeros of the steering joints to the angular position at which the failed wheel got stuck.
- (3) Switch the controller back to the nominal mode path-following controller.

This reconfiguration procedure of Figure 11 can be applied to multiple faults, that is, from F1 to F6, as long as a virtual hinge and a favorable longitudinal direction are suitably chosen. This choice is intuitive in the case of F1 (whether corner or middle wheels) but not in the remaining cases. In the sequence, rules to determine the virtual hinge and the favorable longitudinal direction are stated.

Before a favorable direction is determined, a virtual hinge has to be chosen to point the chassis in the desired direction. Our first approach to the problem was an unconstrained optimization problem: maximizing the driving velocity in the failed wheel(s) by changing the virtual hinge location. But the number of iterations or even the convergence cannot be predicted; these characteristics make real-time implementation unfeasible since time duration and correctness of computations are not deterministically provided. The conceived solution is a generalization of the reconfiguration strategy for the fault mode F1 as shown in Figure 11. In that case, a fault implies in the Ackermann steering having the failed wheel as the center of a turning circle. If two wheels fail, two Ackermann steering configurations are computed, each one having a failed wheel as the center of a turning circle and considering the other one still functioning. Hence, one weighting vector Λ is computed as a metric of how far the amplitude of the failure (failed steering angles) is from the computed Ackermann steering under the assumption that this wheel is still working:

$$\Lambda_2 = \begin{bmatrix} \delta_1^F - \delta_1^2 \\ \delta_2^F - \delta_2^1 \end{bmatrix}. \quad (16)$$

δ_i^F is the amplitude of failure in wheel i ; δ_i^j is the angle of wheel i computed for Ackermann's steering considering j the failed wheel and consequently center of turning circle. The general form of Λ including n_F faults is the following n_F column matrix:

$$\Lambda_{n_F} = \begin{bmatrix} \max_{j \in \{2, \dots, n_F\}} (\delta_1^F - \delta_1^j) \\ \vdots \\ \max_{j \in \{1, \dots, n_F-1\}} (\delta_{n_F}^F - \delta_{n_F}^j) \end{bmatrix}. \quad (17)$$

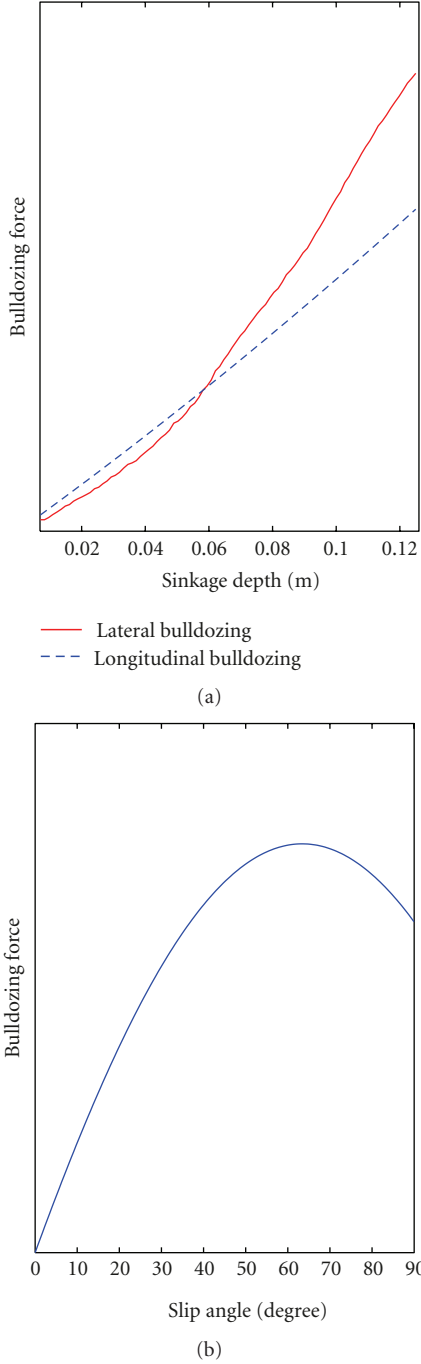


FIGURE 9: Bulldozing resistance force as a function of sinkage (a) and slip angle β (b).

This weighting vector is used to compute two virtual hinge candidates (γ_1 and γ_2) as a function of geometry of the vehicle and severity of fault. Another n_F column vector is given with the distances of each failed wheel from the center of mass of the rover; this is d_{n_F} . The candidates are $\gamma_1 = \Lambda^T d_{n_F} / n_F$ and $\gamma_2 = \Lambda^T d_{n_F}$; they are used to compute two modified Ackermann steering configurations. A steering configuration is composed of a pair of vectors $\delta = [\delta_1 \cdots \delta_{n_W}]^T$ and $\omega = [\omega_1 \cdots \omega_{n_W}]^T$, steering angles

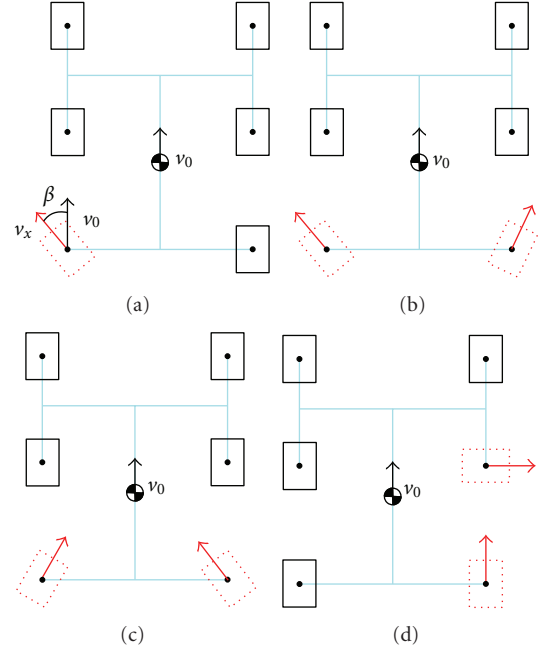


FIGURE 10: Failed steering motors stuck at different angles: (a) rear-left wheel; (b) rear-left and rear-right wheels pointing outwards; (c) rear-left and rear-right wheels pointing inwards; (d) rear right and middle right pointing perpendicular to each other.

and driving velocities for each of the n_W wheels. The turning velocity of the vehicle is a function of the nominal mode parameter v_0 ; all angular velocities are computed having this value as reference. The modified Ackermann steering instead gives $\delta_M \subseteq \delta$, $\omega_M = \omega \otimes g(\Lambda)$; \otimes denotes elementwise multiplication. A design parameter k_4 is then introduced inside the nonlinear vector weighting function $g(\Lambda)$, a $n_W \times 1$ vector defined element by element as

$$g_j(\Lambda_j) = \begin{cases} 1 - k_4 |\Lambda_j|, & \text{if } j = i, \\ 1, & \text{otherwise.} \end{cases} \quad (18)$$

The elements of $g(\Lambda)$ are always equal to one for functioning wheels and between zero and one for failed wheels. Thus, a virtual hinge γ can be definitely chosen:

$$\gamma = \begin{cases} \gamma_1, & \text{if } \min |\omega_M(\gamma_1)| > \min |\omega_M(\gamma_2)|, \\ \gamma_2, & \text{otherwise.} \end{cases} \quad (19)$$

This is sufficient to execute the first step in Figure 11(b) in the general case (single or multiple failures). Turn point placed in the failed wheel location, point turn, Ackermann-like, and skid steering are possible solutions as Figure 12 shows.

Now, a new favorable direction has to be chosen in order to shift the coordinate system of the chassis and the zeros of the steering joints properly. The bulldozing profile in Figure 9(b) is very important to search the favorable direction and define it as the direction where the bulldozing resistance is minimized. When F1 happens, the solution is obvious and is that of Figure 11; the favorable direction in case of F2 is defined by the smallest faulty amplitude. In

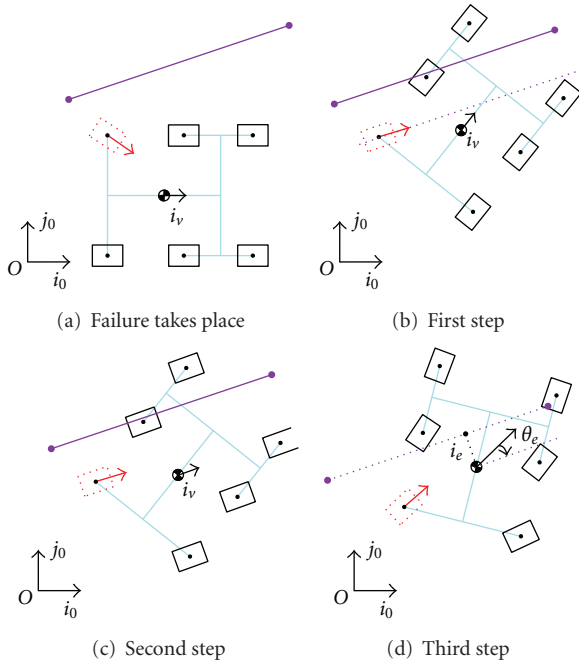


FIGURE 11: Three-step reconfiguration strategy in the case of F1.

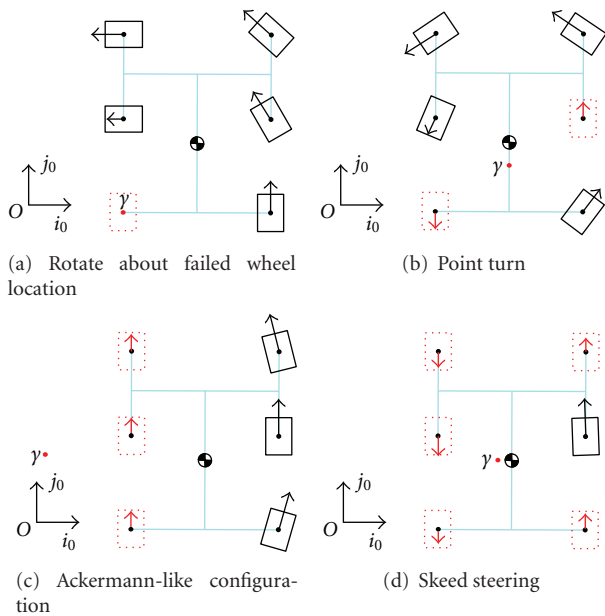


FIGURE 12: Steering configurations generated by the algorithm.

F2 just one of the faulty wheels will experience excessive bulldozing; the other wheels can be steered to the favorable direction and avoid excessive bulldozing. In practice, both angles are acceptable for F2, but the method leads to the angle closer to the trajectory slope. The general solution for all fault modes requires the numerical search of a minimum. Figure 13 shows the bulldozing resistance when three wheels fail at the arbitrarily chosen amplitudes -12° , -50° , and

-60° while the vehicle drives straight ahead with no specific steering configuration.

In this case, 23° would be the shift to define the favorable direction and switch the controller back to the nominal mode. The shifted nominal controller is able to track the path about the favorable direction with the help of the driving capability of the failed wheels. Abrupt changes in the reference path could require the functioning wheels to steer beyond the operational range; because of this, nonsymmetrical saturations are imposed in the steering commands. This dynamic saturation avoids increasing of the bulldozing resistance, keeping the system just about the nominal operating point. Figure 14 shows a scheme of the reconfiguration algorithm; the right-top input in the multiplexer is the shifted nominal controller (crab mode); the right-bottom input is the steering configuration used to point to the new favorable direction; the decision input of the multiplexer is the function which computes deviation of current orientation θ_0 from shifted orientation θ_s . Deviation inside the range $\pm\theta_c$ implies switching to the crab mode, otherwise the alternative steering provides steering angles and angular velocities.

After conceived, the fault-tolerant controller was verified by simulation in Matlab/Simulink. The controller was modeled in Simulink and the plant is the MBS model previously described; see Figure 15. The simulation model implements exactly the block diagram of Figure 14, but substituting the real rover on testbed by its dynamic MBS model. Linear and angular displacements of each mechanical part can be accessed from this model, environment properties are also included in the model. This means that the MBS model is not only a vehicle model but also a simple environment model, a simplification of a more complex environment used in [18]. All measured variables present in the real vehicle can also be accessed in the compiled MBS model; all other functions (path reference generation, Frénet frame transformation, reconfiguration, and switching logic) are considered as the control algorithm. Rover-measured variables are inputs to the controller; steering angles and driving velocities are outputs from the controller. Exactly this simulation model is used on testbed with a small modification, the MBS model is substituted by the proper hardware interfaces to command and acquire measurements from the real rover.

Plant and controller were developed in different software packages but integrated in Simulink to simulate a fault mode F3 (rear-left, middle-left, and rear-right wheels failed at -24° , -10° , and -38° resp.); see Figure 16.

Normally, reconfiguration depends on information provided by a fault detection and diagnosis scheme; in this case we are assuming prompt detection and perfect diagnosis in order to design and evaluate just the fault tolerant control strategy.

6. Experimental Results

6.1. Testbed Description. The PEL of DLR is a test environment for the characterization of soil and dynamic tests with a full rover in hard and soft sand [19]. A bevameter is used to

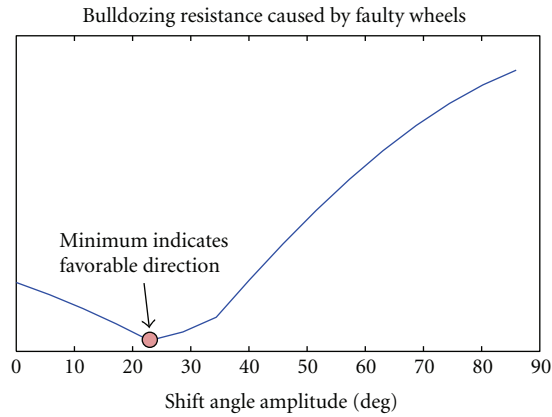


FIGURE 13: Bulldozing resistance force as a function of different shift angles to the crab mode.

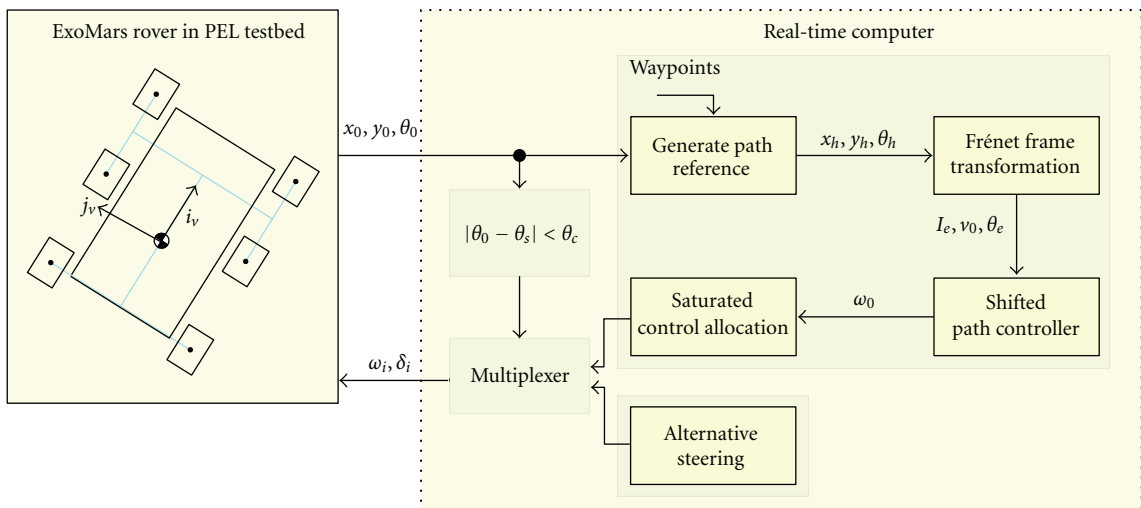


FIGURE 14: Block diagram of the path following control in faulty mode.

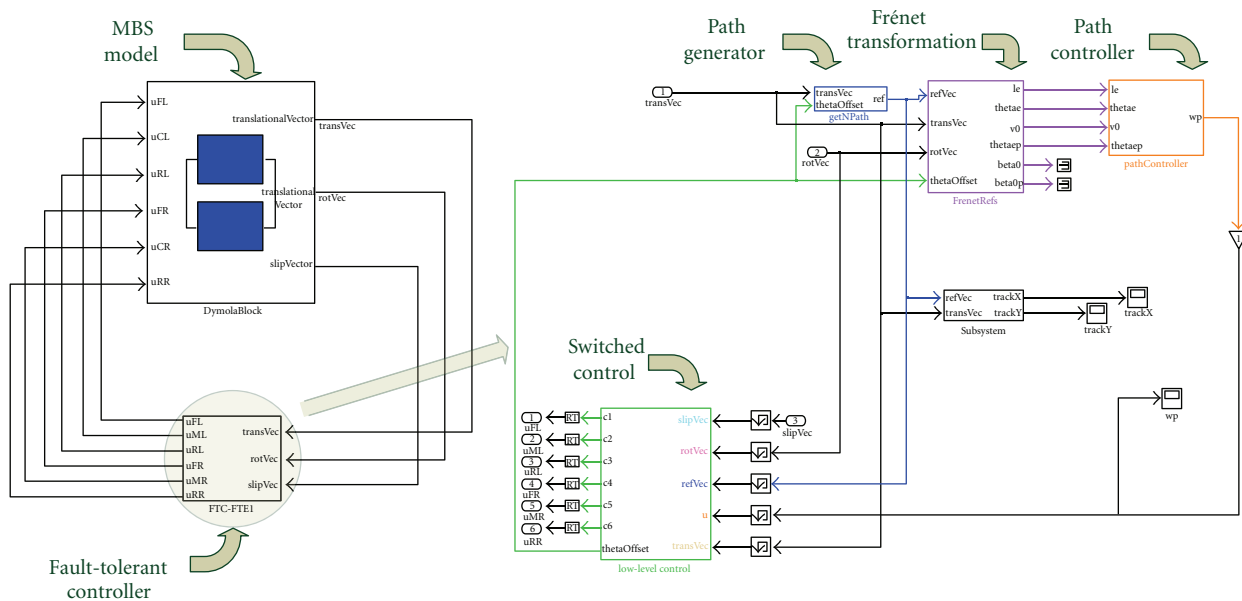


FIGURE 15: Matlab/Simulink implementation of the control system in Figure 14.

TABLE 2: Characteristics of injected fault modes.

Fault mode	Fault configuration	Amplitude
F1	Rear-left wheel	-74°
F2	Rear left and rear right	-24° (rear left) and -38° (rear right)
F3	rear left, rear right, and middle left	-24° (rear left), -10° (middle left), and -38° (rear right)

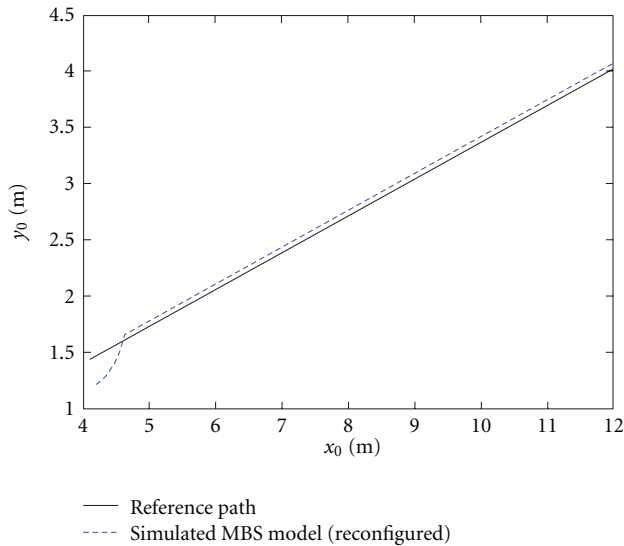


FIGURE 16: Fault-tolerant controller of the ExoMars rover simulated.

characterize soil properties and a testbed filled with two types of sand. Soft sand and hard sand are placed side by side but not mixed; their occupied volume is soft sand (5.5 m width, 4 m width, 0.5 m height) and hard sand (5.5 m width, 6 m width, 0.5 m height). It is equipped with a passive tracking system to measure the actual rover position (accuracy less than 3 mm) and orientation (accuracy less than 1 deg). The ExoMars rover is our main breadboard model used in dynamic tests in the testbed; this vehicle has three bogies equipped with angular position sensors, six wheels with independent driving and steering capabilities, force/torque sensors in each wheel, voltage and current measurements of all motors, and a real-time computer; see Figure 17.

All measurement variables and input signals are managed by the DLR’s “agile Robotics Development” (aRD) concept using internally developed tools [20]. Complex controllers are implemented meeting real-time constraints; storage of a large amount of data is also safely employed to subsequent analysis.

6.2. Tested Scenarios. Three fault scenarios were tested under faults F1, F2, and F3. The corresponding distribution and amplitudes injected to produce these fault scenarios are in Table 2. These amplitude values were chosen in order to avoid excessive motor effort during maneuver to achieve favorable direction. Note that it is not a limitation of the proposed method since the required motor effort is always much smaller than in the case without reconfiguration. The

entire amplitude range can be covered by the proposed method as long as the driving motors are able to provide enough power to the wheels; alternative steering is a relief to the driving motors and structure, but it is still an undesired situation where the vehicle drives against its own structure amplifying terrain resistance. There is some threshold regarding the admissible fault amplitudes; this can be handled by k_4 , but the investigation of the limiting control volume is not the purpose here.

6.3. Results and Discussion of Tested Scenarios. Figure 18 shows the performance of the vehicle under F1 with and without reconfiguration. The small rover in the figure shows the starting configuration; it converges in different ways according to the control system employed. A reconfigured rover drives with shifted orientation in a crab-like mode tracking the reference path. On the other hand, the rover without reconfiguration would need a space which is not available in the testbed to proceed the maneuver.

The situation depicted in Figure 18 (rover without reconfiguration) makes the structure vibrate and spend too much effort in making trenches without significant translation or rotation. Because of this behavior and additionally pushing the rover to the limits of the testbed, the test was manually stopped. In crab-like maneuvers (after reconfigured), the rover did not produce deep tracks, just drove smoothly to reach the reference path.

Two failures, that is, fault mode F2, may allow the nonreconfigured rover to converge to the reference path. Amplitude of fault has a very important impact in performance and can be more severe than a fault distributed in more than 2 wheels. Figure 19 shows this case; both control systems force the rover to converge to the given path. But not that the reconfigured version is much more accurate and fast. The nonreconfigured version leaves deep tracks and requires high traction to perform steering maneuvers with the failed wheels as they disturb the maneuver instead of helping to perform it.

Fault mode F3 was also very satisfactorily handled by the control system; see Figure 20. A favorable orientation is chosen and is not equal to any of the failed steering angles. This choice permits the rover to drive again smoothly and accurately to reach the reference path. Deep tracks and high traction needs are again drawbacks of handling the faulty rover with the nominal controller.

Figures 18–20 illustrate the exact rover orientation, but the orientation error must be considered with respect to the new favorable direction determined by the VHA. Figure 21 shows the error profiles of the heading angles in time domain. A peak can be noted before convergence to zero;



FIGURE 17: ExoMars Rover in the testbed.

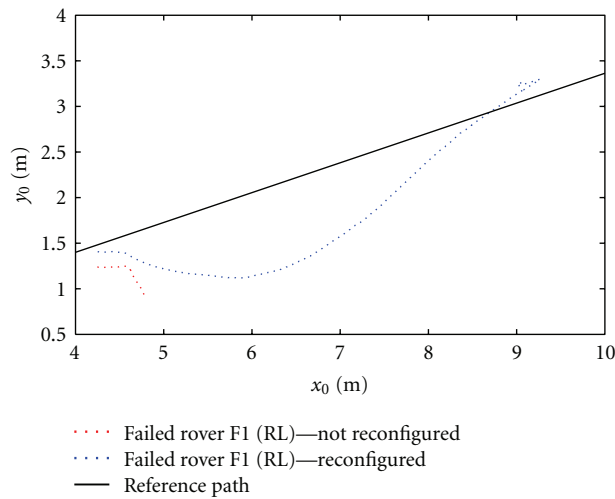


FIGURE 18: Fault-tolerant control performance for ExoMars rover under fault F1.

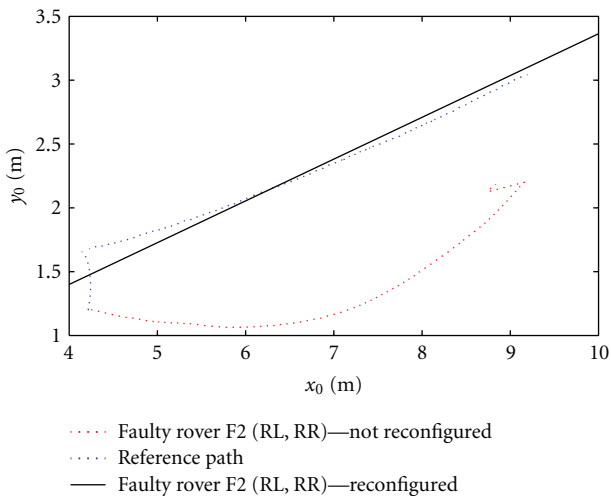


FIGURE 19: Fault-tolerant control performance for ExoMars Rover under fault F2.

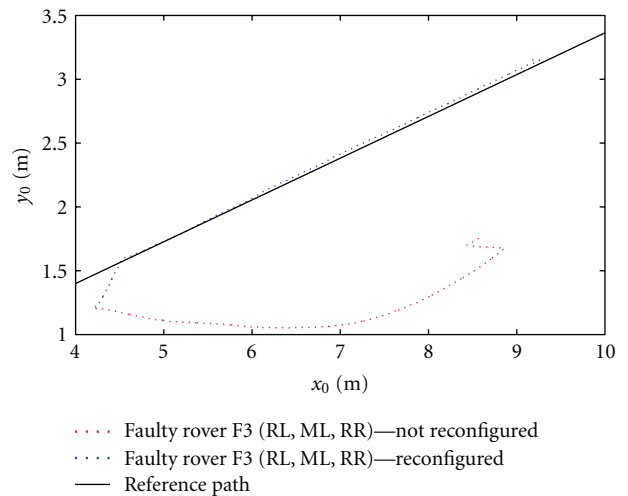


FIGURE 20: Fault-tolerant control performance for ExoMars Rover under fault F3.

this peak is due to the change in the coordinate system. The coordinate system is shifted and the reference also has to be

transformed and take the new shifted trajectory slope into account.

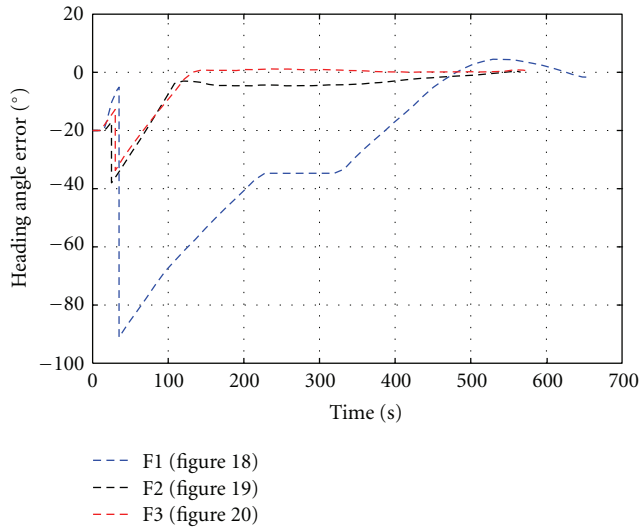


FIGURE 21: Errors in heading angles of the experiments.

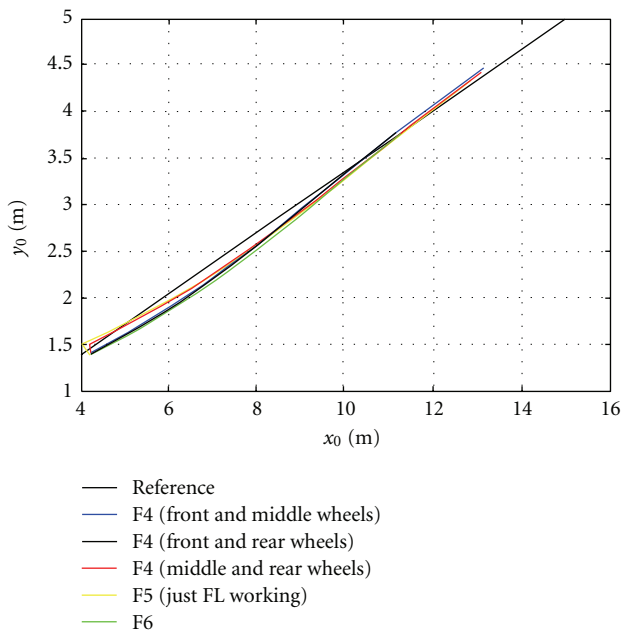


FIGURE 22: Simulation of cases F4 to F6 with control reconfiguration.

It would be tough to experimentally perform all the 63 cases of steering failure, but the remaining failure modes can be simulated. Higher failure modes (F4 to F6) were simulated to clarify the applicability of the proposed control; see Figure 22. The amplitude of the failures is always zero degree. Three configurations of F4 are shown; they achieve almost the same transient and steady-state behavior. The same happens to F5; F6 has just one possible configuration. The similarity of dynamic behavior when comparing different configurations of the same fault mode motivated the reduced set of simulations shown in Figure 22.

7. Conclusion

A fault-tolerant control system for the ExoMars rover was presented in this paper. It proved to be very powerful after experiments with the real vehicle in the PEL testbed. The conceived fault-tolerant control strategy can handle six fault modes in a total of 63 different joint failure distributions. Amplitude of the fault mode affects the performance in several manners: favorable direction changes; available envelope to steer the wheels is not symmetrical; usage of driving capability corresponding to failed steering joints may become not applicable. However, these characteristics are inherent to the problem and cannot be completely circumvented in a scenario where actuators lose functionality. In other words, performance degradation is always expected as a normal effect in the presence of faults and its handling is limited by the available power of the functioning actuators.

This work involves modeling, simulation, parameter estimation, control design, and experimentation. However, the main goal is fault-tolerant control. A considerable effort was spent in experimentation and model tuning in order to have a reliable model to verify the preliminary control concepts before test it in the real plant.

In spite of the sufficiently general fault-tolerant controller for steering maneuvers, other issues still have to be investigated like most unfavorable situations (including distribution and amplitude of failures), combined faults in steering and driving, favorable direction changing as a function of estimated sinkage, and performance optimization as a function of controller parameter k_2 .

Acknowledgments

Parts of the work on contact modeling development were performed in conjunction with the HGF alliance project “Planetary Evolution and Life” (HGF: “Helmholtz-Gemeinschaft der Großforschungsanstalten”). And parts of the work leading to the parameter estimation results have received funding from the European Community’s Seventh Framework Programme (FP7/2007–2013) under Grant Agreement no. 262744. The authors wish to gratefully acknowledge the support by HGF and EC.

References

- [1] S. A. Bøgh and M. Blanke, “Fault-tolerant control—a case study of the Ørsted Satellite,” in *Proceedings of the IEE Colloquium on Fault Diagnosis in Process Systems*, pp. 11.1–11.13, London, UK, April 1997.
- [2] T. Steffen, *Control Reconfiguration of Dynamical Systems*, Springer, New York, NY, USA, 2005.
- [3] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*, Springer, New York, NY, USA, 2nd edition, 2006.
- [4] H. Noura, D. Theillio, J. C. Ponsart, and A. Chamseddine, *Fault-Tolerant Control Systems: Design and Practical Applications*, Advances in Industrial Control, Springer, New York, NY, USA, 2009.

- [5] Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annual Reviews in Control*, vol. 32, no. 2, pp. 229–252, 2008.
- [6] M. Apfelbeck, S. Kuß, B. Rebele et al., "ExoMars Phase B2 breadboard locomotion sub-system test campaign," in *Proceedings of Advanced Space Technologies for Robotics and Automation (ASTRA '11)*, Nordwijk, The Netherlands, April 2011.
- [7] P. C. Leger, A. Trebi-Ollennu, J. R. Wright et al., "Mars exploration Rover surface operations: driving spirit at gusev crater," in *Proceedings of the IEEE Systems, Man and Cybernetics Society*, pp. 1815–1822, Waikoloa, Hawaii, USA, October 2005.
- [8] D. Zhuo-hua, C. Zi-xing, and Y. Jin-xia, "Fault diagnosis and fault tolerant control for wheeled mobile robots under unknown environments: a survey," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3428–3433, Barcelona, Spain, April 2005.
- [9] M. Bisgaard, D. Vinther, and K. Z. Østergaard, *Modelling and Fault-Tolerant Control of an Autonomous Wheeled Robot*, Institute of Control Engineering, University of Aalborg, 2004, Project Group 04gr1030a.
- [10] B. Schäfer, A. Gibbesch, R. Krenn, and B. Rebele, "Planetary rover mobility simulation on soft and uneven terrain," *Journal of Vehicle System Dynamics*, vol. 48, no. 1, pp. 149–169, 2010.
- [11] K. Iagnemma, C. Senatore, B. Trease et al., "Terramechanics modeling of Mars surface exploration rovers for simulation and parameter estimation," in *Proceedings of the ASME International Design Engineering Technical Conference*, 2011.
- [12] G. Ishigami, A. Miwa, K. Nagatani, and K. Yoshida, "Terramechanics-based model for steering maneuver of planetary exploration rovers on loose soil," *Journal of Field Robotics*, vol. 24, no. 3, pp. 233–250, 2007.
- [13] R. Rajamani, *Vehicle Dynamics and Control*, Springer, New York, NY, USA, 2006.
- [14] G. Ishigami, K. Nagatani, and K. Yoshida, "Path planning for planetary exploration rovers and its evaluation based on wheel slip dynamics," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 2361–2366, April 2007.
- [15] C. C. D. Wit, H. Khenouf, C. Samson, and O. J. Sordalen, *Nonlinear Control Design For Mobile Robots*, vol. 11, World Scientific, Singapore, 1993.
- [16] G. Ishigami, K. Nagatani, and K. Yoshida, "Slope traversal controls for planetary exploration rover on sandy terrain," *Journal of Field Robotics*, vol. 26, no. 3, pp. 264–286, 2009.
- [17] M. Scharringhausen, D. Beermann, O. Krömer, and L. Richter, "A wheel-soil interaction model for planetary applications," in *Proceedings of the 11th European Regional Conference of the International Society for Terrain-Vehicle Systems*, Bremen, Germany, October 2009.
- [18] A. C. Leite and B. Schäfer, "A comprehensive wheel-terrain contact model for planetary exploration rover design optimization," in *Proceedings of the Joint 9th Asia-Pacific ISTVS Conference and Annual Meeting of Japanese Society for Terramechanics*, Sapporo, Japan, September 2010.
- [19] M. Apfelbeck, S. Kuß, A. Wedler, A. Gibbesch, B. Rebele, and B. Schäfer, "A novel terramechanics testbed setup for planetary rover wheel-soil interaction," in *Proceedings of the 11th European Regional Conference of the International Society for Terrain-Vehicle Systems*, Bremen, Germany, October 2009.
- [20] B. Bäuml and G. Hirzinger, "Agile Robot Development (aRD): a pragmatic approach to robotic software," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pp. 3741–3748, Beijing, China, October 2006.