Terrain Mapping and Control Optimization for a 6-Wheel Rover with Passive Suspension

Author(s):
Strupler, Pascal; Pradalier, Cédric; Siegwart, Roland

Publication date:
2012

Permanent link:
https://doi.org/10.3929/ethz-a-010034720

Terrain Mapping and Control Optimization for a 6-Wheel Rover with Passive Suspension

Pascal Strupler, Cédric Pradalier, and Roland Siegwart

Autonomous Systems Lab, ETH Zürich, Switzerland
cedric.pradalier@mavt.ethz.ch

Summary. Rough terrain control optimization for space rovers has become a popular and challenging research field. Improvements can be achieved concerning power consumption, reducing the risk of wheels digging in and increasing ability of overcoming obstacles. In this paper, we propose a terrain profiling and wheel speed adjustment approach based on terrain shape estimation. This terrain estimation is performed using sensor data limited to IMU, motor encoders and suspension bogie angles. Markov Localization was also implemented in order to accurately keep track of the rover position. Tests were conducted in- and outdoors in low and high friction environments. Our control approach showed promising results in high friction environment: the profiled terrain was reconstructed well and, due to wheel speed control, wheel slippage could be also decreased. In the low friction sandy test bed however, terrain profiling still worked reasonably well, but uncertainties like wheel slip were too large for a significant control performance improvement.

1 Introduction

Since the first landing of a rover on the moon in 1970 by the Soviet Union, these semi-autonomous, mobile explorers enjoy an increase in popularity. In 1997, the first successful rover named Pathfinder rolled over the Mars surface. On Mars, this is still the only possibility to collect scientific data in such a mobile and interactive manner. Since space rovers are a relatively new way to explore extraterrestrial terrain, mission durations still vary a lot, but the latest missions have been brought to an end due to the rover wheels getting stuck in sand. The two current Mars rovers Spirit and Opportunity were already able to stay operational for more than 5 years, which is 20 times the originally planned mission duration. Nevertheless, they occasionally bogged themselves down in the sand and Spirit was given up and stays immobile because of this issue.

One way to reduce this problem is to minimize wheel slip. During wheel slip the wheels don't move as far as they are supposed to according to their

rotational speed. On a sandy surface, this can result in wheels digging themselves in. One of the cause of wheel slippage is often wheels fighting each other because of lack of knowledge about the involved terrain shape. In this paper, we therefore propose a method to adapt the individual wheel speeds of a rover according to the terrain profile. This leads to reduced wheel slippage as well as reduced chances of wheels digging into sandy soil. Because of the complexity of developing new and advanced sensors for space rover wheels, our method is based on sensor input of commonly used and reliable rover sensor technology like IMU (Inertial Measurement Unit), wheel encoders and angle measurements of the bogie suspension system.

1.1 Related Work

Optimizing rough-terrain control for space rovers is a popular field of research. One approach by Iagnemma et al. proposes to estimate force distribution on the wheels by using approximated wheel-ground contact angles ([1] and [2]). By computing the force distribution of a rover, it is possible to optimize the torques applied on the wheels and therefore reduce wheel slip and power consumption. The estimation of the wheel-ground contact angles is done using simple on-board sensors like IMU inclinometer, joint angle sensors and wheel encoders. Its accuracy strongly depends on dynamic angle measurements and therefore no estimation can be computed when the rover is still. Furthermore, wheel slip and smooth terrain profiles also result in poor wheel contact angle estimation.

Thus Lamon et al. from ETH Zürich developed tactile wheels to measure these wheel-ground contact angles instead of performing an estimation ([3] and [4]). This method was first implemented on the rovers Octopus [5] and Solero [6]. Later it was also applied to the 6-wheel Crab rover [7]. Although the approach shows promising results [8], embedded wheel sensors are still too complex and unreliable to be used in extraterrestrial environments.

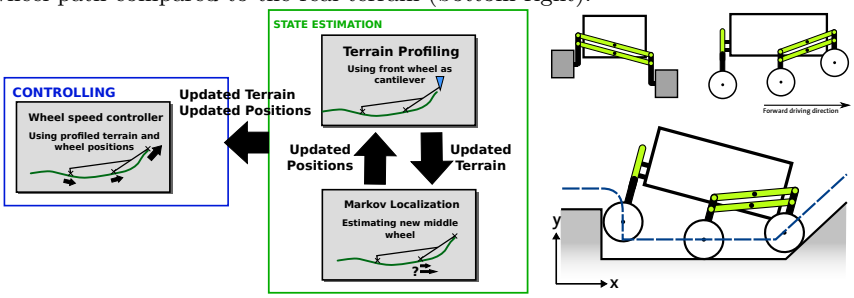
1.2 Goals and Limitations

Our objective is to develop an alternative approach on reducing wheel slip and optimizing control of space rovers in rough terrain. In contrast to the work by Iagnemma et al. mentioned above, our control should also yield good results in smooth terrain. On the other hand, we want to avoid using tactile wheels and other complex sensor systems in order to deliver a realistic approach for current space rovers. Our core idea relies on profiling the terrain shape using commonly used rover sensors such as IMU, wheel encoders and angle measurements of the bogie suspension system. The terrain shape can then be used to achieve wheel speed optimization.

2 Simultaneous Mapping and Control (SMAC)

Our approach proposes a velocity controller based on on-line terrain profiling, called SMAC (Simultaneous Mapping And Control). An overview of this controller is shown in figure 1. In the state estimation part, the terrain shape and the rover position are estimated. On one hand, terrain shape estimation highly depends on the rover position, but on the other hand, the rover position estimate can be improved significantly by accounting for the terrain shape in a probabilistic filter. Finally, knowing the terrain and rover position, a wheel controller can be proposed which optimizes the wheel speed to minimize theoretic slippage.

Fig. 1. State estimation and controlling (left), rover model (top right) and profiled wheel path compared to the real terrain (bottom right).



In the following, our implementation is explained using a parallel suspension bogie rover model, as shown in figure 1. However it is possible to adapt the controller to other suspension systems by modifying the geometry equations accordingly. Furthermore, we take the following assumptions:

1. We decouple both rover sides from each other and apply our method to each side independently.
2. We do not actually profile the real terrain, but the path traversed by the center of the wheels (see figure 1). From now on the term terrain designates this wheel center path. Note that recovering the real terrain shape is not possible due to ambiguities in corners.
3. The rover is assumed to drive straight and does not roll sideways. This allow reducing the profiling problem to 2 dimensions. As another consequence, all the wheel centers will follow the same terrain path which is included in a vertical plane in 3D space (designated as wheel movement plane, also take a look at figure 4 in section 3).

In this paper, we focus straight trajectories as a proof of concept for simultaneous mapping and control. This also allows applying the 2 dimensional

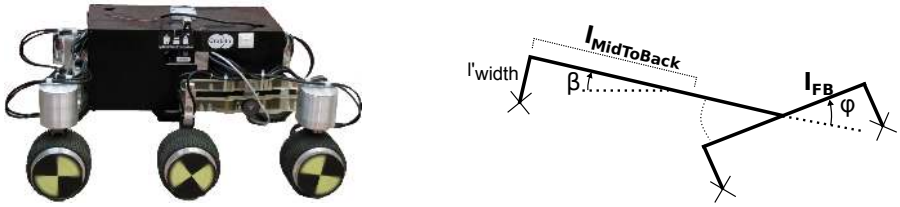
modeling to each side of the rover independently. However, it is clear that considering curved trajectory would require to consider the full 3D complexity of the problem, for which we cannot propose a solution at this stage.

2.1 Terrain Profiling

Terrain profiling allows us to approximate the terrain shape and - knowing the rover position - to optimize the wheel speed. Our objective is to use space-realistic sensors to achieve this goal: IMU, angle measurements of suspension bogies and motor encoder readings. First of all, we assume that, lacking visual or other distant sensing devices, there is no possibility to foreknow the terrain shape. The latter needs to be profiled in the instant the front wheels are traversing it. Therefore, these front wheels can be seen as cantilever-based tactile sensors. They can be used as profiling sensors while the two middle wheels use the former profiled terrain for propagation estimation. During the next iteration, the propagated middle wheels acts as new reference points for the front wheels to profile the next terrain points and so on. Hence, the terrain can be iteratively built up. This procedure is illustrated in the state estimation part of figure 1. However, one can easily observe that errors in profiling will accumulate since there are no measurements with absolute reference. To partially mitigate that, the middle wheel position is estimated through a probabilistic filter that reduces the displacement errors along driving direction and thus also improves the quality of the terrain profile. This is described in the next subsection 2.2.

The illustration of a parallel bogie rover in an arbitrary configuration is shown in figure 1. The front and the middle wheels are connected with a parallel bogie (to be called front left/right bogie). In the rear view, one can see that the two back wheels are also connected with a parallel bogie (rear bogie). A simplified model used for the upcoming computations is illustrated in figure 2. Note that the parallel bogie-wheel connectors can be disregarded since we only depend on relative wheel positions.

Fig. 2. The Crabli rover and its simplified rover model with bogie angle φ and IMU angle β (crosses represent wheel positions).



In order to profile the terrain at the front wheel, the position of the middle wheel has to be defined first. Assuming its x-position is iteratively propagated,

we can find the y-position by placing the middle wheel on our current terrain profile:

$$x_{MW}(x) = x \tag{1}$$
$$y_{MW}(x) = Terrain(x) \tag{2}$$

The position of the front wheel can then be found using the IMU tilt angle β and the front bogie angle φ:

$$x_{FW}(x) = x_{MW}(x) + l_{FB}\cos(-\varphi - \beta) \tag{3}$$
$$y_{FW}(x) = y_{MW}(x) + l_{FB}\sin(-\varphi - \beta) \tag{4}$$

where l_{FB} denotes the distance between the middle and the front wheel (length of the front bogie). Hereby, it is possible to profile new terrain points using the current terrain, the middle wheel x-position and the configuration of the rover.

To initialize the system, it is assumed that the rover starts on a flat terrain. Unfortunately, from proprioceptive measurements only, it is not possible to guess the shape of the terrain. As a result, even though the flat terrain assumption is far from perfect it seems to be the only one available. Alternatively, one could use exteroceptive measurements (e.g. from a stereo camera) as a starting point for the terrain profiling. Although very relevant this has not been addressed in the context of this paper.

2.2 Markov Localization

Markov Localization is used to estimate and propagate the middle wheel x-position by fusing the wheel speed information and the rover configuration. The general Markov Localization equation derived from Bayes' law is given by:

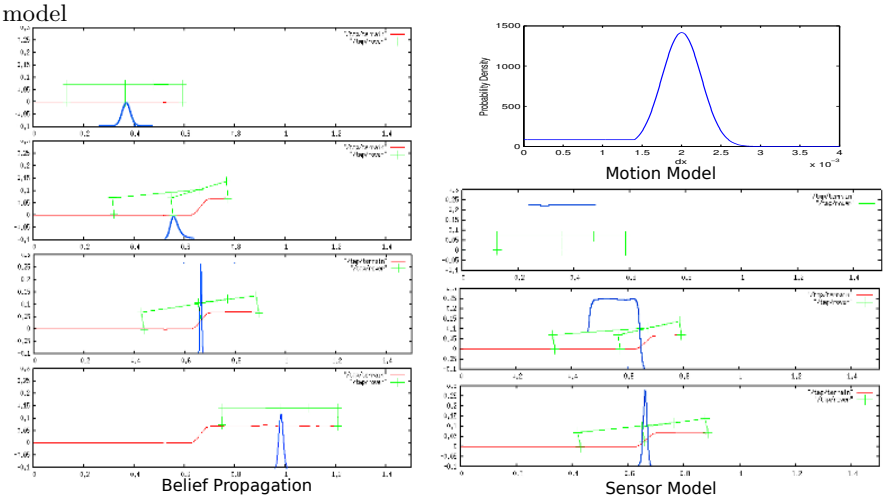
$$Bel(x_t|z_t...z_0) = \eta\, p(z_t|x_t) \int p(x_t|u_{t-1}, x_{t-1}) Bel(x_{t-1}) dx_{t-1} \tag{5}$$

where $p(z_t|x_t)$ corresponds to the observation model predicting the observation z_t (see below) given the middle wheel position x_t, and $p(x_t|u_{t-1}, x_{t-1})$ is the rover motion model giving the rover displacement for a control input u_{t-1}.

Belief

The belief $Bel(x_t|z_t...z_0)$ is an arbitrary probability distribution for the middle wheel x position. Since we do not account for the kidnapped rover problem and the updates are quite frequent, the belief does not have to be computed along the whole terrain. We can keep the width of the belief relatively narrow (about 1-2 times the length of the rover) centered around the middle wheel position. This saves a lot of computing power since the belief is involved in the discrete convolution for the action update.

Fig. 3. Markov localization: belief propagation based on sensor model and motion model



A sequence of beliefs resulting from our test data is shown in figure 3 (left). In the beginning, the belief is relatively wide - the position is not known very well yet. As soon as the rover reaches distinctive terrain, the belief gets more narrow (which is due to the sensor model as we can see later). In the second picture of the sequence, the belief gets steeper on the left side. Since the front wheel of the rover is situated higher than the other wheels, it would not make any sense for the rover to be placed more than 0.1 m to the left. In the third illustration, the rover is in a very distinctive configuration. This configuration can only appear if the middle wheel is positioned at the middle of the step slope. Hence the resulting belief is very narrow and precise - the rover is localized. Later, when flat terrain is reached, the belief starts to widen again. This is caused by the convolution with the motion model.

Motion Update

During the motion update, the old belief $Bel(x_{t-1})$ is convolved with the action model $p(x_t|u_{t-1}, x_{t-1})$. The latter defines the expected rover displacement given the motor input. In our case, this is the result of combining two probability distributions:

- A normal distribution centered around the value $u_{t-1}dt$ of the distance traveled since the last update. This distribution models the ordinary motion of the rover without wheel slip.
- A sigmoid distribution (approximation of the uniform distribution) from 0 to the center of the normal distribution and modeling the uncertainty resulting from slippage.

The equation of the motion model is given by:

$$p(x_t|u_{t-1}, x_{t-1}) = \sum_{S_r} p(x_t|u_{t-1}, x_{t-1}, S_r)p(S_r) \tag{6}$$

$$p(x_t|u_{t-1}, x_{t-1}, S_r) = \begin{cases} h(1 - \frac{1}{e^{-x_{end}(x_t - m)}}), & \text{if } S_r = 1 \\ \mathcal{N}\left(x_{t-1} + u_{t-1}dt, \sigma^2\right), & \text{if } S_r = 0 \end{cases} \tag{7}$$

Where S_r is an indicator for wheel slip ($S_r = 1$ if the wheel is slipping, $S_r = 0$ otherwise). Although the slip indicator is binary, the use of a sigmoid distribution means that any movement up to the desired one is considered equally likely in case of slippage, thus covering the cases from the wheel slipping just a bit up to 100% slippage.

Sensor Update

The sensor model $p(z_t|x_t)$ describes the probability of measuring the observations z_t given x_t. In our case, the measurements are the IMU tilt angle β, the front bogie angle φ and the back bogie angle θ. But in order to reduce computational complexity, we transform these measurements into the y position (height) of the front wheel y_{FW} and the back wheel y_{BW}. Using these transformed measurements, we can write:

$$p(z_t|x_t) = p(y_{FW,t}, y_{BW,t}|x_t) = p(y_{FW,t}|x_t)p(y_{BW,t}|x_t) \tag{8}$$

Where we introduce the reasonable approximation of the probability distributions of $y_{FW,t}$ and $y_{BW,t}$ being independent and normally distributed around the position that can be predicted from the terrain shape.

A sequence of sensor models from our test data is shown in figure 3 (more precisely, the plots correspond to the posterior localization using the sensor model and a uniform prior). The sensor model is depicted blue, the current rover configuration green and the profiled terrain red. In the first diagram of the sequence, one can see that there is more or less the same probability of this configuration to be placed around the middle wheel position neighborhood. In the next diagram the sensor model probability narrows a little bit due to the front wheel position on the step. Eventually in the third diagram, the configuration appears to be very distinctive and can only be placed in the center of the step. This is also indicated by its narrow sensor model.

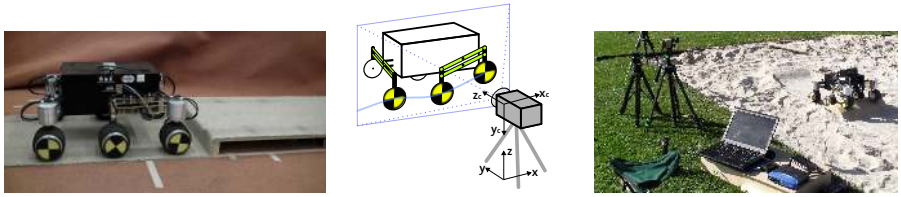
2.3 Wheel Speed Controller

Based on the approximated terrain and the current wheel positions, an optimal speed for the front, middle and back wheels can be derived. The principle of our approach is to compute the movement of the front wheel that would result from applying the desired speed for a control interval. From this movement, based on the known length of the suspension element, we can compute the resulting displacement of the middle and rear wheel, and deduce their optimal

speed. The challenge is that the terrain is necessarily unknown in front of the front wheel, and consequently, the front wheel movement must be computed by extrapolating the known terrain. Given the low speed we are considering, we currently use a linear extrapolation based on the last measured terrain slope.

3 Simulation and Testing

Fig. 4. The indoor test setup (left), the wheel tracking setup with HD webcam and markers (center) and the outdoor setup (right)



3.1 Goals

To measure a possible performance gain due to the new wheel speed controller compared to the former situation, we intend to evaluate if the controller helps decreasing wheel split and power consumption as well as improving the ability to overcome obstacles.

3.2 Wheel Tracking System

In order to measure wheel slip, the difference between actual wheel velocity (ground truth) and commanded velocity needs to be determined. Additionally, we would like to compare the profiled terrain of the controller with the actual path traversed by the wheels. Therefore we decided to implement an external optical wheel tracking system. It is based on a camera recording the test runs from aside the track. Markers placed on the wheels assure reliable detection of the wheel centers. The origin of the 3D coordinate frame is placed below the camera on the ground; y axis aligned with the camera. This setup is illustrated in figure 4. For most of the tracking implementation, functionalities from the OpenCV library were utilized. See [9] for more details on the tracking system and its calibration.

3.3 Testing Environment

Indoor

In a first approach, tests were performed indoor in the ASL Robolab at ETH Zürich (see fig. 4). The HD webcam was set at the distance of 1.35m from the track, which appeared to be a reasonable trade-off between track length and resolution of wheel tracking. With this distance to the track, a resolution of about 5.75 pixels/cm was achieved.

The track surface consisted of carpet to reduce wheel slip at first. The test runs were performed using 3 different obstacles on the track: a step obstacle (carpet surface), a hill obstacle (wooden surface) and a smooth step obstacle (Wooden surface).

Outdoor

In a second stage, tests were also done outdoors in a sand pool (see fig. 4). The distances and lengths of the track were similar to the indoor setup. Before each run, the sand surface had to be made smooth and level. Tests with slack sand where also performed to compare the influence of the different types of sand consistencies.

The outdoor obstacles were slightly different from the indoor setup: a step obstacle (bricks), a hill obstacle (wooden surface) and an asymmetric, irregular obstacle (stones).

3.4 Result Discussion

In this subsection we would like to discuss the results of a part of the test runs performed indoors and outdoors. As can be seen later, it is really hard to find a right measure for performance. In our current setup, the performance improvement on wheel slippage due to the controller is small with respect to the measurement noise. Therefore, it has not been possible to end up with quantitative performance metrics describing wheel slip performance or power consumption. In the following, a qualitative analysis of the performance is proposed.

Indoor

The obstacle traversed by the Crabli rover was a 8 cm high step covered by carpet. The situation is shown in fig. 4.

The resulting terrain shape profiled by the controller is plotted in diagram 5 as a blue line. The red line originates from the wheel tracking and is seen as ground truth. The most noticeable difference is the slope at the step. The real terrain features a sharp rise followed by an even curve. By contrast, the profiled terrain has an almost constant slope which is more flat. This is

actually due to the short wheel slip that occurs when the front wheel touches the step. At that instant, the rover stands still before the front wheel moves up, causing the middle and rear wheels to slip. Our controller is not able to sense this slip though and assumes that the rover is still moving forward. Combined with the new velocity component in vertical direction according to the changing rover configuration, the slope angle appears to be flatter than the one from the ground truth. This effect can also be shown using the diagram of the y and x-position over time: As one can see in diagram 6 the y position of the terrain can be followed quite nicely only having a small lag. Whereas the x position of the profiled terrain deviates from the ground truth in the moment of a wheel reaching the step (diagram 6). Other than that, the qualitative appearance of the profiled terrain matches the one from tracking.

Fig. 5. Profiled terrain compared to tracked terrain (front, middle and back wheel)

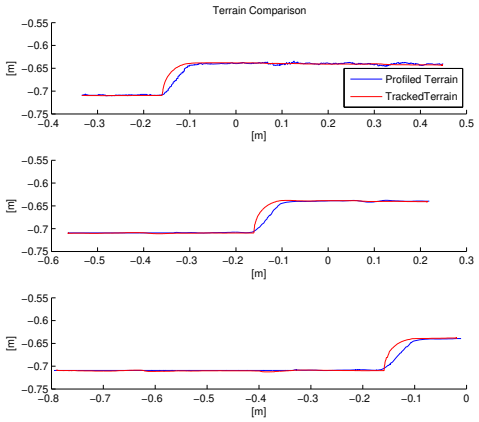
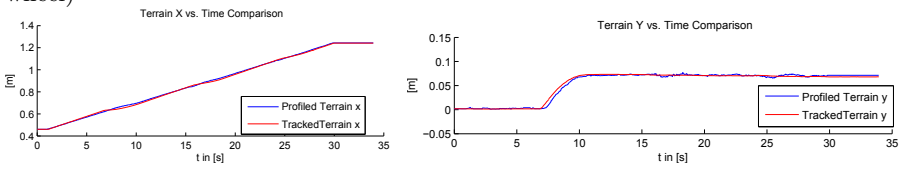


Fig. 6. Profiled terrain compared to tracked terrain, x and y position vs. time (front wheel)



Incorporating this profiled terrain shape, the controller adapted the wheel speeds accordingly. This can be seen in diagram 7. There are some key differences between the computed speed by the controller and the real wheel

speeds. First of all, it can be observed that the decelerations of the controller are about 30% lower than in reality. This condition is due to the lower slope gradient of the profiled step. Next the motor speed slow-down lags behind the real wheel deceleration. This is mostly caused by the delay of the front wheel moving up when reaching the step. At that time, the rover stands still, which cannot be sensed by the controller. For the front wheel, in contrast to the middle and back wheel, the controller does not know yet the shape of the terrain ahead. Hence it is not possible to introduce a foresighted controlling for the front wheel. At around 16s, the middle wheel reaches the terrain. Here the lag of the front and back wheel deceleration appears to be smaller. Finally at 24s, the back wheel moves up. One can observe that the deceleration lag seems to be larger again. This issue is more complex: It is probably due to a small dent, located at 0.07 to 0.11m in the profiled terrain. This dent was formed by the profiling front wheel during the middle wheel moving up the step face. In the current situation, the middle wheel is situated at this asperity. At the instant when the back wheel starts to move up, the rover configuration change is interpreted differently by the controller. This configuration change is equivalent to the back wheel remaining level and the middle and front wheel moving down. Whereas the middle wheel moves down anyway due to the dent. At this stage, it is not clear how this perceptual ambiguity can be identified and solved.

Fig. 7. Front, middle and rear wheel speed

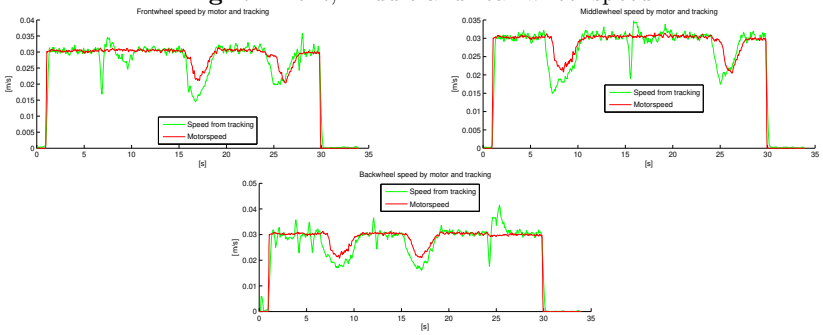
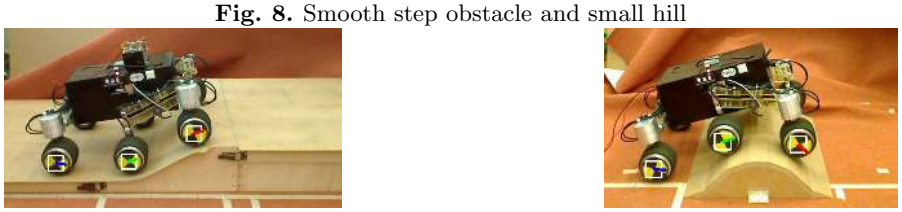


Fig. 8. Smooth step obstacle and small hill

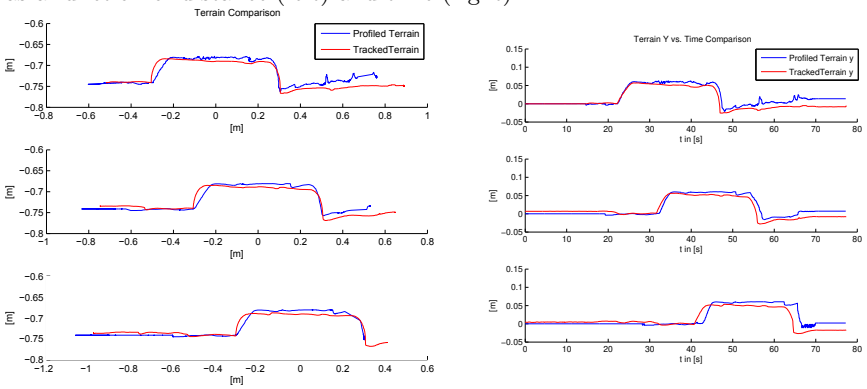


Tests run on a smooth step obstacle and a small wooden bump lead to similar results, with an even better terrain profiling: because there are no sudden change of direction such as on the step obstacle, no wheel slip occur and the terrain profiling performance improves. More details on these test results can be found in [9].

Outdoor tests on sand

During the indoor tests, we were able to achieve good terrain profiling and to some extend good wheel speed controlling in high friction environment. The next question was about how the controller would perform on low friction surfaces like sand. One test to be discussed incorporated a step obstacle shown in figure 10. Looking at the corresponding terrain diagram (fig. 9, left) the terrain profiling seems to be reasonably good. However when plotting the y wheel position against time (see fig. 9, right), lag can be noticed. This increasing lag is responsible for the late wheel speed decelerations observed in the wheel speed diagrams (fig. 10). One may note that some of the wheel decelerations contributed just about 30% of what would have been needed to match speeds. This issue can be once again explained by the lower gradient of the step slope in the profiled terrain. As during the indoor test, this happened due to the fact that the rover stood still as the front wheel touched the step face. Now it even takes more time for the slipping middle and rear wheel to build up enough normal force for the front wheel moving up. Additionally the front wheel also slips when moving up - thus using extra time which flattens the slope even more. The high velocity peak at the end is caused by the wheels falling of the second step.

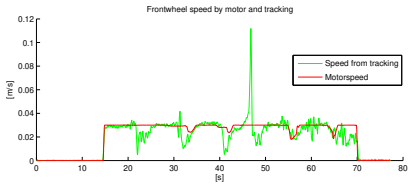
Fig. 9. Profiled terrain compared to tracked terrain (front, middle and back wheel), as a function of distance (left) and time (right)



During the outdoor tests, a lot more test runs were performed and analyzed, with similar results: in most cases, the terrain profiling works well,

but a significant resulting control performance improvement did not follow. This probably stems out from the lag and strong coupling between the control input and the terrain estimation, as well as out of the impossibility to sense wheel slippage. Should simple sensors such as desktop mouse movement estimator be integrated in future rover, our approach could naturally integrate their input to improve both the terrain profiling and as a result the wheel speed control. More details about this test can also be found in [9].

Fig. 10. Illustration of a step obstacle on a sand surface and measured front wheel speed while passing the step. Note in particular the dig-in that occurs each time a wheel climbs the step



4 Conclusion and Outlook

This paper presented an approach for simultaneous terrain profiling and control for 6-wheel rovers with passive suspension. The approach is based on a probabilistic filtering of the vehicle suspension deformation to jointly estimate the rover displacement and the shape of the terrain. Based on this terrain, a wheel speed controller was implemented to minimize the discrepancy between the ideal speed of a wheel following the terrain without slipping and the speed applied by the wheel controller.

Experiments indoor and outdoor have shown that the terrain profiling approach is sound and behave well even in presence of slip. However, at this stage of the implementation, it was not possible to demonstrate a significant control performance improvement resulting from the terrain profiling. The main reason for this lack of performance is mostly the delay introduced in by the terrain estimation, and the absence of wheel-slip sensing leading to noise in the terrain profile.

As for computational load, the SMAC controller was only occupying 10-15% of the capacity of an Atom-based embedded PC (FitPC2). In cases where the computational load would have to be reduced, there would be enough parameters to influence the controllers computational need. Especially narrowing down of the updated belief yields large differences.

On the hardware side, future work will need to consider integrating ground tracking sensors (e.g. optical mouse sensors) close to the wheels to detect and estimate wheel slippage. On the software side, it would be theoretically

feasible to put the terrain estimation in the same Bayesian framework as the current wheel localization. Such a joint estimation would be similar to what is currently implemented in the state of the art of simultaneous localization and mapping (SLAM). Although the computational cost would certainly be higher, a more robust estimation might be possible. Integrating a sensor for wheel sinkage would also help improving the profiling and mitigate the multi-pass effect when 3 wheels drive on the same track.

As a last remark, we would like to point out an alternative application for our SMAC implementation: Since the terrain profiler performs relatively well, one could think of using it for improving odometry or just helping localization on pre-planned path. Let's think of the rover Opportunity on Mars: These rovers plan paths ahead while standing still. Then in a second phase they try to follow this path using mostly odometry. Integrating the terrain estimator or just the terrain-base wheel localization would certainly improve the odometry performance, without having to resort to a full 6-degrees-of-freedom odometry.

References

1. K. Iagnemma and S. Dubowsky: *"Mobile Robot Rough-Terrain Control (RTC) for Planetary Exploration"*. Proceedings of the 26th ASME Biennal Mechanisms and Robotics Conf., DETC 2000, 2000.
2. K. Iagnemma, H. Shibley, S. Dubowsky: *"On-Line Terrain Parameter Estimation for Planetary Rovers"*. IEEE Int. Conf. on Robotics and Automation, Washington D.C, USA, 2002.
3. P. Lamon, A. Krebs, M. Lauria, S. Shooter and R. Siegwart: *"Wheel torque control for a rough terrain rover"*. IEEE Int. Conf. on Robotics and Automation, New Orleans, USA, 2004.
4. P. Lamon and R. Siegwart: *"Wheel torque control in rough terrain - Modeling and simulation"*. IEEE Int. Conf. on Robotics and Automation (ICRA), Barcelona, Spain, 2005.
5. M. Lauria and Y. Piguet and R. Siegwart: *"Octopus - An Autonomous Wheeled Climbing Robot"*. In Proceedings of the Fifth Int. Conf. on Climbing and Walking Robots, 2002
6. R. Siegwart, P. Lamon, T. Estier, M. Lauria, R. Piguet: *"Innovative design for wheeled locomotion in rough terrain"*. Journal of Robotics and Autonomous Systems, Elsevier, vol 40/2-3 p151-162.
7. T. Thueer, P. Lamon, A. Krebs and R. Siegwart: *"CRAB - Exploration Rover with Advanced Obstacle Negotiation Capabilities"*. 9th ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA), Noordwijk, The Netherlands, 2006.
8. A. Krebs, F. Risch, T. Thueer, J. Maye, C. Pradalier, R. Siegwart: *"Rover control based on an optimal torque distribution - Application to 6 motorized wheels passive rover"*. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems(IROS), Taipei, Taiwan, 2010.
9. P. Strupler: *"Control Optimization for Space Rovers Using Suspension Angles"*. Master Thesis, ETH Zürich, Switzerland, 2010.