10th ESA Workshop on Advanced Space Technologies for Robotics and Automation - ASTRA 2008

Strategy for adaptive rover behavior based on experiment

A. Krebs, C. Pradalier, R. Siegwart
Swiss Federal Institute of Technology
Institute of Robotics and Intelligent Systems
Autonomous Systems Lab
CH-8092 Zurich (Switzerland)
{*ambroise.krebs, cedric.paralier,roland.siegwart*}*@mavt.ethz.ch*

Abstract:
Due to the fundamental nature of exploration in rough-terrain, a rover accomplishing this task is naturally confronted with an unknown environment. It is especially true regarding its interaction with the soil, as the nature of it is uncertain. This work aims at creating a strategy allowing the rover to learn from its interaction with the terrains encountered, with the goal of optimizing its behavior. In practice, the information characterizing the terrains, obtained form remote sensors (such as camera), is associated with local sensors (e.g. IMU), characterizing the rover-terrain interaction. Correspondence between those data is learned and used, through the path planner E*, to influence the rover trajectory. The CRAB platform is used in this project for the implementation and testing of this approach.

1. INTRODUCTION

In the all-terrain mobile robotics field, the locomotion performance of a rover is mainly due to two factors. First, there are the mechanical aspects of the locomotion system, which include, for example, the number of wheels, the suspension system, and the size of the rover (1). Second, the locomotion performance of a given rover is highly dependent on its ability to interact with the environment. This is part of fields such as *control*, *path planning*, *obstacle avoidance*, and *position estimation*. These subjects are all linked to the capacity of the rover to *sense* and *represent* its environment. The more this information is complete and precise, the more the rover can deal with it confidently and behave efficiently.

In recent years, the fields of *terrain classification* and *terrain characterization* have received more attention. In the literature, the two fields are clearly separated most of the time (2). The *terrain classification* aims at associating terrain with well-defined categories (3; 4; 5; 6) whereas the *terrain characterization* tends to determine the driving performance corresponding to a terrain (7; 8; 9).

1.1 Terrain classification

The field of *terrain classification* was already the subject of an earlier work (10). In this publication, a Smart[1] vehicle was driven on various terrain, such as *grass*, *sand*, *asphalt*, *gravel* and *stony*. The data from an Inertial Measurement Unit (IMU) as well as a navigation LIDAR were logged and these two different data sources formed the core of that work. Our approach used techniques from the learning community, namely the AdaBoost algorithm (11). Therefore, after a supervised learning, it was possible to successfully classify terrain samples, on the sole and independent basis of each one of those two sensors.

It is interesting to note that the sensors used in this work were very different as the LIDAR provides information related to the terrain, which is ahead of the rover. This type of exteroceptive sensor is named *remote*. On the other hand, the IMU gives this information directly and locally. It corresponds to a sensor type named *local*, sensors which are mostly proprioceptive. As both classification method, named *proprioceptive* and *exteroceptive* in (10), worked well, it strengthens the natural belief that one can be inferred on the other. Ideally this relation should be learned online, as in (12), to provide a rover with the interesting capability of learning from the new terrain it encounters.

[1] http://www.smart-team.ch

Figure 1: CRAB rover, platform used in the context of this project. The pictures on the right hand side depict the compliance offered by its mechanical suspension mechanism.

1.2 Problem definition

On the 29th of April 2005, one of the two Martian rovers of NASA, Opportunity, got almost stuck in a sand dune. It took the scientists five weeks to cautiously extract the rover from this delicate situation and allow the mission to continue its course. This is a good example showing uncertainties that automatically come with an exploration mission. Although MER rovers were extremely well designed and even if their behavior were cautiously characterized and tested in many situations, such unexpected event can still occur. In the end, the terrain on which the rover will have to operate is extremely difficult to characterize beforehand, and most of the time, completely unknown. This is even more true in case of application in rough-terrain. This is the reason why it would be interesting to provide a rover with the capability to learn from its experience while it operates in a mission. A roadmap of this work is presented here.

The following section presents an overview of the method proposed to address the problem defined here. Section **3.** shows concretely how the method can be implemented. The next section presents the results while the section **5.** conclude the paper and give an insight on the future work.

2. APPROACH OVERVIEW

The objective of our work is to implement an online terrain classification and prediction algorithm. Its goal is to modify the rover behavior according to a given metric. The rover Roomba[2] is a very famous domestic robot aiming at cleaning the house floors. The rover is behaving regardless of the type of floor (parquet, carpet, and so on...) of the divers area it has to clean. It would be nice to make this rover able to infer the divers type of floor, initially unknown, from the noise the rover make cleaning them. Roomba could then optimize its cleaning pattern according to the preferences of their owners. Further more, the metric driving the rover behavior could be the amount of noise of the Roomba, or the slippage of the MER rover. Fig. 2 depicts the various elements that are part of this kind of approach which is similar in principle to the techniques at the core of the DARPA-funded LAGR project (13). The next subsections will describe them in more details.

2.1 Rover behavior

The rover behavior, can be influenced in many different ways as it is influenced by the various levels of control. Thus, the rover behavior can be affected by a change of controller type (lateral movement, wheel velocity synchronization, and so on...), or a modification of the parameters of a given controller. Here, what is meant by rover behavior is in fact a higher level and corresponds to the planned trajectory. For this reason, the E* online path replanner is used to determine the rover path according to a given metric. Section **3.** describes it in more details.

[2]http://www.irobot.com

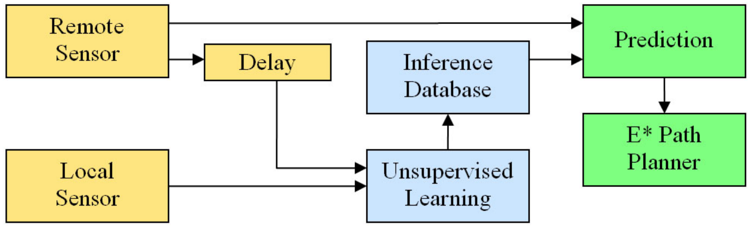


Figure 2: Detailed schematic describing the approach presented in this paper.

2.2 Sensors

The capability of a rover to learn from its environment is linked with the ability to acquire data from it, ability which is directly linked with the rover sensors. To summarize, this work aims at being able to use data characterizing the rover terrain interaction in order to influence its behavior. Therefore, the two different types of sensors, *local* and *remote*, have been used in this work. First, we require to be able to characterize aspects of the rover terrain interaction with the *local* sensors. Then, in order to influence the rover behavior, i.e. its path, it is needed to obtain information of the terrain ahead of the vehicle with the *remote* sensors.

2.3 Inference

The *predictive* sensor observes the terrain ahead of the vehicle and extracts some terrain characterization. Later on, when driving over the same terrain patch, the *behavioral* sensor records another terrain characterization. It is important to link both these data, in order to have samples that can be processed and used for learning. A predictive model can then be built, associating predictive terrain observation and behavioral terrain characterization (roughness, slippage, softness, and so on...).

2.4 Path planning metric

The metric Λ is the function that drives how the path is planned. It can be related to the noise generated (e.g. Roomba), the slippage (e.g.MER), the vibration in the rover structure, and so on... This metric is dependent on the data generated by one or many *behavioral* sensors.

$$\Lambda \in [0;1]$$

Λ takes a low value for a terrain having a good rover interaction. Hence, a value of 0 for Λ means a perfect behavior of the rover according to the metric defined, while a value of 1 corresponds to a terrain to be avoided.

3. IMPLEMENTATION

The work presented in this paper focus on exploration rovers such as CRAB depicted in Fig. 1. Hence, this section shows, the implementation of the various elements of Fig. 2 in the context of exploration in rough-terrain. First, the next subsection expose the assumptions that are made in the context of this work. The following subsections describe each step of the strategy in more details.

3.1 Assumptions

- While numerous works are identifying the *traversability* (T) of the surrounding of the rover, here, we aim at learning and identifying the best soils on which the rover can drive. The goal is to express this characteristic, called rover-terrain characteristics (RTC).
- As these experiments are done within the context of an exploration rover in rough-terrain, the rover speed isn't taken into account in the system as the velocity of those rover is usually very slow (5 to 10 $^{cm}/_{s}$). Hence, the velocity is considered constant in the system but can be considered for future work.
- At first, no prior knowledge exist in relation with the terrain type the rover can encounter. It means the number of terrain type, or their corresponding rover-terrain interaction have to be determined online.

3.2 *Probabilistic model*

The model of the system is considering the following elements. First, the *traversability* T, which is only dependant on the terrain shape, or geometry G. T can be considered as a simple boolean value expressing whether the terrain is traversable or not. Then the local features, F_l, express what the rover is able to characterize from the RTC. F_l is in fact a vector containing all the features used, computed based on the data measurable from the rover, i.e. acquired from the *local* sensors. On the other hand, F_r^t corresponds to the *translated* features computed based on the *remote* sensors available on the rover. F_r^t is also a vector containing all the features used, e.g. such as average rgb values of the terrain, and so on... Note that F_r^t are not acquired at the same time than F_l and therefore the features need to be translated in order to be consistent. The terrain soil is represented through its type (or class) with the variable μ_t. This corresponds to the classes we are able to distinguish based on F_l and F_r^t and it is a discrete variable. To each one of those classes, corresponds a given set of properties, referred to by μ_p. Hence μ_t are groups of similar F_r^t tagged *terrain 1*, *terrain 2*, and so on... μ_p would be the corresponding physical properties of those terrain classes, such as viscosity, friction coefficient, and so on... The resulting probability distribution and its joint distribution is the following:

$$P(G,T,\mu_t,\mu_p,F_l,F_r^t) = P(G)P(T|G)P(\mu_t)P(\mu_p|\mu_t)P(F_l|\mu_t)P(F_r^t|\mu_t) \tag{1}$$

Note that T depends solely on G, and μ_p, F_l and F_r^t depend solely on μ_t. This makes sense because the classification of the various terrain is done based on the rover experience, which is the combination of F_l and F_r^t. The remaining distribution of interest is the following:

$$P(\mu_t)P(F_l|\mu_t)P(F_r^t|\mu_t) \tag{2}$$

3.3 *Learning*

As a first step, and considering the fact that no prior knowledge about the rover-terrain interaction is known, it is essential to be able to learn from the data acquired. Using unsupervised learning techniques, the data F_l and F_r are clustered and labeled with different μ_t. This step corresponds to part **a** in Fig. 2. The technique proposed is part of the artificial neural network family and is called Growing When Requested (GWR). The algorithm will not be described here in detail, but more information can be found in (14). One can note that the overall technique isn't strictly unsupervised as it depends on the features. Those are defined according to what can be expected of the rover behavior on various terrain. For example, in the case of an IMU, one might want to minimize the amount of vibration in the rover structure. This can be done with feature measuring the RMS value of the signal, or just feature measuring the peak acceleration for a sample. Note finally that the GWR offers, as a result, only a topology of clusters. This is not really a problem in the sense that they can reused as mentioned below.

3.4 *Prediction*

Once the rover is able to infer its behavior with the predictive observation, this knowledge is reused to assess the metric of the terrain ahead of the rover. It corresponds to part **b** in Fig. 2. Thus, the goal is to find the expected value of F_l corresponding to a measured F_r.

$$E(F_l) = \sum_{i\in\wp(\mu_t)} P(\mu_{t=i})E(F_l|\mu_{t=i}) \tag{3}$$

$$P(\mu_t) \cong P(\mu_t|F_r) \tag{4}$$

Thus, probability associated with the terrain class μ_t is defined based on the measured *remote* features. It's value can be found through the result of the GWR as a gaussian distribution can be fit onto the cluster associated with F_r. The remaining unknown is $E(F_l|\mu_{t=i})$ which can be found through the characteristic of the associated cluster. In the end, F_r is known as the terrain is observable and F_l is found through the probabilistic model described above. Thus, Λ can be computed.

3.5 E* path planner

The E* algorithm is a path planner which is based on a weighted region approach. Thus, the underlying technique is expressed within the continuous domain, and corresponds to a wavefront propagating from the goal, toward the rover. The environment in which the rover is evolving is represented as a grid, constituted of node, or cells c. Several interesting properties are linked with c:

- $v(c)$ which represent the "height" of the navigation function of the cell.
- $r(c)$ is the difficulty or cost of traversing given cell. This parameter corresponds intuitively to the wavefront speed.

The trajectory of the rover, to head toward the goal, is then computed using the gradient descent on the navigation function resulting from the wave propagation. In the end, E* offers a trade off between the movement cost and the path length. Thus, in the context of the present research, the speed of the wave, depending on $r(c)$, is defined relatively to the metric to determine the path of the rover. Thus:

$$r(c) = f(\Lambda, T) \tag{5}$$

with

$$T = g(G) \tag{6}$$

$$\Lambda = h(F_l, F_r^t) \tag{7}$$

In Fig. 3, the navigation function is depicted. Two areas are identified, one in white and the other with a greyish color. In this situation, the behavior of the rover is better on the white area and it means a wavefront evolving faster in this area than in the greyish one.

$$\Lambda_{white} < \Lambda_{grey}$$

Another important feature of E* is that this algorithm is capable to dynamically replan the path. This is necessary as both the terrain perception and the characterization of the rover evolve online. It is possible that the rover doesn't have a complete perception of the area leading to the goal. Then the wavefront propagation will evolve as new area can be observed. Similarly, new terrain type can be inferred while the rover is evolving and this might result in changing the wavefront propagation as well. More information about the E* algorithm can be found in (15) and (16). Fig. 3 depicts the gradient descent of the navigation function. The trajectory is provided as a vector corresponding to a series of waypoint leading to the goal. The trajectory is called *trace*.

3.6 Rover control

Considering a differential rover, its control can be achieved according to (17). The translational (v_{trans}) and the rotational (v_{rot}) velocities are computed as follow:

$$v_{trans} = k_\rho \rho \tag{8}$$

$$v_{rot} = k_\alpha \alpha \tag{9}$$

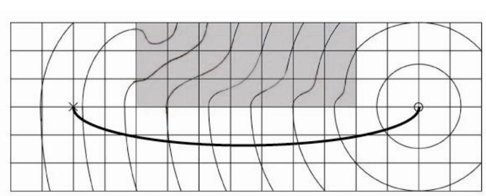
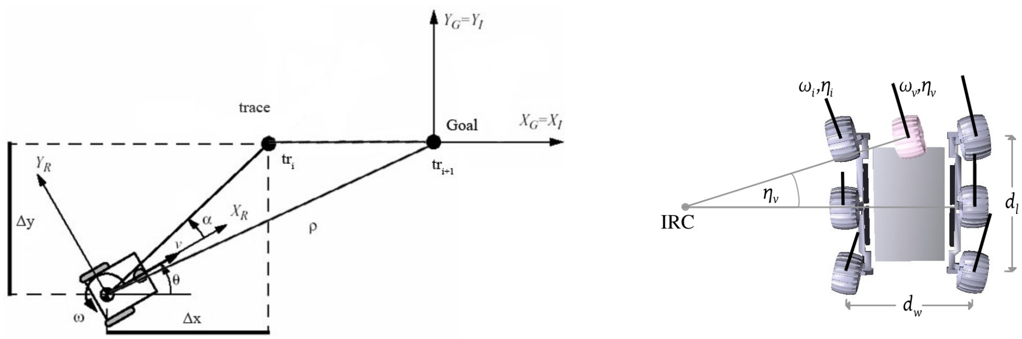


Figure 3: Wavefront propagation with E*, from the goal (marked with a circle) to the robot (marked with a cross). The *Trace* proposed by E* is defined using the gradient descent after the wavefront propagation.



(a) (b)

Figure 4: (a) Differential rover kinematics and its frame and control variables of interest.
(b) CRAB description in relation with its control.

with, ρ the distance to the goal and α the angle between the rover orientation and the direct trajectory to the next trace waypoint, tr.

$$tr = \begin{cases} tr_i & \text{if } \sqrt{\Delta x^2 + \Delta y^2} > \delta \\ tr_{i+1} & \text{otherwise} \end{cases} \tag{10}$$

tr is defined as the next waypoint, tr_i except if it is closer than a threshold δ. In this case, the following waypoint is used. Thus, it allows the rover to reach the goal, following the *trace* provided by E*.

4. RESULTS

The work presented being an ongoing research, the results are limited at the moment as the technique is not fully implemented. Thus, this section presents preliminary results.

4.1 Setup

The platform used in the context of this project is the CRAB rover, depicted on Fig. 1. It is a six motorized wheels rover with a passive suspension mechanism. Its is composed of a double parallel bogie mechanism on each side, connected via a differential. The first parallel bogie is between the front and the middle wheel while the second link the middle and the back wheel. The following sensors are available on the platform:

- An IMU, placed at the chassis level.
- Four angular sensors, positioned on the parallel bogies.
- Two angular sensors, placed on the differential.
- A stereo camera is being interfaced. This is the only sensor providing *predictive* data.

4.2 CRAB control

The control of the CRAB is as presented in section *3.6* with additional control limitation. In fact, the CRAB, due to its middle wheels, can be considered as an *extended* differential robot. The consign have just to consider additional constraints, especially to deal with high v_{rot}. A differential robot can almost instantaneously change its orientation whereas the CRAB has to handle the steering of its front and back wheels. This takes time and these transitions have to be considered carefully in order to achieve the desired trajectory. As displayed in Fig. 4, the CRAB is basically controlled via a virtual wheel placed at the front of the rover. It's steering angle η_v and velocity ω_v are computed as follow:

$$\eta_v = \arctan\left(\frac{v_{rot} * d_l/2}{v_{trans}}\right) \tag{11}$$

$$\omega_v = \frac{v_{trans}}{\cos(\eta_v)} \tag{12}$$

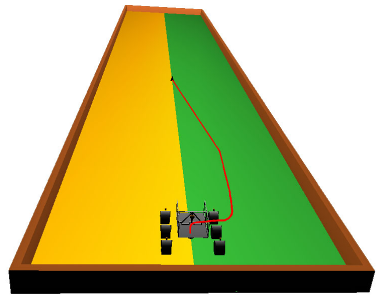


Figure 5: Simulation environment within the Webots simulators.

The corresponding consigns, η_i and ω_i, for the six wheels are derived from η_v and ω_v using the geometry of the rover. The steering angles of the wheels depend only on η_v but as these angles cannot be reached instantaneously, the steering velocity is adapted to reach the η_v configuration of the robot at the same time. For the same reason, the velocities of the wheels ω_i is computed based on ω_v and on the measured steering angles η_i^m.

$$\eta_i = F_{steering}(\eta_v) \tag{13}$$

$$\omega_i = F_{velocity}(\omega_v, \eta_i^m) \tag{14}$$

4.3 Simulation environment

In order to ease the development of CRAB rover, the simulation tool Webots[3] is used. This mobile robotics simulation software allows a rapid prototyping environment for modeling, programming and simulating mobile robots. Fig. 5 represent a test environment where the rover has to reach a target position (marked with a black cone). In this test, the CRAB has a better RTC on the green surface than on the yellow.

$$\Lambda_{green} < \Lambda_{yellow}$$

This information is transmitted to E* which computes the trajectory of the rover. First the navigation function is computed for each cell of the grid and then, the *trace* is generated on its gradient descent. The red line on Fig. 5 corresponds to the trajectory executed by the rover, following this trajectory. This example shows how the rover behavior can be influenced based on the strategy proposed in this paper.

5. CONCLUSION AND FUTURE WORKS

5.1 Conclusion

In this paper, a strategy to influence a rover behavior based on its rover-terrain interaction was present. The approach is based on the fusion of two types of sensors: *local* and *remote*. The inference between the data of both sensors type is planned to be executed online to improve the rover ability to deal with its unknown environment. The rover behavior is defined through a metric Λ which is computed based on the local and remote data acquired. This metric is an input to the E* planner which compute the best path possible for the rover, taking the metric into account. The control of the CRAB rover, it's implementation as well as preliminary tests have also been presented.

5.2 Future Works

Obviously, the work remaining is of importance. It can be mainly summarized in three parts. First, the learning part, grouping the F_l and F_r into clusters have to be integrated. The technique mentioned in the paper (GWR) was tested, but has to be integrated to the rover. Then, the Λ information transmitted to E* is manually set up, at the moment. Therefore the analysis of the samples F_r and the attribution of the corresponding $r(c)$ to the E* grid cell have to be perform online. Finally, and this is also linked to the first part, more field tests are necessary and will be performed in the upcoming month.

[3]http://www.cyberbotics.com

6. ACKNOWLEDGMENTS

The authors gratefully acknowledge Dr. Rolo Philippsen, Cunégonde, as well as Mark Höplfinger for their help and expert advices.

References

[1] T. Thueer and A. Krebs, "Performance comparison of rough-terrain robots simulation and hardware," *Journal of Field Robotics*, vol. 24, pp. 251–271, 2007.

[2] K. Iagnemma, "Terrain estimation methods for enhanced autonomous rover mobility," in *Intelligence for Space Robotics*, 2006.

[3] L. Ojeda and J. Borenstein, "Terrain characterization and classification with a mobile robot," *Journal of Field Robotics*, vol. 23, pp. 103–122, 2005.

[4] C. Brooks and K. Iagnemma, "Vibration-based terrain analysis for mobile robots," in *ICRA*, Barcelona, Spain, 2005.

[5] N. Vandapel and D. F. Huber, "Natural terrain classification using 3-d ladar data," in *ICRA*, New Orleans, CA, USA, 2004.

[6] C. Weiss, H. Froehlich, and A. Zell, "Vibration-based terrain classification using support vector machines," in *IROS*, Beijing, China, 2006.

[7] G. Bekker, *Introduction to Terrain-Vehicle Systems.* The University of Michigan Press, 1969.

[8] J. Wong, *Theory of ground vehicles.* New York: Wiley-Intersience, 2001.

[9] K. Iagnemma and H. Shibly, "On-line terrain parameter estimation for planetary rovers," in *ICRA*, Washington, DC., 2002.

[10] A. Krebs, C. Pradalier, and R. Siegwart, "Comparison of boosting based terrain classification using proprioceptive and exterioceptive data," in *ISER*, Athen, Greece., 2008.

[11] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, pp. 119–139, 1997.

[12] D. Kim and J. Sun, "Traversability classification using unsupervised on-line learning," in *ICRA, Orlando, FL, USA.*, 2006.

[13] J. Mulligan and G. Grudic, "Editorial for journal of field robotics − special issue on machine learning based robotics in unstructured environments," *Journal of Field Robotics*, vol. 23, no. 11-12, pp. 943–944, November-December 2006.

[14] S. Marsland, J. Shapiro, and U. Nehmzow, "A self-organising network that grows when required," *Neural Netw.*, vol. 15, no. 8-9, pp. 1041–1058, 2002.

[15] R. Philippsen, "A light formulation of the e* interpolated path replanner," Autonomous Systems Lab, Ecole Polytechnique Federale de Lausanne, Tech. Rep., 2006.

[16] R. Philippsen, B. Jensen, and R. Siegwart, *Autonomous Navigation in Dynamic Environments.* Springer Tracts on Advanced Robotics, 2006, ch. Towards Real-Time Sensor-Based Path Planning in Highly Dynamic Environments.

[17] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots.* The MIT Press, 2004, ch. Mobile Robot Kinematics.