Selection and Monitoring of Navigation modes for an Autonomous Rover

Thierry Peynot and Simon Lacroix

LAAS-CNRS
University of Toulouse
7, av. du Colonel Roche
31077 Toulouse Cedex 4
France
Thierry.Peynot@laas.fr

1 Introduction

Considering the wide spectrum of situations that it may encounter, a robot navigating autonomously in outdoor environments needs to be endowed with several operating modes, for robustness and efficiency reasons. Indeed, the terrain it has to traverse may be composed of flat or rough areas, low cohesive soils such as sand dunes, concrete road etc... Traversing these various kinds of environment calls for different navigation and/or locomotion functionalities, especially if the robot is endowed with different locomotion abilities, such as the robots WorkPartner, Hylos [4], Nomad or the Marsokhod rovers.

Numerous rover navigation techniques have been proposed, each of them being suited to a particular environment context (*e.g.* path following, obstacle avoidance in more or less cluttered environments, rough terrain traverses...). However, seldom contributions in the literature tackle the problem of selecting autonomously the most suited mode [3]. Most of the existing work is indeed devoted to the passive analysis of a single navigation mode, as in [2]. Fault detection is of course essential: one can imagine that a proper monitoring of the *Mars Exploration Rover* Opportunity could have avoided the rover to be stuck during several weeks in a dune, by detecting non-nominal behavior of some parameters.

But the ability to *recover* the anticipated problem by switching to a better suited navigation mode would bring higher autonomy abilities, and therefore a better overall efficiency. We propose here a probabilistic framework to achieve this, that fuses environment related and robot related information in order to actively control the rover operations.

2 Approach

The aim of our system is to select a *navigation mode*, that specifies either the combination of the main three perception, decision and action processes, or some parameters of each of these processes. For instance, a *path following* mode is defined by a path detection process and a servoing control law (no decision process here), whereas a *rough terrain traverse* mode is defined by a fine terrain geometric modeling process, a trajectory generation process and a trajectory following process. In the latter case, the trajectory following process may be achieved in various ways, especially if the rover is endowed with a complex articulated chassis (*e.g.* that allows both simple rolling or *wheel walking* motions).

To select the navigation mode to apply, two different sources of information are available :

- Environment related knowledge, or *context data*, that specify the suitability of the available navigation modes for given areas. Such information can be provided by an analysis of the terrain based on prior available aerial data for instance, by the rover's own terrain perception abilities (exteroceptive sensors), or by a combination of both. Given the nature of the processes that provide such knowledge, a description that expresses the partial probabilities of each considered navigation mode to be efficient is particularly adapted.
- On-line execution knowledge, provided by processes that evaluate the efficiency of the current navigation mode. These *on-line monitoring* processes check the evolution of some parameters with respect to pre-defined nominal behaviors.

These pieces of information are exploited to estimate the most suited mode thanks to a Hidden Markov Model (figure 1): the role of this HMM is to compute on-line the probability that each available mode is the the most adapted to the current situation [8]. Each state x_k corresponds to the proposition: "mode m_k is the best mode to apply" among the available applicable modes, and the chain is designed to survey the evolution in time of the robot behavior.

2.1 Conditional Estimation

The framework is designed as a Markov *conditional estimation* system [1]: the goal is to estimate the conditional state $x_{k,t}$ at time t, knowing context observation until time t, $O_{1:t}$, and behavior information $B_{1:t}$, provided by the on-line monitoring processes. Let's consider the robot is endowed with N different modes. The probability that mode m_k is the one to be apply at time t can be written, $\forall k \in [\![1, N]\!]$:

$$P(x_{k,t}|O_{1:t}, B_{1:t}) = \eta \, P(O_t|x_{k,t}) \sum_{i=1}^{N} P(x_{k,t}|x_{i,t-1}, B_{1:t}) P(x_{i,t-1}|O_{1:t-1}, B_{1:t-1})$$

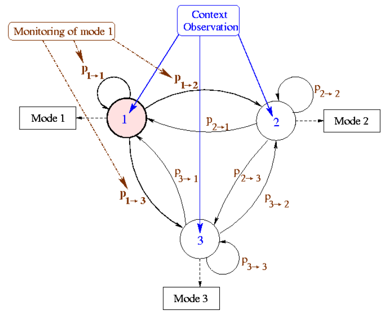


Fig. 1. Example of a HMM for the mode selection, with three modes available. The active mode here is m_1.

Fig. 2. The rover Dala

where:
• $P(O_t|x_{k,t})$ is an observation probability (context data) obtained through an analysis of the terrain (*initial classification data*).
• $P(x_{k,t}|x_{i,t-1}, B_{1:t})$ is a conditional probability of the transition from state x_i to state x_k, knowing the behavior $B_{1:t}$ until time t.
• η is a normalization coefficient.

2.2 Transition Probabilities

The expression of the conditional probability of transition from state x_c to state x_k (with $k \neq c$), knowing the behavior $B_{1:t}$ of the current mode m_c, is:

$$P(x_{k,t}|x_{c,t-1}, B_{1:t}) = P(x_{k,t}|x_{c,t-1}) + (1 - P(x_{k,t}|x_{c,t-1}))Q_{c,k}(B_{1:t}) \quad (1)$$

Where $Q_{c,k}(B_{1:t})$ is a *pseudo-probability of bad behavior* of mode m_c and $P(x_{k,t}|x_{c,t-1})$ is an *a priori* transition probability, from a fixed dynamic model. That way, if bad behavior of the current mode is detected by monitors while another suited mode is available, $Q_{c,k}(B_{1:t})$ will tend to raise, increasing the probability of the transition to the alternative mode (compared with the initial *a priori* probability $P(x_{k,t}|x_{c,t-1})$).

If no behavior data is available (it is generally the case for all modes different from the current active one), $Q_{c,k}(B_{1:t}) = 0$, so the transition probability is no more conditional and: $P(x_{k,t}|x_{c,t-1}, B_{1:t}) = P(x_{k,t}|x_{c,t-1})$

3 Navigation modes

The experiments are made with the robot Dala (figure 2), an *iRobot* ATRV equipped with a stereo-vision bench, a SICK 2D scanning laser rangefinder, an inertial measurement unit, odometry encoders, and a fiber-optics gyrometer. Two navigation modes have been implemented on this rover:

- The *Flat Terrain Navigation Mode (FlatNav)*, adapted to flat terrains, with possible high speeds. The mode is a reactive collision avoidance method based on the information provided by the laser rangefinder, that identifies the navigation situation and applies the corresponding motion heuristics with a divide and conquer strategy [6].
- The *Rough Terrain Navigation Mode (RoughNav)*, dedicated to uneven terrains, with rather low speeds. It uses a local trajectory selection based on predicted placements of the rover's structure on a Digital Elevation Map (DEM) built on-line [5]. The trajectory selected is the one that optimizes an *interest/cost* criteria, where the interest is a distance to the goal and the cost represents an integration of difficulties associated to the predicted attitude and configuration of the rover obtained by the placement function.

.

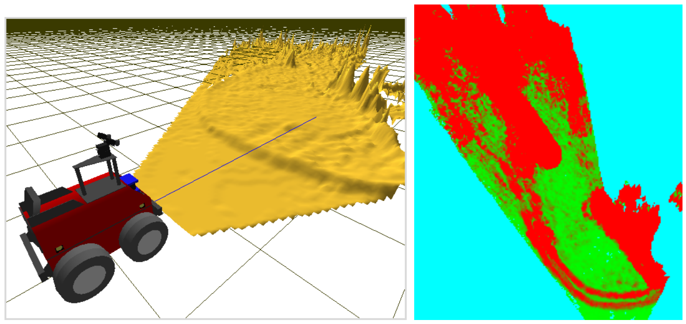
4 Context observation data: terrain classification

The first kind of data used to estimate the mode to apply is *context data*. Such data is obtained thanks to the robot placement algorithm on the DEM built on-line from stereovision data. On each cell of the map, the predicted configurations of the robot placed with different orientations are computed, and then combined to generate a *difficulty* associated to that cell (an example of *difmap* obtained can be seen in figure 3). This *difficulty* is clearly uncertain: a standard deviation is associated to each configuration and also consequently to the difficulty associated to each cell. An observation model enables to compute the context observation probabilities for the three classes associated to the modes *FlatNav*, *RoughNav* and *Stop*, which are: *Flat Terrain* (*difficulty* close to zero), *Rough Terrain* and *Obstacle* (*difficulty* close to 1 = maximum difficulty). Applying that observation model enables to obtain the probability densities $P(O|FlatNav)$, $P(O|RoughNav)$ and $P(O|Stop)$ (here, observation O = difficulty $diff$).

Other ways of getting context information are studied, such as the combination with aerial data from a blimp over the area, and/or with initial information provided by an operator.

5 On-line monitoring

The role of monitors is to check the behavior of the current mode by comparing a model of the nominal behavior with on-line gathered data, in order to provide probabilities that are used to compute the actual transition probabilities between the available modes. That way, if a monitor estimates that the current behavior of the active mode is not nominal, it will tend to provoke a transition to an alternative suited mode.



(a) DEM and robot's trajectory (b) *difmap*

Fig. 3. DEM and *difmap* corresponding to an area with a pavement.

Various monitors can be defined, and we have developed three different ones for the experiments that will be presented.

5.1 Uneven Terrain Detector (*UTD*)

This monitor is to be applied when the current mode is *FlatNav* to verify that the currently traversed terrain is actually flat. It uses an energy function computed with roll/pitch rates and the vertical acceleration provided by the IMU – this energy should indeed remain close to zero [8].

5.2 Locomotion Efficiency Monitor (*LEM*)

This monitor detects significant slippage situations, and especially locomotion faults, on the basis of *speeds coherence indicators*, used in a probabilistic classification procedure (see [7]). These indicators are the differences between various ways of estimating angular and linear speeds on board the rover. If the behavior of the robot is nominal, the various estimations should be similar: the higher the differences, the worse the locomotion behavior. Using these indicators as features, a Bayesian classification procedure based on a preliminary supervised learning stage enables to compute partial probabilities that the robot is in each of these following states: *Efficient Locomotion*, *Slipping* and *Locomotion Fault*.

5.3 Attitude Monitor

The *Attitude Monitor* uses the on-line comparison between the robot's attitude predicted and observed by the on board sensors. This monitor is applied when the *RoughNav* mode is active: the attitude prediction is made on-line thanks to the placement algorithm, applied at the current location of the rover on the computed DEM. Associated standard deviations are also computed, on the basis of the altitude standard deviations encoded in the DEM. These predicted attitude angles are then compared to the ones estimated using the IMU data, thus providing information about the behavior of *RoughNav*: in nominal behavior, predicted and observed angles should be similar.

Indeed, it is crucial to be able to check the validity of the angle prediction (and of the DEM computed for that purpose), as *RoughNav* relies on that operation. The main possible errors made by that prediction algorithm are due to: localization errors (leading to an inaccurate model of the environment), stereovision errors (including miscalibration of the stereovision bench) and model assumptions (e.g. assuming a rigid terrain).

Figure 4 shows an illustration of that comparison between prediction and observation of the pitch angle of the robot, and the consequent pseudo-probability of bad behavior obtained from it after applying the shaping function illustrated on figure 5. That function (corresponding to a pseudo-probability density) takes into account the uncertainty of the prediction and adds an influence of the maximum angle tolerated ϕ_{max}: indeed, the closer to the limitation the observed angle is, the more critical a rather small difference between prediction and observation might be.

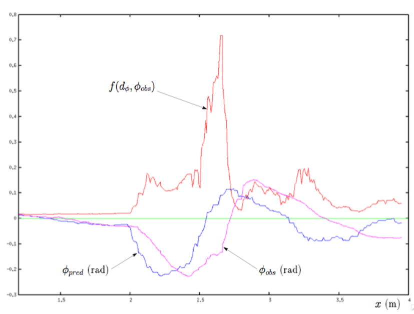


Fig. 4. Comparison between predicted pitch ϕ_{pred} and on-line observed pitch ϕ_{obs} while the robot is going over a pavement step, and consequent "bad-behavior" pseudo-probability computed.

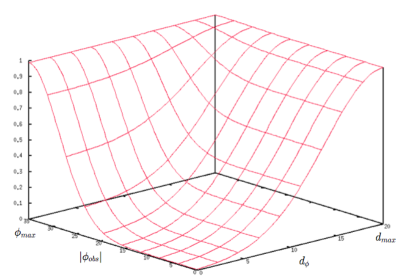


Fig. 5. Behavior pseudo-probability distribution over the difference $d_\phi = |\phi_{pred} - \phi_{obs}|$ and the observation $|\phi_{obs}|$.

6 Integration and experimentation

6.1 Multi-modes navigation system

The multi-modes navigation system implemented on the rover Dala is illustrated on figure 6. It is composed of three navigation modes: *FlatNav*, *Rough-*

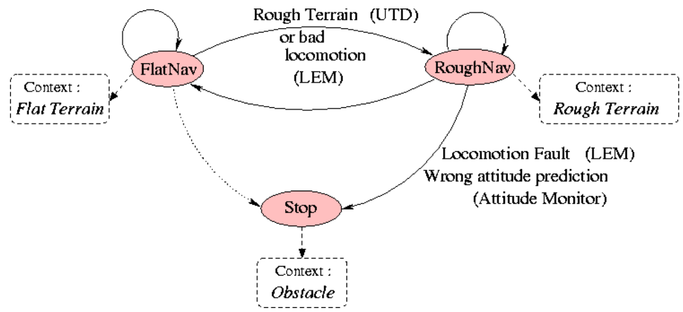


Fig. 6. Multi-modes navigation system for Dala, using *FlatNav*, *RoughNav* and a *Stop* mode in which all navigation actions are stopped. Node named *FlatNav* corresponds to the assumption: "*FlatNav* is the navigation mode to apply".

Nav and *Stop*, in which navigation is stopped if there is no better available alternative or no way for the robot to travel through the present area.

FlatNav is monitored by the *Uneven Terrain Detector* and by the *Locomotion Efficiency Monitor* (*LEM*). Indeed, if an area of uneven terrain or a locomotion difficulty is respectively detected by those monitors, it means that the terrain is not so flat and "easy" than expected (regarding only the context data). Consequently, *RoughNav* should be preferred, as it uses a far more complete environment model for motion planning: transition from $x_{FlatNav}$ (i.e. "*FlatNav* is the mode to apply") to $x_{RoughNav}$ (i.e. "*RoughNav* is the mode to apply") should be encouraged.

RoughNav is monitored by *LEM* too (*Stop* will tend to be preferred to *RoughNav* in case of locomotion fault), but also by the *Attitude Monitor*. As Dala is not endowed with any other navigation mode adapted to rough terrains, the probability of transition from *RoughNav* to *Stop* will increase if *Attitude Monitor* shows evidence of a bad behavior.

6.2 Monitors and Transition Probabilities

The current mode (m_c) pseudo-probability of bad behavior, $Q_{c,k}(B_{1:t})$, which has been introduced in equation 1, is provided by the active monitors. If there is only one active monitor to check the behavior of mode m_c, and that mode m_k is an available alternative, the element $Q_{c,k}(B_{1:t})$ is directly the probability of bad behavior according to that monitor. For example, figure 7 shows behavior data and transition probabilities obtained with *RoughNav* being the current mode and *Attitude Monitor* the only active monitor. The influence of behavior data provided by monitors can be seen clearly when context observation was wrong.

If several monitors are active, the pieces of information they provide need to be combined. Several combination strategies can be considered, depending on the relations between the events detected. Indeed, the events can be the

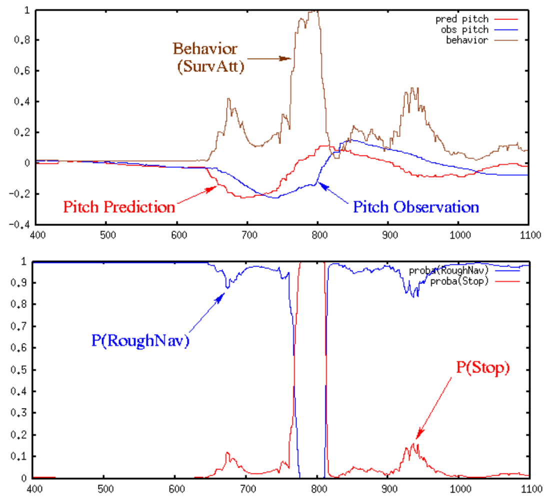


Fig. 7. Example of probabilities of mode to apply with the *Attitude Monitor* active, while context data is fixed by a human observer to: $p(O|Flat) = p(O|FlatNav) = 0.09$, $p(O|Rough) = p(O|RoughNav) = 0.8$ and $p(O|Obstacle) = p(O|Stop) = 0.11$. In spite of context data privileging *RoughNav*, the active monitor contributes to rather privilege several times *Stop*, as it detects evidence of bad behavior of *RoughNav* (inconsistencies between prediction and observation of the attitude angles).

same, but detected using different signals (enabling to use the Bayes formula), or they may be linked by a causal relation, or a logical combination.

In the present case, the behavior of *FlatNav* is checked by two monitors: *UTD* (Uneven Terrain Detector) and *LEM* (Locomotion Efficiency Monitor). The events they detect can be considered as independent, and the transition towards *RoughNav* should be privileged if an uneven area is detected by *UTD* *or* if there is a slipping situation seen by *LEM*. Thus, we make a *logical* combination, leading to:

$$Q_{FlatNav,RoughNav}(B) = P(Slipping) + P(UTD) - P(Slipping).P(UTD)$$

Figure 8 illustrates the result of the combination of behavior information provided by the two monitors *UTD* and *LEM*, while the active mode is *FlatNav* and the context observation probabilities have been set by a human observer. It shows that although context data assume the terrain is flat, behavior data provided by the combined monitors can lead to prefer *RoughNav* several times.

Finally, Figure 9 shows an illustration of a complete experiment: the context data are computed by building a *difmap*, and the adequate monitors are active to provide behavior information.

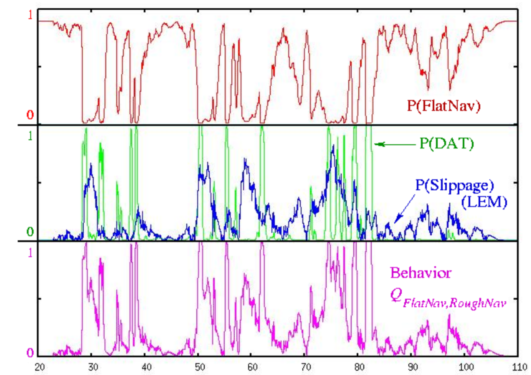
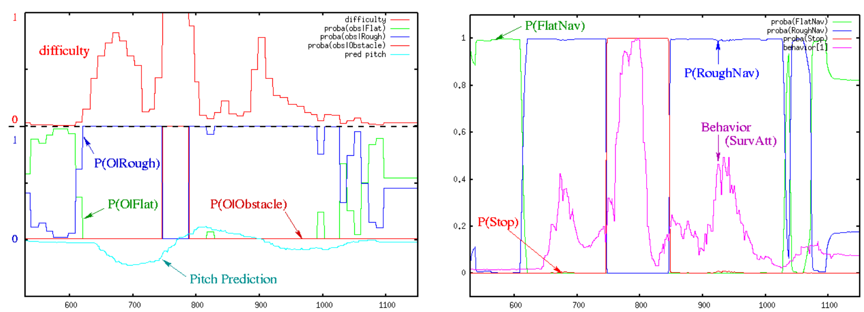


Fig. 8. Example of probabilities of the modes to apply with combination of monitors *UTD* and *LEM*, while context data is fixed by a human observer to: $p(O|Flat) = 0.75$, $p(O|Rough) = 0.20$ and $p(O|Obstacle) = 0.05$. In spite of context data privileging the *FlatNav* mode, the two combined monitors contribute to rather privilege several times *RoughNav*, as they detect evidence of rough terrain areas above the robot and/or inefficient locomotion.



(a) Context observation probabilities computed from the *difmap*

(b) Modes application probabilities after integration of behavior data

Fig. 9. Illustration of results in the situation introduced in figure 3. (a) shows the difficulty read in the *difmap* along the trajectory of the robot, which is going up a pavement step (see the predicted pitch angle), and the context data generated from it ($P(O|TerrainClass)$). (b) illustrates the final probabilities after integration of behavior data.

7 Conclusion

This paper presents a method for estimating on-line the navigation mode to be apply on a rover endowed with several ways of achieving navigation in an outdoor environment. It is based on a Hidden Markov Model which uses two kinds of information: context observation (classification of the terrain on the basis of the evaluation of a *difficulty*) and behavior information provided by monitors. Three of them have been developed and presented, many others may be added for the benefit of the whole system. The experimentation show the interest of such an approach, especially when context observation was not able to make the right assumption about the terrain, which is exhibited by the detection of a behavior issue. There are numerous perspectives to that work, including the development of many other monitors to benefit from more behavior information, the use of additional navigation modes such as path following, and a thorough experimentation campaign to compare performances with other rover navigation methods.

References

1. Elise Arnaud, Etienne Memin, and Bruno Cernuschi-Frias. Conditional filters for image sequence based tracking - application to point tracking. *IEEE Transactions on Image Processing*, 14(1):63–79, 2005.
2. R. Dearden and Dan Clancy. Particle filters for real-time fault detection in planetary rovers. In *Proceedings of the 12th International Workshop on Principles of Diagnosis*, 2002.
3. S. Funiak and B. Williams. Multi-modal particle filtering for hybrid systems with autonomous mode transitions. In *Proceedings of the 13th International Workshop on Principles of Diagnosis*, 2003.
4. Ch Grand, F BenAmar, and Ph. Bidaud. Kinematic analysis and stability optimisation of a reconfigurable legged-wheeled mini-rover. In *SPIE'02 : Unmanned ground-vehicle technology IV*, 2002.
5. S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury, M. Herrb, and R. Chatila. Autonomous rover navigation on unknown terrains: Functions and integration. *International Journal of Robotics Research*, 2002.
6. J. Minguez, J. Osuna, and L. Montano. A "divide and conquer" strategy based on situations to achieve reactive collision avoidance in troublesome scenarios. In *IEEE International Conference on Robotics and Automation*, 2004.
7. T. Peynot and S. Lacroix. Enhanced locomotion control for a planetary rover. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.
8. T. Peynot and S. Lacroix. A probabilistic framework to monitor a multi-mode outdoor robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.