A Probabilistic Framework to Monitor a Multi-Mode Outdoor Robot

Thierry Peynot and Simon Lacroix
LAAS-CNRS
7, av. du Colonel Roche
31077 Toulouse Cedex 4 - France
{*Thierry.Peynot,Simon.Lacroix*}@*laas.fr*

***Abstract*— This paper presents an approach to autonomously monitor the behavior of a robot endowed with several navigation and locomotion modes, adapted to the terrain to traverse. The mode selection process is done in two steps: the best suited mode is firstly selected on the basis of initial information or a qualitative map built on-line by the robot. Then, the motions of the robot are monitored by various processes that update mode transition probabilities in a Markov system. The paper focuses on this latter selection process: the overall approach is depicted, and preliminary experimental results are presented.**

***Index Terms*— Navigation and Locomotion Modes, Markov Chain, Probabilistic Monitors.**

I. Introduction

The autonomous navigation of an outdoor robot imposes to consider a wide spectrum of situations. For robustness and efficiency reasons, it is highly desirable that the robot is endowed with various operating modes, adapted to the kind of terrain it has to cross: indeed, evolving on perfectly flat terrains, in rough areas, or following an existing path call for different perception, decision and action processes [4]. This is all the more true when the considered chassis is able to evolve in various locomotion modes, such as the robots WorkPartner, Hylos [6], Nomad or the Marshokhod rovers.

If the literature now abounds with definitions of approaches to autonomous navigation in the context of path following [9], obstacle avoidance in poorly or highly cluttered environments [10] or rough terrain traverse [7][2], there are not many contributions that deal with robots endowed with several operating modes: usually the selection of these modes is made by an operator. Besides, the evaluation of the current mode of a system is generally used to make a passive state estimation, most of the time dedicated to the issue of fault detection, as in [5] and [3].

This paper proposes a selection process of the most suited operating mode to be applied on board an outdoor mobile robot, mainly on the basis of qualitative data illustrating its behavior. This is a two-step process:

- An *a priori* selection is performed on the basis of information related to the terrain to traverse,
- An *a posteriori* selection is performed on the basis of information provided by processes that evaluate on-line the efficiency of the current mode or the accuracy of the current context.

The paper focuses on this latter selection process. It presents a probabilistic framework to monitor the execution of the robot motions by checking if the current operating mode is adapted to the current situation, and possibly switching to a different mode. The next section depicts the approach: it specifies the notion of *operating mode*, the two decision steps to select the most suited mode, and describes how monitoring processes are used to update mode transition probabilities in a Markov system. Sections III and IV show the system currently under construction on two robots, with more precise examples of modes available and monitors used, and some experiments that have been made so far.

II. Framework

A. Modes to be selected

We distinguish two levels of operating modes: *navigation modes* that specify the perception/decision/action loop performed, and *locomotion modes* that specifies the way the motions are achieved.

- *Navigation mode:* This mode specifies both perception and decision functionalities that define the motions to execute. The perception functionalities are various methods to represent the environment or to detect obstacles, and the decision about the motion consists in choosing the best trajectory or elementary motions to achieve. Some examples are: road following, elementary trajectory planning, potential field navigation.
- *Locomotion mode:* Once the *Navigation mode* is selected, the *Locomotion mode* has to be chosen among those available. This mode specifies the way the robot will follow the trajectory or execute the motion previously selected (*e.g.* adjust the speed to be applied, use simple rolling or *wheel walking* [1]).

These two modes levels correspond to loops with different frequencies: navigation loops can typically run once every few seconds, whereas locomotion loops are run at several tens of Hertz. They are organized along a hierarchy (fig. 1), several locomotion modes being defined for each navigation mode.

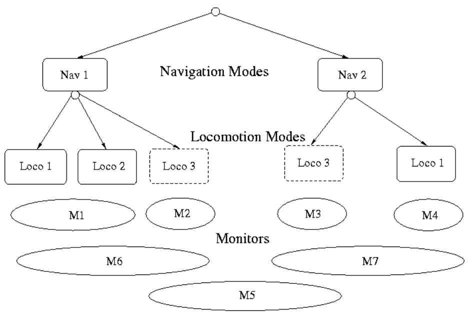

Fig. 1. The two levels of modes (Navigation and Locomotion) and associated monitors.

B. Monitoring

To select the most suited modes, two different processes that rely on different kinds of information are used: *a priori* information about the environment and information provided by an *on-line monitoring*.

1) "A priori" selection: Given an area to traverse, the partial probabilities that a given mode is suited are considered as provided. Such probabilities can be set by an operator on the basis of initial knowledge about the terrain, or autonomously set by a decision level that relies on the analysis of information gathered by the robot itself. For instance, this can be done using a probabilistic environment model that describes the terrain in terms of *navigability classes* [4].

2) On-line Monitoring: At each level, each mode has one or several corresponding *monitors*, activated when the mode is applied (see fig. 1). The goal of these monitors is to evaluate on-line the behavior of the current mode (*e.g.* locomotion efficiency) or to evaluate if the current mode remains adapted to the situation (*e.g.* checking that the conditions which led to the selection of this mode still hold). Some "generic" monitors can be applied to several different modes, others are specific to a given mode. Besides, several monitors can be applied for a given particular mode (see fig. 1). Each monitor computes a "bad behavior" probability for the current mode or a probability that this mode is actually not adapted to the current situation. The combination of probabilities provided by the monitors is used by the mode selection process to update transition probabilities towards other modes.

C. Mode Selection Process

At each level, the *a priori* and *on-line* probabilities provided by the monitoring are used in a Markov Chain [13] which calculates on-line the probability that each mode is the most adapted to the current situation (for example, see figure 2 for the locomotion mode selection). In this Markov Chain, states correspond to modes that could be applied. The best mode is the one associated to the state with the highest probability. The *a priori* probabilities are used as probability densities ($p(O_t|m_k)$ in eq. (1)) whereas the information from the *on-line monitoring* is used to compute transition probabilities, after the combination of the probabilities provided by the various monitors available.

A probabilistic Markov system is used because the evolution in time of the robot situation gives useful information (one can assume that the situation will not change too significantly in just one period of time), and also to avoid too frequent mode changes: the current mode should not change unless enough evidence that it is not the best suited has been collected.

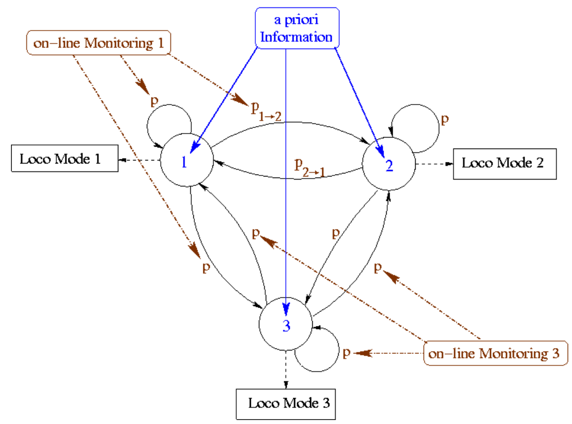

Fig. 2. Example of Markov Chain for the locomotion mode selection, with three locomotion modes available.

If the observation O_t is made at time t, the probability that m_k is the most appropriate mode to be applied is:

$$p(m_k|O^t) = \eta_t \, p(O_t|m_k) \sum_{i=0}^{K-1} p_{ik} \, p(m_i|O^{t-1}) \qquad (1)$$

where:

- O^t are all the observations made until time t.
- p_{ik} is the probability of the transition from mode i to mode k. $p_{ik} = q_{ik} \, cost_{ik}$, where q_{ik} is a transition probability provided by the *on-line* monitoring and $cost_{ik}$ is a cost associated to switching from mode i to mode k.
- K is the number of modes.
- $p(O_t|m_k)$ is the probability that observation O_t is made knowing that the rover is in mode m_k. This information is provided by the analysis of the terrain (*a priori* selection).
- η_t is a normalization coefficient

Too frequent and unnecessary mode changes should be avoided, as they might require more or less heavy operations, such as re-initializations or even a robot stop. It is also not desirable that the robot "jitters" from one mode to another.

Hence, $cost_{ik}$, a *mode change cost*, is introduced in the transition from mode i to mode k. These costs, determined *a priori* by an operator, are also a way for the operator to specify the desired behavior of the robot on given areas (*e.g.* the robot should be rather quicker or safer).

Some modes are adapted to some contexts. Consequently, we should estimate on-line the current context (which is done by the monitors) and select the right mode according to this information. Furthermore, we want the system to find a better solution (another mode) when the current behavior is not satisfying, which is also evaluated by on-line monitors. Thus, the q_{ik} probability for transition from mode i to k will be a combination of probabilities provided by the monitors estimating the current context and behavior of the rover, using the Bayes formula.

At any time, the current working mode m_c is known: the only transitions that can occur are from this mode to another, and the only monitors that can be applied are the ones associated to m_c. Therefore, only some of the transitions depicted in figure 2 are actually computed, and all the other transition probabilities are set to zero. However, the *a priori* probabilities for the modes to be applied are always available. Hence, if m_c is the current mode at time $t-1$, the probability that m_k is the most appropriate mode to be applied at time t is:

$$p(m_k|O^t) \;=\; \eta_t \, p(O_t|m_k) \, p_{ck} \, p(m_c|O^{t-1}) \qquad (2)$$

At time t, the best mode (thus the selected mode), is m_s if $p(m_s|O^t) = max_k(p(m_k|O^t))$.

The two following sections illustrate two implementations of the mode selection process, each on a different outdoor robot. The first one concerns the selection and monitoring of locomotion modes and the second one of navigation modes.

III. Locomotion Modes Monitoring

This section presents locomotion modes selection when the robot is in the *RoughNav* navigation mode (see IV-B.2). After a short presentation of the robot Lama, locomotion modes and an associated monitor are illustrated, and preliminary results are shown.

A. The robot Lama

Lama is a Marsokhod rover (fig. 3). Its chassis is composed of three pairs of independently driven *non-directional* wheels, mounted on 3 axles that can roll relatively to one another, thus giving high obstacle climbing skills (at least on cohesive soils). In addition to the traditional rolling mode, Lama has another locomotion mode, called *peristaltism* (wheel walking) [1]. In this mode, each axle moves one after the other, making the robot *crawl* as a caterpillar. It is particularly well adapted to low cohesion soils (e.g. sand, gravels).
Lama is endowed with the following sensors:
• 6 high resolution optical encoders,
• a 2 axis inclinometer providing the robot attitude by measuring the roll and pitch,
• potentiometers measuring the angles α_1, α_2, β_3 that define the robot chassis configuration (fig. 3),
• a precise fiber-optics gyrometer providing the yaw rate,
• a stereo-vision bench mounted on a mast.

Fig. 3. Lama

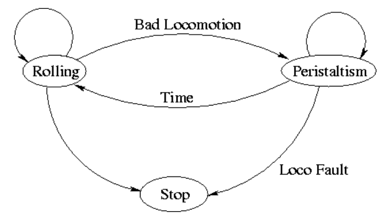

Fig. 4. Locomotion Modes Markov chain selection

B. Locomotion Modes

The two locomotion modes that can be used on a rover like Lama are *rolling* (with an enhanced wheels speed command [11]), and *peristaltism*.
On robots which have several reconfiguration abilities other locomotion modes may be exploited. Some modes can also correspond to different speeds (high speeds on easy and rather flat terrains, slower speeds on uneven and rough terrains).

C. Modes Selection

Figure 4 shows the locomotion modes selection on board Lama. The two available modes are *rolling* and *peristaltism*. The only monitor used for now is the *locomotion efficiency estimator* presented below. If *usual rolling* is not efficient, the robot switches to *peristaltism*. The corresponding transition probability is: $p(Rolling, Peristaltism) = p(LocoFault|O^t)$, provided by that monitor (with a corresponding cost set to 1.0). When *peristaltism* mode is active, if there is no locomotion fault detected, the system simply switches back to *rolling* mode after an elapsed time previously decided by the operator, as in this mode the robot has to drive very slowly. On the other hand, if a locomotion fault is detected by the monitor, we switch to the Stop mode, the default mode in case of problem or any unknown situation.

D. On-line Monitoring

1) Locomotion Efficiency Monitoring: This method is based on a locomotion monitoring process presented in [11]. It provides partial probabilities of being in one of the three following states: efficient locomotion, slipping situation, and locomotion fault (which means that the current locomotion of the rover is no more efficient at all). It is particularly useful to detect situations when there are significant slippages on some wheels. If such a locomotion fault is detected (fig. 5), a rover like Lama should switch to the *peristaltism* mode.

The probabilities are computed as follows. Three *speeds coherence indicators* are used as features in a probabilistic classification procedure: a Markov process evaluates the robot situation on-line according to the previous probabilities calculated, the current values of the features, and their comparison with prototypes recorded during a supervised learning stage. These *speeds coherence indicators* are obtained by comparing different evaluations and measurements of linear and rotation speeds on board the robot that should be equivalent in the absence of slippages (see [11] for more details).

Fig. 5 shows an example of the results of locomotion efficiency monitoring (and locomotion fault detection) obtained on Lama in comparison with the opinion of a human operator.

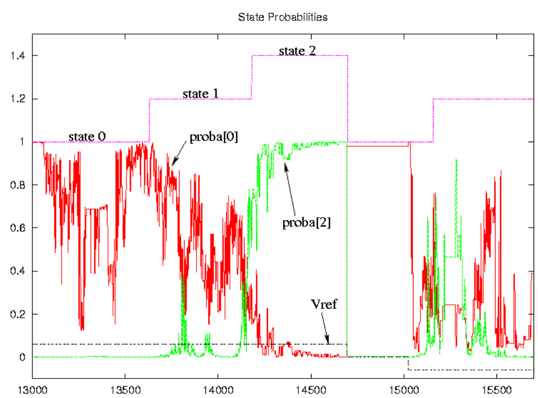

Fig. 5. Locomotion Fault Detection: comparison between the state probabilities calculated and the current state according to the operator. State 2 corresponds to a locomotion fault, state 0 to a good behavior of the current locomotion.

E. Results

Preliminary tests have been made with arbitrary fixed values for the *a priori* probabilities. The data used by the monitors to compute the transition probabilities were recorded on-line on board the robots but the computation of the probabilities for each mode has been made off-line. In order to evaluate the results of the mode selection, observations made by a human observer were recorded with these data.

Figure 6 shows an example of a locomotion mode switch as a consequence of the detection of a locomotion fault on Lama. The initially selected mode is *rolling*. Even though the *a priori* probability of that mode was set to 0.8, the system switches to *peristaltism* because of the evidence of a locomotion fault found by the monitor (which causes a high transition probability).

As one could expect, the lower the "*a priori*" probability for the initial mode, the quicker the system tends to switch modes.

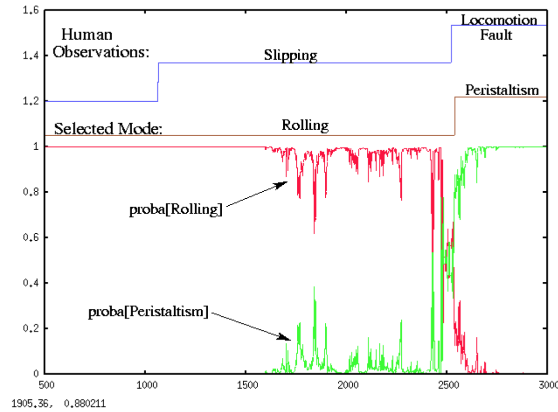

Fig. 6. Probabilities for a two locomotion modes system: *rolling* and *peristaltism*, and the corresponding mode selection. To illustrate the actual behavior of the robot, a human operator gives his observations at the same time.

Even though preliminary results already obtained are satisfying, the system should benefit from the implementation of more monitors. Indeed, other monitors can be considered, such as the comparison of position estimations provided by various independent means (*e.g. visual motion estimation* [8] vs. odometry).

IV. Navigation Modes Monitoring

This section presents an example of navigation modes selection used on board the robot Dala.

A. The Robot Dala

It is an *iRobot* ATRV (figure 7) equipped with the following sensors:

- a stereo-vision bench,
- a SICK 2-D laser,
- an Inertial Measurement Unit (IMU) measuring the three accelerations and angular speeds,
- odometry encoders,
- a fiber-optics gyrometer.

Dala does not have very high obstacle climbing skills, but its speed can reach 2 m/s.

Fig. 7. Dala

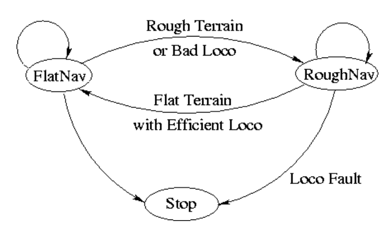

Fig. 8. Navigation Modes Markov chain selection on board Dala. The Stop mode is selected when a fault or an unknown situation is detected.

B. Navigation Modes

A navigation mode is a couple composed of a movement method and an associated way to perceive and represent the environment. The two modes currently used on Dala are:

1) Flat Terrain Navigation Mode (FlatNav): This mode corresponds to a reactive motion strategy based on the perception of obstacles around the robot provided by the 2-D SICK laser [10]. Hence, it is only adapted to flat terrains. High speeds may be applied to the robot in that mode.

2) Rough Terrain Navigation Mode (RoughNav): This mode is rather dedicated to uneven terrains. It combines a trajectory selector named *P3d* [2][4] with a stereo-vision algorithm which builds a Digital Elevation Map (DEM) on-line [4]. *P3d* chooses a trajectory which optimizes an interest/cost criteria. The interest is a distance to the goal, and the cost represents an integration of difficulties associated to the chassis attitude and internal configurations predicted thanks to a geometric placement function of the robot on the DEM (figure 9). Therefore, in addition to the selection of the *best* trajectory, *P3d* also provides *predicted attitude and configuration angles* for the robot along the chosen path. Contrary to the previous case, rather slow speeds are usually applied to the robot in this mode.

Fig. 9. A predicted placement of Lama on a DEM (left) and the corresponding placement on the real terrain (right)

C. Modes Selection

The *navigation modes* selection chain on board Dala is presented on figure 8. It involves the two navigation modes introduced above and two monitors: the *non-flat terrain detector* presented below and the *locomotion efficiency evaluator* (section III-D.1). When the Flat Terrain Navigation Mode (*FlatNav*) is active, the system tends to switch to *RoughNav* when the monitors estimate that the robot is driving over a non-flat area (rough area) or that the current locomotion is inefficient. On the contrary, it tends to switch from *RoughNav* to *FlatNav* when the terrain is flat and the current locomotion rather efficient. Let RT stand for *Rough Terrain*, FT for *Flat Terrain*, BL for *Bad Locomotion* (inefficient locomotion) and GL for *Good Locomotion* (efficient locomotion). Considering that the monitors are independent, the probability $q(FlatNav, RoughNav)$ (transition probability from *FlatNav* to *RoughNav* without cost) is: $P(RTorBL) = P(RT) + P(BL) - P(RT)P(BL)$.

And the opposite probability $q(RoughNav, FlatNav)$: $P(FTandGL) = P(not(RTorBL)) = 1 - P(RTorBL)$.

D. On-line Monitoring

We need some monitors to check if the current mode is well adapted by evaluating its behavior and verify the current context, and then decide whether it should be changed.

1) No Flat Terrain Detection: In the *Flat Terrain Navigation Mode* (*FlatNav*) a monitor should check whether the terrain on which the robot is navigating is flat. For that purpose, we have developped a monitor that uses information provided by a 3-axis IMU (Inertial Measurement Unit) to detect non-flat areas on the terrain. Indeed, as long as the terrain is flat, roll and pitch gyrometers should only measure some noise (with bias), and the z accelerometer the gravity perturbated with noise and bias. If some other patterns are detected in at least one of the three signals, it means that the robot has met some uneven region (fig. 10). Besides, an area should not be classified as non-flat too quickly to avoid mode switching just because of a single little rock lost in the area. On those accounts, the data used to detect non-flat terrains is the integration of an "energy" of roll/pitch rates and z acceleration over a one second time window: $\sum_{i=1}^{K} x^2/K$, where K is the number of samples. Figure 10 shows an illustration of that energy with flat and uneven areas.

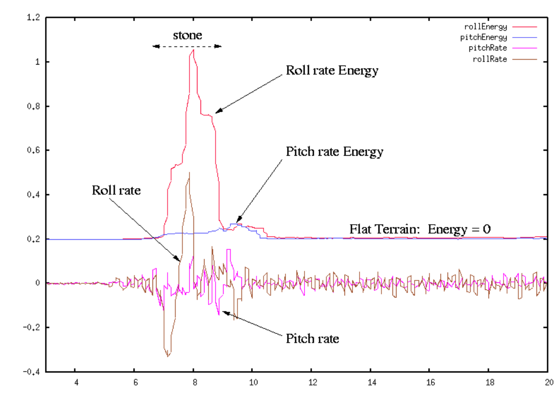

Fig. 10. Roll and pitch rates and their "energy" when the robot drives over a stone, on an otherwise flat terrain.

As the detection of non-flat areas may be exhibited only by one of the energies computed, their maximum at each time sample is extracted. Knowing the standard energy level when the terrain is flat and situations proved to be uneven terrains, we can compute a value similar to a probability that the terrain is non-flat (the higher the energy level, the higher the probability that the current terrain is uneven). The result obtained is used in the transition probability computation. towards modes different from *FlatNav*.

When the probability to detect a non-flat area is significant, it means the *FlatNav* mode is no more relevant and that the system should switch to another navigation mode, adapted to uneven terrain.

This monitor can be also activated with rough terrains navigation modes: the more the terrain is estimated to be flat, the higher the transition probability to a flat terrain navigation mode.

2) Locomotion Efficiency Monitoring: Even though that monitor has been designed to evaluate the behavior of locomotion (see III-D.1), the information provided is also used in the navigation mode selection process, as an unadapted navigation mode might lead to inefficient locomotion. For instance, if monitors reveal such a problem in the *FlatNav* mode, the robot should switch to *RoughNav*, because usually that mode is better adapted to difficult situations for the locomotion.

3) Configurations Incoherence Detection: This monitor can be useful to evaluate both the behavior of the robot's locomotion and the efficiency of the Rough Terrain Navigation Mode. It uses the configurations and attitudes predicted by *P3d* (section IV-B.2) and associated to positions on the DEM, comparing them on-line to configurations and attitudes directly measured or estimated on board the robot. On board Dala, they are derived from the IMU measurements, using an estimator (x-observer) based on the back-stepping technique (adapted from [12]). The result of that comparison provides a probability that the robot's attitude is the one predicted. The further the current situation from the prediction the more the behavior of the active mode is considered as bad: it could mean that the present locomotion of the robot is not efficient or that the navigation mode is not well adapted to the actual situation. This monitor is currently being developped.

E. Results

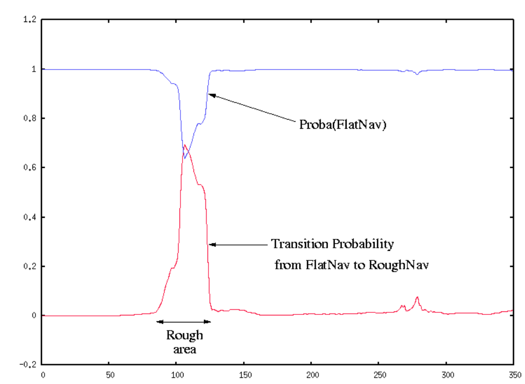

Fig. 11. Navigation modes selection: transition probability from *FlatNav* to *RoughNav* and probability that *FlatNav*, the initial selected mode, remains the best one.

Figure 11 shows an example of the result of a Navigation mode selection in the same situation as in figure 10. The initial mode selected is *FlatNav*, with an *a priori* probability of 0.8. Even though a non-flat area has been detected by the monitors (with an efficient locomotion), there was not enough elements to cause a mode switch to *RoughNav* in that situation, as the robot has only met a few stones in an otherwise flat terrain.

V. Conclusion

We have presented preliminary work on an approach to monitor the behavior of an outdoor robot endowed with several navigation and locomotion modes. We believe that such a system is necessary to improve the autonomy of a mobile robot, considering the variety of situations it has to deal with. The system presented aims at selecting on-line the most adapted modes, according to the terrain, on the basis of a *qualitative* evaluation of the *actual behavior* of the robot, so as to improve its *global* behavior over a complete mission, and avoid failures. To provide that information, efficient monitors are needed; some have been proposed, but the system should benefit from the implementation of as much relevant monitors as possible (the more monitors, the better the information on the robot's behavior). It should also benefit from the implementation of other navigation and locomotion modes. Such developments and an evaluation of the system on board the robot during long-term navigation missions are currently being investigated.

References

[1] G. Andrade, F. Ben Amar, P. Bidaud, and R. Chatila. Modeling robot-soil interaction for planetary rover motion control. In *IROS*, 1998.

[2] D. Bonnafous, S. Lacroix, and T. Simeon. Motion generation for a rover on rough terrains. In *IROS*, 2001.

[3] 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.

[4] S. Lacroix et al. Autonomous rover navigation on unknown terrains: Functions and integration. *International Journal of Robotics Research*, 2002.

[5] 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.

[6] 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.

[7] K. Iagnemma, F. Genot, and S. Dubowsky. Rapid physics-based rough-terrain rover planning with sensor and control uncertainty. In *IEEE International Conference on Robotics and Automation*, May 1999.

[8] A. Mallet, S. Lacroix, and L. Gallo. Position estimation in outdoor environments using pixel tracking and stereovision. In *IEEE International Conference on Robotics and Automation, San Francisco, Ca (USA)*, pages 3519–3524, April 2000.

[9] D. Mateus, G. Avina, and M. Devy. Robot visual navigation in semi-structured outdoor environments. In *ICRA*, 2005.

[10] J. Minguez, J. Osuna, and L. Montano. A "divide and conquer" strategy based on situations to achieve reactive collision avoidance in troublesome scenarios. In *ICRA*, 2004.

[11] T. Peynot and S. Lacroix. Enhanced locomotion control for a planetary rover. In *IROS*, 2003.

[12] Jean Michel Pflimlin, Tarek Hamel, Philippe Soures, and Najib Metni. Non linear attitude and gyroscope's bias estimation for a vtol uav. In *16th IFAC world congress, Prague*, 2005.

[13] V. Verma, R. Simmons, and J. Fernandez. Probabilistic models for monitoring and fault diagnosis. In *Joint Workshop on Technical Challenge for Dependable Robots in Human Environments*, 2002.