A Case Study in Machine Intelligence: Adaptive Autonomous Space Rovers

Raja Chatila and Simon Lacroix
LAAS / CNRS
7, Ave du Colonel Roche
31077 Toulouse Cedex 4 - France

Abstract

Planetary exploration by robots is a demanding challenge for roboticists: the constraints of this application and the requirement to autonomously achieve long traverses raises major requirements on the robot conception. A robot capable of performing such an exploration would definitely be an exemplary instance of an "intelligent machine". We present here our approach to autonomous long range navigation in poorly known unstructured environments: we show how it is necessary to produce representations at several levels of abstraction, and that a certain degree of deliberation is necessary for the robot in order to anticipate events, take efficient decisions, and react adequately to unexpected events.

1 Introduction

An autonomous mobile robot offers a challenging and ideal field for the study of machine intelligence. Decisional and operational autonomy, as it could be measured by the robot's effectiveness and robustness in carrying out tasks in different environment conditions, raises fundamental questions to the roboticists: it especially requires suitable interactions between perception, deliberation and action.

Among the wide range of real-world applications for intelligent machines, special interest has been devoted to intervention mobile robots that have to perform tasks in ill-known environments, which are often remote or of difficult or dangerous access (demining, civil security...). Outstanding case studies are certainly rovers for planetary exploration: communication constraints (bandwidth, delays and communication windows) voids the possibility to efficiently teleoperate the machine. Moreover, this application brings forth specific constraints that calls for a very high level of autonomy:

- Scientists on Earth should be able to control the robot, *i.e.* to send him specific missions, that might require *long range* traverses.
- The environment is complex and ill-known: missions cannot be planned *a priori*, and can only be defined a quite high level of abstraction ("reach that hill", "map this area"...): the robot must interpret them according to its actual context.
- Finally, the robot engineers may need to troubleshoot the robot (it could fall into difficult situations where its capabilities can not help), but also to reconfigure it according to previous mission execution reports.

The problem of long range navigation in unknown outdoors environments has not been very frequently addressed yet. Important achievements are Robbie [Weisbin *et al.*, 1992], Ambler [Simmons and Krotkov, 1993] and the navigation of the UGV [Stentz and Hebert, 1994]. At LAAS, we have tackled various aspects related to this problem, and experimented some in realistic conditions.

In this paper, we summarize our approach and present our main contributions to autonomous long range navigation, which exhibits the features which we believe an intelligent machine should be endowed with. In the next section we present our approach to address the autonomous long range navigation problem, in the context of planetary exploration: it strongly relies on the ability to *(i)* autonomously build environment representations, and to *(ii)* reason on these representations in order to plan and execute actions. Section 3 is devoted to the presentation of an integrated architecture allowing a mobile robot to plan its tasks, taking into account temporal and domain constraints, and to perform corresponding actions and to control their execution in real-time, while being reactive to possible events. A discussion concludes the paper.

2 Autonomous long range navigation

Among the various tasks a planetary exploration robot should be able to carry out autonomously, navigation is of course one of the most important. By navigation, we understand the task of reaching a distant goal (*long range navigation*), the path being not predefined or given to the robot. The complexity of the processes involved by this task depend on the general context in which it is to be executed: in the case of planetary exploration, the fact that the environment is initially poorly known, varying from one area to the other and sometimes very rough has strong implications on perception, action planning and motion execution processes. We present here an approach to autonomous long range navigation we have been developing and experimenting with the robot Adam until 1995, and now with the robots Eve and Lama[1].

2.1 An Adaptive Approach

According to a general "economy of means" principle due to limitations of on-board processing capacities, memory and energy, and to achieve an efficient behavior, the robot should be *adapt* to the nature of the terrain [Lacroix *et al.*, 1994; Chatila *et al.*, 1995]. Basically, two kinds of modes are considered:

- **Reflex** modes: when the environment is "easy", *i.e.* rather flat and lightly cluttered, the robot can efficiently move just on a basis of a goal to reach[2] and informations provided by "obstacle detector" sensors. In essentially flat environments, informations returned by the observation of a laser stripe might be sufficient; when the environment is known to contain obstacles, a broader perception (as provided by stereovision) is required in order to safely execute avoiding maneuvers.
- **Planned** modes: reflex modes become inefficient when applied in more complex environments, in which the robot can be trapped in dead-ends for instance. In such cases, trajectory planners that reason on a model of the environment are required. Depending on the difficulty of the area to cross, a 2D planner can find out ways between obstacles, using a description that exhibits traversable and non-traversable areas, or a 3D planner that checks stability and collision constraints thanks to a fine 3D model of the environment has to be run.

The existence of different motion modes enables to choose a well adapted and efficient behavior depending on the terrain, but it complicates the system, that has to deal with several different terrain representations and motion planning processes. And especially, it requires the ability to select the adequate motion mode to apply: this is realized thanks to a specific planning level, the *navigation planner* [Lacroix *et al.*, 1994]. This planner is *systematically* activated at each step of the incremental execution of the task: each time 3D data are acquired, they are analyzed to provide a description of the perceived zone in terms of navigation classes. This description is fused to maintain a *global qualitative representation* of the environment (the *region map*), on which the navigation planner reasons to select a sub-goal, a motion mode to apply and the next perception task to execute. The introduction of this planning layer defines a particular instance of the usual "perception-decision-action" loop, in which the "decision" part is split into two distinct and hierarchically layered processes: navigation and trajectory planning.

2.2 Environment representations

Each of the different motion modes requires a particular terrain representation[3], the navigation planner also requires a specific terrain representation, and during navigation, an exteroceptive localization process has to be activated frequently to update robot position with respect to environment features, which obviously also requires a terrain representation. Aiming at building a "universal" terrain model that contains all the necessary informations for these various processes is extremely difficult, inefficient, and moreover not really necessary. It is more direct and easier to build different representations adapted to their use: the environment model is then *multi-layered and heterogeneous*, and perception is *multi-purpose*: several modeling processes coexist in the system, each dedicated to the building of specific representations.

The region map

The region map is intended to be the basis on which the navigation layer reasons. Each time 3D data are acquired, it is build thanks to a fast classification process that produces a description of the perceived areas in term in *terrain classes.* It relies on a specific discretisation (figure 1) of the perceived area, that defines "cells" on which different relevant characteristics (attributes) are determined. A non-parametric Bayesian classification procedure is used to label each cell: a learning phase based on prototypes classified by a human lead to the determination of probability density functions, and the classical Bayesian approach is applied, which provides an estimate of the probability for each cell to correspond to one of the defined terrain classes ({*Unknown, Flat, Rough, Obstacle*}). A decision function that privileges false alarms instead of the non-detections is performed.

When used with stereovision, the geometric classifier is enriched by a *terrain nature* classifier that reasons on texture attributes computed on "luminance cells" (defined as regions in the original image from the geometric

[1] For informations concerning these robots, see www.laas.fr/˜matthieu/robots

[2] not necessarily the distant goal, it can be a sub-goal that the robot selected

[3] besides local informations, the terrain representation required by reflex modes is just the description of the borders of the region within which they can be applied

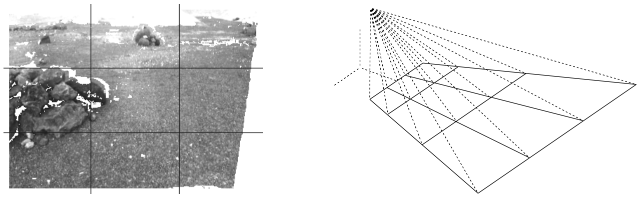

Figure 1: *Discretisation of a 3D stereo image: regular Cartesian discretisation in the sensor frame (left - only the correlated pixels are shown), and its projection on the ground (right - the actual discretisation is much finer)*

cells using the correspondence 3D points/pixels). The procedure can also be used as an obstacle detector for rather flat terrains, using a two classes ({*Flat, Obstacle*}) data base (figure 2).

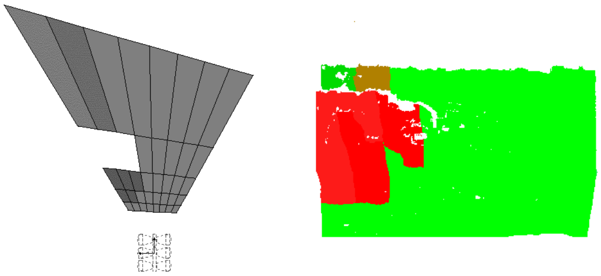

Figure 2: *Binary classification of the stereo image of figure 1: perceived area (left) and reprojection in the sensor frame (right). The grey levels represent the partial probability $P(Obstacle)$*

This technique proved its efficiency and robustness on several hundreds of 3D images, using various prototype data bases. Its main interest is that it provides an estimate of the confidence of its results: this information is given by the *entropy* of a cell. Moreover, a statistical analysis of the cell labeling confidence as a function of its label and distance to the sensor defines a predictive model of the classification process (section 2.3).

The partial probabilities of a cell to belong to a terrain class allow to perform a fusion procedure of several classified images. The fusion procedure is performed using a bitmap: for the purpose of navigation planning, the bitmap model is structured into a global region map that defines a connection graph, whose nodes are on their borders, and whose arcs correspond to a region crossing. To each region are associated the terrain classes partial probabilities, the decided label, the confidence on this labeling and elevation informations (figure 3).

Geometric representations

To plan trajectories, a precise geometric model is required. The binary representations (Traversable/Non-Traversable) required by a 2D trajectory planner can be easily built on the basis of the output of any obstacle detection procedure [Matthies *et al.*, 1995]. Note that there are some advantages to enrich such representations with a score that represents a confidence in the partition (as can be done using our classification procedure for instance): it allows to consider costs based on a *risk* to plan paths or to generate motion commands.

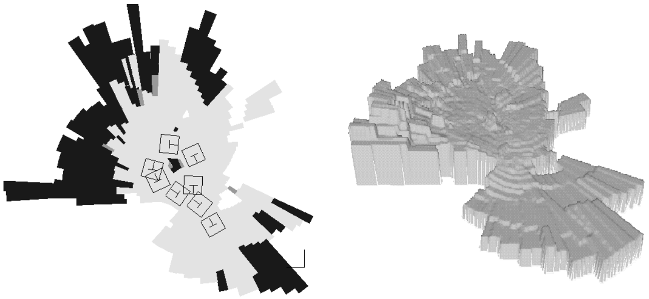

Figure 3: *A region map build by the fusion of 8 classified images: terrain classes (left) and elevation (right). Note that this is not a numerical representation.*

When it comes to plan trajectories on rough terrains, a *numerical terrain model* is required (figure 4). Although there has been several contributions to this problem [Kweon and Kanade, 1992], the problem has still not been addressed in very satisfactory way [Hoffman and Krotkov, 1991]: the main difficulty comes from the uncertainties on the 3D input data, that can be fairly well estimated, but hardly propagated throughout the computations and represented in the grid structure.

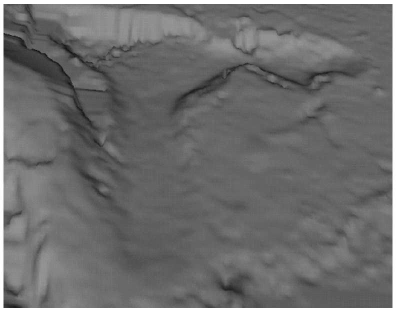

Figure 4: *A portion of a numerical terrain model built on the basis of several stereo images*

Localization

The ability for the robot to self-locate with respect to its environment is a key problem that must be solved to tackle long range navigation: otherwise, reaching a given goal (that is not in sight) will be impossible in general, considering the uncertainties of odometry or inertial sensors. Moreover, a good estimation of the robot position is mandatory to ensure the consistency of all the environment representations build from sensor data. To address this critical (and difficult) problem, *all* the proprioceptive and exteroceptive sensors are useful.

We split this problem in three sub-problems, according to the level of abstraction of the data to be processed:

- Motion estimation: performed continuously while the robot is moving, this process actually do not requires any terrain representation. It is mainly performed using the robot proprioceptive sensors, but can benefit from the processing of exteroceptive sensors (optical flow, image tracking...).
- Exteroceptive dead-reckoning (an initial position estimate is known): it is unrealistic to consider that the motion estimation will give sufficiently precise data to guaranty a good estimation of the robot position. One must correct this estimation on the basis of *landmarks* extracted in the environment as often as possible.
- Full localization (no initial position estimate is known): even with exteroceptive dead-reckoning, one can not make sure that the robot do not get lost. This can happen after traveling a large landmark-free area for instance. In such a case, when the robot comes back to previously visited areas, a localization without any initial estimate calls for *recognition capacities*, not only for landmarks, but also for *places*. Such an ability can be of a good help to maintain a global spatial consistency of the environment. Indeed, one can not make sure that the robot is always precisely positioned with respect to a certain global frame; but a large environment model can be described as a set of areas, to each of which a frame is attached, but loosely relatively positioned.

A lot of work has been devoted to the exteroceptive dead-reckoning problem. In outdoor terrains, correlation techniques on relevant parts of a numerical terrain model can be used. However, these techniques suffer from the difficulty to represent the data uncertainties in such models. *Object-based representations* appear to be more robust and efficient to tackle this problem. We have developed a peak detection procedure[Betge-Brezetz *et al.*, 1995], applicable when the terrain has been classified as flat but cluttered with obstacles: we extract by a segmentation process the objects that are lying on the ground. The salient objects are selected, and the coordinates of their peak is computed, along with an accuracy. This defines a *2D localization model* (figure 5), which is used to refine the robot position using a Kalman filtering technique.

Reducing the landmarks to the coordinates of their peaks can be sufficient for dead-reckoning, but is mot of a good help to recognize them. Object modeling and recognition has not been very studied in the context of outdoor unstructured environments. We are currently considering the adaptation of contour based modeling techiniques [Mallet and Lacroix, 1998] and of free-form modeling techniques used in the field medical 3D imaging [Delingette *et al.*, 1992] to represent rocks (figure 6).

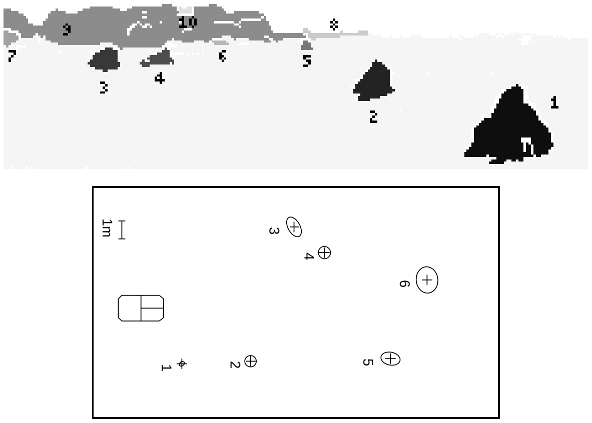

Figure 5: *Segmented objects (top) and 2D localization model (selected landmarks with their uncertainty according to sensor noise and resolution and object shape -bottom)*

A deformable mesh is applied to the 3D data points covering rocks (extracted thanks to a segmentation procedure), and attributes are computed on the obtained mesh (luminance and texture, normal to the surface when the robot attitude is known, and curvature otherwise). A spherical attribute image [Delingette *et al.*, 1992] is then determined: spherical attribute images obtained from various view-points can be compared thanks to correlation techniques, and the analysis of the correlation scores can help to recognize the rocks, or to estimate the relative view-points positions when the meshes match.

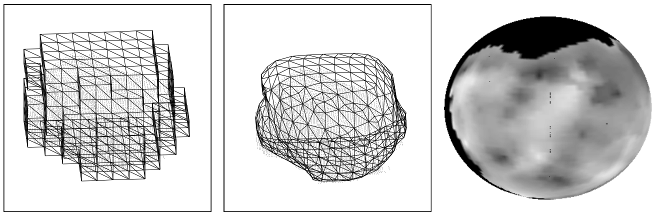

Figure 6: *Modeling a rock thanks to free-form surfaces: initial mesh (left), mesh after convergence to the data points (center), and mapping on a sphere of the curvatures extracted from the mesh (right)*

2.3 Planning

Navigation planning

Each time 3D data are acquired, classified and fused in the global region map, the robot has to answer autonomously the following questions:

- Where to go? (sub-goal selection)
- How to go there? (motion mode selection)
- Where to perceive? (data acquisition control)
- What to do with the acquired data? (perception task selection)

For that purpose, the navigation planner reasons on the robot's capabilities (action models for perception and motion tasks) and the global region map. A straightforward fact is that motion and perception tasks are strongly interdependent: executing a motion requires to have formerly modeled the environment, and to acquire some specific data, a motion is often necessary to reach the adequate observation position.

Planning motion tasks in an environment modeled as a connection graph is quite straightforward: finding paths in the graph that minimizes some criteria (time and energy for instance) is easily solved by classical search techniques, using cost functions that express these criteria. Planning perception tasks is a much more difficult issue: one must be able to predict the results of such tasks (which requires a model of the perception processes), and the *utility* of these results to the mission to achieve:

- Localization processes can be modeled by a function that expresses the gain on the the robot position precision, depending on the number and distance of perceivable landmarks;
- With the confidence model of the 3D data classification process, one can predict the amount of information a classification task can bring. But it is much more difficult to express the utility of a classification task to reach the goal: the model of the classification task cannot not predict *what* will be effectively perceived. It is then difficult to estimate the interest of these tasks.

A direct and brute force approach to answer the former questions would be to perform a search in the connection graph, in which *all* the possible perception tasks would be predicted and evaluated at *each* node encountered during the search. Besides its drastic algorithmic complexity, this approach appears unrealistic because of the difficulty to express the utility of a predicted classification task to reach the goal. We therefore choose a different approach to tackle the problem: the perception task selection is *subordinated* to the motion task. A search algorithm provides an *optimal* path, that is analyzed afterwards to deduce the perceptions tasks to perform. The "optimality" criterion takes here a crucial importance: it is a linear combination of time and energy consumed, weighted by the terrain class to cross *and the confidence of the terrain labeling.* Introducing the labeling confidence in the crossing cost of an arc comes to consider *implicitly* the modeling capabilities of the robot: tolerating to cross obstacle areas labeled with a low confidence means that the robot is able to acquire easily informations on this area. The returned path is not executed directly, it is analyzed according the following procedure:

1. The sub-goal to reach is the last node of the path that lies in a traversable area;
2. The label of the regions crossed to reach this sub-goal determines the motion mode to apply;
3. And finally the rest of the path that reaches the global goal determines the aiming angle of the sensor.

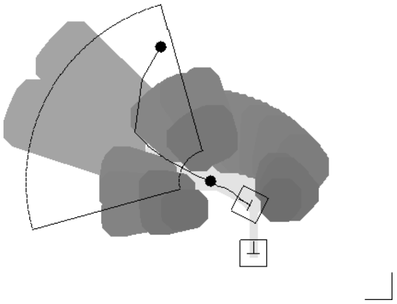

Figure 7: *A result of the navigation planner on a region map: the result of the analysis of the shortest path found can be interpreted as the answer to the question "what area should I perceive to reach the goal ?"*

The terrain modeling procedures and navigation planning algorithm have been intensively tested with the mobile robot Adam. We performed experiments on the Geroms test site in the French Space Agency CNES[4], where Adam achieved several `''Go To [goal]''` missions, traveling over 80 meters, avoiding obstacles and getting out of dead-ends (figure 8.

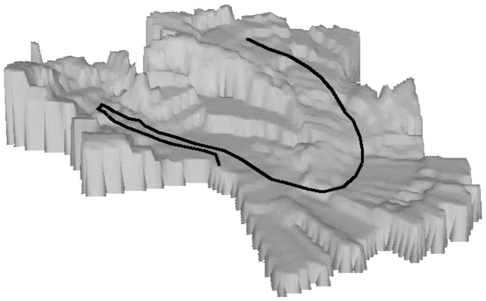

Figure 8: *Elevations encoded in the region map built during a dead-end exploration, and trajectory executed to reach the global goal (80 meters run).*

Trajectory planning

Planning geometric paths on flat terrains has been extensively addressed. But to cope with perception and motion uncertainties, it is more robust to plan the motion in terms of closed-loop sensor-based processes directly, rather than executing a geometrical trajectory relying on odometry or inertial data. Another approach is to compute a trajectory that takes into account possible error reduction by sensing. These "new generation" path planners [Latombe *et al.*, 1991] produce more realistic paths.

[4] Centre National d'Études Spatiales

On uneven terrain, irregularities are important enough and the binary partition into *free/obstacle* areas is not anymore sufficient: the notion of obstacle clearly depends on the capacity of the locomotion system to overcome terrain irregularities and also on specific constraints acting on the placement of the robot over the terrain (figure 9).

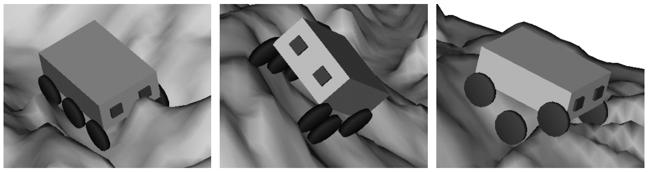

Figure 9: *The constraints considered by the 3D planner. From left to right: collision, stability and terrain irregularities*

We developed a planner [Simeon and Wright, 1993] that computes motions verifying such constraints by exploring a three dimensional configuration space $CS = (x, y, \theta)$ on a numerical terrain model. The planner incrementally builds a graph of discrete configurations that can be reached from the initial position by applying sequences of discrete controls during a short time interval. The minimum-cost trajectory returned by the planner realizes a compromise between the distance crossed by the vehicle, the security along the path and a small number of maneuvers (figure 10).

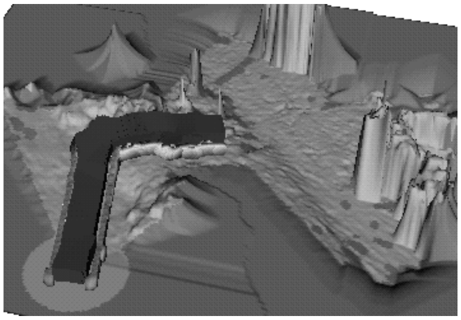

Figure 10: *A 3D trajectory planned on a real elevation map*

2.4 Motion execution

To guaranty reliable and safe motions, the robot must servo on its proprioceptive sensors, but also on its exteroceptive sensors [Miller *et al.*, 1992]. This is what happens during reflex motions, where the robot maneuvers are determined on the basis of a visual goal tracker and local terrain informations. Figure 11 presents an instance of reflex motions, using an artificial potential fields technique on local maps produced by a binary classifier [Haddad *et al.*, 1998].

Executing motions that have been determined thanks to a trajectory planner is a quite more difficult issue: one must then guaranty that the robot follows precisely the planned trajectory (which can sometimes be risky, especially on rough terrains). The adaptation in unstructured environment of visual servoing techniques, that become well mastered in structured environments, is here a critical point.

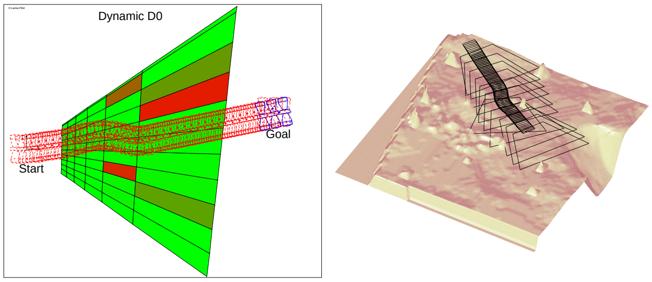

Figure 11: *Reflex motions using potential fields on a single local map (left), and integration of successive perceptions (left)*

3 An Architecture for Autonomy

The organization of the robot capacities is clearly a central issue. In order to reach a high level of autonomy, a robot control structure should have the following properties:

- **Programmability:** a useful robot cannot be designed for a single environment or task, programmed in detail. It should be able to achieve multiple tasks described at an abstract level.
- **Autonomy and adaptability:** the robot should be able to carry out its actions and to refine or modify the task and its own behavior according to the current goal and execution context as perceived.
- **Reactivity:** the robot has to take into account events with time bounds compatible with the correct and efficient achievement of its goals, including its own safety.
- **Consistent behavior:** the reactions of the robot to events must be guided by the objectives of is task.
- **Robustness:** the control architecture should be able to exploit the redundancy of the processing functions, which requires the control to be decentralized to some extent.

We briefly present here the generic concepts of an architecture that allows the integration of both decision-making and reactive capabilities, while satisfying these properties (a detailed presentation can be found in [Alami *et al.*, 1997]). The architecture is presented in figure 12, and has been fully instantiated in multi-robot cooperation experiments [Alami, 1996] .

3.1 The functional level

This level includes all the basic built-in robot action and perception capacities. These processing functions and control loops (image processing, obstacle avoidance,

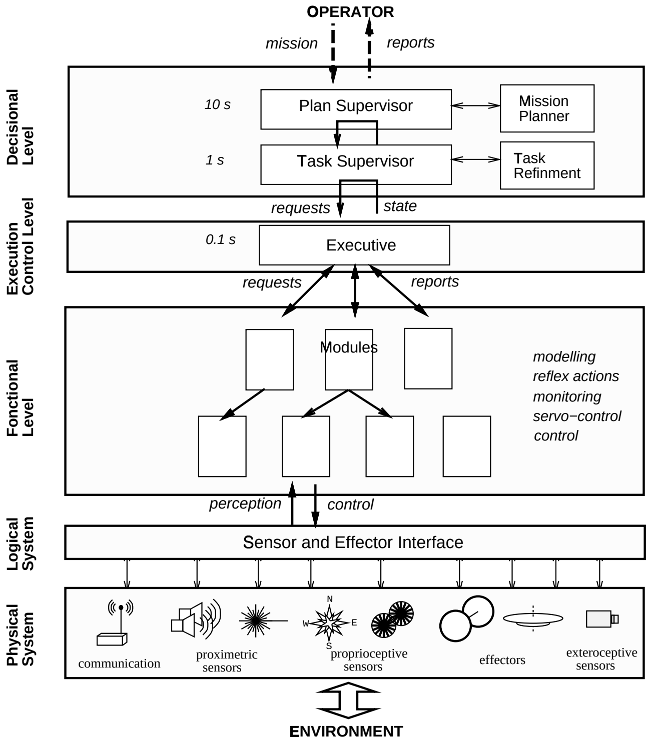

Figure 12: *An architecture for robot autonomy.*

motion control, etc.) are encapsulated into controllable communicating *modules* [Fleury *et al.*, 1994]. A module may read data exported by other modules, and output its own processing results in exported data structures. The organization of the modules is *not fixed*, their interactions depend on the task being executed and on the environment state. This is an important property that enables to achieve a flexible, reconfigurable robot behavior. Modules fit a standard structure, and are implemented thanks to a development environment, GenoM.

Note that in order to make this level as hardware independent as possible, and hence portable from a robot to another, the functional level is interfaced with the sensors and effectors through a *logical robot level*.

3.2 The Executive

This level controls and coordinates the execution of the functions distributed in the modules according to the task requirements. It is a pivot interface between the decision and functional levels. Thus it has the tricky function to actually fill the gap between decision and action, i.e., between the *slow rate* logical reasoning on *symbolic* data (0.1 s to few seconds), and the *higher bandwidth* computation on *numerical* data (10 Hz to 100 Hz).

The Executive is a purely reactive system, with no planning capability. It receives from the decision level the sequences of actions to be executed, and must select, parameterize and synchronize dynamically the adequate functions of the functional level. It sends *requests* to the functional level, generally by redirecting data previously produced by others activities (for instance a trajectory computed by a trajectory planner which is now to be executed): the Executive manages both the control flow and the data flow of the functional level, and solves possible conflicts, using priorities. Previously instantiated with a procedural reasoning system (PRS) during our first navigation experiments, the Executive is currently being developed using Kheops [Medeiros *et al.*, 1996], a compiler that transforms a set of propositional rules into an optimized decision network.

3.3 The decision level

This level includes the capacities of producing the task plan and supervising its execution, while being at the same time reactive to events from the previous level. This level may be decomposed into two or more layers, based on the same conceptual design, but using different representation abstractions or different algorithmic tools, and having different temporal properties. This choice is mainly application dependent.

In the case of long range navigation, the planner is IxTeT, a temporal planner, that produces a plan from a description of the state of the world and a goal. The "quality" of the produced plan is related to the cost of achievement of a given task or objective (time, energy...), and to its robustness, i.e., its ability to cope with non nominal situations. This last aspect is one of the motivations of our approach: besides providing a plan, the planner also provides a set of execution "modalities" expressed in terms of:

- constraints or directions to be used by a lower planning level if any;
- description of situations to monitor and the appropriate reactions to their occurrence; such reactions are immediate reflexes, "local" correcting actions (without questioning the plan), or requests for re-planning.

These "modalities" provide a convenient (and compact) representation for a class of conditional plans. However, the generation of modalities still remains to be investigated, we have no generic method yet for automatic generation of modalities in a planning algorithm.

4 Discussion

We believe that our approach exhibits important characteristics of a machine capable to behave "intelligently", *i.e.* in a rational way that can be measured by the robot's effectiveness and robustness in carrying out its missions. The two characteristics are the following:

- **Ability to build environment representations**: we are convinced that such an ability is mandatory to achieve long range navigation. A bunch of arguments can be developed (requirement to localize the robot, ability to memorize and recognize - a first step to learning, ability to anticipate...) to assess that a good understanding of the environment makes the robot more robust and efficient. Our approach exhibits that there is a need to deal with several adequate (purposive) representations. However, the problem of consistency management among these various representations is still an open issue.

- **Action planning and executing**: On-line decisional capacities for analyzing the context, anticipating situations, deciding of the relevant events to be expected (and focus some attention on them), and prepare the adequate reactions to them are necessary [Chatila, 1995]. Communicating a task to the robot at an abstract level clearly implies that it possesses the reasoning abilities to rationally select the actions to be executed to accomplish the objectives. To do this, the robot must *anticipate* the evolution of the environment. It must predict the outcome of its own actions and be able to compare it to the desired state, and this at a more or less long term, not just based on immediate stimuli. Therefore, there is a need in general to implement a *planning* capacity.

Adaptiveness is of course a key component of autonomy, if not the only one. Our approach is adaptive in essence, since we want the robot to be able to choose among a set of different motion modes. But choosing to plan actions instead of only reacting to sensor data gives also more adaptiveness to the robot, allowing him to anticipate events, and therefore to tackle them in a more efficient way. Finally, executing motions by servoing on exteroceptive data is a basic step to adaptiveness, especially required by safety.

References

[Alami *et al.*, 1997] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand. An architecture for automy. *To be published in International Journal of Robotics Research*, 1997.

[Alami, 1996] R. Alami. A multi-robot cooperation scheme based on incremental plan-merging. In *Robotics Research: The 7th International Symposium, Spinger Verlag*, 1996.

[Betge-Brezetz *et al.*, 1995] S. Betge-Brezetz, R. Chatila, and M.Devy. Object-based modelling and localization in natural environments. In *IEEE International Conference on Robotics and Automation, Nagoya, Japan*, 1995.

[Chatila *et al.*, 1995] R. Chatila, S. Lacroix, T. Siméon, and M. Herrb. Planetary exploration by a mobile robot : Mission teleprogramming and autonomous navigation. *Autonomous Robots Journal*, 2(4):333–344, 1995.

[Chatila, 1995] R. Chatila. Deliberation and reactivity in autonmous mobile robots. *Robotics and Autonomous Systems*, 16:197–211, 1995.

[Delingette *et al.*, 1992] H. Delingette, M. Hebert, and K. Ikeuchi. Shape representation and image segmentation using deformable surfaces. *Image and Vision Computing*, 10(3), april 1992.

[Fleury *et al.*, 1994] S. Fleury, M. Herrb, and R. Chatila. Design of a modular architecture for autonomous robot. In *IEEE International Conference on Robotics and Automation, San Diego (USA)*, 1994.

[Haddad *et al.*, 1998] H. Haddad, M. Khatib, S. Lacroix, and R. Chatila. Reactive navigation in outdoor environments using potential fields. In *Submitted to International Conference on Robotics and Automation (Belgium)*, 1998.

[Hoffman and Krotkov, 1991] R. Hoffman and E. Krotkov. Perception of rugged terrains for a walking robot : True confessions and new directions. In *International Conference on Intelligent Robots and Systems, Japan*, 1991.

[Kweon and Kanade, 1992] I.S. Kweon and T. Kanade. High-resolution terrain map fron multiple sensor data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), 1992.

[Lacroix *et al.*, 1994] S. Lacroix, R. Chatila, S. Fleury, M. Herrb, and T. Siméon. Autonomous navigation in outdoor environment : Adaptive approach and experiments. In *Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, (USA)*, May 1994.

[Latombe *et al.*, 1991] J.C. Latombe, A. Lazanas, and S. Shekkar. Robot motion planning with uncertainties in control and sensing. *Artificial Intelligence*, 52(1), 1991.

[Mallet and Lacroix, 1998] A. Mallet and S. Lacroix. Toward real-time position estimation in unstructured environments. In *Submitted to International Conference on Robotics and Automation, Leuven (Belgium)*, May 1998.

[Matthies *et al.*, 1995] L. Matthies, A. Kelly, and T. Litwin. Obstacle detection for unmanned ground vehicles: A progress report. In *International Symposium of Robotics Research, Munich (Germany)*, Oct. 1995.

[Medeiros *et al.*, 1996] A. Medeiros, R. Chatila, and S. Fleury. Specification and validation of a control architecture for autonomous mobile robots. In *IEEE International Conference on Intelligent Robots and Systems, Osaka, Japan*, Nov. 1996.

[Miller *et al.*, 1992] D.P. Miller, R.S. Desai, E. Gat, R. Ivlev, and J. Loch. Reactive navigation through rough terrain : Experimental results. In *10th National Conference on Artificial Intelligence, San Jose, Ca (USA)*, 1992.

[Simeon and Wright, 1993] T. Simeon and B. Dacre Wright. A practical motion planner for all-terrain mobile robots. In *IEEE International Conference on Intelligent Robots and Systems, (Japan)*, July 1993.

[Simmons and Krotkov, 1993] R. Simmons and E. Krotkov. Autonomous planetary exploration : From ambler to apex. In *International Conference on Advanced Robotics, Tokyo, Japan*, pages 429–434, 1993.

[Stentz and Hebert, 1994] A. Stentz and M. Hebert. A complete navigation system for goal acquisition in unknown environments. Technical Report CMU-RI-TR-94-07, Carnegie Mellon University, 1994.

[Weisbin *et al.*, 1992] C.R. Weisbin, M. Montenerlo, and W. Whittaker. Evolving directions in nasa's planetary rover requirements end technology. In *Missions, Technologies and Design of Planetary Mobile Vehicules. Centre National d'Etudes Spatiales, France*, Sept 1992.