# 20

# Physical Models for Solving Off-Road Vehicle Motion Planning

*Moëz Cherif*
*LIFIA & INRIA Rhône-Alpes*
*46, av. Félix Viallet, 38031 Grenoble Cedex 1, FRANCE*
*Tel: (33) 76574583    Fax: (33) 76574602*
*email:* Moez.Cherif@imag.fr

**Abstract**

This paper deals with motion planning for a mobile robot moving on a hilly three dimensional terrain and subjected to strong physical interaction constraints. The main contribution of this paper is a planning method which takes into account the dynamics of the robot, the robot/terrain interactions, the kinematic constraints of the robot, and more classical geometric constraints. The basic idea of our method is to integrate geometric and physical models of the robot and of the terrain in a two-level motion planning process consisting in combining a *discrete search strategy* and a *continuous motion generation method.* It will be shown how each planning level operates and how they interact in order to generate a *safe* and *executable* motion for the all-terrain vehicle.

**Keywords**

motion planning, physical models, mobile robots, off-road vehicles, interaction constraints

## 1 INTRODUCTION

This paper addresses the problem of motion planning for a robot moving on a hilly three dimensional terrain, and subjected to dynamic constraints due to the interactions with its environment. During the last decade, most of works in robot motion planning have been focused on solving the problem of generating collision-free trajectories on planar areas considering non-holonomic kinematic constraints (see (Latombe, 1990)). Recently, few results have been obtained when additional dynamic constraints have to be processed, and when the robot has to move in a natural environment (for instance an off-road vehicle, or a planetary rover). Nevertheless, despite the ability of the proposed methods to solve some instances of such a planning problem (Shiller and Gwo, 1991)(Siméon and Dacre Wright, 1993), the automatic generation of *safe* and *executable* motions for a mobile robot subjected to strong physical constraints is far to be fully accomplished.

This comes from the fact that both the physical vehicle/terrain interactions and the dynamic constraints to satisfy cannot be processed using purely geometric and kinematic models. Indeed, such parameters play a major role in this context —because friction, slid-

J. Rix et al. (eds.), *Virtual Prototyping*

ing and skidding phenomena may strongly modify the behavior of the vehicle—. Moreover, this behavior results from the combination of various geometric and physical criteria: the mechanical architecture of the vehicle, the characteristics of the motion control law which is applied, the vehicle/terrain interactions, and the strategic orders given to the robot. This means that appropriate ***physical models*** have to be combined with more classic *geometric* and *kinematic models*, in order to integrate such complex constraints within the motion planning scheme (i.e. vehicle/terrain interactions and dynamic characteristics have to be accurately modelled and processed at the planning time).

# 2 THE APPROACH

## 2.1 The problem

Let $\mathcal{A}$ be the robot, $\mathcal{T}$ be the terrain, and $Q_{start}$ and $Q_{goal}$ be respectively the initial and the final configurations of $\mathcal{A}$. We denote in the sequel the workspace by $\mathcal{W}$, the configuration space of $\mathcal{A}$ by $\mathcal{CS}_{\mathcal{A}}$, and its state space by $\mathcal{SS}_{\mathcal{A}}$. The problem to solve is to find a safe and executable continuous motion $\Gamma(Q_{start}, Q_{goal})$ and the corresponding sequence of controls $U$ allowing to move $\mathcal{A}$ from $Q_{start}$ to $Q_{goal}$ while respecting the constraints of the task. $\Gamma$ is said to be safe if takes into account non-collision and contact relation constraints. These last constraints express the fact that contacts between several wheels of $\mathcal{A}$ and $\mathcal{T}$ have to be maintained, and cases of tip-over of $\mathcal{A}$ have to be avoided. Besides, $\Gamma$ is considered to be executable if it verifies the constraints resulting from the non-holonomy and the dynamics of $\mathcal{A}$, and the constraints imposed by the set of forces and torques created by both the vehicle/terrain interactions and the control strategy to apply.

## 2.2 The Two-level Planning Method

The robot $\mathcal{A}$ considered in this paper is an articulated non-holonomic vehicle having a locomotion system composed of three axles having each one two motorized wheels (see Figure 1). $\mathcal{A}$ is equipped with a set of joint mechanisms on its axles allowing it to be constantly in contact with the irregular surface of $\mathcal{T}$. However, this leads $\mathcal{CS}_{\mathcal{A}}$, and consequently $\mathcal{SS}_{\mathcal{A}}$, to be of a high dimension since a full configuration $Q$ of $\mathcal{A}$ is given by $6 + n_\delta$ parameters: *six* parameters $(x, y, z, \theta, \varphi, \psi)$ specifying the position/orientation (yaw,roll and pitch) of the main body in the reference frame of the workspace $\mathcal{W}$, and $n_\delta$ parameters specifying the values of the set of joint mechanisms. Besides, dealing with dynamics and physical vehicle/terrain interactions when solving for $\Gamma$ may lead to heavy computational burden. This is due to the fact that we have to operate in the state space $\mathcal{SS}_{\mathcal{A}}$ in order to cope with both complex differential equations depending on the execution constraints of the task.

The main idea of the approach we propose consists in solving the planning problem by combining and integrating the geometric and physical models of the task with a two-level technique in order to make the motion planning problem more tractable and to deal uniformly with the above mentioned constraints (Cherif et al, 1994a).

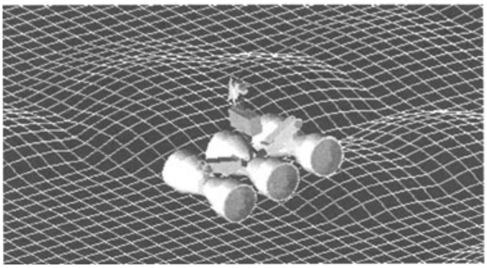

**Figure 1 The six-wheeled robot on the terrain.**

### *The Geometric Level*

As mentioned earlier, only the configurations $Q$ of $\mathcal{A}$ satisfying the safety constraints are considered. In that case, the parameters of $Q$ are not independent and the inter-relations between them depend on the vehicle/terrain relation (i.e. distribution of the contact points according to the geometry of the terrain). A practical consequence of this property is that it allows to reduce the search space into a reduced subset $\mathcal{C}_{search}$ of $\mathcal{CS}_{\mathcal{A}}$ defined on $(x, y, \theta)$ —the horizontal position and the heading angle of the main body of $\mathcal{A}$ denoted from now by $q$. Afterwards, the solution is iteratively computed between $q_{start}$ and $q_{goal}$ corresponding to $Q_{start}$ and $Q_{goal}$, by applying a discrete search technique (an $A^*$ algorithm for instance) through an incrementally generated directed graph $\mathcal{G}$ representing the explored configurations of $\mathcal{C}_{search}$. Such an approach has already been used in (Barraquand and Latombe, 1989) and (Siméon and Dacre Wright, 1993) to find non-holonomic paths for a robot moving on planar and $3D$ areas, respectively. Two nodes $N(q)$ and $N(q_{next})$ are connected in $\mathcal{G}$ if $\mathcal{A}$ a simple non-holonomic path composed of a circular arc or a straight line segment may be generated from the current configuration $q$ towards the next sub-goal $q_{next}$.

### *The Physical Level*

Since the construction of $\mathcal{G}$ (i.e. generation of the nodes $N(q)$ and the non-holonomic paths) does not account neither the geometric shape of $\mathcal{T}$ nor the dynamics of the task, we process the second level in order to cope with such features. This consists in computing locally a continuous motion of $\mathcal{A}$ satisfying the execution constraints between $q$ and $q_{next}$. $q_{next}$ corresponds to the configuration of the best successor node of $N(q)$ when searching $\mathcal{G}$. This is solved by formulating the planning problem in $\mathcal{SS}_{\mathcal{A}}$ and using a physical model of the task (Cherif, 1994b).

Unlike methods operating in two stages: (1) processing a geometric path, and afterwards (2) generation of a full trajectory of the robot when considering the task execution constraints, our approach allows to introduce dynamics and vehicle/terrain interactions at the planning time. Indeed the exploration of $\mathcal{C}_{search}$ is mainly used to guide the search process, and to give us only potential intermediate configurations of $\mathcal{A}$ approximating the final solution (see Figure 2). The real trajectory of the robot is provided by the sequence of the local motions computed when the physical model of the task are processed. The main advantage of such an approach is to cope locally with the dynamic constraints of the task

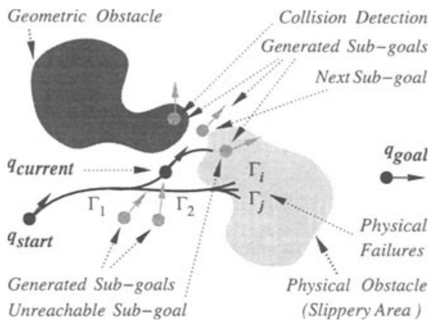


**Figure 2 The general scheme of the planning approach.**

as it has been already proposed in (Cherif et al, 1994a). At a given step of the algorithm, the potential successors of the current configuration $q$ are explored in order to verify the safety constraints of the task (e.g. no-collision with the geometric obstacles). Afterwards, a motion $\Gamma(q, q_{next})$ is computed to move $\mathcal{A}$ from $q$ towards its best safe neighbor $q_{next}$. The solution $\Gamma$ is incrementally built until reaching the goal configuration $q_{goal}$. For instance, the motion solution $\Gamma$ to be processed in Figure 2 is given by the previously computed motion $\Gamma_1$ between $q_{start}$ and $q$, and the concatenation of the sub-motions $\Gamma(q, q_{next})$ and $\Gamma(q_{next}, q_{goal})$ which have to be processed. If the $\Gamma(q, q_{next})$ cannot be computed because of sliding or skidding of $\mathcal{A}$ on slippery areas (such as when computing $\Gamma_i$ and $\Gamma_j$), the algorithm backtracks in order to select an other potential intermediate configuration to reach and to guide the search until $q_{goal}$. We present in §3 and §4 the physical modeling of the task and the physical level of the planning algorithm, respectively.

# 3 TASK REPRESENTATION USING PHYSICAL MODELS

## 3.1 Physical Modeling of the Vehicle $\mathcal{A}$

The dynamics of $\mathcal{A}$ is formulated using a mixed model combining the physics of particles and the mechanics of solids (see (Cherif et al, 1994a)). It is described by a network of interconnected rigid bodies $\Omega_i$ (corresponding to the components of $\mathcal{A}$ such as the wheels, the chassis or the axles) linked on specific points by visco-elastic relations (Cherif et al, 1994a). This enables to couple directly each element of the robot to those others that are in contact with it. It is different from the dynamic model based on a joint space formulation where every link is related to the one immediately before it, and every motion is sensed in relative and require successive transformations processing which can be time consuming.

Let $r_i(t)$ and $\alpha_i(t)$ be respectively the position and the orientation parameters at time $t$ of a given $\Omega_i$ of $\mathcal{A}$. The motions of $\Omega_i$ are specified by the Euler/Newton equations: $F_i = m_i \ddot{r}_i(t)$ and $T_i = \dot{L}_i(t) = I_{\Omega_i} \ddot{\alpha}_i(t)$, where $F_i$ and $T_i$ are respectively the sums of forces and torques applied on $\Omega_i$, $L_i(t)$ is the angular momentum about the center of mass $G_{\Omega_i}$, and $I_{\Omega_i}$ is the inertia tensor of $\Omega_i$ about the frame axes. $\dot{L}_i(t)$ is also related to

Coriolis and centrifugal terms (Goldstein, 1983), but we will make the assumption that such terms are negligible. Then, $F_i$ and $T_i$ can be computed using the *Euler*'s principle of superposition: $F_i = F_d + \sum_{force\,j} F_{i,j}$ and $T_i = U_i + \sum_{force\,j}(G_{\Omega_i}P_{i,j} \times F_{i,j})$. $F_{i,j}$ are the forces acting on $\Omega_i$, $P_{i,j}$ are the points where the forces $F_{i,j}$ are applied, $G_{\Omega_i}P_{i,j}$ is the vector from $G_{\Omega_i}$ to $P_{i,j}$, and $\times$ is the outer product. The term $F_d$ includes the gravity forces and additional viscous forces of the environment. The set of forces $F_{i,j}$ results from the the physical interactions of $\Omega_i$ with the other components of $\mathcal{A}$ —through the joints of $\mathcal{A}$— and with the involved components of the terrain —through the wheel/ground contact interactions. When $\Omega_i$ is a wheel, $U_i$ corresponds to the torques generated by the control mechanisms (i.e. the "physical effector") applied on $\Omega_i$, otherwise this term vanishes.

Each articulated mechanism associated with the joint $\delta_k$ of $\mathcal{A}$ is represented by a network $\Phi(\delta_k)$ combining a set of connectors $c_r$ and a set of specific points selected on the rigid bodies $\Omega_i$ corresponding to the joint $\delta_k$. The connectors are defined in terms of visco-elastic laws (i.e. combination of springs and dampers). For instance, we have represented $3D$ rotoïd joint mechanisms by two rigid bodies connected through two pairs of points respectively selected on them and belonging to the rotation axis (see Figure 3).

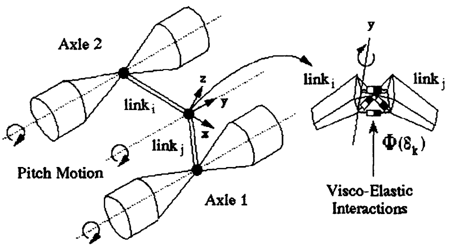


**Figure 3 The physical model of the vehicle.**

## 3.2 The Virtual Physical Model of the Terrain $\mathcal{T}$

Coping with robot/terrain interactions requires to build a virtual model $\Phi(\mathcal{T})$ of the terrain which is able to capture both the geometric and the physical properties of $\mathcal{T}$ and which allows the formulation of such interactions. For that purpose, we have implemented a model based upon the concept of *"deformable physical models"*, initially proposed for COMPUTER GRAPHICS (see (Luciani et al, 1991)(Terzopoulos et al, 1987)).

According to this concept, the terrain is represented by a set of interconnected particles $\Phi(P_i)$ having the following properties (Jimenez, Luciani and Laugier, 1991)(Luciani et al, 1991): (1) each particle is seen as a point mass $m_i$ which obeys Newtonian dynamics —given by the equation $F_{P_i} = m_i \ddot{r}_{P_i}$ where $r_{P_i}$ is the position of $\Phi(P_i)$ in $\mathcal{W}$— and which is surrounded by a spherical non-penetration "elastic" area; (2) the set of particles corresponds to the inertial and spatial occupancy characteristics of the modeled area of $\mathcal{T}$; (3) the particles are interconnected using interaction components referred to as the "connectors". Each connector corresponds to a type of interaction, and is modeled using appropriate physical laws allowing several types of behaviors (e.g. visco-elastic or elastic cohesion, dry friction interactions).

The discretization of the terrain in terms of such elementary physical components accounts several criteria such as the terrain surface shape, the average distribution of the contact points between the wheels and the ground, and the complexity of $\Phi(\mathcal{T})$ (i.e. the number of the processed particles). In the current implementation of the system, the particles distribution is determined by computing a set of spheres $S_i$ whose profile approximates the surface of $\mathcal{T}$ given by the set of the initial geometric patches. This is done in such a way that each point of the terrain surface should be located on the surface of at least one $S_i$. Afterwards, a dynamic behavior is "given" to the computed set of spheres by placing a particle $\Phi(P_i)$ at the center of each $S_i$ (Figure 4). The main advantage of such a representation is related to the fact that it allows us to maintain the geometric features of the motion planning problem (i.e. checking the geometric constraints as the non penetration in the ground), and to uniformly process the physical behavior of $\mathcal{T}$ and its interactions with $\mathcal{A}$.

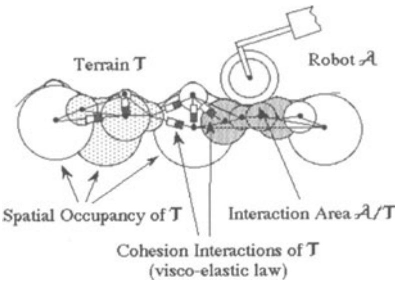


**Figure 4 The physical model of the terrain.**

## 4 THE PHYSICAL MOTION PLANNING LEVEL

The purpose of the physical planning level function) is to check for the existence of an executable motion allowing $\mathcal{A}$ to move between two successive configurations $q_{i,p}$ and $q_i$, i.e. a motion which satisfies the kinematic/dynamic constraints of the task. Task constraints consist of the kinematic/dynamic constraints of $\mathcal{A}$, the constraints imposed by the vehicle/terrain interactions, and the constraints coming from the applied control strategy.

### 4.1 Coping with Vehicle/Terrain Interactions

The interactions between $\mathcal{A}$ and $\mathcal{T}$ are based on the formulation of a set of dynamic laws depending on both the distribution of the contact points and the type of the surface-surface interactions. Describing $\mathcal{T}$ in terms of spheres requires to handle a smaller amount of information to represent its geometric shape (as shown in Figure 4). Furthermore, the combination of such simple primitives with an appropriate hierarchical description of the wheels allows us to compute easily the distribution of the contact points using a fast distance computation algorithm involving the structured sets of the considered spheres (Hopcroft, Schwartz and Sharir, 1983). Once a contact between the wheels of $\mathcal{A}$ and a sphere $S_i$ corresponding to $\mathcal{T}$ is detected, the corresponding physical interaction is easily computed by activating the associated dynamic laws.

The surface-surface interactions are processed using two types of constructions: a visco-elastic law associated with the set of $S_i$ involved in the contact, and surface-surface interactions. In order to solve the second point, we make use of a *finite state automaton* since complex phenomena like dry friction basically involves three different states: no contact, gripping, and sliding under friction constraints. The commutation from one particular state to another is determined by conditions involving gripping forces, sliding speed, and relative distances. Each state is characterized by a specific interaction law. For instance, a visco-elastic law between the interacting points of the wheels of $\mathcal{A}$ and $\mathcal{T}$ is associated with the gripping state, and a Coulomb equation is associated with the sliding state (Jimenez, Luciani and Laugier, 1991).

## 4.2 The Physical Planning Algorithm

The main advantage of the virtual model of $\mathcal{T}$ and the physical model of $\mathcal{A}$ is to exhibit consistent numerical (and graphical) behavior of the robot which can be useful in predicting the resulting motion and planning safe and executable trajectories. Thanks to this concept, the motion of $\mathcal{A}$ can be computed by uniformly integrating the differential equations of motion corresponding to the components of the physical models of the robot and the terrain, when controlling the wheels. In order to achieve such a process, we use a motion generation scheme proposed and described in (Cherif, 1994b).

The main difficulty is to find an appropriate way to generate a motion $\Gamma(q_{i,p}, q_i)$ which allows $\mathcal{A}$ to move from $q_{i,p}$ to the next subgoal $q_i$. Let $\delta t$ be the time increment of the motion generation process, $s_{i,p}$ be the state of $\mathcal{A}$ corresponding to the configuration $q_{i,p}$, and let $s(n\delta t)$ be the state of $\mathcal{A}$ obtained after having applied $n$ elementary motion steps when starting from $s_{i,p}$ (i.e. after having applied a sequence of $n$ controls on the "physical effectors" of $\mathcal{A}$). Determining the required sequence of controls $U$ to apply to $\mathcal{A}$ can be done by executing an iterative algorithm involving two complementary steps. The first step consists in hypothesizing a nominal sub-path $\mathcal{P}_i^k$ between the current configuration $q(n\delta t)$ and the next sub-goal represented by $q_i$, and the second step allow to track $\mathcal{P}_i^k$ while processing the physical vehicle/terrain interactions, as illustrated in Figure 5.

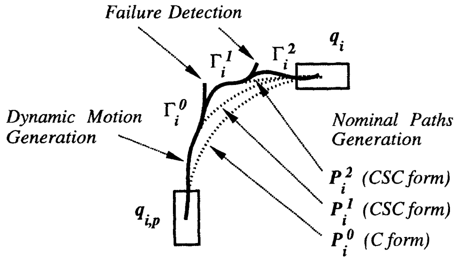


**Figure 5 The local motion generation scheme.**

The nominal path $\mathcal{P}_i^k$ is constructed using a technique derived from the Dubins' approach (Dubins, 1957). The obtained sub-path is a smooth curve made of straight line segments $S$ and circular arcs $C$ (of the form $CSC$). The tracking function operates on the dynamic model of $\mathcal{A}$ and the *virtual* representation of $\mathcal{T}$. It takes as input the velocity controls applied on each controlled wheel during a time increment $\delta t$. These controls are computed from the linear and steering speeds which are associated with the reference point of $\mathcal{A}$ when moving on $\mathcal{P}_i^k$. They are converted into a set of torques $U(t)$ to be applied to the wheels of $\mathcal{A}$. Since the motion generation step accounts physical phenomena like sliding or skidding, the configuration $q^\star(n\delta t)$ corresponding to the state $s^\star(n\delta t)$ of $\mathcal{A}$ obtained after having applied $n$ successive controls may be different from the nominal configuration $q(n\delta t)$. The processed motion generation step will be considered as a failure when $q^\star(n\delta t)$ is too far from its nominal value $q(n\delta t)$. The previous algorithm is iterated until the neighborhood of $s_{i,p}$ is reached or until a failure is detected (see Figure 5). Figure 6 shows a local trajectory obtained when $\mathcal{A}$ is controlled to cross an irregular area of the terrain.

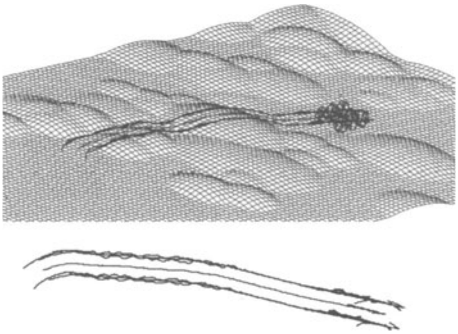

**Figure 6 Local motion of the vehicle $\mathcal{A}$.**

## 5 SIMULATION RESULTS AND CONCLUSION

We have shown in this paper how physical models can help in solving some classes of complex motion planning problems of an articulated all-terrain vehicle. A method to plan safe and executable motion for a rover moving in a natural environment and strongly subjected to physical interactions with the terrain has been also presented. This is based on the use of a "virtual" model of the terrain, and combines a continuous motion technique and a discrete search strategy in order to deal with several non-trivial features such as collision avoidance, kinematics and dynamics of the vehicle and its physical interactions with the terrain. The approach presented in the paper has been implemented on a Sun Sparc workstation. Several experiments for a non-holonomic six-wheeled rover $\mathcal{A}$ have been successfully performed in simulation. For instance, Figures 7 and 9 show the trajectories generated by the planner when $\mathcal{A}$ moves on an irregular area of the terrain. In figure 8 and 10, we show the resulting motions when the robot has to move on slippery areas.

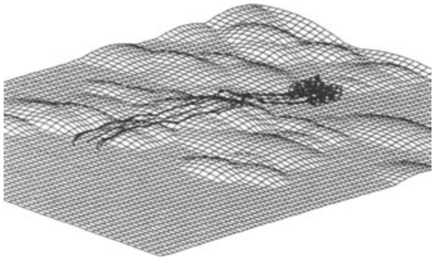

**Figure 7** Moving on irregular areas.

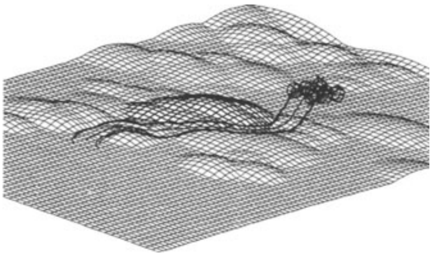

**Figure 8** Avoiding slippery areas.

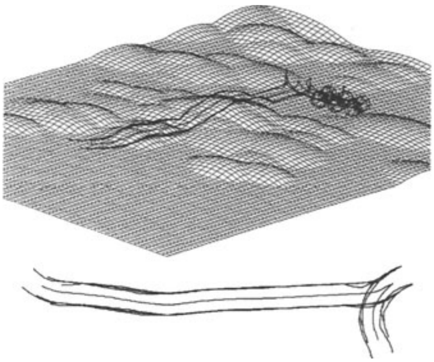

**Figure 9** Maneuvering on irregular areas.

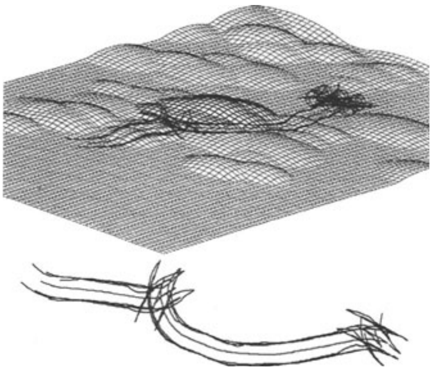

**Figure 10** Maneuvering on slippery areas.

*Acknowledgements*

This work has been partly supported by the CNES (Centre National des Etudes Spatiales) through the RISP national project and the MRE (Ministère de la Recherche et de l'Espace) . It has been also partly supported by the Rhône-Alpes Region through the IMAG/INRIA Robotics project SHARP.

## REFERENCES

Barraquand, J. and Latombe J.-C. (1989). On non-holonomic mobile robots and optimal maneuvering. *Revue d'intelligence Artificielle* **3**(2), 77–103.

Cherif, M., Laugier Ch., Milési-Bellier Ch. and Faverjon B. (1994a). Planning the motions of an all-terrain vehicle by using geometric and physical models. In: *Proc. of the IEEE*

*Int. Conf. on Robotics and Automation.* San Diego, CA (USA). pp. 2050–2056.
Cherif, M. (1994b). Dynamic motion planning of autonomous off-road vehicles. Research Report RR-2370. INRIA Rhône-Alpes. Grenoble (F).
Dubins, L.E. (1957). On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics* **79**, 497–516.
Goldstein, H. (1983). *Classical Mechanics.* Addison-Wesley.
Hopcroft, J.E., Schwartz J.T. and Sharir M. (1983). Efficient detection of intersections among spheres. *Int. Journal of Robotics Research.*
Jimenez, S., Luciani A. and Laugier Ch. (1991). Physical modeling as an help for planning the motions of a land vehicle. In: *Proc. of the IEEE/RSJ Int. Workshop on Intelligent Robots and Systems.* Osaka (J).
Latombe, J.-C. (1991). *Robot motion planning.* Kluwer Academic Press.
Luciani A. *et al* (1991). An unified view of multiple behaviour, flexibility, plasticity and fractures: balls, bubbles and agglomerates. In: *IFIP WG 5.10 on Modeling in Computer Graphics.*
Shiller, Z. and Gwo Y.R. (1991). Dynamic motion planning of autonomous vehicles. *IEEE Trans. Robotics and Automation.*
Siméon, T. and Dacre Wright B. (1993). A practical motion planner for all-terrain mobile robots. In: *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems.* Yokohama (J).
Terzopoulos D. , Platt J., Barr A., Fleischer K., Elastically deformable models, *Computer Graphics*, 21(4), July 1987.

## BIOGRAPHY

The author received the engineering degree from the University of Tunis, Tunisia, in 1990, and the M.S. degree in computer science from the Polytechnic Institute of Grenoble (INPG), France, in 1991. He is currently preparing a Ph.D. in computer science within the robotics group of LIFIA-IMAG Laboratory and INRIA Rhône-Alpes. His research involves mobile robots, motion planning, dynamic modeling and simulation, and virtual reality.