Motion Planning for All-Terrain Vehicles: A Physical Modeling Approach for Coping with Dynamic and Contact Interaction Constraints

Moëz Cherif, *Member, IEEE*

***Abstract*— This paper addresses modeling and global motion planning issues for an autonomous wheeled mobile robot moving on an uneven three-dimensional (3-D) terrain. We focus particularly on the issue of dealing with dynamic and wheel/ground interaction constraints. A key feature of our approach is that it incorporates appropriate physical models to cope with the task dynamics in the motion planning paradigm. The planner is based on a two-level scheme. The high level considers a simplified two-dimensional (2-D) instance of the motion task and searches a subset of the configuration space of the robot in order to generate nominal subgoals through which the robot is steered. The local level solves for continuous feasible trajectories and actuator controls to move the robot between neighboring subgoals in the presence of the entire task constraints. To the best of our knowledge, this is the first implemented planner that solves for feasible trajectories to be performed by a wheeled vehicle on quite complex terrains. Simulation results are presented for the case of a six-wheeled articulated robot.**

***Index Terms*— Motion planning, off-road vehicles, nonholonomic robots, kinematics, dynamics, physical models.**

I. Introduction

A. Overview of the Problem

MOTION planning for nonholonomic wheeled robots has attracted considerable attention during the last decade. Much work has focussed on the case of a car-like robot moving on a flat surface among static obstacles [5], [7], [20], [34], [40] (see [33] for a good survey). More recently, mobile robots are being used in challenging new applications such as planetary exploration and navigation in off-road sites. Main projects that were or are still concerned with these applications include the French Programme RISP-VAP [8], [10], NASA's Pathfinder project [54] and CMU's Ambler project [4]. Other projects concern intervention robotics such as the BUGS project for mine countermeasures [15]. In the context of off-road mobile robotics, the robot is often a wheeled (redundant) articulated system and the surfaces on which it has to move are irregular and complex. These features exhibit new constraints and modeling/computational issues that make the motion planning problem much more difficult than in the 2-D case.

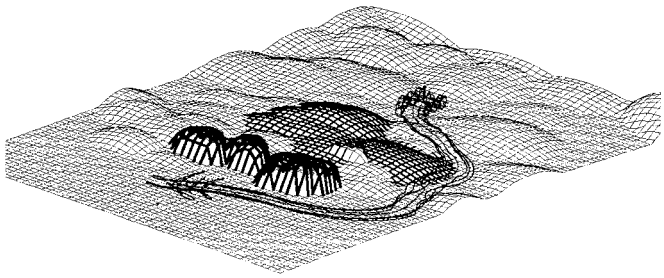

Fig. 1. Steering motions when avoiding static obstacles and sliding on the terrain.

In this paper, we address aspects of modeling and global motion planning for a wheeled vehicle moving on an uneven terrain. The basic problem we address is *starting from a given initial configuration of the robot in contact with the terrain, find a feasible trajectory (configurations/velocities and the control torques to be applied on the robot wheels) that moves the robot toward a given desired configuration*. For illustration, consider the case of a six wheeled vehicle as shown in Fig. 1. The environment is composed of a set of static obstacles (shown by the dark meshes in front of the figure), a sticky area (the finer mesh) and a slippery region (dark mesh at the center). The robot is initially located at the front of the static obstacles and the final desired configuration is located at the back of the sliding region. The planned motion (shown in Fig. 1 by the traces of the center of the robot and of its wheels) avoids the static obstacles and tip-over of the robot and minimizes slippage at the wheels. We also deal with additional constraints arising from the robot dynamics and its physical interactions with the terrain. This issue is fundamental for characterizing feasible motions in the case of an all-terrain vehicle. In this paper, we discuss this issue and show how it can be addressed in the framework of global motion planning by introducing specific and appropriate physical models.

For a better focus on motion planning under kinematic, dynamic and contact interaction constraints, we make the assumption that a complete description of the environment is available when planning. This description includes the geometry of the environment as well as the physical features of

Manuscript received November 12, 1997; revised October 27, 1998. This work was supported in part by the French Programme RISP-VAP (of CNES and CNRS) and the Rhône-Alpes Region through the IMAG & INRIA Robotics Project Sharp, 1991–1993. This paper was recommended for publication by Associate Editor J. Wen and Editor V. Lumelsky upon evaluation of the reviewers' comments.

The author was with the Robotics Group Sharp, INRIA, Rhône-Alpes, France. He is now with the School of Engineering Science, Simon Fraser University, Burnaby, B.C. V5A 1S6, Canada.

Publisher Item Identifier S 1042-296X(99)01228-8.

the terrain such as cohesion/deformation, friction, and inertia[1]. The problems related to sensing and modeling the environment are beyond the scope of this paper. However, issues such as coping with incomplete knowledge of the environment and uncertainty (due to inaccurate sensing, modeling and control) merit further investigation in relationship with our planning framework to make it useful and applicable in real contexts.

While building geometric elevation maps of the environment from sensory data is an active research area in outdoor mobile robotics [23], [31], the use of appropriate models for dealing with soil mechanics is limited. Advanced analysis of the texture of the different surfaces may permit qualitative estimation of some of the environment physical properties (e.g., rigidity/softness and adherence). This qualitative data can possibly be used to (locally) refine the environment map and to generate a description of the workspace in terms of primitives or components having different physical properties. The quantitative identification of mass, deformation, and friction parameters of these various components is, however, very difficult. We believe that this modeling problem can be overcame by incorporating uncertainty in the geometric and dynamic formulation of all-terrain motion planning and by developing appropriate algorithms for dealing with incremental contact-based sensing and model refinement. These issues are likely among the key problems to be investigated in future research in motion planning for autonomous off-road mobile robots.

B. Overview of Our Approach

A first and intuitive planning scheme consists of decoupling path planning and trajectory planning [18], [28]. This can be achieved by:

1) planning a nonholonomic path when accounting for the three-dimensional (3-D) relief of the terrain;
2) transforming the resulting path into a trajectory when coping with the dynamic constraints.

Because of the discrepancy of the models and the analyses considered at each stage, such a scheme can be time-consuming since, depending on the task, a large number of paths must be planned before finding a solution satisfying the task dynamics.

Our approach copes simultaneously with kinematic and dynamic constraints. Basically, it consists in interleaving two reasoning levels. The high level—a *discrete grid search*—operates globally and expands a tree of subgoals over the space of position/orientation in the plane of the robot. The subgoals are computed assuming that the robot can move, for a coarse period of time, along a canonical nonholonomic 2-D path in absence of the dynamic constraints. At each iteration, the best subgoal (in terms of a cost function such as the distance to the final goal) is chosen and is checked if it is locally reachable from a subgoal adjacent to it. This is performed by the local level which operates *continuously* in the state space of the robot and solves for effective feasible smooth motions for the robot between adjacent subgoals. A key feature of our approach is the introduction, in the motion planning paradigm, of specific and appropriate physical models which help in coping with the task dynamics and solving for feasible instantaneous motions. The two levels are iteratively interleaved until the final goal is reached or no solution can be found. The planner has been implemented and applied in simulation to several motion tasks showing the power and the promise of our approach. To the best of our knowledge, this is the first implemented global motion planner for all-terrain wheeled robots that copes with non simplified dynamic and physical interaction constraints.

Section I-C presents relevant previous works in all-terrain robot motion planning. In Section II, we outline the problem to be addressed. Section III presents the instantaneous forward motion solution used by the local planner. Section IV describes the two-level planning algorithm. Section V presents simulation results for a six-wheeled vehicle moving on various terrains. Section VI presents concluding remarks and future work.

C. Related Works in All-Terrain Vehicle Motion Planning

Little work has addressed motion planning for all-terrain wheeled robots. The existing literature can be classified into two classes. In the first class, the problem is to find *itineraries* (gross motions) for the robot when it is reduced to a point and it is constrained to move along pre-defined directions determined from an approximation of the terrain [21], [36], [39]. In the second class, a more accurate model of the robot and/or of the terrain is considered, and kinematic and/or simplified dynamic constraints are incorporated in motion planning [2], [14], [47], [49]–[51]. Shiller *et al.* presented a global time-optimal trajectory planner that accounts for kinematic and dynamic constraints [47], [49]. The relief of the terrain is assumed to be smooth and of the size of the vehicle, enabling the robot to be reduced to a point. The planner uses an iterative three-stage approach based on several optimization schemes to find the optimal solution. Because of the necessary condition of the smoothness of the terrain, this planner is not applicable when contact distribution at the wheels is not uniform or when the size of the relief is smaller than the robot. Siméon described an algorithm to plan nonholonomic paths for a polygonal vehicle having three wheels reduced to points on a 3-D terrain [50]. As our high planning level, the planner operates as a grid search on the 2-D position/orientation space (x, y, θ) of the robot and uses a sampling of the extreme velocity controls of the wheels to steer the robot. In [14] and [51], the planner has been extended for dealing with a car-like robot having an arbitrary number of axles. In these works, no robot dynamics and no friction effects are dealt with. Despite the efficiency of this approach, the reduction of the wheels and the contacts to pure rolling points, however, may be restrictive to generate realistic solutions. Other relevant works have considered modeling and simulation aspects without addressing motion planning [1], [6], [27].

Very recently, uncertainty has been addressed by some researchers in the framework of off-road navigation and path planning [11], [22], [30]. Hait and Siméon have extended the planner described in [51] in order to cope with uncertainty

[1] In this article, the terrain refers to the areas of the environment on which the robot can possibly move. Three types of terrain components are considered: static rigid areas, deformable areas whose surface can move locally under the effect of a contact (e.g., sandy or muddy areas) and movable components (e.g., stones or debris when exploring damaged sites).

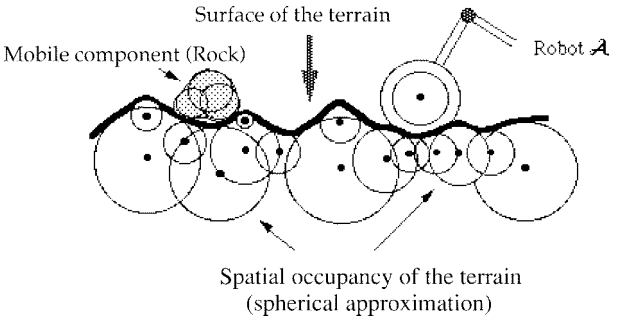

Fig. 2. The spherical approximation of $\mathcal{T}$.

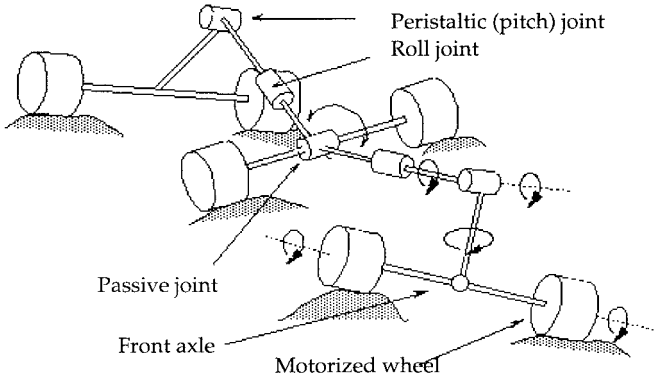

Fig. 3. Schematic description of $\mathcal{A}$ when $n_a = 3$.

in the terrain model by incorporating error intervals in the elevation map [22]. Kubota *et al.* [30] and Chen and Kumar [11] have addressed path planning aspects for a rover and a multi-legged walking robot, respectively, by incorporating "traversability" probabilities in the terrain elevation map. All these works have focussed on the static behavior of the robot and have not investigated uncertainty effects (related to geometry and contact interactions) on the kinematics and dynamics of the robot. Uncertainty in control in the presence of nonholonomic constraints has been also investigated by Fraichard and Mermond for path planning in the simple case of a mobile robot moving on a planar surface [19]. All these techniques are relevant for future extensions of our planning framework to cope with uncertainty.

II. Statement of the Problem

A. Workspace

The workspace, $\mathcal{W}$, is composed of a 3-D terrain, $\mathcal{T}$, and a set, $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \cdots, \mathcal{B}_{n_\mathcal{B}}\}$, of static obstacles located on $\mathcal{T}$. The terrain surface, $\partial\mathcal{T}$, is divided into several areas, $\mathcal{T}_i$, having different properties (i.e., friction, deformation, etc.). We describe the geometry of $\mathcal{W}$ by a collection, $\mathcal{ST} = \{S_1, \cdots, S_{n_{\mathcal{ST}}}\}$, of spheres approximating the relief of $\mathcal{T}$ and the $\mathcal{B}_i$'s (see Fig. 2). Such a collection, $\mathcal{ST}$, can be obtained by computing a set of tangent spheres that approximate accurately an elevation map of $\mathcal{W}$ [41]. In addition, a 2-D hierarchical model approximating the shape of each obstacle, $\mathcal{B}_i$, is considered. For a given $\mathcal{B}_i$, this model is composed of a set of discs, denoted by $\mathcal{DB}_i$, covering the projection of $\mathcal{B}_i$ in the plane (x, y) of a frame, $\mathcal{F}_W$, fixed in $\mathcal{W}$.

B. Vehicle

Many existing wheeled robots devoted to planetary exploration and/or off-road intervention tasks are composed of articulated structures (e.g., the six-wheeled robot Marsokhod [29], JPL's mobile robots: the Sojourner (or Rocky) robot [53], Robby vehicle and Surveyor Lunar Rover (SLRV) [52], and the BUGS vehicles for mine countermeasures in uneven sites [15]). For developing our approach, we consider a robot structure derived from the model of the Marsokhod rover. However, other nonarticulated structures may be taken into account (e.g., a car-like robot).

The mechanical structure of the robot, $\mathcal{A}$, is composed of a main body, $n_a \geq 2$ wheeled axles connected through an articulated chassis composed of a set of passive and active rotoid joint mechanisms that allow the different axles to achieve roll and pitch motions (see Fig. 3). $\mathcal{A}$ is steered by applying a series of torque vectors, $U \in \mathbb{R}^{2n_a}$, on its $2n_a$ wheels. Such a structure allows $\mathcal{A}$ to adapt its configuration to the geometry of the terrain and to have a better distribution of the contacts at its wheels. The roll/pitch motions of the axles are performed through the passive joints of the chassis which are considered to be compliant (spring-damper mechanisms). The active joints used for controlling the length of $\mathcal{A}$ are set to an arbitrary constant value. Thus, the length of $\mathcal{A}$ can vary only under the effect of the passive pitch mechanisms. Finally, only the front axle of $\mathcal{A}$ may have yaw motions within a limited range.

A full configuration, Q of $\mathcal{A}$, is defined by the 3-D position/orientation, $(x, y, z, \theta, \varphi, \psi)$, of a frame, $\mathcal{F}_\mathcal{A}$, fixed on the main body of $\mathcal{A}$ with respect to (w.r.t.) the reference frame $\mathcal{F}_\mathcal{W}$, and the vector of compliant passive joints. (The active joints are fixed to constant values). θ, φ, and ψ are the yaw, roll, and pitch angles of $\mathcal{F}_\mathcal{A}$, respectively. Let n_p equal the number of passive joints of the chassis of $\mathcal{A}$. We denote the C-space and the state space of $\mathcal{A}$ by $\mathcal{CS}_\mathcal{A}$ and $\mathcal{SS}_\mathcal{A}$, respectively. In the following, the vector $q = (x, y, \theta)$ of Q is referred to as the subconfiguration of $\mathcal{A}$, and we denote the space of subconfigurations q by $\mathcal{SCS}_\mathcal{A}$.

C. Task Constraints

1) No-Collision Between $\mathcal{A}$ and the Obstacles $\mathcal{B}_i$: At each instant, $\forall j, \mathrm{Proj}_{(x,y)}(\mathcal{A}) \cap \mathcal{DB}_j = \emptyset$, where $\mathcal{DB}_j$ is the set of discs used in 2-D to model $\mathcal{B}_j$ (c.f. Section II-A), and $\mathrm{Proj}_{(x,y)}$ is the projection map into the plane (x, y) of $\mathcal{F}_\mathcal{W}$.

2) No-Collision Between $\mathcal{T}$ and $\mathcal{A}$: Collision between $\mathcal{T}$ and parts of $\mathcal{A}$ other than the wheels, W_i, must be avoided. Hence, $(\mathcal{A} \setminus \{W_i, i = 1 \cdots 2n_a\}) \cap \mathcal{ST} = \emptyset$, where $\mathcal{ST}$ is the set of spheres modeling $\mathcal{T}$.

3) Contact Distribution: Because of the irregularities of $\mathcal{T}$, the contact can be broken at one or several wheels for a short period of time without drastically affecting the steerability of $\mathcal{A}$. It is difficult to analyze the steerability of an all-terrain vehicle in the general case (i.e., for all possible robot configurations and terrain features) and to clearly formulate

the conditions under which it is steerable. From a perspective of motion planning, we are more interested in detecting (by the local planner) if $\mathcal{A}$ is able or not to achieve certain configurations (subgoals) provided during the search. For this purpose and as a first simplification, we consider as possibly admissible configurations only those where at least a wheel on the left side and a wheel on the right side are in contact with $\mathcal{T}$. Depending on the geometry and friction of the crossed regions, $\mathcal{A}$ may remain steerable (for a short period of time) when it has only one contact per side since each wheel is independently actuated. The analysis of all-terrain vehicle steerability is a key issue and merits further investigation in the future.

4) No Tip-Over: No tip-over reduces to keep the roll and the pitch angles (φ, ψ) within the interval $]-\varphi_{\max}, \varphi_{\max}[$, where $\varphi_{\max} \leq \pi/2$ is an arbitrary upper-bound. We have deliberately simplified the no tip-over condition by considering constant bounds on φ and ψ. The previous constraint in Section II-C3 contributes somewhat to avoiding tip-over since breaking all the contacts at one side of $\mathcal{A}$ corresponds to the beginning of a tip-over.

5) Friction Constraints: We consider both sticky or frictionless contacts between $\mathcal{A}$ and components of $\mathcal{T}$. For frictional contacts, we use a Coulomb model incorporating kinetic effects. This is detailed in Section III-C.

6) Velocity Constraints: When $\mathcal{A}$ moves on a horizontal flat surface, the passive joints of the chassis and the roll/pitch parameters (φ, ψ) are all null. Since only the front axle can perform a yaw motion, w.r.t. $\mathcal{F}_{\mathcal{A}}$, sliding occurs at the wheels of the remaining axles each time $\mathcal{A}$ executes a turn. Under the condition that summation of the sliding velocities at the wheels is instantaneously minimized, one can derive a relationship between the velocity parameters of $\mathcal{A}$ that can be shown to be nonholonomic[2]. This relationship has the form $F(q, \dot{q}, t) = \dot{x}\sin\theta - \dot{y}\cos\theta + \alpha\dot{\theta} = 0$, where α depends on the length, L, of $\mathcal{A}$ and n_a. For the case of $n_a = 3$,[3] one can get the following nonholonomic constraint: $\dot{x}\sin\theta - \dot{y}\cos\theta + L_r\dot{\theta}/2 = 0$, where L_r is the distance between the middle and the rear axles. Fig. 4 illustrates this constraint and shows the location of the center of instantaneous rotation, $G_{\mathcal{A}}$. For a given $\dot{\theta}$, the y components w.r.t. $\mathcal{F}_{\mathcal{A}}$ of the velocities of the central and rear axles are equal and of opposite signs. In addition, we consider, at each instant, that $|v| = |\dot{\boldsymbol{r}}| \leq V_{\max}$ and $|\phi| \leq \phi_{\max}$, where v is the translation velocity of the reference point, $C_{\mathcal{A}}$ of $\mathcal{A}$ (the origin of $\mathcal{F}_{\mathcal{A}}$), and ϕ is the steering angle (yaw angle w.r.t. $\mathcal{F}_{\mathcal{A}}$) of the front axle. This leads to the following constraint on the yaw velocity: $|\dot{\theta}| \leq \dot{\theta}_{\max} = f(V_{\max}, \phi_{\max})$, where f is a function of the maximum translation velocity, $V_{\max}$, and steering angle, $\phi_{\max}$. For $n_a = 3$, we have $\dot{\theta}_{\max} = V_{\max}(\sin\phi'_{\max}/L_r/2)$, where $\phi'_{\max}$ is the maximum angle between the velocity vector of $\mathcal{A}$ and its longitudinal axis. $\phi'_{\max}$ is related to the maximum steering angle, $\phi_{\max}$, by $\tan\phi'_{\max} = L_r\tan\phi_{\max}/(2L - L_r)$.

[2]In the 3-D case, the formulation of such kinematic constraints is not straightforward. This depends on the 3-D structure of the contacted region of $\mathcal{T}$ and the distribution of the contacts at the wheels.

[3]For $n_a = 3$ and when $\mathcal{A}$ moves on a horizontal flat surface, the minimum lateral sliding velocities at the middle and rear axles satisfy that their summation is null. In general, this condition is not guaranteed when $\mathcal{A}$ moves on uneven surfaces.

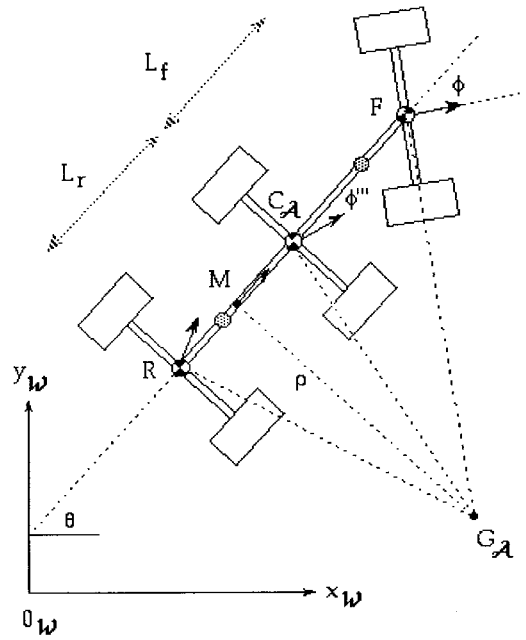

Fig. 4. The 2-D kinematic model of $\mathcal{A}$ $(n_a = 3)$.

7) Control Torque and Acceleration Constraints: The wheels are driven by a bounded control vector U. We have, at each instant, $|u_i| \leq u_{\max}, i = 1 \cdots 2n_a$, where u_i is the torque control of the ith wheel, and $u_{\max}$ is an arbitrary positive upper-bound. Minimizing sliding in the presence of friction contributes additional dynamic constraints that limit the range of admissible instantaneous translation/yaw accelerations (and wheel controls) to be applied on $\mathcal{A}$. We describe these constraints in more detail in Section IV-B.

D. The Motion Planning Problem

The motion planning problem is formally stated as: *Given an initial configuration, Q_{start} in $\mathcal{CS}_{\mathcal{A}}$ of $\mathcal{A}$, in contact with $\mathcal{T}$ and a subconfiguration, q_{goal} in $\mathcal{SCS}_{\mathcal{A}}$, find a feasible (piecewise-) smooth trajectory satisfying the task constraints in Section II-C (a curve, $\Gamma(t)$ in $\mathcal{SS}_{\mathcal{A}}$, and the corresponding controls, U in $\mathbb{R}^{2n_a}$), that takes $\mathcal{A}$ from the state $(Q_{\text{start}}, 0) = \Gamma(0)$ to a state $(Q, 0) = \Gamma(t_f)$ in a finite period of time, t_f, such that $q(t_f) = q_{\text{goal}}$. This means that the goal to achieve is a subset of $\mathcal{CS}_{\mathcal{A}}$ so that $q = q_{\text{goal}}$.*

In this paper, we search for solutions Γ that are of submin-imal length. For consistency, the suboptimality concerns only the length of the projection of the resulting paths on the plane (x, y) of $\mathcal{W}$.

III. The Instantaneous Forward Motion Solution

A. Physical Modeling of the Deformable Regions of $\mathcal{T}$

Classical techniques for analyzing the structure of 3-D deformable surfaces/structures are geometry-based and often time-consuming such as finite elements based methods [56]. In this section, we briefly describe a simple physics-based representation to cope with deformable regions of $\mathcal{T}$. The basic idea consists of synthesizing the deformations of these

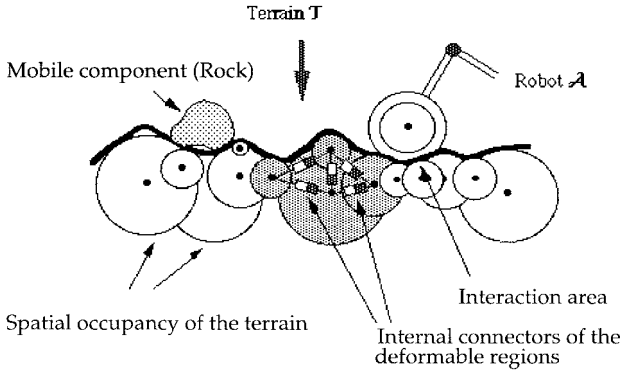

Fig. 5. The physical model of the terrain shown in Fig. 2.

regions by using a discrete physical model [37], [55]. For a simple presentation, we consider only visco-elastic behavior in response to the applied forces. Other types of behavior [55] (e.g., plastic, elasto-plastic, or visco-plastic) can conceptually be modeled and incorporated in our treatment. The physical model of a deformable region $\mathcal{T}_i$ is obtained by interconnecting the centers of the spheres $S_{i,j}$'s describing it (cf. Section II-A) by physical (spring-damper) connectors and considering that each center behaves as if it were a particle, $P_{i,j}$, of mass, $m_{i,j}$, (see Fig. 5). Each particle obeys Newton's dynamics and its motion is given by solving the equation $\boldsymbol{F}_d + \boldsymbol{F}_{P_{i,j}} + \boldsymbol{R}_{P_{i,j}} = m_{i,j}\ddot{\boldsymbol{r}}_{P_{i,j}}$, where $\boldsymbol{r}_{P_{i,j}}$ is the position of $P_{i,j}$ w.r.t. $\mathcal{F}_{\mathcal{W}}$, $\boldsymbol{F}_d$ is the gravity force, $\boldsymbol{F}_{P_{i,j}}$ is the resultant of the penalty forces provided by the spring-damper connectors, and $\boldsymbol{R}_{P_{i,j}}$ is the resultant of the contact forces between the sphere $S_{i,j}$ and the robot wheels and/or other movable components of $\mathcal{T}$. In our implementation, the connectivity structure and the stiffness/damping parameters are chosen arbitrarily.

B. Physical Modeling of $\mathcal{A}$

1) Basic Idea and Motivation: The formulation of the dynamics of rigid-body articulated chains involves complex nonlinear and highly coupled second-order differential equations because the system kinematics is imposed explicitly [3], [25], [26], [38]. In such a formulation, the joints are generally idealized and considered to be noncompliant. For reducing the set of coupled equations of the chain of $\mathcal{A}$ and coping explicitly with its compliance, we introduce a *hybrid* model that combines the mechanics of rigid-body chains and the physics of particle-based compliant systems. For the particular case of $\mathcal{A}$, the basic idea consists of using a rigid-body dynamic formulation for the parts of $\mathcal{A}$ which are actively controlled and/or involved in the computation of the contact interaction with $\mathcal{T}$ (i.e., the wheeled axles) and of using compliant discrete physical structures to model the passive joint mechanisms and links of the chassis. The model of $\mathcal{A}$ is seen as a collection of the n_a subchains of the wheeled axles and a network of interconnected particles acting to maintain the cohesion and the kinematic structure of the chassis. The kinematics is incorporated in the dynamic formulation by substituting the geometric and relative motion constraints between neighboring components of $\mathcal{A}$ with a set of penalty force/torque constraints (modeled by spring-dampers). The hybrid model permits formulating the dynamics of each basic subsystem as if it were independent of the others and permits coupling its dynamics only to the neighboring parts through the discrete mechanisms. This has the effect of decreasing the set of coupled dynamic equations of the entire robot system and simplifies the instantaneous forward motion problem since the dynamic equations of the different subsystems can be solved independently and in any order. In addition, our model makes it easy to cope with closed loops due to the contact between $\mathcal{A}$ and $\mathcal{T}$. This has been shown by Shih and Frank who described a model slightly similar to our model in order to study the gaits of a legged robot [45], [46].

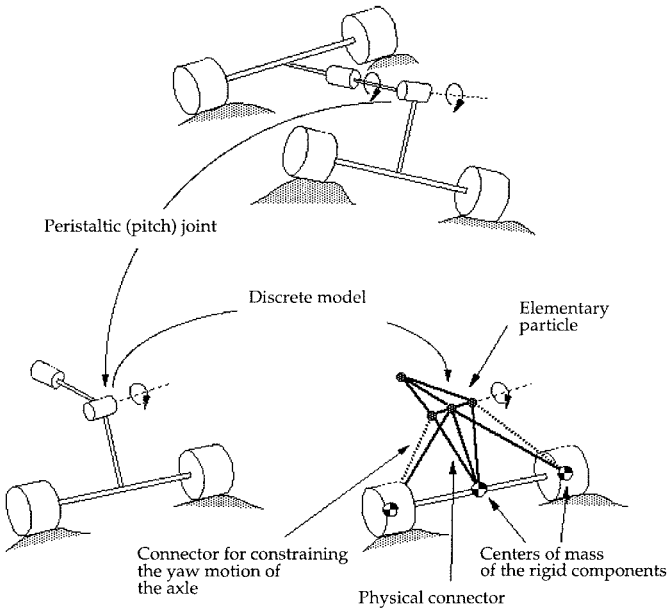

Fig. 6. Discrete physical model of the peristaltic (pitch) joint.

For illustration, let us consider the case of a passive pitch rotoid joint linking two given axles as depicted in Fig. 6. We first select a set of points (masses), $\{P_{ik}\}$ and $\{P_{jk'}\}$, located on the intersecting part of each link, respectively. The constraint of having only 1 degree of freedom between the two links is satisfied by connecting couples of points chosen from $\{P_{ik}\} \times \{P_{jk'}\}$ so that they remain on the joint axis. As shown in Fig. 6, further connectors may be added between the particles and the wheeled axles in order to enforce the constraints on the degrees of freedom of the chassis joints. Other specific structures may be applied for modeling a rotoid joint (see [12] for further examples). The resulting hybrid physical model of $\mathcal{A}$ is shown in Fig. 7.

2) Dynamics of $\mathcal{A}$: Let $\mathcal{E}_i$ be an axle, and let $\mathcal{F}_{\mathcal{E}_i}$ be a fixed frame on its center of mass. The axes of $\mathcal{F}_{\mathcal{E}_i}$ are chosen to be the principal inertia axes of $\mathcal{E}_i$ and the y-axis is chosen to be parallel to the axis of $\mathcal{E}_i$. The position/orientation vector of $\mathcal{E}_i$ w.r.t. $\mathcal{F}_{\mathcal{W}}$ is $(\boldsymbol{r}_{\mathcal{E}_i}, \Theta_{\mathcal{E}_i})$. Let θ_l and θ_r be the rotation angles of the wheels W_l and W_r w.r.t. $\mathcal{F}_{\mathcal{E}_i}$, respectively. $(\dot{\boldsymbol{r}}_{\mathcal{E}_i}, \omega_{\mathcal{E}_i}, \dot{\theta}_{l_i}, \dot{\theta}_{r_i})$ is the velocity vector of $\mathcal{E}_i$ ($\omega_{\mathcal{E}_i}$ is the rotation velocity of $\mathcal{E}$ written w.r.t. $\mathcal{F}_{\mathcal{E}}$). Let $n_{\text{part},\mathcal{A}}$ be the number of the particles, P_j, modeling the chassis. The state, E, of $\mathcal{A}$ is defined by: $E = \{(\boldsymbol{r}_{\mathcal{E}_i}, \Theta_{\mathcal{E}_i}, \theta_{l_i}, \theta_{r_i}, \dot{\boldsymbol{r}}_{\mathcal{E}_i}, \omega_{\mathcal{E}_i}, \dot{\theta}_{l_i}, \dot{\theta}_{r_i}), i = 1 \cdots n_a\} \cup \{(\boldsymbol{r}_j, \dot{\boldsymbol{r}}_j), j = 1 \cdots n_{\text{part},\mathcal{A}}\}$. The passive joints

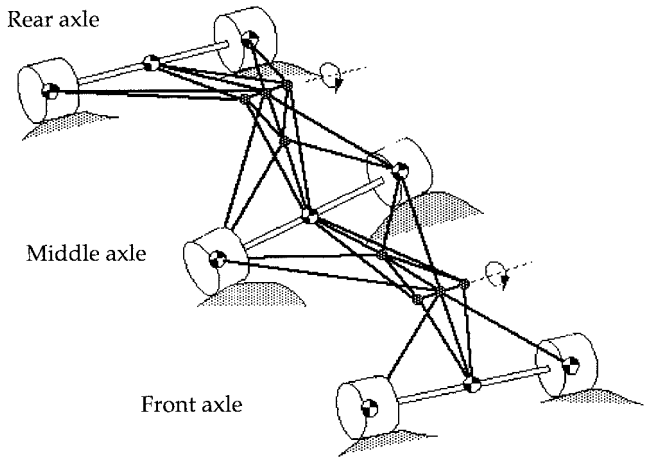

Fig. 7. The hybrid model of $\mathcal{A}$.

are implicit in E and can be determined, at each instant, by considering the spatial distribution of the P_j's.

The dynamics of the chassis is derived from the behavior of the collection of particles composing it. Each particle P_i obeys Newton's dynamics and its motion is given by solving the equation $\boldsymbol{F}_d + \boldsymbol{F}_{P_i} = m_i \ddot{\boldsymbol{r}}_i$, where $\boldsymbol{r}_i$ and m_i are the position and the mass of $P_i, \boldsymbol{F}_d$ is the gravity force, and $\boldsymbol{F}_{P_i}$ is the resultant of the penalty forces provided by the spring-damper connectors connected to P_i.

Let $\mathcal{E}$ be a given wheeled axle of $\mathcal{A}$, and let $m_\mathcal{E}$ and $\boldsymbol{I}_\mathcal{E}$ be its mass and its inertia matrix, respectively. The translation motion, w.r.t. $\mathcal{F}_\mathcal{W}$, is given by solving the following differential equation:

$$\boldsymbol{F}_d + \boldsymbol{R}_\mathcal{T} + \boldsymbol{F}_\mathcal{A} = m_\mathcal{E}\, \ddot{\boldsymbol{r}}_\mathcal{E}(t) \tag{1}$$

where $\boldsymbol{F}_d$ is the gravity force, $\boldsymbol{R}_\mathcal{T}$ is the contact forces applied by $\mathcal{T}$ on the wheels, and $\boldsymbol{F}_\mathcal{A}$ is the net forces applied on $\mathcal{E}$ by the other neighboring components of $\mathcal{A}$. Let $\boldsymbol{T}_\mathcal{E}$ be the net torque applied on $\mathcal{E}$ (w.r.t. $\mathcal{F}_\mathcal{E}$). The rotation motion of $\mathcal{E}$ is solved using Euler's equation $\boldsymbol{T}_\mathcal{E} = \dot{\boldsymbol{L}}_\mathcal{E}(t) = \dot{\boldsymbol{I}}_\mathcal{E}\boldsymbol{\omega}_\mathcal{E} + \boldsymbol{I}_\mathcal{E}\dot{\boldsymbol{\omega}}_\mathcal{E} = \boldsymbol{\omega}_\mathcal{E} \times (\boldsymbol{I}_\mathcal{E}\boldsymbol{\omega}_\mathcal{E}) + \boldsymbol{I}_\mathcal{E}\dot{\boldsymbol{\omega}}_\mathcal{E}$, i.e.,

$$\begin{aligned} T_{\mathcal{E},x} &= I_{\mathcal{E},x}\dot{\omega}_{\mathcal{E},x} + (I_{\mathcal{E},z} - I_{\mathcal{E},y})\omega_{\mathcal{E},y}\omega_{\mathcal{E},z} \\ T_{\mathcal{E},y} &= I_{\mathcal{E},y}\dot{\omega}_{\mathcal{E},y} + (I_{\mathcal{E},x} - I_{\mathcal{E},z})\omega_{\mathcal{E},x}\omega_{\mathcal{E},z} \\ T_{\mathcal{E},z} &= I_{\mathcal{E},z}\dot{\omega}_{\mathcal{E},z} + (I_{\mathcal{E},y} - I_{\mathcal{E},x})\omega_{\mathcal{E},x}\omega_{\mathcal{E},y}. \end{aligned} \tag{2}$$

Knowing the translation/rotation motion of $\mathcal{F}_\mathcal{E}$, we compute the wheel velocities by solving $T_{j,y} + u_j = I_y \ddot{\theta}_j, j = l, r$, where $T_{j,y}$ and u_j are the contact torque and control torque applied on the wheel W_j, I_y is the y-component of the inertia matrix of W_j, and $\ddot{\theta}_j$ is its rotation acceleration.

C. Modeling the Wheels/Ground Contact Forces

The interaction force applied on W is given by the summation of the contact forces computed at each contact occurring between W and different components of $\mathcal{T}$. In the following, we describe the interaction model at a single contact, c_i. Let $c_{W,i}$ and $c_{\mathcal{T},i}$ be the contact points located on W and $\mathcal{T}$, respectively, and let S_j be the sphere of the geometric model of $\mathcal{T}$ on which $c_{\mathcal{T},i}$ is located. We define a direct contact frame, $\mathcal{F}_{c_i} = (\mathbf{n}, \mathbf{t}, \mathbf{s})$, having its origin at $c_{\mathcal{T},i}$, where $\mathbf{n}$ is the outward normal to S_j at point $c_{\mathcal{T},i}$, and $\mathbf{t}$ and $\mathbf{s}$ are the tangent vectors to S_j. The vector, $\boldsymbol{t}$, is chosen to be perpendicular to the axis of the wheel as shown in Fig. 8. The contact interaction is modeled using a combination of a visco-elastic law (i.e., a spring/damper between the points $c_{W,i}$ and $c_{\mathcal{T},i}$) and/or a Coulomb friction model along each axis of $\mathcal{F}_{c_i}$ (see Fig. 8). At a given instant, let x_n, x_t, and x_s be the displacements of $c_{W,i}$ relative to $c_{\mathcal{T},i}$ along the axes of $\mathcal{F}_{c_i}$. Let κ_n, κ_t, and κ_s be the stiffness of the springs along each axis, respectively, and let $\eta_\mathrm{n}, \eta_\mathrm{t}$, and η_s be their damping coefficients. The contact force components are computed as follows.

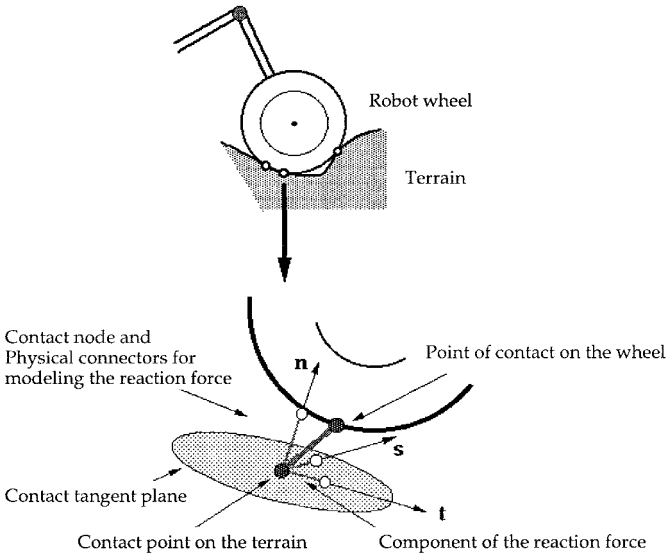

Fig. 8. Physical modeling of a contact reaction force.

Modeling the ground normal reaction, $\boldsymbol{N}_i$:

The reaction, $\boldsymbol{N}_i$, is modeled by a spring-damper acting only in the compression phase along $\mathbf{n}$. We have $\boldsymbol{N}_i = -\kappa_\mathrm{n} x_\mathrm{n} + \eta_\mathrm{n} \dot{x}_\mathrm{n}$ when $x_\mathrm{n} < 0$; otherwise, $\boldsymbol{N}_i$ is zero, and the contact is broken at point $c_{W,i}$.

Modeling the friction force, $\boldsymbol{f}_i$, between a wheel and $\mathcal{T}$:

When $\boldsymbol{N}_i$ is positive, the contact, $c_{\mathcal{W},i}$, may be either sticking or sliding on the surface of $\mathcal{T}$ and a Coulomb model of friction is used to solve for the tangential components, $\boldsymbol{f}_{i,\mathrm{t}}$ and $\boldsymbol{f}_{i,\mathrm{s}}$. In order to introduce the visco-elastic effect during the sticking phase, we assume that a static model of friction is valid when the tangential velocity of P_i relatively to P_i' is inferior to an arbitrary constant threshold, $\underline{\dot{x}}$ [42]. The friction forces are obtained by the following multivalued systems:

$$f_{i,\mathrm{r}} = \begin{cases} -\mu_{k,\mathrm{r}} N_i, & \text{if } \dot{x}_\mathrm{r} > \underline{\dot{x}}_\mathrm{r} \\ f'_{i,\mathrm{r}}, & \text{if } \dot{x}_\mathrm{r} \in [-\underline{\dot{x}}_\mathrm{r} \underline{\dot{x}}_\mathrm{r}] \\ \mu_{k,\mathrm{r}} N_i, & \text{if } \dot{x}_\mathrm{r} < -\underline{\dot{x}}_\mathrm{r} \end{cases} \tag{3}$$

where the subscript, $\mathrm{r} \in \{\mathrm{t}, \mathrm{s}\}$, and the case, $|\dot{x}_\mathrm{r}| > \underline{\dot{x}}_\mathrm{r}$, corresponds to the sliding phase of point, P_i, on the terrain. When this condition is not satisfied, P_i sticks on the contact surface, and the corresponding friction force $f'_{i,\mathrm{r}}$ is given by

$$f'_{i,\mathrm{r}} = \begin{cases} \mu_{s,\mathrm{r}} N_i & \text{if } \kappa_\mathrm{r} x_\mathrm{r} > \mu_{s,\mathrm{r}} N_i \\ \kappa_\mathrm{r} x_\mathrm{r} & \text{if } \kappa_\mathrm{r} x_\mathrm{r} \in [-\mu_{s,\mathrm{r}} N_i, \mu_{s,\mathrm{r}} N_i] \\ -\mu_{s,\mathrm{r}} N_i & \text{if } \kappa_\mathrm{r} x_\mathrm{r} < -\mu_{s,\mathrm{r}} N_i \end{cases} \tag{4}$$

where $\mu_{s,\mathrm{r}}$ and $\mu_{k,\mathrm{r}}$ are the static and kinetic coefficients of friction.

D. Generation of Instantaneous Forward Motions

Given the state $E(t)$ of $\mathcal{A}$, the state of $\mathcal{T}$, and a wheel control vector U, the instantaneous forward solution (routine `InstantaneousMotion`) aims to compute a new admissible state, $E(t+\delta t)$, reached by $\mathcal{A}$ after a small period of time, δt, and a new state of the movable/deformable components of $\mathcal{T}$. Section IV-B describes how this solution is used by the local planner to find feasible trajectories (including U).

algorithm `InstantaneousMotion`$(\mathcal{A}, \mathcal{T}, E(t), U, \delta t)$

1. For each wheel, $W_i, i = 1 \cdots 2n_a$ of $\mathcal{A}$, compute the contact force, $\boldsymbol{R}_i$ (cf. Section III-C).
2. Compute $E(t+\delta t)$ by solving the equations of motions of $\mathcal{A}$ (cf. Section III-B).
3. Compute the new state of (the particles of) the deformable regions of $\mathcal{T}$ (cf. Section III-A).
4. **if** $\mathcal{A}$ is not collision-free, **return**$(\emptyset)$.
5. **if** the contact is not maintained between $\mathcal{A}$ and $\mathcal{T}$, **return**$(\emptyset)$.
6. **if** the no tip-over constraint is not satisfied, **return**$(\emptyset)$.
7. **return**$(E(t+\delta t))$.

endalgorithm

In the following, we present an approximate analysis of what would be the worst-case complexity of the instantaneous solution in term of the complexity of the geometric and physical models of $\mathcal{A}$ and of the environment. Our analysis accounts only for computations of higher cost done for performing the routine `InstantaneousMotion` (i.e., solving the task dynamics and the distance computations needed for collision and contact detection tests). Our analysis assumes that a collision checking between two spheres, or a sphere and a convex geometric primitive modeling a wheel or a link of $\mathcal{A}$, takes a unit cost. It also assumes that the computation of a force/torque vector due to a contact or a physical connector takes a unit cost. We consider that the collection, $\mathcal{ST}$ of $n_{\mathcal{S}}\mathcal{T}$ spheres approximating $\mathcal{T}$, is represented by independent sets, $\mathcal{ST}_i$, of spheres, each of them representing a component, $\mathcal{T}_i$ of $\mathcal{T}$.

1) Collision Detection Between the Components of $\mathcal{T}$: This reduces to check a collision between each movable component, $\mathcal{T}i$, and the set, $\mathcal{T} \setminus \mathcal{T}_i$, which is $O(n_{\mathcal{ST}} n_{\mathrm{sph,mov},\mathcal{T}})$, where $n_{\mathrm{sph,mov},\mathcal{T}}$ is the total number of spheres representing the movable components.

2) Processing the Physical Model of the Terrain: The dynamics of a discrete model composed of n particles and m connectors is $O(m)$. (Note that $m > n$ in most cases). Processing a rigid object described by n spheres and updating the positions of these spheres is $O(n)$. Processing the physical model of the terrain is then $O(m_{\mathrm{def},\mathcal{T}} + n_{\mathrm{sph,mov},\mathcal{T}})$, where $m_{\mathrm{def},\mathcal{T}}$ is the number of connectors used to model the deformable regions of $\mathcal{T}$.

3) Collision Between $\mathcal{A}$ and $\mathcal{W}$: This stage uses the 2-D description of the obstacles, $\mathcal{B}_i$, and the set of spheres, $\mathcal{ST}$, to check for collision between $\mathcal{T}$ and parts of $\mathcal{A}$ other than the wheels. It is $O(n_{\mathrm{disc},\mathcal{B}} + n_{\mathrm{prim},\mathcal{A}} n_{\mathcal{ST}})$, where $n_{\mathrm{prim},\mathcal{A}}$ is the number of geometric primitives modeling $\mathcal{A}$ and $n_{\mathrm{disc},\mathcal{B}}$ is the number of discs used for approximating the $\mathcal{B}_i$'s.

4) Processing the physical model of $\mathcal{A}$: This includes solving the dynamics of the n_a axles (including the contact forces at the wheels) and the physical model of the chassis. The contact distribution depends on the considered local irregularities of $\mathcal{T}$. In [12], we have described a discretized model of the wheels enabling us to approximate surface contacts and upper-bound the number of contacts at a single wheel. In this case, the dynamics of an axle is solved in a bounded time and processing the entire dynamic model of $\mathcal{A}$ is $O(n_a + m_{\mathrm{part},\mathcal{A}})$, where $m_{\mathrm{part},\mathcal{A}}$ is the number of connectors used to model the chassis.

The required time t_{fs} of the instantaneous solution amounts to $O(n_{\mathcal{ST}}(n_{\mathrm{sph,mov},\mathcal{T}} + n_{\mathrm{prim},\mathcal{A}}) + n_{\mathrm{disc},\mathcal{B}} + m_{\mathrm{def},\mathcal{T}} + m_{\mathrm{part},\mathcal{A}} + n_a)$.

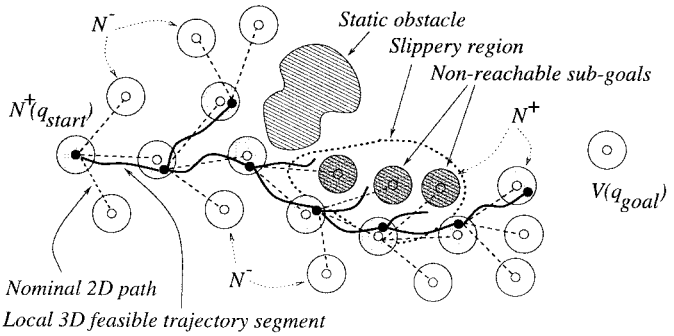

Fig. 9. The searched graph $\mathcal{G}$.

IV. The Motion Planning Algorithm

A. High Planning Level

The high planning level operates as a heuristic graph search (an A^*-type algorithm in our implementation) to find a near-optimal solution in a directed graph, $\mathcal{G}$, defined on $\mathcal{SCS}_{\mathcal{A}}$. Each node N of $\mathcal{G}$ is defined by a subconfiguration, q, reached by the planner and a neighborhood, $V(q)$ in $\mathcal{SCS}_{\mathcal{A}}$, centered in q. For a given $q_0, V(q_0) = \{(x,y)/d(x,y,x_0,y_0) \leq h_{xy}\} \times \{\theta/|\theta_0 - \theta| \leq h_\theta\}$, where h_{xy} and h_θ are two arbitrary positive constants, and $d(\ )$ is the Euclidean distance. We distinguish two types of nodes in the graph $\mathcal{G}$:

1) nodes denoted by N^+ which have been already processed by the local planner;
2) nodes denoted by N^- which have not been locally processed.

An arc pointing to a node, $N^+(q)$, corresponds to a feasible trajectory, Γ_i, moving $\mathcal{A}$ to $V(q)$ or an empty curve (i.e., q is not locally reachable). An arc pointing to a node, N^-, corresponds to a 2-D nominal path. The nodes, N^- (respectively, N^+), generated during the search are maintained in two lists, LN^- (respectively, LN^+). LN^- is maintained sorted according to the cost function, f_{cost}, in an ascending order. The basic algorithm is described by the routine `GlobalPlanner` (see Fig. 9, and [12] for a complete presentation).

algorithm `GlobalPlanner`$(Q_{\text{start}}, 0, q_{\text{goal}}, 0)$
  Let q_{start} *be the subconfiguration in* $\mathcal{SCS}_{\mathcal{A}}$ *(from* Q_{start}*),*
  $LN^+ \leftarrow \{N^+(q_{\text{start}})\}$;
  Compute new subgoals from q_{start};
  For each unvisited subgoal, create a node, N^-,
  and store it in LN^-;
  while $(LN^- \neq \emptyset)$
    $q_i \leftarrow$ `FirstSubGoal`(LN^-);
    $LN^- \leftarrow LN^- - \{N^-(q_i)\}$;
    $LN^+ \leftarrow LN^+ \cup \{N^+(q_i)\}$;
    if (`CollisionFree`(q_i))/* using the 2-D model of the $\mathcal{B}_i$'s */
      Let $q_{i,p}$ be the parent subgoal of q_i;
      Let $E(q_{i,p})$ be the state of $\mathcal{A}$ at $q_{i,p}$,
      Let $(\tilde{v}(q_i), \tilde{\omega}(q_i))$ be the velocities hypothesized at q_i,
      $(\Gamma_i, U_i, t_{\Gamma_i}) \leftarrow$ `LocalPlanner`$(E(q_{i,p}), (q_i, \tilde{v}(q_i), \tilde{\omega}(q_i)))$;
      Let $E(q_{i,r})$ be the state reached by the local planner,
      if ($q_{i,r}$ is within $V(q_{\text{goal}})$ and $\mathcal{A}$ has null velocities)
        $(\Gamma, U, t_\Gamma) \leftarrow$ `ExtractSolution`$(\mathcal{G}, (\Gamma_1, U_1, t_{\Gamma_1}), \cdots, \cdots, (\Gamma_i, U_i, t_{\Gamma_i}))$;
        return$((\Gamma, U, t_\Gamma))$;
      endif;
      if ($q_{i,r}$ is in th neighborhood $V(q_i)$)
        Compute new subgoals from $q_{i,r}$;
        For each unvisited subgoal, create a node, N^-,
        and store it in LN^-;
      endif;
    endif
  endwhile;
  return$(\emptyset)$;
endalgorithm;

When q_{goal} is reached, the trajectory Γ goes smoothly from Q_{start} to $V(q_{\text{goal}})$ through the trajectory segments, Γ_i, planned locally between the subgoals, q_i, lying on the near-shortest path in $\mathcal{G}$. For limiting the search, we have considered that the neighborhood of each subgoal can be visited only once with the exception that only $V(q_{\text{goal}})$ can be visited more than once for a good convergence.

1) Cost Function f_{cost}: At a subgoal q_i, f_{cost} is given by the summation of the length of the trajectory between q_{start} and $q_{i,p}$, the length of the nominal path between $q_{i,p}$ and q_i, and the (heuristic) distance between q_i and q_{goal}. This distance accounts for the kinematic constraints of $\mathcal{A}$ by using Reeds and Shepp's shortest paths [43].

2) Computing the Subgoals: Let $q_{i,p}$ be the configuration to be expanded in $\mathcal{G}$, and let us assume that the spent time at $q_{i,p}$ is 0 [*i.e.*, $q_{i,p} = q(0)$]. For the case of the three-axle robot and considering the nonholonomic kinematic constraints introduced in Section II-C6, we obtain $\dot{q} = (\dot{x}(t), \dot{y}(t), \dot{\theta}(t)) = (v\cos(\theta+\phi'), v\sin(\theta+\phi'), v(2\sin\phi'/L_r))$ which gives after integration for a period of time $\Delta T_G \gg \delta t$,

$$
\begin{aligned}
x(\Delta T_G) &= x(0) + \frac{L_r}{2\sin\phi'}\left(\sin(\theta(\Delta T_G)+\phi') - \sin(\theta(0)+\phi')\right)\\
y(\Delta T_G) &= y(0) - \frac{L_r}{2\sin\phi'}\left(\cos(\theta(\Delta T_G) + \phi') - \cos(\theta(0)+\phi')\right)\\
\theta(\Delta T_G) &= \theta(0) + v\,\frac{2\sin\phi'}{L_r}\,\Delta T_G\\
\tan\phi' &= \frac{L_r\tan\phi}{2L - L_r}.
\end{aligned}
\tag{5}
$$

This integration is done when assigning (v, ϕ) with constant extreme controls chosen among $\{-V_{\max}, V_{\max}\} \times \{-\phi_{\max}, 0, \phi_{\max}\}$. This leads to 3 forward subgoals (left and right turns, and straight motion) and three similar backward subgoals. In our results, ΔT_G is chosen so that the distance between $q_{i,p}$ and the q_i's is of the size of $\mathcal{A}$. For each q_i, we specify the translation/yaw velocity vector to be achieved by the local planner at q_i by either a nominal vector $(\tilde{v}_i, \tilde{\omega}_i)$ or hypothetical intervals where the velocity parameters must lie. These nominal values depend on whether q_i is a reversal point or not, and they aim to guarantee a good convergence of the local planner and the smoothness of Γ. Let $\mathcal{P}_i^0$ and $\mathcal{P}_{i,\text{goal}}$ be the curve used to generate q_i from $q_{i,p}$ and the first segment of the Reeds and Shepp's curve between q_i and q_{goal}, respectively. $(\tilde{v}_i, \tilde{\omega}_i)$ is determined as follows.

1) q_i **is a reversal point:** $\mathcal{P}_i^0$ and $\mathcal{P}_{i,\text{goal}}$ are of opposite sign (see left of Fig. 10), $(\tilde{v}_i, \tilde{\omega}_i) = (0, 0)$.
2) q_i **is not a reversal point:** When the velocity of $\mathcal{A}$ at $q_{i,p}$ is positive (respectively, negative), $\tilde{v}_i \in [0, V_{\max}]$ (respectively, $[-V_{\max}, 0]$). In both cases, $\tilde{\omega}_i$ is estimated as follows.
   a) $\mathcal{P}_i^0$ is a straight line path, $\tilde{\omega}_i = 0$.
   b) $\mathcal{P}_i^0$ and $\mathcal{P}_{i,\text{goal}}$ are both forward right or backward left turns, $\tilde{\omega}_i \in [-\dot{\theta}_{\max}, 0]$.
   c) $\mathcal{P}_i^0$ and $\mathcal{P}_{i,\text{goal}}$ are both forward left or backward right turns (see right of Fig. 10), $\tilde{\omega}_i \in [0, \dot{\theta}_{\max}]$.
   d) $\mathcal{P}_i^0$ is a left (respectively, right) turn and $\mathcal{P}_{i,\text{goal}}$ is a right (respectively, left) turn, $\tilde{\omega}_i = 0$.

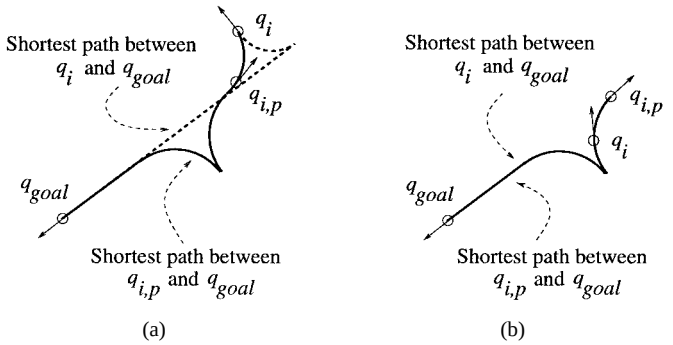

Fig. 10. subgoal q_i considered as (a) a reversal point or (b) a turning point.

B. Local Planning Level

Starting from the current state, $E_{i,p}$ of $\mathcal{A}$, the local planner uses the instantaneous forward motion solution to find a local feasible trajectory segment, Γ_i, moving $\mathcal{A}$ to a state E such that q is within $V(q_i)$ the velocity is close to $(\tilde{v}_i, \tilde{\omega}_i)$. In addition, we constrain Γ_i to have no backup maneuver (i.e., a sequence of a forward and a backward motion or the inverse) when $v_{i,p}$ and $\tilde{v}_i$ are of the same sign; otherwise, only a single backup is allowed. The latter case occurs when $v_{i,p}$ and $\tilde{v}_i$ are of opposite signs.

1) Search Algorithm: As in [9], [16], [17], [20], [24], [44], [48], we formulate locally motion planning as a graph search problem. The basic idea consists in applying a search process over a discrete representation of a subspace, $\mathcal{SSS}_{\mathcal{A}}$, of the state space, $\mathcal{SS}_{\mathcal{A}}$, that considers at its core the instantaneous solution. We define $\mathcal{SSS}_{\mathcal{A}}$ as a five-dimensional (5-D) space parameterized in terms of the vector $(x, y, \theta, v, \omega)$, where (v, ω) are the translation and yaw rotation velocities of $\mathcal{F}_{\mathcal{A}}$, respectively.

The local level operates as a *best-first search* algorithm. Considering the discrete representation—a cell-based representation—of $\mathcal{SSS}_{\mathcal{A}}$, it builds a tree where the nodes are the cells in which a substate of $\mathcal{A}$ has been reached by the search and the arcs are segments of feasible motions of $\mathcal{A}$. The tree is expanded from the cell containing the substate corresponding to $E_{i,p}$. The cell expansion is described in Section IV-B3. At each iteration of the search, the cell containing the closest substate to the subgoal is selected and expanded. The proximity to the subgoal is measured by the length of a simple nonholonomic 2-D path of type CSC ending at the subgoal, q_i, (where C is a circular arc of minimum admissible radius and S is a straight line). Furthermore, we consider that $\mathcal{A}$ is converging to q_i if such a path lies in the vicinity of the canonical path, $\mathcal{P}_i^0$, used by the high level. The search is iterated until the subgoal is reached [within the same cell or within the neighborhood, $V(q_i)$] or the set of cells has been fully explored without reaching the subgoal. As in Section IV-A, we limit the search by allowing each cell to be visited only once. In the first case, the local planner returns the local trajectory, Γ_i, the corresponding sequence of control vector, U, and the period of time, t_{Γ_i}, needed to move $\mathcal{A}$ along Γ_i. In the second case, the subgoal, q_i, is said nonreachable, and no new subgoal can be generated from it in the global graph, $\mathcal{G}$.

2) Discretized Representation of $\mathcal{SSS}_{\mathcal{A}}$: The discrete representation of $\mathcal{SSS}_{\mathcal{A}}$ is given by a collection, CL, of 5-D cells corresponding to a regular discretization of the domains of $(x, y, \theta, v, \omega)$. We denote by $\delta_x, \delta_y, \delta_\theta, \delta_v$, and δ_ω the discretization along each dimension. (We consider $\delta_x = \delta_y$). CL is defined on the following domains. The parameters θ and ω are varying in the intervals, $[0, 2\pi[$ and $[-\dot{\theta}_{\max}, \dot{\theta}_{\max}]$, respectively. If no backup maneuver has to be performed by $\mathcal{A}$ (i.e., $v_{i,p}$. $\tilde{v}_i \geq 0$), only cells of CL so that $v.\tilde{v}_i \geq 0$ and $|v| \leq V_{\max}$ are considered. If a backup motion has to be performed, CL includes cells so that v is bounded by $v_{i,p}$ and $-\mathrm{sgn}(v_{i,p})V_{\max}$. We constrain the motion of $\mathcal{A}$ to take place locally in a small region, $\mathcal{W}_i$ of $\mathcal{W}$, containing the positions of $\mathcal{A}$ at q_i and $q_{i,p}$. In the current implementation, $\mathcal{W}_i$ is given by a circular area having a center located at the middle point of the positions $(x_{i,p}, y_{i,p})$ and (x_i, y_i) and having a radius given by $d/2 + h_{xy} + L + \sqrt{2}\delta_x/2 + \epsilon$, where d is the distance between $(x_{i,p}, y_{i,p})$ and (x_i, y_i), h_{xy} is the radius of $V(q_i)$, L is the length of $\mathcal{A}$. ϵ is chosen arbitrarily equal to $d/2$ if $q_{i,p}$ is a reversal point; otherwise, it is set to 0.

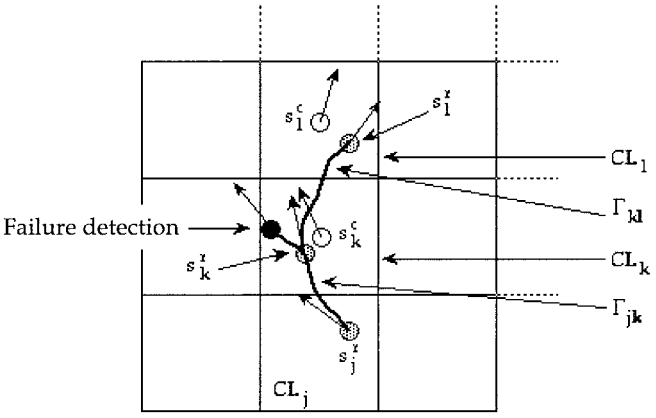

Fig. 11. Expansion of a cell CL_k.

3) Cell Expansion—Moving Between Adjacent Cells: Let s_k^r be a substate previously reached within a cell, CL_k, and let E_k^r be the complete state of $\mathcal{A}$ at s_k^r. We denote by s_k^c the substate corresponding to the center of CL_k. The expansion of CL_k (or s_k^r) follows the routine, `CellExpansion` (see Fig. 11).

---

algorithm `CellExpansion`$(E_k^r, CL_k, u_{\max}, \mathcal{A}, \mathcal{T}, \delta t)$

1 Starting from E_k^r, estimate the range of instantaneous admissible translation/yaw accelerations of $\mathcal{A}$;
2 Select a discrete set of nominal accelerations;
3 **for** each of these nominal acceleration vectors:
Let $(\tilde{v}_k, \tilde{\omega}_k)$ be the selected vector;
4 **do**
5 Knowing $(\tilde{v}_k, \tilde{\omega}_k)$, compute a bounded control torque vector, $U_k(t)$, to apply on the wheels;
6 Apply `InstantaneousMotion` $(\mathcal{A}, \mathcal{T}, E(t), U_k(t), \delta t)$;
7 **while** (**not** `end_condition`);
8 **if** (a non visited cell, CL_l, is reached)
9 CL_l becomes adjacent to CL_k in the local graph;
10 **endfor**;
11 **return** (the set of adjacent cells, CL_l, and trajectory segments, Γ_{kl}, connecting CL_k to them);

endalgorithm;

When no unvisited cell has been reached, CL_k cannot be expanded in the local graph, and the system is deemed to be in a local minimum located in CL_k. The graph search provides then another cell to be processed. Steps 1, 2, and 5 of the routine, `CellExpansion`, will be detailed in Section IV-B.4. The predicate, `end_condition`, used in step 7 defines

the conditions for which the motion generation is processed (steps 5 and 6). Let Γ_{kl} be the trajectory segment being generated. The conditions are

1) The resulting state, $E(t)$ of $\mathcal{A}$, is admissible.
2) The number of time the instantaneous solution is applied is upper-bounded by an arbitrary positive integer, n_{fs}[4].
3) The reached substate s is within the region of $\mathcal{SSS}_{\mathcal{A}}$ defined by the cell-decomposition (constraint on the local workspace, $\mathcal{W}_i$, and the velocity bounds).
4) The part of Γ_{kl} within the cell, CL_k, is so that $\mathcal{A}$ is converging toward the subgoal, q_i (i.e., $\text{length}(\mathcal{P}_i(t)) \leq \text{length}(\mathcal{P}_i(t-\delta t))$) and the lateral and longitudinal sliding velocities are bounded.
5) The part of Γ_{kl} within the adjacent cell, CL_l, is so that the sliding velocities of $\mathcal{A}$ are bounded and decreasing. When this condition is satisfied, the substate, s_l^r, at which the motion generation is stopped corresponds to the first substate at which the type of the path, $\mathcal{P}_i$, changes, or to the closest substate to s_l^c (center of CL_l) if the type of $\mathcal{P}$ remains the same.

We should point out that although $(\tilde{\dot{v}}_k, \tilde{\dot{\omega}}_k)$ is maintained unchanged during the motion between two adjacent cells (loop defined by step 4–7), $U_k(t)$ is updated at each instant in order to incorporate the current contact interaction with the terrain. A more elaborate scheme may consist in characterizing acceleration bounds so that the robot dynamics, which is state-dependent, remains locally constant [17], [24]. For instance, this key property has been used in [24] for designing a provably good approximation solution to kinodynamic motion planning of an open kinematic chain. In our case, the dynamic equations of $\mathcal{A}$ depend also on the contact interactions and possibly on the state of movable components of the terrain. This feature makes it much difficult to estimate such bounds. However, our scheme is fairly reasonable since the forward motion of $\mathcal{A}$ and its actual accelerations are computed at each increment, δt, while the control torques are guaranteed to remain bounded (steps 5 and 6).

4) Computing Nominal Accelerations and Wheel Controls:

Acceleration Range (Step 1 in `CellExpansion`*):* The approximation done for characterizing the range of instantaneous admissible accelerations of $\mathcal{A}$ assumes that all the wheels are contacting rigid surfaces and that the center, $C_{\mathcal{A}}$ of $\mathcal{A}$, remains instantaneously in a plane defined by the axes, $x_{\mathcal{A}}$ and $y_{\mathcal{A}}$) of $\mathcal{F}_{\mathcal{A}}$. In Section III-B, the dynamics of $\mathcal{A}$ is described by a hybrid system allowing us to solve for instantaneous motions of $C_{\mathcal{A}}$ by only considering the dynamics of the middle axle, $E_{\mathcal{A}}$. Let a_x and a_y be the translation accelerations of $E_{\mathcal{A}}$ along $x_{\mathcal{A}}$ and $y_{\mathcal{A}}$ written w.r.t. $\mathcal{F}_{\mathcal{A}}$, respectively. We have $\dot{v} = (a_x^2 + a_y^2)^{1/2}$. a_x and a_y are obtained by re-writing (1) w.r.t. $\mathcal{F}_{\mathcal{A}}$ as

$$m_{\mathcal{E}} a_x = g_x + F_{c,x} + f_x^r + f_x^l \tag{6}$$

$$m_{\mathcal{E}} a_y = g_y + F_{c,y} + f_y^r + f_y^l \tag{7}$$

[4]This upper-bound is considered in order to guarantee that a trajectory segment between the cells is found in a finite period of time. Due to the 3-D irregularities of $\mathcal{T}$, it is quite difficult to characterize precisely what n_{fs} would be in the general case. We consider that n_{fs} is $\lceil 5\delta_x / V_{\max}\delta t \rceil$ (i.e., about five times the time required to move along a cell edge in the plane, δ_x, with a maximum translation velocity).

where $m_{\mathcal{E}}$ is the mass of the middle axle, $E_{\mathcal{A}}, g$ is the gravitational force, and F_c is the resulting force applied by others parts of $\mathcal{A}$ on $E_{\mathcal{A}}$ (cf. Section III-B). f^l and f^r are the contact friction forces applied on the left and the right wheels, respectively. The yaw acceleration is given by rewriting the z component of (2) w.r.t. $\mathcal{F}_{\mathcal{A}}$ as

$$I_{\mathcal{E},z}\dot{\omega} = (f_x^r - f_x^l)l + M(F_c, \omega_{\mathcal{E},x}, \omega_{\mathcal{E},y}) \tag{8}$$

where $I_{\mathcal{E},z}$ is the moment of inertia of $E_{\mathcal{A}}$ along the z axis, l is the distance between the centers of the wheels and $M(F_c, \omega_{\mathcal{E},x}, \omega_{\mathcal{E},y})$ is the summation of the net torque applied by the forces involved in the computation of F_c and $(I_{\mathcal{E},x} - I_{\mathcal{E},y})\omega_{\mathcal{E},x}\omega_{\mathcal{E},y}$. In (8), the torque due to f_y was neglected assuming that the contact is a single point on each wheel during the displacement of $\mathcal{A}$.

The conditions of bounded wheel controls and no (longitudinal) sliding yield, respectively

$$a_x \geq \frac{1}{m_{\mathcal{E}}}\left[-2\,\frac{u_{\max}}{R} + g_x + F_{c,x}\right]$$

$$a_x \leq \frac{1}{m_{\mathcal{E}}}\left[2\,\frac{u_{\max}}{R} + g_x + F_{c,x}\right] \tag{9}$$

$$a_x \geq \frac{1}{m_{\mathcal{E}}}\left[-\mu_t^l N^l - \mu_t^r N^r + g_x + F_{c,x}\right]$$

$$a_x \leq \frac{1}{m_{\mathcal{E}}}\left[\mu_t^l N^l + \mu_t^r N^r + g_x + F_{c,x}\right] \tag{10}$$

where R is the radius of a wheel, μ_t^r and μ_t^l are the static coefficients of (longitudinal) friction at the contact point located on the right and the left wheels, respectively, and N^r and N^l are the reaction forces of the ground at these wheels. The friction coefficient at a given wheel is chosen to be the highest one among those involved in the computation of the contact forces at the wheel.

The constraints of no (lateral) sliding and of bounded curvature of the paths of $\mathcal{A}$ yield, respectively

$$a_y \geq \frac{1}{m_{\mathcal{E}}}\left[-\mu_s^l N^l - \mu_s^r N^r + g_y + F_{c,y}\right]$$

$$a_y \leq \frac{1}{m_{\mathcal{E}}}\left[\mu_s^l N^l + \mu_s^r N^r + g_y + F_{c,y}\right] \tag{11}$$

$$-v^2/\rho_{\min} \leq a_y \leq v^2/\rho_{\min} \tag{12}$$

where μ_s^r and μ_s^l are the static coefficients of (lateral) friction at the right and the left wheels, respectively, and $\rho_{\min} = L_r/(2\sin\phi'_{\max})$ is the minimum turning radius of $\mathcal{A}$ with $\tan\phi'_{\max} = \tan\phi_{\max} L_r/(2L - L_r)$ (cf. Section II-C6).

As for (9), bounding the wheel controls yields

$$\dot{\omega} \geq \frac{1}{I_{\mathcal{E},z}}\left[-2\,\frac{u_{\max}}{R}\,l + M(F_c, \omega_{\mathcal{E},x}, \omega_{\mathcal{E},y})\right]$$

$$\dot{\omega} \leq \frac{1}{I_{\mathcal{E},z}}\left[2\,\frac{u_{\max}}{R}\,l + M(F_c, \omega_{\mathcal{E},x}, \omega_{\mathcal{E},y})\right]. \tag{13}$$

The dynamic constraints given by inequalities (9)–(13) define the range of admissible accelerations that $\mathcal{A}$ can have at a given state, E.

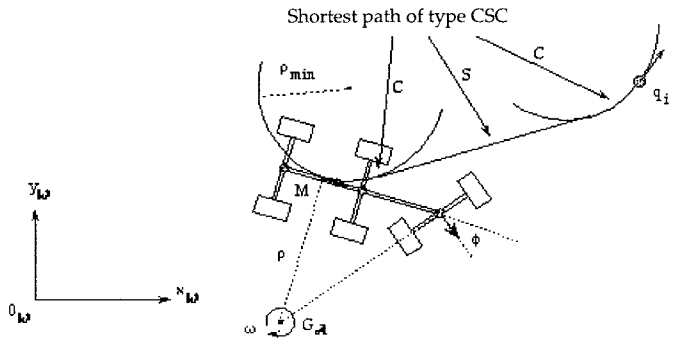

Fig. 12. Case of $\omega_{\mathcal{P}} > 0$ and $\omega < 0$.

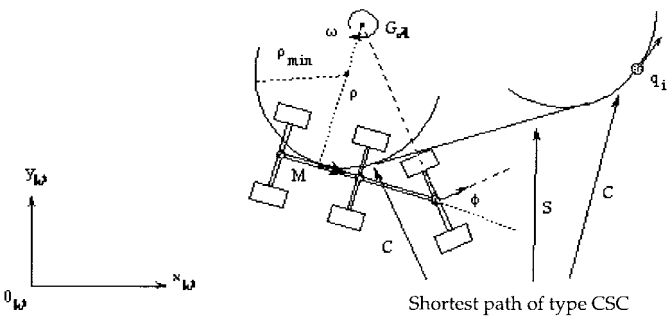

Fig. 13. Case of $\omega_{\mathcal{P}} > 0$ and $\omega > 0$.

Choosing the Nominal Accelerations (Step 2 in `CellExpansion`*):* Let $s = (q, v, \omega)$ be the substate to be expanded. For practicality, we consider only a subset of nominal accelerations chosen among the extreme bounds defined by (9)–(13). Such a reduction is achieved by analyzing the type of shortest path, $\mathcal{P}$ of type CSC, connecting q to the subgoal, q_i, and the direction of instantaneous lateral sliding velocity observed at the current state, $E(s)$. *W.l.o.g.,* we present in the following such an analysis when the shortest 2-D path, $\mathcal{P}_i^0$ (with end-points $q_{i,p}$ and q_i), corresponds to a forward motion (i.e., the velocity $\tilde{v}_i$ at q_i is positive).

The lateral sliding velocity, $\boldsymbol{v}_t$, is given by the projection of the translation velocity vector of $\mathcal{A}$ on the y axis of $\mathcal{F}_{\mathcal{A}}$. Let us consider first the case of a motion without a backup maneuver. Let $\mathcal{P} = C_1SC_2$ be the 2-D path computed between q and q_i, and let $\omega_{\mathcal{P}}$ be the rotation velocity needed to track the first portion of $\mathcal{P}$ (i.e., C_1, or S if $C_1 = \emptyset$, or C_2 if $C_1 = S = \emptyset$). $(\tilde{v}_a, \tilde{\omega}_a)$ (respectively, $(\tilde{v}_d, \tilde{\omega}_d)$) refers to the extreme acceleration (respectively, deceleration) vector selected at s. We reduce the set of nominal accelerations as follows.

1) $\omega_{\mathcal{P}} > 0$ **and** $\omega < 0$: $\tilde{\omega}$ is chosen to be equal to $\tilde{\omega}_d$. When no sliding is observed, $\tilde{v} \in \{\tilde{v}_d, \tilde{v}_a\}$ (as in Fig. 12). Otherwise, $\tilde{v}$ is chosen by comparing the directions of $\boldsymbol{v}_t$ and the vector, $\boldsymbol{x}_i$ of $\mathcal{F}_{\mathcal{A}}$, when $\mathcal{A}$ is at q_i. When $\boldsymbol{v}_t.\boldsymbol{x}_i < 0$, only 0 and $\tilde{v}_d$ are considered, otherwise $\tilde{v} \in \{\tilde{v}_d, 0, \tilde{v}_a\}$. The case where $\omega_{\mathcal{P}} < 0$ and $\omega > 0$ is processed in an analog way.
2) $\omega_{\mathcal{P}} > 0$ **and** $\omega > 0$: $(\tilde{v}, \tilde{\omega}) \in \{\tilde{v}_d, 0, \tilde{v}_a\} \times \{\tilde{\omega}_d, 0, \tilde{\omega}_a\}$ (as in Fig. 13). The same choice is considered if $\omega_{\mathcal{P}} < 0$ and $\omega < 0$.
3) $\omega_{\mathcal{P}} > 0$ **and** $\omega = 0$: $(\tilde{v}, \tilde{\omega}) \in \{\tilde{v}_d, 0, \tilde{v}_a\} \times \{0, \tilde{\omega}_a\}$.
4) $\omega_{\mathcal{P}} < 0$ **and** $\omega = 0$: $(\tilde{v}, \tilde{\omega}) \in \{\tilde{v}_d, 0, \tilde{v}_a\} \times \{\tilde{\omega}_d, 0\}$.
5) $\omega_{\mathcal{P}} = 0$ **(i.e.,** $C_1 = \emptyset$**):** If the current rotation velocity is negative (respectively, positive), $(\tilde{v}, \tilde{\omega})$ is chosen as in the case of $\omega_{\mathcal{P}} > 0$ and $\omega < 0$ (respectively, $\omega_{\mathcal{P}} < 0$ and $\omega > 0$). When ω is in the vicinity of 0, $\tilde{v} \in \{\tilde{v}_d, 0, \tilde{v}_a\}$. When no sliding is observed, $\tilde{\omega} = 0$, otherwise $\tilde{\omega}$ is chosen according to the direction of $\boldsymbol{v}_t$. Thus, $\tilde{\omega} \in \{0, \tilde{\omega}_a\}$ (respectively, $\{\tilde{\omega}_d, 0\}$) if $\boldsymbol{v}_t < 0$ (respectively, $\boldsymbol{v}_t > 0$).

For a reversal motion, $(\tilde{v}, \tilde{\omega}) \in \{\tilde{v}_a\} \times \{\tilde{\omega}_d, 0, \tilde{\omega}_a\}$.

Nominal Wheel Control U (Step 5 in `CellExpansion`*):* Let $(\tilde{\dot{v}}, \tilde{\dot{\omega}})$ be the nominal acceleration of the center of a given axle, $\mathcal{E}$, and W_l and W_r be its left and right wheels, respectively. We denote by θ_l (respectively, θ_r) the angle of rotation of W_l (respectively, W_r) w.r.t. its axis (i.e., Y axis of $\mathcal{E}$). Under the assumption that the motion of the wheels is of pure rolling and that robot system remains, during δt, in a plane parallel to the (x, y) plane of $\mathcal{F}_{\mathcal{A}}$, we have: $\tilde{\dot{v}} = (\tilde{\ddot{\theta}}_r + \tilde{\ddot{\theta}}_l)R/2$ and $\tilde{\dot{\omega}} = (\tilde{\ddot{\theta}}_r - \tilde{\ddot{\theta}}_l)R/d$. Knowing $\tilde{\ddot{\theta}}_l$ and $\tilde{\ddot{\theta}}_r$, the control parameters, u_l and u_r of U, to be applied on both the wheels are computed from the inverse dynamics: $u_j = -T_{j,y} + I_y\tilde{\ddot{\theta}}_j, j = l, r$, where $T_{j,y}$ is the torque applied on W_j by the terrain $\mathcal{T}$. When $|u_j| > u_{\max}$, we set u_j to $\mathrm{sgn}(u_i).\ \max(|u_i|, u_{\max})$. When the contact is broken at a wheel, no torque is applied and we have $u_j = 0$.

C. Algorithm Analysis

In this section, we give a somewhat informal analysis of what would be the expected overall complexity of the planner (in terms of the complexity of the geometric/physical models of the task and the discretization used by the two planning levels), and we comment on it completeness.

1) Complexity: The high planning level constrains the subgoals of the graph, $\mathcal{G}$, to be visited only once. The maximum number of subgoals that are generated is equal to the maximum number, n_{cyl}, of cylindrical regions (i.e., neighborhoods $V(q)$) that can cover $\mathcal{SCS}_{\mathcal{A}}$. This number is of the form $n_{cyl} = n_{xy}n_\theta$, where $n_\theta = \lceil 2\pi/2h_\theta \rceil$. n_{xy} is the maximum number of circles of radius h_{xy} that can cover (x, y)-plane of $\mathcal{W}$ with the constraint that the distance between each pair of centers has to be greater or equal to h_{xy}. Determining n_{xy} can be reduced to the problem of determining the maximum number of discs of radius $h_{xy}/2$ to be used for *packing* a rectangular area [13]. In our case, n_{xy} is of the form $n_{xy} = \lceil 0.906\,90 L_{\mathcal{W},x}L_{\mathcal{W},y}/h_{xy}^2 \rceil$, where $L_{\mathcal{W},x}$ and $L_{\mathcal{W},y}$ are the length and the width of $\mathcal{W}$. Hence, n_{cyl} is $O(L_{\mathcal{W},x}L_{\mathcal{W},y}/(h_{xy}^2 h_\theta))$. As each node of $\mathcal{G}$ is visited only once, we have, in the worst-case, a graph with the same number of arcs and nodes (i.e., n_{cyl}). We have used a locally consistent heuristic based on the length of Reeds and Shepp's shortest paths. (This defines a metric in the space $R^2 \times S^1$ [35] which is diffeomorphic to the reduced C-space of $\mathcal{A}$). Hence, the A^* search is in the worst-case, $O(t_{lp}(L_{\mathcal{W},x}L_{\mathcal{W},y}/h_{xy}^2 h_\theta \log(L_{\mathcal{W},x}L_{\mathcal{W},y}/h_{xy}^2 h_\theta)$, where t_{lp} is the worst-case time required by the local planner.

In the following, we discuss t_{lp}. The best-first search applied at the local level is, in the worst-case,

$O(t_{l\text{cells}} n_{l\text{cells}} \log n_{l\text{cells}})$, where $t_{l\text{cells}}$ is the worst-case time to move $\mathcal{A}$ between two adjacent cells and $n_{l\text{cells}}$ is the number of cells representing the searched regions of the five-dimensional space, $\mathcal{SSS}_{\mathcal{A}}$. As presented in Section IV-B2, the number of cells along x or y is $O(\Delta T V_{\max}/\delta_x)$ ($\delta_x = \delta_y$), and along the translation/yaw velocities is $O(V_{\max}/\delta_v)$ and $O(\dot{\theta}_{\max}/\delta_\omega)$, respectively. Hence, $n_{l\text{cells}}$ is $O(\Delta T^2 V_{\max}^3 \dot{\theta}_{\max}/(\delta_x \delta_y \delta_\theta \delta_v \delta_\omega))$. $t_{l\text{cells}}$ is of the form $n_{fs} t_{fs}$, where t_{fs} is the time required to process the instantaneous forward solution (cf. Section III-D) and n_{fs}, which is $O(\lceil \delta_x / V_{\max} \delta t \rceil)$, is the (input) upper-bound on the number of times this solution is processed between two adjacent cells of $\mathcal{SSS}_{\mathcal{A}}$ (cf. Section IV-B3). The worst-case time overall the planner takes is $O(m \log(L_{\mathcal{W},x} L_{\mathcal{W},y}/h_{xy}^2 h_\theta) \log(\Delta T^2 V_{\max}^3 \dot{\theta}_{\max}/\delta_x \delta_y \delta_\theta \delta_v \delta_\omega))$, where $m = (n_{fs}\ t_{fs} L_{\mathcal{W},x} L_{\mathcal{W},y} \Delta T^2 V_{\max}^3\ \dot{\theta}_{\max}/h_{xy}^2\ h_\theta\ \delta_x \delta_y\ \delta_\theta \delta_v\ \delta_\omega)$.

In our implementation, collision/contact checking is performed using an incremental scheme based on a hierarchical description of the sets of spheres [12]. During local planning, it is performed only on a subset of the components located in the neighborhood of local workspaces, $\mathcal{W}_i$ [cf. Section IV-B2)]. This has the effect of applying the collision checker for only a smaller number of geometric primitives (spheres and polygons). As it will be presented in Section V, the run time of the planner is of the order of one to a few hours, depending on the problem size. Such performance must be expected if one deals with dynamic and contact interaction constraints for an articulated mechanical chain such as $\mathcal{A}$, and is likely to be common to most dynamic motion planning algorithms. For instance, Shiller and Gwo reported the same order of processing times in [49] when the robot is reduced to a point and the terrain is smooth. We believe, however, that the use of more efficient collision checking schemes will contribute to significantly enhancing the current performance of the planner. In Section VI, we report on other possible practical extensions.

2) Completeness: Due to the approximations/discretization we have made, and since the two planning levels search only subsets of the C-space and the state space of $\mathcal{A}$, the planner is not complete but it solved often several nontrivial tasks.

The high level searches only the reduced C-space, $\mathcal{SCS}_{\mathcal{A}}$, and does not account for the velocity space of the robot. This clearly yields a lost in the completeness of the planner. Indeed, the robot is constrained to steer toward each subgoal, only once, with a velocity lying in a heuristic range. In some cases, this range may be different from the actual set of feasible velocities, and the subgoal is declared to be nonreachable although a local trajectory may be planned when it is processed with a different velocity range. For overcoming somewhat this limitation, we have allowed the nominal velocities at certain subgoals to lie within large intervals (e.g., all positive or negative velocities). In some cases, such an estimation is, however, not sufficient. For instance, it is difficult and sometimes impossible to plan a backup maneuver at a subgoal which has nonzero velocity. A straightforward extension to make the planner account globally for the robot velocity space consists of applying directly the local planner to solve for the entire problem. (In this case, the search operates on the entire reduced state space $\mathcal{SSS}_{\mathcal{A}}$). For such a scheme and when the same discretization scale is used, global planning is expected to be $O((n_{fs} t_{fs} L_{\mathcal{W},x} L_{\mathcal{W},y} V_{\max} \dot{\theta}_{\max}/\delta_x \delta_y \delta_\theta \delta_v \delta_\omega) \log(L_{\mathcal{W},x} L_{\mathcal{W},y} V_{\max} \dot{\theta}_{\max}/\delta_x \delta_y \delta_\theta \delta_v \delta_\omega))$. The run times in practice may be, however, significantly larger than the performances of our two-level scheme.

In order to enhance the current completeness of the planner, it would be possible 1) to consider, at the local level, all possible nominal accelerations during the motion generation between adjacent cells[5], and/or 2) to choose a fine cell-based representation of the state space at the local level. These points, together with the extension of the planner to search directly a solution in the robot reduced state space, lead unavoidably to increasing the complexity of planning and yields a trade-off between the completeness and the desired practicality of the planner.

V. Simulation Results

The described models and the algorithms were implemented in C on a SUN Sparc 10 workstation, and several task examples have been successfully performed in simulation for a nonholonomic six-wheeled vehicle moving on different terrains. The following parameters are considered.

1) $L = 0.9$ m, $l = 0.75$ m, and $L_r = 0.45$ m.
2) Physical model of $\mathcal{A}$: eight particles have been used to model the chassis of $\mathcal{A}$ as depicted in Fig. 7, each of them has a mass of 10 kg. The stiffness and damping coefficients of the connectors between them are 2000 N/m and 500 Ns/m, respectively. The mass of each wheeled axle is 100 kg.
3) Physical model of the terrain:
   a) The stiffness and damping coefficients of the connectors used to compute the reaction forces are 2000 N/m and 500 Ns/m, respectively.
   b) The stiffness and damping coefficients of connectors used to model the deformable regions vary in the intervals [500 N/m, 1000 N/m] and [250 Ns/m, 500 Ns/m], respectively.
4) For non slippery areas, the static (respectively, kinetic) coefficient of friction varies within the interval [0.4, 0.9] (respectively, [0.2, 0.6]).
5) $V_{\max} = 5$ m/s and $\phi_{\max} = 0.4$ radian.
6) $u_{\max} = 300$ Nm
7) $\varphi_{\max} = \Pi/2$.
8) $\Delta T = 0.2$ s and $\delta t = 0.005$ s. δt has been chosen to be too small in order to avoid divergence of the differential equations of motion due to the stiffness of the connectors of the physical models.
9) Neighborhood $V(q)$: $h_{xy} = 0.25$ m and $h_\theta = 0.2$ radian.
10) Local cells: $\delta_x = \delta_y = 0.1$ m, $\delta_\theta = 0.1$ radian, $\delta_v = 0.25$ m/s, and $\delta_\omega = 0.25$ radian/s.

[5]Because in our implementation, the subgoals were not too distant from each other, we reduced heuristically the set of nominal accelerations when expanding a cell. Some solutions may be lost when ΔT grows.

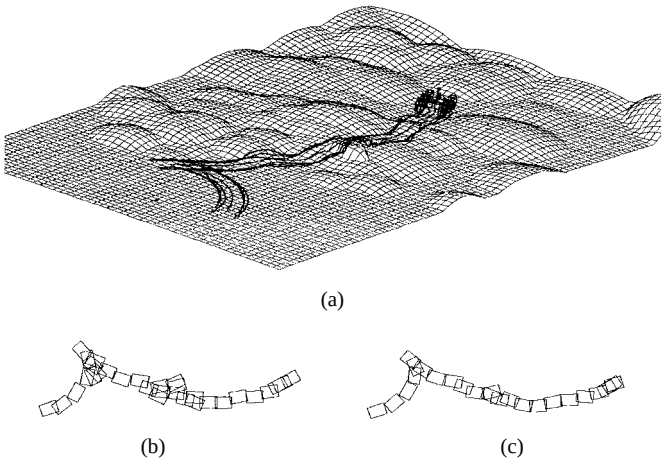

Fig. 14. (a) Planned trajectory crossing a static rock, (b) subgoals generated by the global planner and processed locally, and (c) subgoals reached locally.

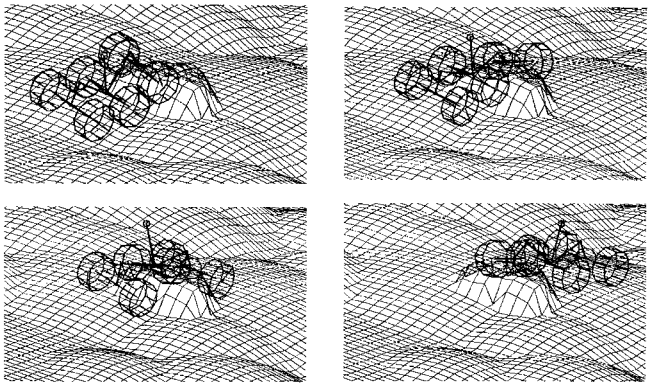

Fig. 15. Motion of $\mathcal{A}$ on a static rock.

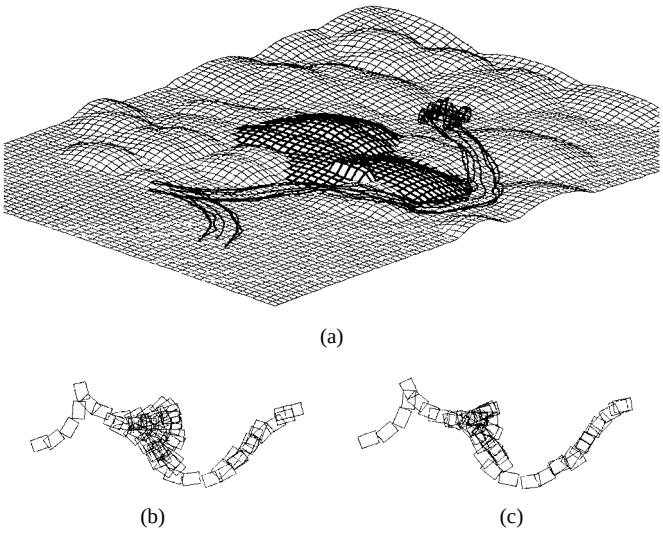

Fig. 16. (a) Planned trajectory avoiding slippery areas (dark regions), (b) subgoals of $\mathcal{SCS}_{\mathcal{A}}$ processed by the local planner, and (c) corresponding reached configurations.

In the example shown in Fig. 14, the global trajectory planned for $\mathcal{A}$ to reach its final goal is composed of a backup maneuver followed by a displacement on an irregular area corresponding to a rock of the size of a wheel of $\mathcal{A}$. The set of subgoals processed by the local planner and the corresponding configurations that were reached are shown on the (top and bottom) right of Fig. 14, respectively. (For simplifying the illustration, $\mathcal{A}$ is depicted by a rectangle). The comparison of the set of configurations shows that some of the subgoals were detected as nonreachable by the local planner. This was due to important slippage at the wheels. Some other subgoals corresponding to possible reversal points were not reached because $\mathcal{A}$ had a relatively high translation and/or rotation velocities at these configurations. This failure is explained by the fact that the velocity parameters of $\mathcal{A}$ are not incorporated during the expansion of the nodes of the search graph, $\mathcal{G}$. Among the subgoal configurations generated by the high level in $\mathcal{SCS}_{\mathcal{A}}$, 30 of them were processed by the local planner and seven were detected as nonreachable. The planned trajectory, Γ, goes through the neighborhoods of 22 of them. (Γ is shown by the traces of the centers of the wheels and the main body of $\mathcal{A}$). In this example, $\mathcal{A}$ reached its final goal with an error of 0.11 m in position (x, y) and 0.01 radian in the yaw orientation θ. Γ required a run time of about 27 min. Such a time performance is explained by the fact that the time step, δt, considered for solving the differential equations of the dynamic system was deliberately chosen to be very small. Fig. 15 shows some intermediate configurations reached by $\mathcal{A}$ when it crosses the rock.

In the example shown in Fig. 16, the initial and final configurations of $\mathcal{A}$ are chosen so that are they separated by a slippery and uneven region. The coefficients of static and kinetic friction associated with this area were both set to 0. Planning a feasible solution required a run time of about 1 h 23 min. In addition to the above mentioned remark concerning the parameter, δt, a large part of the computational burden is due to the search for local admissible movements performed by the local planner on the slippery area. Depending on the velocity of $\mathcal{A}$ when it is at its starting state and the slope of the terrain, the local planner failed several times in finding admissible trajectories allowing the robot to steer toward its subgoals.

In Fig. 17, we present an example where the slippery area has been enlarged. We also decreased the upper-bounds of the slippage velocities. This led to a significant difference in the explored regions of the search space and in the results of processing the local planner. Table I gives the processing time and the cardinality of the processed subgoals. At the right of Fig. 17, one can see that $\mathcal{A}$ has reached some subgoals located on the slippery area. The reachability of these configurations was dependent on several factors such as the velocity of $\mathcal{A}$ when it started crossing the slippery area and the type of motion it was executing (i.e., a straight line motion or a gyration motion). Other simulation experiments showed that finding admissible controls to steer the robot on uneven slippery regions and along gyration trajectories is a difficult task. This is supported in the presented example by the cardinality of the explored subgoals given in Table I. Most of the configurations reached on the slippery area were performed

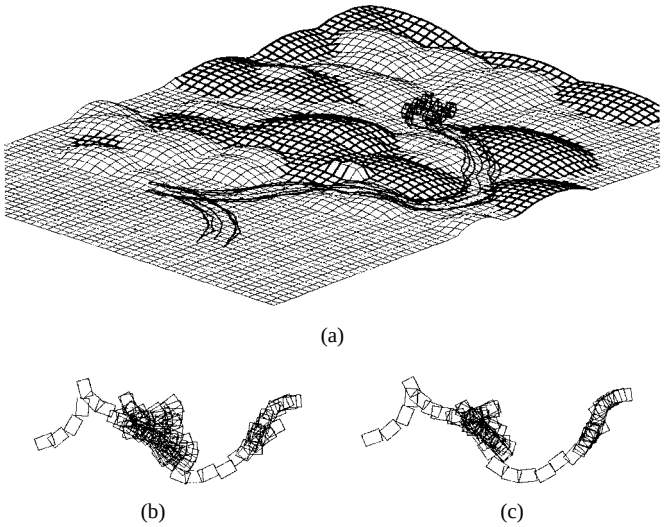

Fig. 17. (a) Trajectory planned in the presence of a larger slippery area (dark regions), (b) subgoals processed by the local planner, and (c) subgoals reached by the local planner.

TABLE I
EXPLORED SUBGOALS AND RUN TIME OF THE PLANNER

Task	Sub-goals processed locally	Sub-goals located on Γ	Non-reached sub-goals	run time
Fig. 14	30	22	7	27 minutes
Fig. 16	72	26	17	83 minutes
Fig. 17	121	23	38	152 minutes
Fig. 18	65	44	11	53 minutes
Fig. 19	76	36	22	73 minutes
Fig. 20	109	44	32	91 minutes

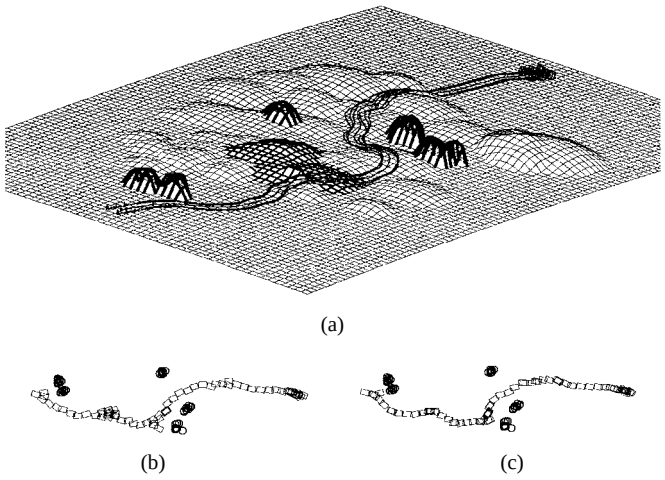

Fig. 18. (a) Motion in the presence of slippery areas and geometric obstacles (dark regions), (b) subgoals processed locally, and (c) reached configurations.

by steering $\mathcal{A}$ along a straight line motion. This observation is also confirmed by the example shown in Fig. 18 where geometric obstacles have been added to the environment. The location of these obstacles relative to the generated subgoals is depicted on the top right of Fig. 18 by the set of disks approximating their shapes, respectively.

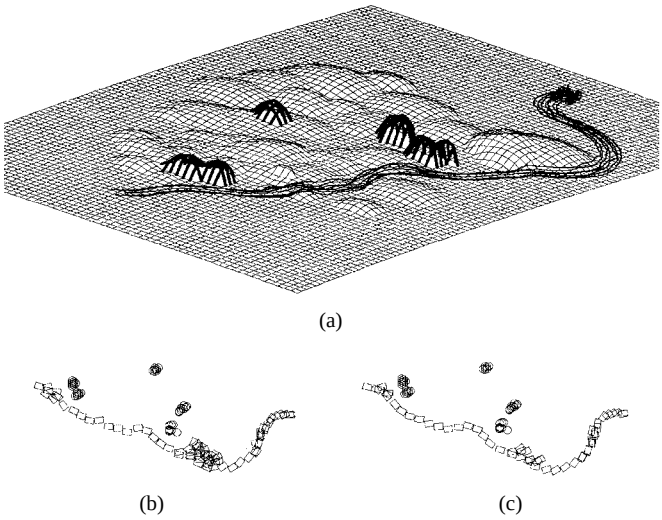

Fig. 19. (a) Planned motion in the presence of geometric obstacles ($d = 1$ m), (b) subgoals processed locally, and (c) reached configurations.

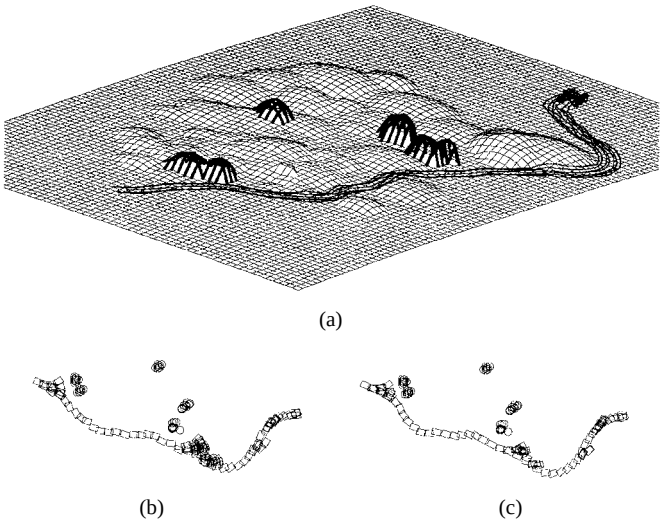

Fig. 20. (a) Planned motion in the presence of geometric obstacles when $d = 0.75$ m, (b) subgoals processed locally, and (c) reached configurations.

In the following, we present further examples solved in the presence of geometric obstacles and/or slippery areas (see Figs. 19 and 20). In the last two examples, both the trajectories were processed considering different discretization at the higher level. The period of time, ΔT_G, is chosen so that the distance between two subgoals and along a straight line motion is equal to 1 m and 0.75 m, respectively. For these examples, the geometry of the two resulting trajectories is nearly the same, but this observation does not hold in general. For a fixed ΔT_G, the distribution of the subgoals made by the higher level depends on the location of the slippery areas and of the obstacles w.r.t. to the starting subconfiguration, q_{start}. For the two examples shown in Figs. 19 and 20, we introduced a penalty term in the heuristic function associated to the presence of the obstacles. The detection of the obstacles is achieved when projecting the nominal 2-D path (a Reeds and Shepp's curve) on the terrain and

computing its geodesic length on the terrain [12]. The introduction of such penalty terms is inspired from Shiller's motion planner which solves for collision avoidance using a penalty method [49]. In our case, this resulted in selecting first the subgoals which are located far from the obstacles. This helps in decreasing the run time of the planner by reducing the number of subgoals to be processed locally in the vicinity of the obstacles. For such subgoals, the local planner may fail often since steering $\mathcal{A}$ close to the boundary of the $\mathcal{B}_i$'s is difficult mainly when the velocity is near the maximum. However, we have to confess that such a gain is not usually observed because the considered penalty weights depend on the distribution of the $\mathcal{B}_i$'s w.r.t. the current and final configurations of $\mathcal{A}$. In some cases, the resulting trajectories may be too far to be optimal in length as shown by Figs. 19 and 20.

VI. Conclusion

We have presented a two-level approach for planning global feasible motions for a wheeled vehicle moving on a 3-D uneven terrain. Basically, it consists in interleaving two complementary reasoning levels:

1) high level that guides the search by expanding a tree of subgoals in a subset of the configuration space of the robot;
2) local level which solves for feasible 3-D trajectories and actuator controls to move the robot between neighboring subgoals.

The novelty of our approach is the introduction, in the motion planning paradigm, of appropriate physical models that are helpful in dealing with dynamic and contact constraints of the task, and in characterizing instantaneous feasible motions of the robot. To the best of our knowledge, the described planner is the first implemented system that solves for global trajectories of an all-terrain vehicle in the presence of kinematic, dynamic and contact interaction constraints. Although the simulation results have been performed for an articulated three-axle vehicle, our approach remains conceptually applicable to a larger variety of mobile robots (e.g., robots with more than three axles, nonarticulated systems such a car-like robot).

In [32], Lacroix *et al.* have presented an adaptive approach for embedding an off-line motion planner (the path planner described in [51] and discussed in Section I-C) within an autonomous navigation system for an all-terrain vehicle. Our planning framework may conceptually be implemented within an architecture adopting the same approach as in [32], but the current performance of the planner must be enhanced and the assumption of complete and perfect knowledge of the task features must be relaxed for making the planned trajectories useful in a real context. Indeed, the bounds on modeling errors are currently required to be very low to obtain output trajectories applicable in practice.

The improvement of the planner efficiency can be achieved by the application, at the high level, of a more elaborate analysis which allows a more practical distribution and pruning of the subgoals. These two points aim to make a better use of the local planner and to possibly minimize the calls to it. For instance, pruning can be improved by checking the slope of the terrain between successive subgoals in order to avoid searching a local trajectory that moves, in all cases, the robot on reliefs which may cause unavoidable tip-overs [50]. A nonuniform generation of the subgoals may also be considered to make it easy to traverse certain important reliefs. For instance, the subgoals can be computed using some particular nominal paths so that the robot is moved first to encounter these reliefs with a uniform contact distribution on it wheels before achieving a stable traversal. Because of the heuristic nature of such analysis, its must be conservative enough so that not much completeness of the planner is lost (i.e., the local planner is applied when it is required).

In Section I-A, we have mentioned that the (exact) identification of certain environment features (e.g., friction and deformation parameters) is difficult and may be achieved incrementally (during execution) by using some qualitative informations on the nature of the regions to be crossed by the robot. We have also pointed out that a solution to this modeling problem consists of incorporating uncertainty (due to modeling, sensing, and control) in planning. Dealing with uncertainty is essential for ensuring the robustness of the planned trajectories. On the other hand, it permits reducing the models accuracy required for planning. This reduction is very important in a real context. Previous works in kinodynamic motion planning [9], [16], [17], [24] have somewhat dealt with the issue of uncertainty by constraining the robot to avoid the obstacles with a speed-dependent safety margin (defined by an affine function whose coefficients are inputs of the problem). In our planning framework, this safety margin can be incorporated as an additional constraint when checking collision with the static obstacles (or more specifically with their disc-based representations). When contact interaction constraints are important, as in our case, planning in the presence of uncertainty is much more complex. Large discrepancies between the models and the real environment (in terms of the geometry and the physical features) may yield to important errors during execution. Because the robot kinematic chain is considered to be compliant, the effects of the errors in the terrain geometry may be partially reduced. Errors in friction and deformation features affect drastically the task dynamics.

Considering more inaccurate models and making our result more tolerant to errors give rise to several modeling and computational issues that merit further investigation in future research:

1) characterization of the required bounds on the model errors which guarantee a correct execution of the planned trajectories;
2) formulation of what the safety margin would be (by analogy to the above-mentioned margin for avoiding static obstacles) with regards to our dynamic representations and how it can be modeled and incorporated in the local planner;
3) how our planning framework can deal with various origins of uncertainty (modeling, sensing, and control)

and error recovering so that the resulting complexity remains reasonable.

ACKNOWLEDGMENT

The author would like to thank C. Laugier, INRIA Rhône-Alpes, for discussions which developed several ideas in this work and for his guidance, T. Fraichard, INRIA, for discussions on kinodynamic motion planning, K. Gupta, Simon Fraser University, for encouragement in writing this paper, C. Richard for helping to draw many pictures, and finally the five anonymous reviewers for their insightful comments and suggestions.

REFERENCES

[1] F. Ben Amar and Ph. Bidaud, "Dynamic analysis of off-road vehicles," in *Preprints 4th Int. Symp. Experimental Robot.*, Stanford, CA, June 1995.

[2] F. B. Amur, Ph. Bidaud, and F. B. Ouezdou, "On modeling and motion planning of planetary vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Yokohama, Japan, 1993.

[3] W. W. Armstrong, "Recursive solution to the equations of motion of an n-link manipulator," in *Proc. 5th World Congr. Theory Mach. Mech.*, July 1979.

[4] J. Bares *et al.*, "Ambler: An autonomous rover for planetary exploration," *IEEE Comput.*, pp. 18–26, June 1989.

[5] J. Barraquand and J.-C. Latombe, "On nonholonomic mobile robots and optimal maneuvering," *Rev. d'Intell. Artif.*, vol. 3, no. 2, pp. 77–103, 1989.

[6] C. Bellier, C. Laugier, and B. Faverjon, "A kinematic simulator for motion planning of a mobile robot on a terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Yokohama, Japan, July 1993.

[7] A. Bicchi, G. Casalino, and C. Santilli, "Planning shortest bounded-curvature paths for a class of nonholonomic vehicles among obstacles," *J. Intell. Robot. Syst.*, vol. 16, pp. 387–405, 1996.

[8] L. Boissier and G. Giralt, "Autonomous planetary rover: The robotic concepts," in *Proc. Int. Adv. Robitics Progr. (IARP)*, Pisa, Italy, June 1991.

[9] J. Canny, B. Donald, J. Reif, and P. Xavier, "On the complexity of kynodynamic planning," in *Proc. IEEE Symp. Foundations Comput. Sci.*, White Plains, NY, Nov. 1988.

[10] R. Chatila *et al.*, "Autonomous navigation in natural environment," in *Preprints 3rd Int. Symp. Experimental Robot.*, Kyoto, Japan, Oct. 1993.

[11] C.-H. Chen and V. Kumar, "Motion planning of walking robots in environments with uncertainty," in *Proc IEEE Int. Conf. Robot. Automat.*, Minneapolis, MN, Apr. 1996, pp. 3277–3282.

[12] M. Cherif, *Motion Planning for an Autonomous All-Terrain Vehicle: An Approach Based Upon Physical Models*, Ph.D. dissertation, Inst. Nat. Polytechnique de Grenoble, Grenoble, France, Oct. 1995, in French.

[13] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*. Munich, Germany: Springer-Verlag, 1993.

[14] B. Dacre-Wright, *Planification de trajectories pour un robot mobile sur terrain accidenté*, Ph.D. dissertation, Lab. d'Automatique et d'Analyze des Systèmes, Toulouse, France, Oct. 1993.

[15] Ch. DeBolt, Ch. O'Donnell, C. Freed, and T. Nguyen, "The bugs 'basic uxo gathering system' project for uxo clearance and mine countermeasures," in *Proc. IEEE Int. Conf. Robot. Automat.*, Albuquerque, NM, May 1997, pp. 329–334.

[16] B. Donald and P. Xavier, "A provably good approximation algorithm for optimal time trajectory planning," in *Proc. IEEE Int. Conf. Robot. Automat.*, Scottsdale, AZ, May 1989, pp. 958–963.

[17] ———, "Provably good approximation algorithms for optimal kinodynamic planning for Cartesian robots and open-chain manipulators," in *Proc. ACM Symp. Computat. Geom.*, Berkeley, CA, 1990, pp. 290–300.

[18] Th. Fraichard and C. Laugier, "Path-velocity decomposition revisited and applied to dynamic trajectory planning," in *Proc. IEEE Int. Conf. Robot. Automat.*, Atlanta, GA, May 1993, vol. 2, pp. 40–45.

[19] Th. Fraichard and R. Mermond, "Path planning with uncertainty for car-like robots," in *Proc. IEEE Int. Conf. Robotics Automat.*, Leuven, Belgium, May 1998, pp. 27–32.

[20] Th. Fraichard and A. Scheuer, "Car-like robots and moving obstacles," in *Proc. IEEE Int. Conf. Robot. Automat.*, San Diego, CA, May 1994, pp. 64–69.

[21] D. Gaw and A. Meystel, "Minimum-time navigation of an unmanned mobile robot in a 2-1/2d world with obstacles," in *Proc. IEEE Int. Conf. Robot. Automat.*, San Francisco, CA, Apr. 1986.

[22] A. Hait and Th. Siméon, "Motion planning on rough terrain for an articulated vehicle in presence of uncertainties," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1996, pp. 1126–1132.

[23] M. Hebert *et al.*, "Terrain mapping for a roving planetary explorer," in *Proc. IEEE Int. Conf. Robot. Automat.*, Scottsdale, AZ, May 1989, pp. 997–1002.

[24] G. Heinzinger, P. Jacobs, J. Canny, and B. Paden, "Time-optimal trajectories for a robot manipulator: A provably good approximation algorithm," in *Proc. IEEE Int. Conf. Robot. Automat.*, Cincinnatti, OH, May 1990, pp. 150–156.

[25] J. M. Hollerbach, "A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-10, pp. 730–736, Nov. 1980.

[26] ———, "Kinematics and dynamics for control," in *Robotics Science*, M. Brady, Ed. Cambridge, MA: MIT Press, 1989, pp. 378–431.

[27] S. Jimenez, A. Luciani, and C. Laugier, "Predicting the dynamic behavior of a planetary vehicle using physical models," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Yokohama, Japan, July 1993.

[28] K. Kant and S. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *Int. J. Robot. Res.*, vol. 5, no. 3, pp. 72–89, 1986.

[29] A. Kemurdjian *et al.*, "Small marsokhod configuration," in *Proc. IEEE Int. Conf. Robot. Automat.*, Nice, France, May 1992, pp. 165–168.

[30] T. Kubota, I. Nakatani, and T. Yoshimitsu, "Path planning for planetary rover based on traversability probability," in *Proc. IEEE Int. Conf. Adv. Robot.*, San Feliu de Guixols, Catalunia, Spain, Sept. 1995, pp. 739–744.

[31] I. S. Kweon and T. Kanade, "High resolution terrain map from multiple sensor data," in *Proc. IEEE/RSJ Int. Workshop Intell. Robots Syst.*, Tsushiura, Japan, July 1990, pp. 127–134.

[32] S. Lacroix and R. Chatila *et al.*, "Autonomous navigation in outdoor environment: Adaptive approach and experiment," in *Proc. IEEE Int. Conf. Robot. Automat.*, San Diego, CA, May 1994, pp. 426–432.

[33] J-C. Latombe, *Robot Motion Planning*. Norwell, MA: Kluwer, 1991.

[34] J-P. Laumond, P. E. Jacobs, M. Taix, and R. M. Murray, "A motion planner for nonholonomic mobile robots," *IEEE Trans. Robot. Automat.*, vol. 10, pp. 577–593, Oct. 1994.

[35] J-P. Laumond, M. Taix, and P. Jacobs, "A motion planner for carlike robots based on a mixed global/local approach," in *Proc. IEEE/RSJ Int. Workshop Intell. Robots Syst.*, Tsuchiura, Japan, July 1990, pp. 765–773.

[36] A. Liégeois and Ch. Moignard, "Optimal motion planning of a mobile robot on a triangulated terrain model," in *Geometric Reasoning for Perception and Action*, Ch. Laugier, Ed. Munich, Germany: Springer-Verlag, 1993, vol. 708 of *Lecture Notes in Computer Science*.

[37] A. Luciani *et al.*, "An unified view of multiple behavior, flexibility, plasticity and fractures: Balls, bubbles and agglomerates," in *IFIP WG 5.10 Modeling Comput. Graph.*, 1991.

[38] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "On-line computational scheme for mechanical manipulators," *ASME J. Dyn. Syst., Meas., Contr.*, vol. 102, pp. 69–76, June 1980.

[39] F. O. Manaoui, "Etude et simulation d'algorithmes de navigation pour robots mobiles autonomes sur terrain inégal," Ph.D. dissertation, Univ. Montpellier II, Montpellier, France, Dec. 1988.

[40] B. Mirtich and J. Canny, "Using skeletons for nonholonomic path planning among obstacles," in *Proc. IEEE Int. Conf. Robot. Automat.*, Nice, France, May 1992.

[41] J. O'Rourke and N. Badler, "Decomposition of three-dimensional objects into spheres," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, pp. 295–305, July 1979.

[42] E. Rabinowicz, "The nature of the static and kinetic coefficients of friction," *J. Appl. Phys.*, vol. 22, no. 11, pp. 1373–1379, 1951.

[43] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forward and backward," *Pacific J. Math.*, vol. 145, no. 2, pp. 367–393, 1990.

[44] G. Sahar and J. H. Hollerbach, "Planning of minimum-time trajectories for robot arms," in *Proc. IEEE Int. Conf. Robot. Automat.*, St. Louis, MO, Mar. 1985, pp. 751–758.

[45] L. Shih and A. A. Frank, "A study of gait and flywheel effect on legged machines using a dynamic compliant joint model," in *Proc. IEEE Int. Conf. Robot. Automat.*, Raleigh, NC, Mar. 1987, pp. 527–532.

[46] L. Shih, A. A. Frank, and B. Ravani, "Dynamic simulation of legged machines using a compliant joint model," *Int. J. Robot. Res.*, vol. 6, no. 4, pp. 33–46, 1987.

[47] Z. Shiller and J. C. Chen, "Optimal motion planning of autonomous vehicles in three-dimensional terrains," in *Proc. IEEE Int. Conf. Robot.*

Automat., Cincinnati, OH, May 1990, pp. 198–203.
[48] Z. Shiller and S. Kubowsky, "Global time optimal motions of robotic manipulators in the presence of obstacles," in *Proc. IEEE Int. Conf. Robot. Automat.*, Philadelphia, PA, Apr. 1988, pp. 370–375.
[49] Z. Shiller and Y. R. Gwo, "Dynamic motion planning of autonomous vehicles," *IEEE Trans. Robot. Automat.*, vol. 7, Apr. 1991.
[50] Th. Siméon, "Motion planning for a nonholonomic mobile robot on 3-dimensional terrains," in *Proc. IEEE/RSJ Int. Workshop Intell. Robots Syst.*, Osaka, Japan, Nov. 1991, pp. 1455–1460.
[51] T. Siméon and B. Dacre-Wright, "A practical motion planner for all-terrain mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Yokohama, Japan, July 1993.
[52] B. Wilcox *et al.*, "Robotic vehicles for planetary exploration," in *Proc. IEEE Int. Conf. Robot. Automat.*, Nice, France, May 1992, pp. 175–180.
[53] NASA www site: http://mpfwww.jpl.nasa.gov/ops/rover.html.
[54] NASA www site: http://www.nasa.gov//hqpao/pathfinder.html.
[55] R. Y. Yong and B. P. Warkentin, *Introduciton to Soil Behavior*. New York: Macmillan, 1966.
[56] O. C. Zienkiewicz, *La méthode des éléments finis*. Paris, France: McGraw-Hill, 1979.

Moëz Cherif (M'98) was born in Tunis, Tunisia, on June 14, 1966. He received the Dip. Ing. degree from the Faculty of Sciences, University of Tunis, Tunisia, in 1990, and the M.S. and Ph.D. degrees from the Institut National Polytechnique de Grenoble, France, in 1991 and 1995, respectively, all in computer science.

From October 1995 to October 1996, he was a Post-Doctoral Fellow at the School of Engineering Science, Simon Fraser University, Burnaby, B.C., Canada (fellowship funded by NSERC). From November 1996 to December 1997, he had a research appointment at INRIA, Rhône-Alpes, France (with funding from CNRS). Since February 1998, he has been a Research Associate at the School of Engineering Science, Simon Fraser University. His research interests are in the areas of spatial reasoning and robot motion planning (particularly when the contact is an issue), mobile robots, automated dextrous manipulation by artificial hands, physical modeling and simulation for robotics, and computer animation.