Proceedings of the 1999 IEEE
International Conference on Robotics & Automation
Detroit, Michigan • May 1999

Kinodynamic Motion Planning for All-Terrain Wheeled Vehicles *

Moëz Cherif
Simon Fraser University
School of Engineering Science, Burnaby, V5A 1S6, B.C., Canada
E-mail: cherif@cs.sfu.ca

Abstract

We present a two-level algorithm that incorporates appropriate physical models of the robot, the terrain and their interaction to cope with kinodynamic aspects in all-terrain vehicle motion planning. The high planning level expands a tree of sub-goals in a low dimensional subset of the robot C-space considering a simplified 2D instance of the locomotion task. The second planning level considers locally the full set of task constraints and makes use of a state space formulation to find feasible trajectories and actuator controls moving the robot between adjacent sub-goals. We demonstrate our approach in the case of a six-wheeled articulated vehicle.

1 Introduction

The application of mobile robots to challenging new tasks such as planetary exploration and navigation in off-road sites is gaining a great interest. In such a context, 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 problems of autonomy and motion planning for all-terrain vehicles much more difficult than for the case of indoor robots moving on flat surfaces. In this paper, we address some of these issues in the framework of kinodynamic motion planning. We are interested in finding global trajectories (including the actuator controls) that move a (wheeled) vehicle between arbitrary states on a terrain composed of various regions such as obstacles and traversable areas that can be rigid or deformable/movable (e.g., sandy or muddy areas, stones, etc.). In addition to geometric/kinematic constraints (e.g., no-collision with the obstacles, no tip-over of the robot, maintaining contact and minimizing slippage at the wheels), we deal with additional constraints arising from the robot dynamics and its physical interactions with the terrain. Such constraints are fundamental for characterizing feasible motions in the case of an all-terrain vehicle and must be coped with simultaneously with geometric/kinematic constraints. Our approach aims to achieve this goal (§4). Basically, it consists in interleaving two complementary reasoning levels for tackling the intrinsic complexity of all-terrain kinodynamic motion planning. The high level –a *discrete grid search*– expands a tree of sub-goals within a low dimensional subset of the robot C-space (i.e., the space of the robot position/orientation in the plane) when considering only a simplified 2D instance of the task. The full set of constraints is coped with by the local level which operates *continuously* in the robot state space for finding feasible motions of the robot between adjacent sub-goals. A key feature of our approach is the introduction, in the motion planning paradigm, of specific and appropriate *physical models* described in §2 which help in coping with the task dynamics. The simulation results presented in §5 for the locomotion of a six-wheeled Marsokhod-like vehicle demonstrate the promise of our modeling and planning approach.

All-terrain vehicle motion planning has received a little attention in the past. Main works incorporates simplified kinematic or dynamic models of the robot and/or of the terrain for analyzing/synthesizing the robot motions [1, 2, 3, 4]. For instance, Shiller and Gwo accounts for kinematic and dynamic constraints for planning time-optimal trajectories of a point robot moving on a smooth terrain [1]. Because the smoothness of the terrain, the planner applies only for uniform contact distribution at the robot wheels and only for terrain relief much larger than the robot size. Siméon described a nonholonomic path planner for a polyhedron moving on a polygonal terrain considering that the contact occurs at 3 point wheels [2]. The planner searches a grid defined on the space of 2D position/orientation of the robot using extreme wheel velocities to steer the robot. This search has inspired our high planning level. [3] presents further extensions dealing with a car-like robot having an arbitrary number of axles. In [2, 3], no dynamics and no friction effects are considered. The planner is efficient, but the assumption of pure rolling at point wheels makes it not applicable to more realistic tasks. Other works have considered simplified *itinerary* (gross motion) planning [5, 6] or only modeling/simulation aspects [7, 8, 9].

For a better focus on kinodynamic motion planning with contact constraints, we have made the assumption that a complete description of the environment (including its geometry and physical features) is available when planning. Issues related to sensing, coping with incomplete knowledge of the environment, and uncertainty are beyond the scope of this paper but merit further investigation in relationship with our planning framework to make it useful and applicable in real contexts. We briefly discuss some of these issues in §6. Works described in [10, 11, 12] are relevant, although they have focussed on the static behavior of the robot and not on the effect of uncertainty on the kinematics and dynamics of the robot.

2 Modeling

2.1 The workspace

The workspace, $\mathcal{W}$, is composed of a three dimensional terrain, $\mathcal{T}$, and a set, $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{n_\mathcal{B}}\}$, 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, $\mathcal{ST} = \{S_1, \ldots, S_{n_{\mathcal{ST}}}\}$, of spheres approximating the relief of $\mathcal{T}$ and the $\mathcal{B}_i$'s (see Figure 1). Such a collection, $\mathcal{ST}$, can be obtained by computing a set of tangent spheres that approximate accurately an elevation map of $\mathcal{W}$. In addition, a 2D 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

* This work has been partly done when the author was with the Sharp Project of INRIA RHône-Alpes, France, and has been partly supported by the French Programme RISP-VAP (of CNES and CNRS) and by the Région Rhône-Alpes. Many thanks to Ch. Laugier of INRIA Rhône-Alpes for discussion and insightful comments, and to K. Gupta of SFU for support.

by $\mathcal{DB}_i$, covering the projection of $\mathcal{B}_i$ in the plane (x, y) of a frame, $\mathcal{F}_\mathcal{W}$, fixed in $\mathcal{W}$.

Discrete physical model of $\mathcal{T}$ The motions deformable region $\mathcal{T}_i$ are synthesized by solving the dynamics of a discrete model [9, 13]. It consists of transforming the spatial model of $\mathcal{T}_i$ (spheres $S_{i,j}$'s describing it) into a network where the nodes are particles located at the centers of the $S_{i,j}$'s and obeying Newton's dynamics and the arcs are connectors corresponding to a particular physical behavior (see Fig. 1). For illustrating our approach, we have considered only visco-elastic behavior in response to the applied forces but plastic, elasto-plastic, or visco-plastic phenomena can conceptually be incorporated. The connectivity structure and the stiffness/damping parameters are chosen arbitrarily. The applied force on each particle (or surrounding sphere) are the resultant of the penalty forces provided by the internal spring-damper connectors and the external forces (gravity, forces contact with the robot wheels and/or other movable components of $\mathcal{T}$).

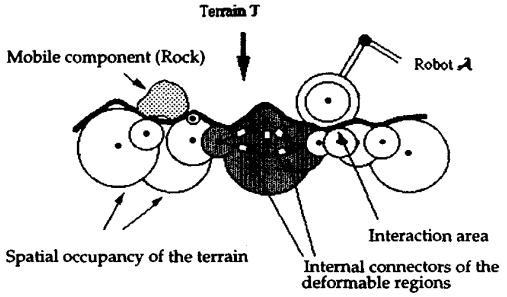

Figure 1: Geometric and physical models of $\mathcal{T}$.

2.2 The vehicle

We illustrate our approach considering a robot model derived from the Marsokhod rover [14]. Other structures may be considered. 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 compliant passive rotoid joint mechanisms. This joints allow $\mathcal{A}$ *w.r.t.* the terrain relief through roll and pitch motions of the axles (see Fig. 2). The front axle of $\mathcal{A}$ can additionally perform yaw motions within a limited range. $\mathcal{A}$ is steered by applying a series of torque vectors, $U \in \mathbb{R}^{2n_a}$, on its $2n_a$ wheels. A full configuration, Q of $\mathcal{A}$, is defined by the 3D position/orientation, $(x, y, z, \theta, \varphi, \psi)$, of a frame, $\mathcal{F}_\mathcal{A}$, fixed on the main body of $\mathcal{A}$ *w.r.t.* $\mathcal{F}_\mathcal{W}$, and the set of its joints. θ, φ and ψ are the yaw, roll, and pitch angles of $\mathcal{F}_\mathcal{A}$, respectively. 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 sub-configuration of $\mathcal{A}$, and we denote the space of sub-configurations q by $\mathcal{SCS}_\mathcal{A}$.

Hybrid physical model of $\mathcal{A}$ The formulation of the dynamics of rigid-body articulated chains involves complex non-linear and highly coupled second-order differential equations because the system kinematics is imposed explicitly [15]. In such a formulation, the joints are generally idealized and considered to be non-compliant. For reducing the set of coupled dynamic equations of the compliant chain of $\mathcal{A}$, we introduce a *hybrid* model that combines the mechanics of rigid-body chains and the physics of particle-based compliant systems. The rigid-body dynamic formulation is used 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). The discrete physical structures (particles + spring-dampers) is used to model the compliant passive joints and the links of the chassis. The model of $\mathcal{A}$ is then seen as a collection of n_a articulated sub-chains (of the wheeled axles) interconnected by a network of discrete mechanisms acting to maintain the cohesion and the kinematic structure of the chassis (see Figure 2). The hybrid model yields a simple solution for the forward motion problem since it permits decreasing the set of coupled dynamic equations of the entire robot system. Indeed, the dynamics of each basic sub-system is coupled only to its neighboring parts. In addition, our model makes it easy to cope with closed loops due to the contact between $\mathcal{A}$ and $\mathcal{T}$ since each contact can be modeled as a compliant joint subject to friction and no-penetration constraints [16]. This has also 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 [17].

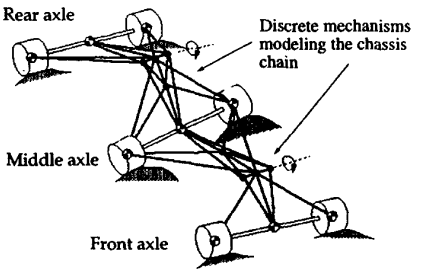

Figure 2: The hybrid model of $\mathcal{A}$.

3 The 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 sub-configuration, q_{goal} in $\mathcal{SCS}_\mathcal{A}$, find a feasible (piecewise-) smooth trajectory (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 $(Q_{start}, 0) = \Gamma(0)$ to a state $(Q, 0) = \Gamma(t_f)$ in a finite period of time, t_f, such that $q(t_f) = q_{goal}$. This means that the goal to achieve is a subset of $\mathcal{CS}_\mathcal{A}$ so that $q = q_{goal}$.*

We search for a solution Γ whose projection on the plane (x, y) of $\mathcal{W}$ is sub-minimal. Γ is feasible if it satisfies the following constraints.

a. No-collision The first constraint is no-collision between $\mathcal{A}$ and the $\mathcal{B}_i$'s, and collision detection is based on the disc-based approximation of the obstacles (cf. §2.1). The second constraint is no-collision between $\mathcal{T}$ and parts of $\mathcal{A}$ other than the wheels. The collection $\mathcal{ST}$ is used for the 3D distance computation with parts of $\mathcal{A}$.

b. Maintaining the contact: 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. Because each wheel is independently actuated, we simplify the analysis considering 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 $\mathcal{T}$. Such choice contributes somewhat to avoiding tip-over of $\mathcal{A}$ since breaking all the contacts at one side of $\mathcal{A}$ corresponds to the beginning of a possible tip-over. In addition, we arbitrarily bound the roll and the pitch angles, i.e., $|\varphi| \leq \varphi_{max}$ and $|\psi| \leq \varphi_{max}$, where $\varphi_{max} \leq \pi/2$ is an arbitrary positive integer.

c. 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.

d. Velocity constraints: 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. Considering an horizontal flat surface, this relationship reduces to 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 $n_a = 3$, the minimum lateral sliding velocities at the middle and rear axles satisfy that their summation is null. Let M be the middle point of the center of the middle and back robot axles. The location of the center of instantaneous rotation, $G_{\mathcal{A}}$, corresponds to the intersection of the axis of the front axle and a line perpendicular to the robot axis at M. The nonholonomic constraint is: $\dot{x}\sin\theta - \dot{y}\cos\theta + L_r\dot{\theta}/2 = 0$, where L_r is the distance between the centers of the middle and the rear axles. In addition, we consider, at each instant, that $|v| = |\dot{\mathbf{r}}| \leq V_{max}$ 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, ϕ_{max}. For $n_a = 3$, we have $\dot{\theta}_{max} = V_{max}\frac{\sin\phi'_{max}}{L_r/2}$, where ϕ'_{max} is the maximum angle between the velocity vector of $\mathcal{A}$ and its longitudinal axis. ϕ'_{max} is related to the maximum steering angle, ϕ_{max}, by $\tan\phi'_{max} = L_r\tan\phi_{max}/(2L - L_r)$.

e. Control torque and acceleration constraints: The wheels are driven by a bounded control vector U. We have, at each instant, $|u_i| \leq u_{max}$, $i = 1\ldots 2n_a$, where u_i is the torque control of the i-th 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 §4.2.

4 The planning algorithm

4.1 Expanding a tree of sub-goals in $\mathcal{SCS}_{\mathcal{A}}$

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 sub-configuration, 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, N^+, which have been already processed by the local planner and the arcs pointing to them are feasible trajectory, Γ_i or empty curve (i.e., q is not locally reachable), and (2) nodes, N^-, which have not been locally processed and the arcs pointing to them are nominal paths in the plane. The heuristic distance to the goal we have used accounts for the kinematic constraints of $\mathcal{A}$ by using Reeds and Shepp's shortest paths [18]. At each iteration, the expansion of a reached node, N^+, aims in generating new sub-goals, q_i, (nodes N^-), located in unvisited neighborhoods, V. The q_i's are computed assuming that $\mathcal{A}$ moves on a planar surface for a coarse period of time ΔT with an extreme kinematic control (v,ϕ). The q_i's correspond to 3 nonholonomic forward motions (left and right turns of minimum curvature radius, and a straight motion) and 3 similar backward motions. For enabling a good convergence of local planning and depending on whether q_i is a reversal point or not, the desired velocity at q_i is specified by either a nominal vector $(\tilde{v}_i,\tilde{\omega}_i)$ or hypothetical intervals where the translation/yaw rotation velocity (v,ω) must lie. The search process is depicted in Figure 3. When $V(q_{goal})$ is reached, the trajectory Γ goes smoothly from $(Q_{start},0)$ to a state, $E(t_f)$ (such that $q \in V(q_{goal})$ and $(v,\omega) = (0,0)$), through the trajectory segments, Γ_i, planned locally between the sub-goals lying on the near-shortest path in $\mathcal{G}$.

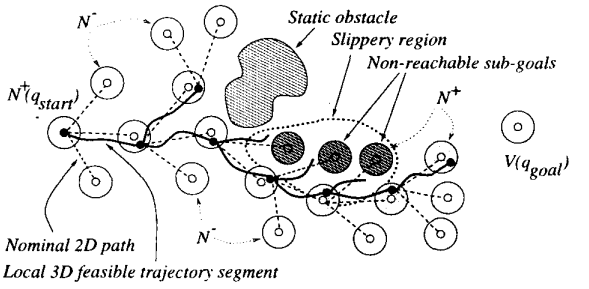

Figure 3: The searched graph $\mathcal{G}$.

4.2 Local planning between the sub-goals

The local problem is to find a feasible trajectory, Γ_i, that moves $\mathcal{A}$ from its current state, $E_{i,p}$ to a state E where $q \in V(q_i)$ and (v,ω) is close to $(\tilde{v}_i,\tilde{\omega}_i)$. We constrain Γ_i to have no backup maneuver when $v_{i,p}$ and $\tilde{v}_i$ are of the same sign; otherwise, only a single backup motion is allowed.

As in [19, 20, 21, 22], we formulate locally kinodynamic motion planning as a graph search problem. The basic idea consists in applying a *best-first search* over a cell-based representation, CL, of the sub-state space, $\mathcal{SSS}_{\mathcal{A}}$ –a 5 dimensional subset of the full state space $\mathcal{SS}_{\mathcal{A}}$ parameterized by (x,y,θ,v,ω). CL is composed of a collection of 5 dimensional cells corresponding to a regular discretization of the local domains of (x,y,θ,v,ω) (see [16]). Starting from the sub-state corresponding to $E_{i,p}$, the local level expands a tree where the nodes are cells of CL in which a sub-state of $\mathcal{A}$ has been reached by the search and the arcs are segments of feasible motions of $\mathcal{A}$. At each iteration of the search, the cell containing the closest sub-state to the sub-goal is selected and expanded as described in §4.2.1. The closeness to the sub-goal is measured by the length of a simple nonholonomic 2D path of type CSC ending at the sub-goal, q_i, (where C is a circular arc of minimum admissible radius and S is a straight line). $\mathcal{A}$ is considered 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. We limit each cell to be visited only once. The search is iterated until the cell corresponding to the sub-goal and its velocity is reached (the local planner returns Γ_i, the sequence of controls U, and the time t_{Γ_i} to the high level) or the set of cells has been fully explored without reaching the sub-goal (the local planner reports a failure).

4.2.1 Moving between adjacent cells

Let s_k^r be a sub-state 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 sub-state corresponding to the center of CL_k. The expansion of CL_k (or s_k^r) follows the routine, `CellExpansion` (see Figure 4).

---

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

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

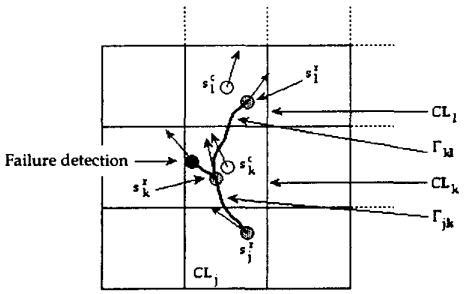

Figure 4: Expansion of a cell CL_k.

torque vector, $U_k(t)$, to apply on the wheels;
6 /* Solve the forward motion of $\mathcal{A}$ and $\mathcal{T}$ [16] */
6.1 Compute the contact forces at each object (wheels and terrain component);
6.2 Compute the new state $E(t+\delta t)$ using $U_k(t)$;
6.3 Compute the new state of the deformable/ movable regions of $\mathcal{T}$;
6.4 Check the task constraints;
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 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 §4.2.2 and §4.2.3. 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}. This upper-bound is considered in order to guarantee that a trajectory segment between the cells is found in a finite period of time. We consider that n_{fs} is $\lceil \frac{5\delta_x}{V_{max}\delta t} \rceil$ (i.e., about 5 times the time required to move along a cell edge in the plane, δ_x, with a maximum translation velocity).
3. The reached sub-state 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 towards the sub-goal, q_i (i.e., $length(\mathcal{P}_i(t)) \leq length(\mathcal{P}_i(t-\delta t))$) 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 sub-state, s_l^r, at which the motion generation is stopped corresponds to the first sub-state at which the type of the path, $\mathcal{P}_i$, changes, or to the closest sub-state to s_l^c (center of CL_l) if the type of $\mathcal{P}$ remains the same.

We should point out that although $(\tilde{v}_k, \tilde{\omega}_k)$ is maintained unchanged during the motion between two adjacent cells (loop defined by step 4 to 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 [23, 22]. For instance, this key property has been used in [22] 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).

4.2.2 Nominal accelerations (steps 1 and 2)

We assume 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 $\mathcal{F}_{\mathcal{A}}$. Let a_x and a_y be the translation accelerations of the middle axle, $\mathcal{E}_c$, 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 considering the dynamic equations ($w.r.t.$ $\mathcal{F}_{\mathcal{A}}$)

$$m_{\mathcal{E}} a_x = g_x + F_{c,x} + f_x^r + f_x^l \quad (1)$$

$$m_{\mathcal{E}} a_y = g_y + F_{c,y} + f_y^r + f_y^l, \quad (2)$$

where $m_{\mathcal{E}}$ is the mass of $\mathcal{E}_c$, g is the gravitational force, and F_c is the resulting force applied by others parts of $\mathcal{A}$ on $\mathcal{E}_c$ (cf. 2.2). 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 the z component of the dynamic rotation equation ($w.r.t.$ $\mathcal{F}_{\mathcal{A}}$)

$$I_{\mathcal{E},z}\dot{\omega} = (f_x^r - f_x^l)l + M(F_c, \omega_{\mathcal{E},x}, \omega_{\mathcal{E},y}), \quad (3)$$

where $I_{\mathcal{E},z}$ is the moment of inertia of $\mathcal{E}_c$ 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 (3), 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,

$$\begin{cases} a_x \geq \frac{1}{m_{\mathcal{E}}}[-2\frac{u_{max}}{R} + g_x + F_{c,x}] \\ a_x \leq \frac{1}{m_{\mathcal{E}}}[2\frac{u_{max}}{R} + g_x + F_{c,x}] \end{cases} \quad (4)$$

$$\begin{cases} a_x \geq \frac{1}{m_{\mathcal{E}}}[-\mu_t^l N^l - \mu_t^r N^r + g_x + F_{c,x}] \\ a_x \leq \frac{1}{m_{\mathcal{E}}}[\mu_t^l N^l + \mu_t^r N^r + g_x + F_{c,x}], \end{cases} \quad (5)$$

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 wheel is chosen to be the highest one among those involved in the computation of the contact forces at its contacts. The constraints of no (lateral) sliding and of bounded motion curvature yield respectively,

$$\begin{cases} a_y \geq \frac{1}{m_{\mathcal{E}}}[-\mu_s^l N^l - \mu_s^r N^r + g_y + F_{c,y}] \\ a_y \leq \frac{1}{m_{\mathcal{E}}}[\mu_s^l N^l + \mu_s^r N^r + g_y + F_{c,y}], \end{cases} \quad (6)$$

$$-v^2/\rho_{min} \leq a_y \leq v^2/\rho_{min}, \quad (7)$$

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. §3). As for equation (4), bounding the wheel controls yields

$$\begin{cases} \dot{\omega} & \geq & \frac{1}{I_{\mathcal{E},z}}[-2\frac{u_{max}}{R}l + M(F_c, \omega_{\mathcal{E},x}, \omega_{\mathcal{E},y})] \\ \dot{\omega} & \leq & \frac{1}{I_{\mathcal{E},z}}[2\frac{u_{max}}{R}l + M(F_c, \omega_{\mathcal{E},x}, \omega_{\mathcal{E},y})]. \end{cases} \quad (8)$$

The dynamic constraints given by inequalities (4) to (8) define the range of admissible accelerations that $\mathcal{A}$ can have at a given state, E.

Let $s = (q, v, \omega)$ be the sub-state to be expanded. For practicality, we consider only a subset of nominal accelerations chosen among the extreme bounds defined by (4) to (8). Such a reduction is achieved by analyzing the type of shortest path, $\mathcal{P}$ of type CSC, connecting q to the sub-goal, q_i, and the direction of instantaneous lateral sliding velocity observed at the current state, $E(s)$ (see [16] for more details).

4.2.3 Nominal wheels control (step 5)

Let $(\tilde{\dot{v}}, \tilde{\dot{\omega}})$ be the nominal acceleration of the center of a given axle, and let θ_l (resp. θ_r) be the angle of rotation of its left wheel (resp. right wheel) *w.r.t.* its axis (i.e., Y axis of $\mathcal{E}$). Assuming that the wheels are purely rolling and that the center of $\mathcal{A}$ 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 $\tilde{\dot{\omega}} = (\tilde{\ddot{\theta}}_r - \tilde{\ddot{\theta}}_l)R/d$, where d is the distance between the centers of the wheels. Having determined $\tilde{\ddot{\theta}}_l$ and $\tilde{\ddot{\theta}}_r$, the control parameters, u_l and u_r of U, are computed by solving 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 a wheel by T. When $|u_j| > u_{max}$, we set u_j to $sgn(u_j).\max(|u_j|, u_{max})$. For wheels at which the contact has been broken, we set $u_j = 0$.

5 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. In the following, the snapshot on the top of each figure shows the robot trajectory found by the planner (shown by the traces of the wheels) and the bottom snapshots show the set of sub-goals processed by the local planner (left) and the corresponding reached configurations (right). (For simplifying the illustration, $\mathcal{A}$ is depicted by a rectangle).

In the example shown in Fig. 5, 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 *hour* 23 *minutes*. 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. In addition, a large part of the computational burden is due to the local search for feasible movements on the slippery area. Depending on the velocity of $\mathcal{A}$ and the slope of the terrain, the local planner failed several times in finding feasible motions steering the robot toward its sub-goals.

In Fig. 6, 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 1 gives the processing time and the cardinality of the processed sub-goals. At the right of Fig. 6, one can see that $\mathcal{A}$ has reached some sub-goals 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 sub-goals given in Table 1. Most of the configurations reached on the slippery area were performed by steering $\mathcal{A}$ along a straight line motion. This observation is also confirmed by the example shown in Fig. 7 where geometric obstacles have been added to the environment. The location of these obstacles relative to the sub-goals is shown in the bottom of Fig. 7 by the corresponding discs.

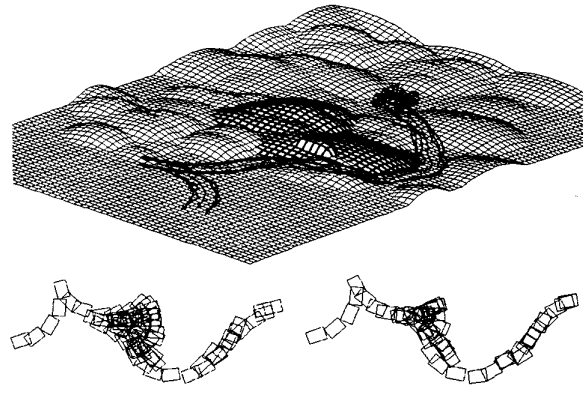

Figure 5: Avoiding slippery areas (dark regions).

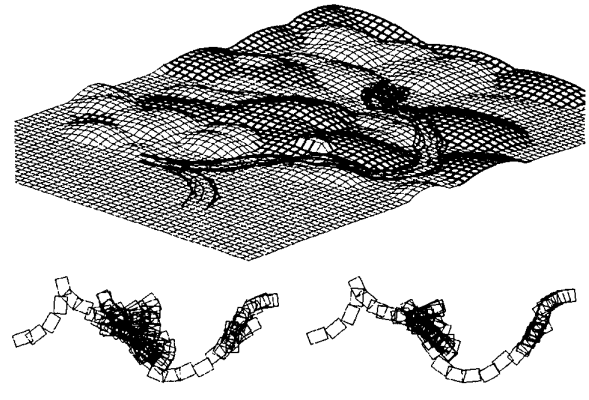

Figure 6: Avoiding larger slippery areas (dark regions).

Task	Sub-goals processed locally	Sub-goals located on Γ	Non-reached sub-goals	run time
Fig. 5	72	26	17	83 min
Fig. 6	121	23	38	152 min
Fig. 7	65	44	11	53 min

Table 1: Explored sub-goals and run time of the planner.

6 Concluding remarks

Depending on the problem size, the run time obtained with our current implementation is of the order of one to a few hours. Such performance must be expected if complex robot structures and non simplified dynamic and contact interaction constraints are considered. It is likely

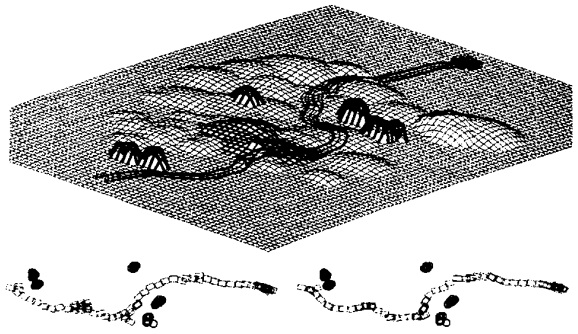

Figure 7: Avoiding slippery areas and static obstacles (dark regions).

to be common to most kinodynamic motion planners. For instance, Shiller and Gwo reported the same order of run time in [1] for a point robot moving on a smooth surface. In [16], we report on possible practical enhancements and on the complexity and completeness of the planner.

In real applications such as planetary exploration or field inspection, the robot is often constrained to move in an area of particular interest around a static plate-form (lander) or some landmarks of the environment. By accounting for possible visibility constraints and avoiding areas of sharp relief, one can a priorly build a roadmap where the nodes are particular regions of simple relief (e.g., regions where complex steering including backup maneuvers can possibly be safely performed) and the arcs are passages in the workspace. The roadmap can be used as an input of the two-level planner to generate global trajectories between couples of states corresponding to robot positioning on adjacent nodes of the roadmap. The goal is to build a connectivity network that can be used later for rapid planning of repetitive point-to-point locomotion tasks. The graphs generated by the high level can be made more connected and the sub-goals sampling can be made at random within each passage as in [24, 25, 26].

Dealing with uncertainty is essential for ensuring the robustness of the planned trajectories and enabling reducing the models accuracy required for planning. When contact interaction constraints are important, as in our case, planning in the presence of uncertainty is expected to be much more complex than collision-free kinodynamic motion planning [19, 20, 22]. 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) the characterization of the required bounds on the model errors which guarantee a correct execution of the planned trajectories, (2) the formulation of what a safety margin would be (by analogy to the safety margin used in [19, 20, 22] for avoiding static obstacles) with regards to our dynamic representations and how it can be modeled and incorporated in the local planner, and (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.

References

[1] Z. Shiller and Y.R. Gwo, "Dynamic motion planning of autonomous vehicles," *IEEE Trans. Robotics and Automation*, vol. RA-7, no. 2, Apr. 1991.

[2] Th. Siméon, "Motion planning for a non-holonomic mobile robot on 3-dimensional terrains," *IEEE/RSJ IROS*, Osaka (J), Nov. 1991, pp. 1455–1460.

[3] Th. Siméon and B. Dacre-Wright, "A practical motion planner for all-terrain mobile robots," *IEEE/RSJ IROS*, Yokohama (J), July 1993.

[4] F. Ben Amar, Ph. Bidaud, and F. Ben Ouezdou, "On modeling and motion planning of planetary vehicles," *IEEE/RSJ IROS*, Yokohama (J), 1993.

[5] D. Gaw and A. Meystel, "Minimum-time navigation of an unmanned mobile robot in a 2-1/2d world with obstacles," *IEEE ICRA*, San Francisco (USA), Apr. 1986.

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

[7] F. Ben Amar and Ph. Bidaud, "Dynamic analysis of off-road vehicles," *Fourth ISER*, Stanford, CA (USA), June 1995.

[8] C. Bellier, C. Laugier, and B. Faverjon, "A kinematic simulator for motion planning of a mobile robot on a terrain," *IEEE/RSJ IROS*, Yokohama (J), July 1993.

[9] S. Jimenez, A. Luciani, and C. Laugier, "Predicting the dynamic behaviour of a planetary vehicle using physical models," *IEEE/RSJ IROS*, Yokohama (J), July 1993.

[10] C-H. Chen and V. Kumar, "Motion planning of walking robots in environments with uncertainty," *IEEE ICRA*, Minneapolis, MN (USA), Apr. 1996, pp. 3277–3282.

[11] A. Hait and Th. Siméon, "Motion planning on rough terrain for an articulated vehicle in presence of uncertainties," *IEEE/RSJ IROS*, 1996, pp. 1126–1132.

[12] T. Kubota, I. Nakatani, and T. Yoshimitsu, "Path planning for planetary rover based on traversability probability," *IEEE ICAR*, Sant Feliu de Guixols, Catalunia (S), Sep. 1995, pp. 739–744.

[13] R. Y. Yong and B. P. Warkentin, *Introduction to soil behavior*, Series in Civil Engineering. The Macmillan Company, 1966.

[14] A. Kemurdjian *et al*, "Small marsokhod configuration," *IEEE ICRA*, Nice (F), May 1992, pp. 165–168.

[15] J.M. Hollerbach, "Kinematics and dynamics for control," *Robotics Science*, M. Brady, Ed., pp. 378–431. MIT Press, 1989.

[16] M. Cherif, "Motion planning for all-terrain vehicles: A physical modeling approach for coping with dynamic and contact interaction constraints," *IEEE Trans. Robotics and Automation*, To appear, 1999.

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

[18] J.A. Reeds and L.A. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Jour. of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.

[19] J. Canny, B. Donald, J. Reif, and P. Xavier, "On the complexity of kynodynamic planning," *IEEE Symp. on the Foundations of Computer Science*, White Plains, NY (USA), Nov. 1988, pp. 306–316.

[20] B. Donald and P. Xavier, "A provably good approximation algorithm for optimal time trajectory planning," *IEEE ICRA*, Scottsdale, AZ (USA), May 1989, pp. 958–963.

[21] Th. Fraichard and A. Scheuer, "Car-like robots and moving obstacles," *IEEE ICRA*, San Diego, CA (USA), May 1994, pp. 64–69.

[22] G. Heinzinger, P. Jacobs, J. Canny, and B. Paden, "Time-optimal trajectories for a robot manipulator: a provably good approximation algorithm," *IEEE ICRA*, Cincinnatti, OH (USA), May 1990, pp. 150–156.

[23] B. Donald and P. Xavier, "Provably good approximation algorithms for optimal kinodynamic planning for cartesian robots and open-chain manipulators," *ACM Symp. on Computational Geometry*, Berkeley, CA (USA), 1990, pp. 290–300.

[24] P. Bessière *et al.*, "Planning in a continuous space with forbidden regions : The ariadne's clew algorithm," *Algorithmic Fondations of Robotics*, A.K. Peters Publisher, Wellesley, MA (USA), 1995.

[25] L. E. Kavraki, P. Svetska, J-C Latombe, and M. H. Overmars, "Probalistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robotics and Automation*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[26] S.M. Lavalle and J.J. Kuffner, "Randomized kinodynamic planning," *IEEE ICRA*, Detroit, MI (USA), May 1999.