ABCM Symposium Series in Mechatronics - Vol. 2 - pp.597-604

ROUGH-TERRAIN ROBOT MOTION PLANNING BASED ON TOPOLOGY AND TERRAIN CONSTITUTION

Alan Ettlin
University of Applied Sciences of Central Switzerland, Institute of Electronics, Technikumstrasse 21, CH-6048 Horw, Switzerland
aettlin@hta.fhz.ch

Patrick Büchler
University of Applied Sciences of Central Switzerland, Institute of Electronics, Technikumstrasse 21, CH-6048 Horw, Switzerland
pbuechler@hta.fhz.ch

Hannes Bleuler
Swiss Federal Institute of Technology, Lausanne (EPFL), Ecublens, CH-1015 Lausanne, Switzerland
hannes.bleuler@epfl.ch

Abstract. *We present a randomised potential-field based motion planning approach for rough-terrain mobile robot navigation. The motion planning component is closely integrated with a real-time capable rigid-body kinematics and dynamics simulation module. The terrain is approximated as a polygonal surface generated from height data. The robot interacts with the terrain, modelled as a series of physically accurate collisions between its geometry and the polygons of the terrain. This model includes physical material properties, such as coefficients of restitution, static and dynamic friction of the underground as well as the robot parts. For each polygon of the terrain, a different set of material properties can be specified to represent variations in the underground constitution. In real rough-terrain settings it is typically not possible to accurately define a binary obstacle area, therefore we have included a continuous obstacle model. Each point of the terrain has a degree of "obstacleness" associated with it. This coefficient depends on the local terrain geometry and material properties. When navigating on the terrain, some areas of the workspace might be unreachable for the robot. Importantly, these areas need not a priori be specified as obstacles, but are a consequence of the physics-based robot-terrain interaction model. A motion plan is generated by performing a randomised search on a regular discreet grid potential computed over the workspace of the robot. Since no binary obstacle area exists, each point of the terrain contributes to the repulsive obstacle potential according to its obstacle coefficient. The computed motion plan allows the robot to navigate through the terrain taking into consideration not only the topology but also the physical underground properties.*

Keywords: *robot motion planning, rough-terrain navigation, continuous obstacle definitions, terrain material properties*

1. Introduction

A major difficulty in rough-terrain robot motion planning lies in the uncertainties of the underground constitution. In real-world settings it is hard to determine the spatial distribution of multiple physical properties of the terrain, controlling them is even harder. A further difficulty lies in the repeatability of experiments. It often proves challenging to recreate the same conditions for multiple algorithm test runs without introducing errors due to the terrain characteristics or robot configuration having changed.

To lessen the impact of these issues, we use a real-time simulation environment for researching rough-terrain motion planning algorithms. In this simulation, we create a physically accurate model of the terrain. Based on CAD data, robot geometries can be imported and complemented with actuator simulation components.

Having direct control over the terrain topology and its physical properties has allowed us to design rough-terrain motion planning algorithms which take full advantage of this additional information. In this publication we present the underlying idea as well as a concrete application. Typically, it is not possible to determine binary obstacle areas in rough-terrain settings. Therefore, we propose to use continuous obstacle definitions based both on terrain topology and physical underground properties. Such definitions are of general interest in rough-terrain motion planning, but due to the affinity of the continuous obstacle function with a potential, they fit naturally into a potential-field approach. The implementation of such a system is also presented in this publication.

In Section 2., we introduce the real-time simulation environment we use for the experiments. The terrain model is described in Section 3.. Section 4. deals with the motion planning algorithms themselves: the continuous obstacle definitions are discussed in Section 4.1, then details of a potential-field based implementation, which makes use of all the terrain information, are given. Section 5. gives an overview of the present state of research and motivates various aspects of future interest based on the introduced concepts.

2. Simulation environment

We use the Ibex simulation framework (e.g. Ettlin, Büchler and Bleuler, 2005) for prototyping and testing motion planning algorithms. Using a simulation for motion planning has a series of advantages, which are discussed in above mentioned publication. Rough-terrain motion planning taking into account physical terrain property distributions only becomes a possibility thanks to such a simulation environment. In the real world, precise control of the underground material properties could only be achieved with prohibitive effort.

Ibex is a modular software framework capable of running in real-time. The framework is based on a generic abstraction layer which defines the functionality of all entities involved in a complete simulation application.

For the experiments described in this document we use an implementation of the Ibex rigid body dynamics abstraction based on the Novodex (Novodex, 2005) physics engine. This module includes rigid body kinematics and dynamics as well as highly optimised collision detection and resolution algorithms. Therefore it is ideally suited to be used for kinodynamic motion planning. The implementation is based on work pioneered e.g. by Baraff (1993) and Mirtich (1998). One key strength of the module is its capability to run in real-time or faster up to a high complexity of the simulated model. Figure 1 shows the vehicle used as example in this document. The two chains of this fictional planetary rover consist of 100 rigid elements each, connected in a closed kinematic loop by as many revolute joints. In the simulation, four of the eight wheels are powered by applying torque to the revolute joints which connect them to the vehicle body through an actuator abstraction. The body itself consists of some 25'000 mesh triangles. All these rigid objects interact with each other and the terrain (consisting typically of over 10'000 triangles) by collisions to reproduce the physical behaviour of the system. Each object is assigned physical material properties, primarily coefficient of restitution, static and dynamic friction. It is also possible to assign such material properties to individual mesh triangles. In the present simulation this technique is used for the terrain, as described in Section 3..

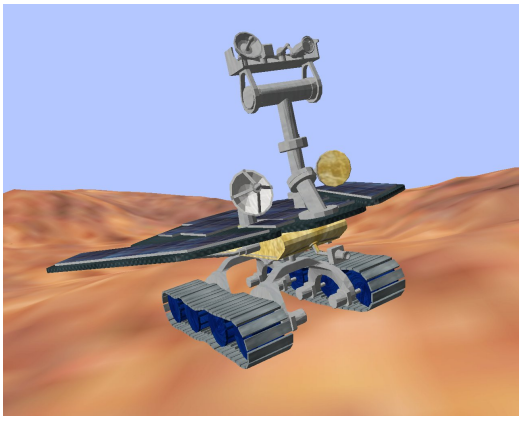

Figure 1. Fictional tracked vehicle used for the experiments described in this document.

To give a user feedback of the state of the simulation, Ibex includes various types of observer modules. Observers are entities which gather information about the state of the simulation and present it to the user. Real-time visual inspection of the simulation is best done by using a 3D graphics observer. Figure 1 is an example of the robot visualised in such an observer.

A simulation as described above runs in real time at 100Hz with some resources to spare, in parallel with a graphics observer. The test machine is a 1.6GHz Intel Pentium M processor notebook with 1GB RAM and an ATI Mobility Radeon 9200 GPU with 64MB VRAM running Microsoft Windows XP.

3. Terrain model

The raw terrain data is provided as set of (x, y, z)-coordinates with associated material properties for restitution (C_r), static friction (C_{f_s}) and dynamic friction (C_{f_d}) coefficients. The original point-based data is transformed into a triangulated polygonal mesh which represents the surface of the terrain. The original data defines a regularly spaced grid in the xy-plane, which allows a straightforward triangulation when generating the mesh. Four neighbouring points span a generally non-planar rectangular surface. Each such surface can be triangulated in two different combinations of three points with either of the two diagonals of the rectangle as common edge. One diagonalisation direction is arbitrarily selected. Vertex ordering within a triangle is by convention selected to be counter-clockwise from the outside of the object, i.e. above the terrain surface. Figure 2 shows part of a triangulated terrain.

The terrain model allows to specify individual material properties for each polygon of the surface. Since originally

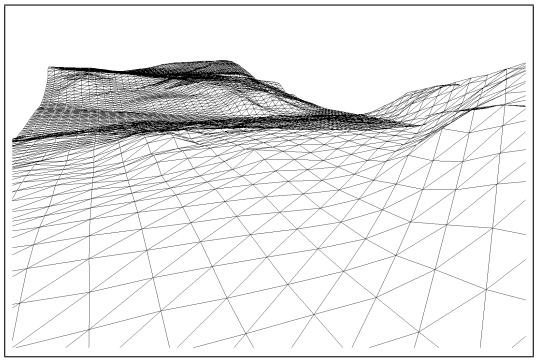

Figure 2. Triangulated surface of a terrain mesh object showing the arbitrary selection of diagonalisation direction.

physical material properties are specified for discrete data points, each terrain triangle obtains the mean value of the properties of its three constituting vertices, as in Eq. (1):

$$C_i(\triangle_{p_1,p_2,p_3}) = \left.\frac{\sum_{j=1}^{3} C_i(p_j)}{3}\right|_{C_i \in \{C_r, C_{fs}, C_{fd}\}} \tag{1}$$

To avoid an artificial uniformness of physical terrain parameters we optionally add noise following a random distribution when computing the set of properties for a mesh triangle.

4. Motion planning

In the simulation, we use a configuration space $\mathcal{C}_{sim} \in \mathbb{R}^N \times SO(N) \subset \mathbb{R}^{N+N^2}$ with $N = 3$, representing a configuration $\mathbf{q}$ of the robot $\mathcal{A}$ as its 3-dimensional position $\mathcal{T}_{sim}$ and orientation Θ_{sim}, cp. Latombe (1991). $\mathcal{T}_{sim}$ is defined as the position of the robot's frame $\mathcal{F}_{\mathcal{A}}$ with respect to the world frame $\mathcal{F}_{\mathcal{W}}$. Θ is the orientation of $\mathcal{F}_{\mathcal{A}}$ with respect to $\mathcal{F}_{\mathcal{W}}$. By convention, it is equal to the $N \times N$ matrix whose columns are the components of the unit vectors along $\mathcal{F}_{\mathcal{A}}$'s axis in $\mathcal{F}_{\mathcal{W}}$, thus belonging to the special orthogonal group $SO(N)$.

For motion planning, we use the projection of the three dimensional translational position $\mathcal{T}_{sim} \in \mathbb{R}^3$ onto $\mathcal{T}_{motion} \in \mathbb{R}^2$ as defined in Eq. (2).

$$\mathbf{q}_{\mathbf{trans}\,sim} = (x, y, z)^T \mapsto \mathbf{q}_{\mathbf{trans}\,motion} \equiv (x, y, 0)^T \tag{2}$$

This projection is justified since in our scenario the robot is all but bound to the terrain surface. The robot has no controlled way of moving along the height axis of $\mathcal{T}_{sim}$ other than following the terrain which determines its position. Likewise, the orientation Θ_{sim} of the robot $\mathcal{A}$ is mapped from $SO(3)$ onto an orientation $\Theta_{motion} \in SO(2)$ in the xy-plane. This orientation can further be reduced to a single variable $\theta \in [0, 2\pi)$ which represents the orientation as a single angle, cp. Eq. (3).

$$\mathbf{q}_{\mathbf{rot}\,sim} \in SO(3) \mapsto \mathbf{q}_{\mathbf{rot}\,motion} \equiv \theta = \tan^{-1}\left(\frac{\mathbf{e}_{\mathcal{F}_{\mathcal{A}}\mathbf{1}y}}{\mathbf{e}_{\mathcal{F}_{\mathcal{A}}\mathbf{1}x}}\right) \tag{3}$$

With $\mathbf{e}_{\mathcal{F}_{\mathcal{A}}\mathbf{1}}$ being the unit vector of $\mathcal{F}_{\mathcal{A}}$ which points forward (i.e. the heading of the robot). Combining translation $\mathcal{T}_{motion}$ and rotation θ, a configuration space $\mathcal{C}_{motion} \in \mathbb{R}^3$ results for motion planning.

$\mathcal{C}_{motion}$ is already inherently discrete along the translational x and y axes, due to the discrete nature of the raw point-based data. For practical purposes, the orientation axis θ is also discretely sampled, resulting in the overall discrete grid potential $\mathcal{GC}$. Importantly, the simulation and the associated configuration space $\mathcal{C}_{sim}$ remain continuous.

4.1 Continuous obstacle definitions

Often in potential-field based motion planning approaches, a binary obstacle definition is used. Obstacles $\mathcal{B}_i$ represent a clearly defined subspace of the workspace $\mathcal{W}$. These workspace obstacles are transformed to configuration space obstacles $\mathcal{CB}_i = \{\mathbf{q} \in \mathcal{C} | \mathcal{A}(\mathbf{q}) \cap \mathcal{B}_i \neq \emptyset\}$, with $\mathcal{A}(\mathbf{q})$ the subspace of $\mathcal{W}$ occupied by the robot $\mathcal{A}$ at configuration $\mathbf{q}$. This in turn leads to the definition of free space $\mathcal{C}_{free} = \mathcal{C} \setminus \bigcup_{i=1}^{\mathbf{q}} \mathcal{CB}_i$.

Considering rough-terrain navigation such a binary obstacle definition is typically not possible (cp. Iagnemma, Génot and Dubowsky, 1999). The boundary between navigable and impassable terrain is hard to define. Moreover, while in "difficult" terrain navigation might be possible, it can be desirable to display a preference for "easier" regions.

For the above mentioned reasons, we have opted for a continuous obstacle definition, which has led us to the definition of a degree of "obstacleness" $o(\mathbf{q})$, which is associated with every configuration $\mathbf{q} \in \mathcal{C}$. "Obstacleness" specifies the difficulty of navigating through the corresponding area. It ranges from free space with no difficulty whatsoever to impassable terrain. Such an obstacle definition implicitly also results in binary obstacles, in the sense that some configuration regions cannot be reached by the robot. Importantly though, these areas need not be specified beforehand but result as natural continuation of a continuous gradient of impassibility.

In our terrain model, the difficulty of passing a certain small area of the terrain depends not only on the topology but also on the physical material properties of the terrain. Therefore, these values are taken into consideration when computing the "obstacleness". For a given configuration $\mathbf{q} \in \mathcal{GC}$ following components of the "obstacleness" have been identified: the inclination of the terrain C_{incl} as well as the coefficients of restitution C_r, static and dynamic friction (C_{f_s} and C_{f_d}). Additionally, the vehicle used for the experiments (cp. Fig. 1) has marked difficulties with lateral inclinations, which has led us to additionally include a component C_{lat} covering this effect. Therefore the "obstacleness" o at a configuration $\mathbf{q}$ is a function of all these components in the example application:

$$o(\mathbf{q}) = f(C_{incl}(\mathbf{q}), C_{lat}(\mathbf{q}), C_r(\mathbf{q}), C_{f_s}(\mathbf{q}), C_{f_d}(\mathbf{q})) \tag{4}$$

Using a potential field method of motion planning, "obstacleness" can be used as basis for computing the repulsive potential $\mathbf{U}_{rep}$. Every individual component can be considered to generate a component potential field in the domain of $\mathcal{GC}$. The total repulsive potential $\mathbf{U}_{rep\ total}$ is then a function of the component potentials:

$$\mathbf{U}_{rep\ total}(x, y, \theta) = f'(\mathbf{U}_{incl}, \mathbf{U}_{lat}, \mathbf{U}_r, \mathbf{U}_{f_s}, \mathbf{U}_{f_d}) \tag{5}$$

In the following, we focus on the potential-field based implementation of the system making use of "obstacleness". The methods presented for component potentials in Section 4.2 and for the total repulsive potential in Section 4.3.1 are directly applicable to components and the total of $o(\mathbf{q})$ respectively. This allows continuous obstacle definitions based on "obstacleness" to be used for other motion planning approaches. In fact, the concept of "obstacleness" is independent of any specific motion planning implementation.

4.2 Potential components

Terrain inclination is closely related to the difficulty for the robot to navigate on the underlying terrain. Therefore, we introduce the component repulsive potential $\mathbf{U}_{incl}$, which corresponds to the magnitude of the terrain inclination. $T(x, y)$ denotes the terrain height data in the following:

$$\mathbf{U}_{incl}(x, y) = |\nabla T(x, y)| \tag{6}$$

Figure 3 shows the contour plot of an example terrain used for testing. The arrows indicate gradient orientation and magnitude. The inclination potential component $\mathbf{U}_{incl}$ is independent of the robot orientation, hence it is defined on the two-dimensional space spanned by $\mathcal{T}_{motion}$. The isopotential lines of $\mathbf{U}_{incl}$ for the example terrain are shown in Fig. 4.

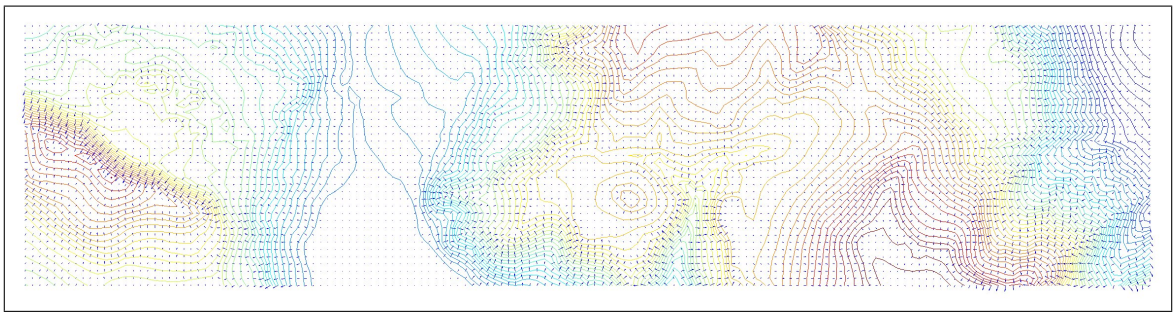

Figure 3. Example terrain topology, displayed as contour lines and gradient vectors.

Navigation over laterally inclined terrain poses problems to the example robot. Therefore, we have introduced the lateral inclination component $\mathbf{U}_{lat}$, which adds a tendency for the robot to avoid laterally inclined terrains to the overall repulsive potential. The configuration space interpretation of $\mathbf{U}_{lat}$ is that the (continuous) workspace obstacles $\mathcal{B}_i$ are mapped to configuration space obstacles $\mathcal{CB}_i$ as function of the robot orientation θ. A possible definition for $\mathbf{U}_{lat}$ is given in Eq. (7):

$$\mathbf{U}_{lat}(x, y, \theta) = -\frac{|\nabla T(x, y)| \cdot (\cos(2(\Phi(x, y) - \theta)) + 1)}{2} \tag{7}$$

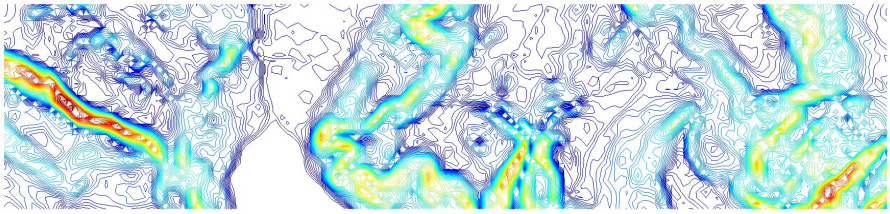

Figure 4. Isopotentials of orientation-invariant components of $\mathbf{U}_{rep\ total}$ computed for the terrain shown in Fig. 3.

$\Phi(x, y)$ is the direction of the terrain inclination (i.e. of its gradient ∇T) projected onto the xy-plane and expressed as an angle, cp. Eq. (8):

$$\Phi(x, y) = \tan^{-1}\left(\frac{\frac{\partial T(x,y)}{\partial y}}{\frac{\partial T(x,y)}{\partial x}}\right) \tag{8}$$

The remaining repulsive potential components represent the linear degree of added difficulty by the respective physical property. The straightforward mappings of the physical terrain parameters to potential values are shown in Eq. (9):

$$\mathbf{U}_{phys}(x, y)|_{U_{phys}\in\{U_r,U_{f_s},U_{f_d}\}} = (1 - C_{phys}(x, y))|_{C_{phys}\in\{C_r,C_{f_s},C_{f_d}\}} \tag{9}$$

4.3 Total potential

4.3.1 Repulsive potential

Before discussing any concrete total repulsive potential functions, it is important to notice the different ranges generated by the individual component functions. The trivial components $\mathbf{U}_r$, $\mathbf{U}_{f_s}$ and $\mathbf{U}_{f_d}$ all map to $[0, 1]$. In theory, $\mathbf{U}_{incl}$ maps to $[0, \infty)$ for perfect vertical inclinations. In practice though, given the discretised terrain model introduced in Section 3., the upper limit of the inclination is bounded and well within numerical range. The lateral inclination component $\mathbf{U}_{lat}$ is defined in Eq. (7) in a way to map to the same range as the inclination component. These differences in range must be taken into consideration when designing a concrete potential function.

Inclination is a direct measure for the difficulty of navigation and hence, the presence of a continuous obstacle based on the degree of "obstacleness". Therefore, it has a orientation-invariant component in the total repulsive potential function. In a first approach, nothing is known about the nature of the motion in terms of the adhesion of the robot to the underground. Therefore, both the static and dynamic friction component are used to scale the inclination component of the total potential in the example.

The reason for introducing the lateral inclination potential component is that the example robot model displays difficulties moving at right angles to the terrain inclination. This effect is caused by the robot not adhering perfectly to the terrain, but rather laterally sliding and consequently entering an uncontrolled rotation. With the example robot, the low-level controller of the robot chains is capable of compensating sliding effects when moving along the terrain gradient to some degree. The non-holonomic motion constraints of its design prevent it from doing so for perpendicular motion. This leads to the robot drifting away from the locally ideal trajectory. The lateral sliding motion, which is to be prevented by $\mathbf{U}_{lat}$, occurs when the dynamic friction model takes over from the static friction model in the simulation. For this reason, the lateral inclination component is scaled with the dynamic friction component $\mathbf{U}_{f_d}$.

Even with no terrain inclination at the current configuration, suboptimal adhesion hinders the progress of the robot. To represent this fact, we have introduced a friction term in the example repulsive potential. The definition of the total repulsive potential function for the example application is shown in Eq. (10):

$$\mathbf{U}_{rep\ total\ example} = a \cdot \mathbf{U}_{f_s} \cdot \mathbf{U}_{f_d} \cdot \mathbf{U}_{incl} + b \cdot \mathbf{U}_{f_d} \cdot \mathbf{U}_{lat} + c \cdot \mathbf{U}_{f_s} \cdot \mathbf{U}_{f_d} \tag{10}$$

where a, b and c are weighting coefficients of the inclination, lateral inclination and friction terms respectively. Figure 5 shows the repulsive potential as defined in Eq. (10) for the example terrain (cp. Fig. 3) with $\mathbf{U}_{incl}$ as shown in Fig. 4. $C_{phys}(\mathbf{q})$ and hence $\mathbf{U}_{phys}$ are constant to improve the clarity of the illustrations. Regions of high potential values are rendered as solid surfaces. Note the π-periodic behaviour of $\mathbf{U}_{lat}$ over the (vertical) orientation axis.

4.3.2 Attractive potential

The attractive potential $\mathbf{U}_{att}$ is not the focus of the research presented in this document, but rather included to describe the complete potential-based motion planner. For convenience, the attractive potential is discussed as two separate func-

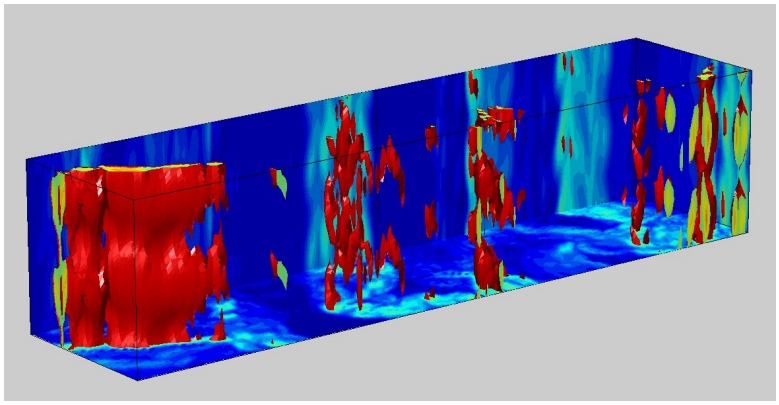

Figure 5. 3D repulsive potential based on the orientation-invariant components shown as 2D potential in Fig. 4. The orientation θ is plotted as vertical axis. Solid areas denote high potential magnitudes. Note the periodicity along the orientation due to the lateral inclination term $\mathbf{U}_{lat}$.

tions for translation $\mathbf{U}_{att\ trans}$ and rotation $\mathbf{U}_{att\ rot}$ with $\mathbf{U}_{att} = d \cdot \mathbf{U}_{att\ trans} + e \cdot \mathbf{U}_{att\ rot}$ the weighted sum of the two.

The translational component has a global minima at the goal configuration $\mathbf{q}_{goal} \in \mathcal{C}_{motion}$ and no further minima in $\mathcal{C}_{motion}$. This is classically achieved e.g. by a parabolic (quadratic) well. We use slightly more general symmetric polynomials of up to third degree, as defined in Eq. (11):

$$\mathbf{U}_{att\ trans}(x, y) = \xi_1 \rho_{goal}(\mathbf{q}) + \xi_2 \rho^2_{goal}(\mathbf{q}) + \xi_3 \rho^3_{goal}(\mathbf{q}) \tag{11}$$

where ρ_{goal} denotes the translational Euclidean distance from the current configuration $\mathbf{q}$ to the goal:

$$\rho(\mathbf{q}) \equiv ||\mathbf{q}_{trans} - \mathbf{q}_{trans\ goal}|| = \sqrt{(x - x_{goal})^2 + (y - y_{goal})^2} \tag{12}$$

The attractive component for orientation is defined taking into account orientational 2π-periodicity, as shown in Eq. (13) with Φ the angle of terrain inclination defined in Eq. (8).

$$\mathbf{U}_{att\ rot}(x, y, \theta) = \left((\Phi(x, y) - \theta)|_{\ \mathrm{mod}\ 2\pi} - \pi \right)^2 \tag{13}$$

In our example, the desired behaviour of the robot is to first attain the optimal orientation, then start translational movement. Therefore $\mathbf{U}_{att\ rot}$ is the dominant potential in $\mathbf{U}_{att}$, creating a tendency for the inverse gradient-following behaviour to first rotate the robot and then move in the reached direction.

Note, that $\mathbf{q}_{goal}$ is defined in an orientation-independent way, i.e. it does not matter from which direction the target position is reached. In terms of the three-dimensional (x, y, θ)-configuration space, this means that the target configuration $\mathbf{q}_{goal}$ is not a single point but rather a line through all points of the target x and y values. Figure 6 shows an example of $\mathbf{U}_{att}$. The main axis of the solid spiral indicates the (x, y)-position of $\mathbf{q}_{goal}$.

The combined potential function $\mathbf{U}$ is the superposition of $\mathbf{U}_{rep\ total}$ and $\mathbf{U}_{att}$. Since we internally store the attractive and repulsive potentials separately, composed trajectories $\tau_{total} = \tau_1 \circ \tau_2 \circ \ldots \circ \tau_n$ can easily be generated dynamically. Assume a motion plan $\tau_i : \mathbf{q}_{goal\ i} \rightarrow \mathbf{q}_{goal\ i+1}$ has been generated and is followed by the robot as described in Section 4.4. τ_i is based on $\mathbf{U}_{rep\ total}$ and $\mathbf{U}_{att}(\mathbf{q}_{goal\ i+1})$. When $\mathbf{q}_{goal\ i+1}$ has been reached, the target can be switched to $\mathbf{q}_{goal\ i+2}$ by substituting $\mathbf{U}_{att}(\mathbf{q}_{goal\ i+1})$ by $\mathbf{U}_{att}(\mathbf{q}_{goal\ i+2})$ in $\mathbf{U}_{rep\ total}$. This algorithm can be repeated until the final goal $\mathbf{q}_{goal\ n}$ has been reached. Note, that the attractive potentials need not be pre-computed, but can also be dynamically generated at run-time if the intermediate targets are not known beforehand.

4.4 Trajectory following

The classical RPP (Randomised Path Planner) approach (Barraquand and Latombe, 1991) assumes perfect robot motion, i.e. no dynamics effects, sliding motions etc.. As described above, our simulation explicitly includes such behaviour. To deal with these effects, we have adopted the basic idea of RPP and reverted it to straightforward inverse-gradient following (e.g. Latombe, 1991), which is more robust in that respect. Instead of pre-computing a motion plan, which might not be able to be satisfied due to disregarded kinodynamic constraints, the robot at each configuration attempts to follow the inverse gradient of the potential. Applying conservative limits on the torque outputs of the actuators, thus keeping the robot in a controllable state, closely follows the path that would be computed by the best-first motions of RPP. In the event

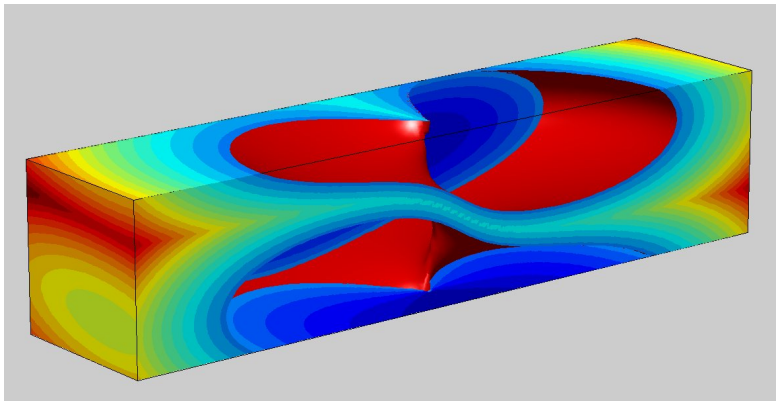

Figure 6. Attractive potential $\mathbf{U}_{att}$ with the orientation θ plotted as vertical axis. The (x, y)-position of the goal configuration $\mathbf{q}_{goal}$ is visible as main axis of the solid spiral.

of an uncontrolled motion (e.g. lateral sliding due to excessive inclination), no new path needs to be computed, but rather the inverse potential gradient at the new configuration followed. This introduces a new element of randomness related to the dynamic behaviour of the robot, which is non-deterministic from the point of view of the motion planner. While the exact implications of this randomness need to be further studied, the effect is negligible in practical setups.

The main drawback of potential-based methods are difficulties caused by the planner being trapped at local minima $\mathbf{q}_{loc}$. RPP tackles this issue by performing a random walk of length t steps. Each step performs a motion of either Δ_i or $-\Delta_i$ along each axis of $\mathcal{GC}$, with Δ_i being the discrete increment along the i-th axis. We have adopted this idea to design a local planner, which takes over when the robot reaches a local minimum of the potential.

A series of t_j random steps are performed and the reached configuration $\mathbf{q}_j$ at distance $\rho_j = |\mathbf{q}_j - \mathbf{q}_{loc}|$ stored. Next, the local planner executes a series of inverse-gradient following moves until a configuration $\tilde{\mathbf{q}}_j$ with distance $\tilde{\rho}_j = |\mathbf{q}_j - \tilde{\mathbf{q}}_j| > \rho_j$ has been reached. If $\tilde{\mathbf{q}}_j$ lies in a ε-neighbourhood of $\mathbf{q}_{loc}$, the random walk has not yet reached a position outwith the influence of the local minima and further t_{j+1} random steps are performed. Otherwise, the path $\tau_j : [0, 1] \mapsto \mathcal{C}_{motion}$ with $\tau_j(0) = \mathbf{q}_{loc}$ and $\tau_j(1) = \mathbf{q}_j$ is followed by the robot. At $\mathbf{q}_j$ normal inverse-gradient following operations can recommence.

In the present document, we assume the robot can rotate on the spot, as is the case with the example robot depicted in Fig. 1. This fact was already drawn upon when defining the attractive potential in Section 4.3.2. For the three-dimensional discrete configuration space $\mathcal{GC}$ with coordinate axes x, y and θ, such a rotation corresponds to a free motion along the vertical θ axis.

The example robot is subject to a motion constraint specifying that it can only perform a controlled translational motion in direction of its main axis. This reduces possible translational motions at a constant orientation θ (figuratively within one horizontal slice of $\mathcal{GC}$) to motions parallel to θ. The general potential function described in Section 4. does not take into account such motion constraints. Either the total potential function $\mathbf{U}$ has to be post-processed to include the motion constraint or an alternative scheme devised. Since post-processing of $\mathbf{U}$ which restricts translations to specific orientations is prone to introduce undesirable local minima, we have devised a heuristic scheme to deal with the issue:

If inverse-gradient following requires a translation along a different direction than that of the present orientation, a rotation is initiated which starts changing the orientation of the robot in the required direction. As a consequence, the current configuration $\mathbf{q}$ translates along the θ-axis of $\mathcal{GC}$, figuratively formulated. This reflects the fact that for the new orientation the potential function in the (x, y)-plane has changed due to the potential components dependent on the orientation, in the example $\mathbf{U}_{lat}$. To some extent, this heuristic carries the risk of introducing new local minima too, when rotational components generated within an (x, y)-plane cancel out the genuine components along the θ-axis. When such a case is recognised, the local planner described above is also employed to escape from the minimum.

5. Outlook and future work

Preliminary experiments with the system described above have led to encouraging results. Using potential functions like the ones introduced in this document, the robot displays the desired behaviour. Nevertheless, more test runs as well as formal reasoning need to be conducted to establish general qualities of the approach. This is particularly true for the trajectory following component which strongly relies on heuristics.

Further development of continuous obstacle definitions based on the concept of "obstacleness" needs to be performed. On the one hand, a complete formalism should be developed. Also, additional non-standard components should be designed, as required by concrete applications. An example is the lateral inclination component introduced in this document, motivated by the difficulties the example robot displayed with such slopes. New components could be based on a wide variety of properties relevant to rough-terrain motion planning, including e.g. the current velocity, the degree of adhesion to the underground or importantly certain cost functions. Cost functions result from applying performance criteria to the motion planning task, e.g. Guo et al. (2003).

For this publication, the continuous obstacles based on the degree of "obstacleness" have been integrated into a potential-based motion planning framework due to the close resemblance between "obstacleness" and a potential function. The same concept could also be used with other motion planning methods to generate continuous obstacles in the configuration space and then compute the motion plan accordingly. This would allow to benefit from the advantages of including additional information in the obstacle definitions, while avoiding the problems associated with potential-based planners.

One important aspect which still needs to be included into a planner based on "obstacleness" is parameter learning. When generating a total potential (or the overall continuous obstacle function for other approaches), a number of weighting coefficients are used which reflect the importance of the constituent components. These parameters, and possibly others within individual obstacle components, should be learnt and updated during robot operation. The goal of such a learning process is that the "obstacleness" should reflect the real difficulty for the robot to traverse the corresponding region. One requirement is that regions which cannot be traversed by the robot have "obstacleness" values associated with them, which reflect this fact (as a sideline: this is a typical example of velocity being a useful "obstacleness" component). The learning process should iteratively determine the difficulty of navigation and update the tunable coefficients accordingly until it converges with the desired values.

Further, there is a need for validation of the results obtained with the simulator in comparison with the real world. This necessarily includes not only a validation of the kinodynamic robot simulation but also of the execution of the motion plan, which explicitly includes the interaction with the terrain model.

6. Conclusion

We have briefly described the simulation environment and terrain model we use for rough-terrain robot motion planning. Then, based on the terrain model, the concept of a degree of "obstacleness" has been introduced in a general form. A concrete randomised potential-field based planner demonstrating the use of "obstacleness" has been described, giving more details of how to apply the general concept.

The components of the degree of "obstacleness" are mapped to component repulsive potentials. These components are combined with an attractive potential to form an overall discrete grid potential. A simple randomised inverse-gradient following approach is presented for determining the trajectory based on the potential.

The concept of "obstacleness" is still at a preliminary state of development; some of the ideas, which we hope will make it a powerful concept for rough-terrain motion planning are outlined and related directions for further research motivated.

References

Baraff, D., 1993, "Non-penetrating rigid body simulation", Proceedings of the First Eurographics Workshop on Virtual Environments.

Barraquand, J., Latombe, J.-C., 1991, "Robot Motion Planning: A Distributed Representation Approach", International Journal of Robotics Research, p. 628–649.

Ettlin, A., Büchler, P., Bleuler, H., 2005, "A Simulation Environment for Robot Motion Planning", RoMoCo'05, 5^{th} International Workshop on Robot Motion and Control.

Guo, Y., Parker, L. E., Jung, D. L. ,Dong, Z., 2003, "Performance-based rough terrain navigation for nonholonomic mobile robots". The 29^{th} Annual Conference of the IEEE Industrial Electronics Society (IECON'03).

Iagnemma, K., Génot, F., Dubowsky, S., 1999, "Rapid Physics-Based Rough-Terrain Rover Planning with Sensor and Control Uncertainty", IEEE International Conference on Robotics and Automation.

Latombe, J.-C., 1991, "Robot Motion Planning", volume SECS 124 of The Kluwer international series in engineering and computer science. Kluwer, cop.

Mirtich, B., 1998, "Rigid body contact: Collision detection to force computation", Technical Report TR-98-01, Mitsubishi Electrical Research Laboratory.

Novodex, 2005, *http://www.ageia.com/novodex.html*