

3D Mobility Learning and Regression of Articulated, Tracked Robotic Vehicles by Physics-based Optimization

Panagiotis Papadakis and Fiora Pirri

ALCOR - Vision, Perception and Cognitive Robotics Laboratory
Department of Control and Management Engineering, University of Rome 'La Sapienza', Italy

Abstract

Motion planning for robots operating on 3D rough terrain requires the synergy of various robotic capabilities, from sensing and perception to simulation, planning and prediction. In this paper, we focus on the higher level of this pipeline where by means of physics-based simulation and geometric processing we extract the information that is semantically required for an articulated, tracked robot to optimally traverse 3D terrain. We propose a model that quantifies 3D traversability by accounting for intrinsic robot characteristics and articulating capabilities together with terrain characteristics. By building upon a set of generic cost criteria for a given robot state and 3D terrain patch, we augment the traversability cost estimation by: (i) unifying pose stabilization with traversability cost estimation, (ii) introducing new parameters into the problem that have been previously overlooked and (iii) adapting geometric computations to account for the complete 3D robot body and terrain surface. We apply the proposed model on a state-of-the-art Search and Rescue robot by performing a plurality of tests under varying conditions and demonstrate its efficiency and applicability in real-time.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Physically based modeling—Cognitive science

1. INTRODUCTION

Parallel to common robotic applications where robots operate within structured environments, there has been an evident interest in advancing robot technology so that they can be deployed into outdoor, off-road, natural, as well as unnatural environments. Robotic applications such as planetary exploration, Urban Search and Rescue (USAR), forestry and mining are made feasible by designing robots with reconfigurable components that adapt to rough terrain. Towards this goal, typically, there exist a number of issues that need to be addressed, namely; (i) estimation of the terrain traversability, (ii) path planning and (iii) adapting the configuration of the robot for the purpose of motion planning.

The focus of this paper relies primarily in alleviating the first and third issue through a unified perspective. Our motivation originates from the fact that while there is extensive work in relation to a number of applications, there is a fairly limited amount of work in the domain of USAR environments that are undoubtedly the most complex in terms of terrain irregularities and span the highest range in the diversity

of terrain classes [KPG*12]. Furthermore, due to the complexity and plurality of challenges that are involved, most research efforts have been dedicated to address these issues distinctively and often by imposing several oversimplifications.

Simulating the physical behaviour of a robotic vehicle before the actual execution of its task (see Figure 1) is frequently considered a paramount component [BLS01], [HJC*08]. In the majority of cases, this is useful in preventing the execution of unachievable or hazarding tasks, in the

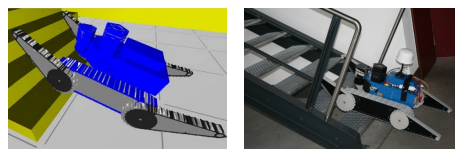


Figure 1: Physics-based simulation of a common robot traversal task, namely, staircase climbing.

sense that if a simulated task eventually results in a failure, then this would also be the expected outcome in the real setting. When the robot and its interaction with the environment can be simulated at sufficiently high fidelity, then the simulated plan can also be used in order to guide the actual robotic vehicle [HK07] or in the context of shared-autonomy operation modes [ONY*11], [CB08].

In this paper, we propose a methodology to consistently and precisely assess static traversability costs for reconfigurable tracked mobile robots operating in 3D terrain and in particular, for USAR environments. We consider both intrinsic robot characteristics and articulating capabilities in combination to the entire terrain surface and model the physical behaviour of a given robot and its interaction inside a given environment, by considering the exact 3D shape of the robot and the terrain surface. The result yields the stable pose of the robot together with a set of corresponding cost measurements, that quantify the difficulty of the robotic vehicle to reside over the given terrain. The general framework for learning and regressing the mobility of a robotic vehicle in 3D terrain through a physics-based simulation was sketched partly in our earlier work [GPPP11]. In the present work we unfold the details behind the theory and implementation of the complete framework, whose contributions are summarized as follows:

- A simulated, physics-based approach to efficiently obtain the stable state of an articulated robot on a given 3D terrain, that accounts for a terrain model, robot model as well as its stability and kinematic constraints.
- A framework for quantifying static, 3D terrain traversability of the robot at the optimally stable state.

As a direct application of the above two contributions, we can endow the robot with the ability to acquire its mobility skills through off-line learning within an exhaustive range of 3D terrain shapes. In turn, the robot is rendered capable of regressing the traversability of an encountered terrain in real-time and suitably adjust its morphology upon traversal.

We organize the remainder of the paper as follows: In Section 2 we review the related work in 3D terrain mobility analysis and in Section 3, we first formulate the problem that we are addressing and continue by describing the proposed methodology. In Section 4 we present our experiments using the proposed approach and in Section 5 we summarize the contributions of this paper.

2. RELATED WORK

The predominant approach for measuring the traversability of 3D terrain concerns the analysis of 2D *Digital Elevation Maps* (DEM) [KK92], originating from *Occupancy Grid* maps whose usage in robotics is accredited to Moravec [ME85]. The majority of methods for quantifying terrain traversability of mobile robots, concerns the computation of a set of features that are based on a simplistic terrain model

and simulating the robot as a point or a basic geometric entity, such as a rectangle or a sphere.

Within this line of thought, one of the earliest approaches corresponds to the work Langer et al. [LRH94] who computed elevation statistics from the set of 3D points within each grid cell, namely, the maximum, minimum and variance of height and slope that were checked by hard thresholds set according to the UGV (Unmanned Ground Vehicle) capabilities. In the pioneering work of Genery [Gen99], a cost function aggregates the elevation, slope, roughness and data point accuracy. Those features were computed by iterative plane-fitting that adaptively weighed the fitted points according to their accuracy, roughness and distance from the cell center. The work of Helmick et al. [HAM09] goes a step further by using several description levels of increasing granularity. They build upon fine terrain descriptions extracted from the GESTALT system [GMM02] that outputs a *goodness map* which quantifies traversability by locally fitting planar patches and using the patch statistics to derive *step*, *roughness*, *pitch* and *border* hazards. In a higher-level, terrain is classified into traversability classes by thresholding the goodness value of each cell. On the other hand, the focus of the work of Singh et al. [SSS*00] was primarily in alleviating uncertainty and error, by assessing traversability jointly through quantifying terrain *goodness* and *certainty*. Goodness is determined as the minimum of the roll, pitch and roughness of planar rover-sized patches that are computed by fitting planes onto the stereo range points and computing the residual of the planes while certainty depends on the number and variance of points within the patch as well as its distance from the position of the UGV.

Notable, earlier approaches that build higher-level robotic representations and take into account robot-dependent variables are comparatively limited. A representative example corresponds to the work of Bonnafous et al. [BLS01] who model traversability as a *danger* attribute, taking into account the robot configuration and stability constraints related to the pitch/roll angles of the articulated components and an uncertainty constraint that accounts for the sparseness of information within the DEM. In [VDH06], one type of map stores a cost incurred by the presence of vegetation encoding the confidence of terrain reconstruction that could be used to plan paths below canopy and a second type of map is derived by superposition of the robot across different directions on the DEM and estimating its roll, pitch and ground clearance. These estimates are then smoothly mapped into a finite fixed interval and the total cost is taken as the least favourable of the three criteria. In [HJC*08], the traversability of unknown terrain is determined by employing forward simulation of a path following within the ROAMS environment and calculating the energy consumption along a path together with the amount of wheel slippage. In the recent work of Ishigami et al. [INY11] terrain traversability is evaluated through the *dynamic mobility index* that considers robot stability, wheel slippage, time duration and energy consumption. Terrain

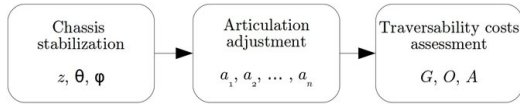


Figure 2: Stable state and cost assessment pipeline.

roughness is computed as the standard deviation of elevation across the robot footprint when projected onto the map at varying yaw angle and wheel slippage is quantified by measuring the terrain roll/pitch inclination. While these approaches assess robot mobility at a finer level, there exists no previous work to the best of our knowledge, for actively adaptable, tracked, mobile robots operating in USAR environments that can reliably model the robot and its interaction with the terrain, further taking into account its kinematic and stability constraints. Earlier work that bears the highest similarity in terms of scenario and robotic vehicle concerns the work of Okada et al. [ONY*11]. In contrast to our approach, their stable pose estimation is based on regressing the slope of the terrain and not by considering its interaction with the UGV while the stability cost does not take into account the complete geometry of the interacting surfaces. Finally, their goal was to develop a shared, human-robot mode of motion while our goal is to allow the robot to learn its mobility skills and allow it to optimize its motion planning autonomously.

3. PROPOSED METHODOLOGY

The problem that we address can be described as follows. We seek to obtain the stable pose of a UGV on top a given terrain considering its mobility capabilities, together with its physical interaction with the terrain involving the effects of surface collisions, gravitational force and potential slipping. We further aim to quantify the capability of the vehicle to reside, statically, over a terrain under a specific state, by taking into account its kinematic constraints together with a set of criteria based on which the optimality of a state can be assessed. The pipeline for obtaining the stable robot state and the corresponding traversability costs is decomposed into three sequential stages as shown in Figure 2.

More formally, we denote the *configuration space* of a robot as $\mathbf{C} = (x, y, z, \phi, \theta, \psi, \alpha_1, \alpha_2, \dots, \alpha_n) \subset \mathbb{R}^{n+6}$, where (x, y, z) denotes its 3D position, (ϕ, θ, ψ) its roll, pitch, yaw

and $(\alpha_1, \alpha_2, \dots, \alpha_n)$ give the rotations of the n articulating components of the robot. The *state space* \mathbf{S} is derived from the kinematic constraints of the particular robot. A sub-configuration space \mathbf{C}_m describes a subspace of \mathbf{C} , through a mapping $m : \mathbf{C} \mapsto \mathbf{C}_m \in \mathbb{R}^d, d < n + 6$ yielding the corresponding sub-state space \mathbf{S}_m .

We represent a terrain surface as a DEM and denoted by $M : M(i, j) \rightarrow \mathbb{R}$ where $i \in \{1, 2, \dots, w\}, j \in \{1, 2, \dots, l\}$, and w, l correspond to the width and length of the grid respectively. The value of $M(i, j)$ is used to capture the height of the supporting terrain at the cell (i, j) according to the global coordinate frame. A DEM is the equivalent of a 2D depth map in the computer graphics domain. Although any 3D surface could be used, we base our description using DEM as they facilitate path planning by using graph-search algorithms.

In general, we could be interested in determining all the possible states of the robot for a cell (i, j) and further order the different cases according to the cost. Most often, however, motion planners do not search exhaustively within a cost map, rather, the search space is constrained through a set of precomputed paths that consider the *maneuvering* capabilities of the vehicle (e.g. as in [LF09], [HK07] and [HGF08]). Under this perspective, e.g. when using an *arc-based* path planner, the search space is constrained by eliminating a number of degrees of freedom from the initial \mathbf{C} space. In particular, we obtain the sub-state space $\mathbf{S}_{path} = (x, y, \psi)$ through the set of fixed candidate paths in front of the robot generated by the path planner that prescribe the position and direction that the robot would follow. Hence, the traversability assessment problem as sketched in Figure 2, decomposes into the following steps:

- Estimation of the sub-state $\mathbf{s}_{ter} = (z, \phi, \theta)$ as a result of the contact of the robot with the terrain (Section 3.1).
- Estimation of the sub-state $\mathbf{s}_{art} = (\alpha_1, \alpha_2, \dots, \alpha_n)$ corresponding to the rotation angles of the articulated components, as a result of the robot kinematic constraints and contact with the terrain (Section 3.2).
- Traversability costs assessment (Section 3.3).

For the complete framework to be applicable, the following main assumptions are primarily accommodated:

- i. The robot acquires a 3D point cloud of the terrain in question, that is subsequently transformed into a corresponding polygonal surface (as shown in Figure 3).

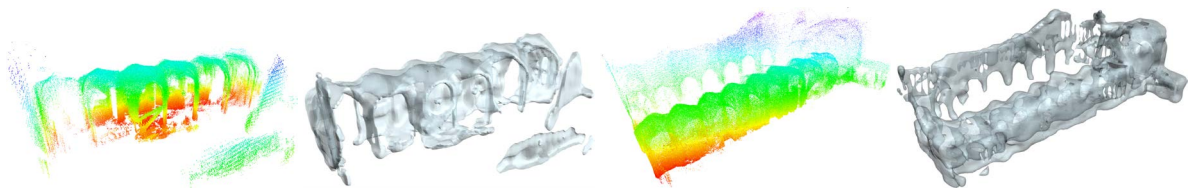


Figure 3: 3D point clouds as acquired from the UGV and reconstructed surfaces of two churches damaged by earthquake.

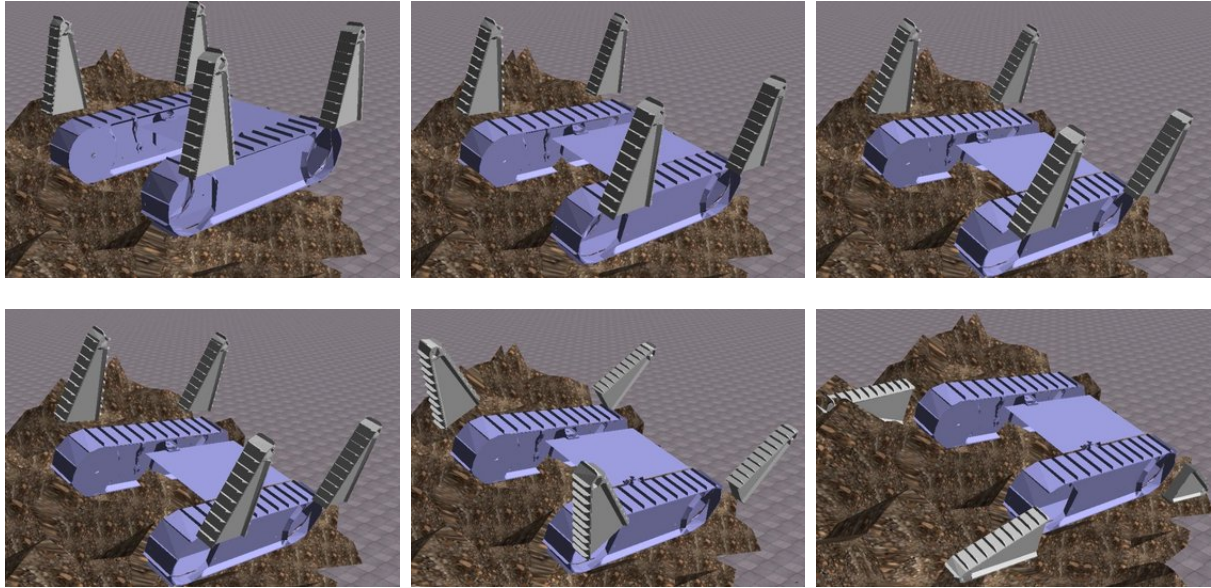


Figure 4: Top row: Snapshots of the stable sub-state estimation process of robot elevation and orientation s_{ter} . From left to right, the robot is shown at its initial given state s_{path} , intermediate state and its final stable state. Bottom row: Snapshots of the sub-state estimation of rotation angles of articulating bodies s_{art} . From left to right, the robot is shown at its stable chassis state, intermediate reconfiguration state and at its final optimal state.

- ii. The polygonal surface is not subject to changes and can be considered as sufficiently rigid to support the weight of the robotic vehicle.

Since our focus is not to develop new methodologies to accommodate these conditions, for our scenarios we found particularly useful the application of optimized kernel-based surface density estimation. In particular, through the use of Support Vector Machines (SVM) [PP11, TD99, SGS05], we obtain the oriented polygonal surface (see Figure 3) as the 0-level set of the hyper-decision surface. Our motivation in using this approach is mainly due to the formulation that allows dealing with noise and error by regulating the *slack* ratio and with uncertainty through the variance of the Gaussian kernel.

3.1. Sub-state estimation of robot elevation and orientation

The estimation of the sub-state $s_{ter} = (z, \phi, \theta)$ is performed through a mapping of the state $s_{path} = (x, y, \psi)$, namely, $s_{ter} = g_R(s_{path})$. Here, g_R is the mapping from the path sub-state space S_{path} to the terrain sub-state space S_{ter} , that models the physical interaction of the main body of the robot under consideration when residing over the terrain at a state s_{path} , taking into account gravity, friction, bounciness and softness of the terrain.

The exact estimation of the latter three parameters requires *appearance-based* terrain classification or the use of

dedicated sensors. Depending on the available robot perception capabilities, these parameters could be incorporated into the physics-based simulation at the cost of increased computational complexity. Typically, however, robot movements within USAR scenarios are of low speed and on rough terrains that implies setting the Coulomb friction coefficients to comparatively high values, while bounciness and softness are trivial. In this context, the only information required by our model is the 3D shape of the vehicle (and that of the terrain), the robot's mass characteristics and its inertia axes, which are acquired directly from the robot specifications [Blu11].

To estimate s_{ter} , we overlay the robot on top of the terrain at the given sub-state s_{path} and iterate in time until its motion (linear and angular) vanishes. At that moment, we consider the robot as stably residing over the terrain. This step can be comprehended as a simulated parallelism of the *natural* stabilization process that would take place if the robotic vehicle laid on the given terrain region.

At the top row of Figure 4, we show snapshots of the three distinct stages of the physically-based simulation of this step, namely, *initial*, *intermediate* and *stable* state and in Algorithm 1 we describe the respective steps. Note that the final stable state may have caused a drift from the initial $s_{path} = (x, y, \psi)$, as a result of the robot stabilization process. In Section 3.3 we explain how we explicitly take this into account into the traversability cost assessment.

Algorithm 1: Sub-state computation of the UGV elevation and orientation

Input: M : DEM of the terrain
 ugv : 3D kinematic model of the vehicle
 $s_{path} = (x, y, \psi)$: Initial position and yaw of the ugv within M
Output: $s_{ter} = (z, \phi, \theta)$

```

begin
  Chassis_Stable = false
  t = 0
  while Chassis_Stable = false do
    Compute (ugv, M, t) interaction
    if  $(v = \dot{p}(x, y, z) \simeq \mathbf{0} \wedge \omega = \dot{r}(\phi, \theta, \psi) \simeq \mathbf{0})$  then
      Chassis_Stable = true
      t = t +  $\Delta t$ 
    end
  end
end

```

In Figure 5 we show how the magnitude of the linear and angular velocity of the robotic vehicle typically evolves until Algorithm 1 converges to the stable state of the vehicle. Characteristically, the vehicle initially attains a small linear momentum by overlaying it above the terrain at a very small height. Upon contact with the terrain, the stabilization process of Algorithm 1 is initiated that is witnessed through a sudden increase in angular velocity that happens concurrently with small, decreasing fluctuations of the linear speed. Once the angular velocity reaches its maximum, then potential slipping starts to vanish together with the vehicle’s linear speed which implies that the vehicle starts to settle on top of the given terrain until it finally remains immobile.

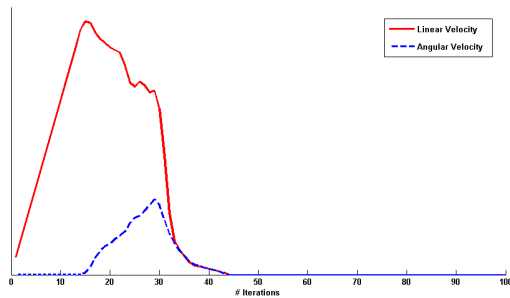


Figure 5: Convergence behaviour of Algorithm 1 for estimating the robot elevation and orientation at the stable state.

3.2. Sub-state estimation of rotations of articulating bodies

Upon completion of the previous stage we further improve the robot’s stability through a suitable configuration of its articulating components. In particular, the robot should stably reside on the terrain as allowed by the contact of its main, *non-articulating* tracked surface, although it might be possible to stably reside within a terrain region by an *acrobat*

Algorithm 2: Sub-state computation of rotations of articulating components

Input: M : DEM of the terrain
 ugv : 3D kinematic model of the vehicle
 $s_{pose} = (x, y, z, \phi, \theta, \psi)$: Stable position and orientation of the UGV chassis within M
 $\Delta\alpha$: Rotation increment
 Δt : Simulation time increment
Output: $s_{art} = (\alpha_1, \alpha_2, \dots, \alpha_n)$

```

begin
  Robot_stable = false, stop_comp[] = false
  while Robot_stable = false do
    for k = 1 to n do
      if stop_comp[k] = false then
        if collision(ugv.comp[k], M, t) = false
           $\wedge \alpha_k + \Delta\alpha \leq lim_k$  then
             $\alpha_k = \alpha_k + \Delta\alpha$ 
          else
            stop_comp[k] = true
          end
        end
      end
    end
    if all(stop_comp) = true then
      Robot_stable = true
      t = t +  $\Delta t$ 
    end
  end
end

```

robot pose where the articulating components raise the robot chassis above the ground. Such a pose, however, could result into a state with severely limited mobility, let alone the exertion of very high forces on the contact points that could undesirably stress the robotic components.

To increase the stability of the robot we adjust its articulating components until they come in contact with the terrain. Hence, stability is increased by augmenting the size of the robot footprint and reducing the chance of tip-over. In parallel, we take care not to violate the robot’s kinematic constraints, which could restrict the rotation limits of the articulating components. If a constraint is met before the component comes in contact with the terrain, then we set the rotation of the respective component to the prescribed limit.

In Algorithm 2 we describe the respective steps that are taken for adjusting the articulating components while at the bottom row of Figure 4, we show snapshots taken during the execution.

3.3. Static 3D traversability cost assessment

An instructive set of cost criteria that we consider in order to assess the optimality of a given static state are the following:

- **Ground Clearance;** the minimum distance between the centre of the robot frame to the terrain below it.
- **Robot orientation;** The roll/pitch of the robot frame with respect to the world frame.
- **Force-angle stability measure [PR96];** the minimum angle required to tip over the vehicle, between the gravitational net force and a tip-over axis normal.

While other criteria could be considered as well (e.g. the Zero Moment Point (ZMP) distance [VB04], traction efficiency or terrain features), the basis of the proposed model for traversability assessment is the usage of the minimal and most common set of perceptual capabilities, where the robot has knowledge of the 3D terrain surface that it resides on as well as its relative pose. In this perspective, we generalize the applicability of the proposed framework by considering only criteria that involve distances and geometric processing that can be assumed as readily applicable given the 3D model of the UGV and that of the terrain surface.

In detail, we augment the standard formulations of *ground clearance*, *orientation* and *angle stability margin* in order to obtain more reliable estimates, as explained in the following subsections. In Figure 6 we provide a picture of the stabilized robot annotated with geometry information to assist in the comprehension of the traversability costs computation.

3.3.1. Ground clearance

We introduce an improved way to estimate the ground clearance of the vehicle, namely, its distance from the ground. Typically this criterion is computed by simply measuring the Euclidean distance from the centre of the vehicle's frame to the ground. The drawback of this approach is that completely different shaped terrain surfaces could give the same estimate of ground clearance, despite the difference in the roughness of the terrain and the particular 3D shape of the robot itself. This in turn could result in inconsistent traversability assessments.

To alleviate this problem, we adopt the *directed Hausdorff distance* from the UGV base to the terrain, in order to obtain a measure of ground clearance that accounts for the shape of the vehicle and the terrain below it, hence, being more

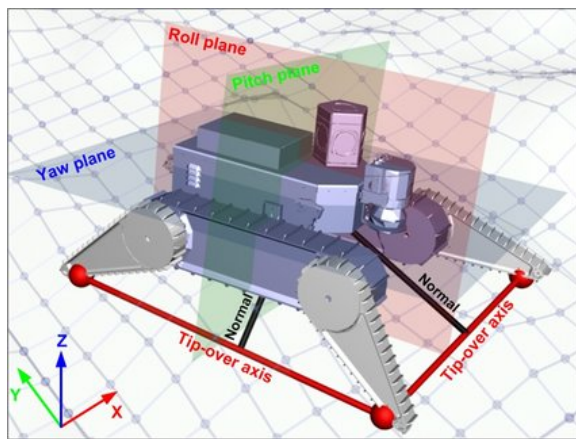


Figure 6: Illustration of the stabilized robot together with coordinate frames, collision points (red spheres), tip-over axes and corresponding normals.

consistent. We define a vehicle's ground clearance cost G at a given state \mathbf{s} as:

$$G = d_h(R, M_{vic}) = \sup_{\mathbf{r} \in R} \inf_{\mathbf{m} \in M_{vic}} d(\mathbf{r}, \mathbf{m}) \quad (1)$$

Here, R corresponds to the set of points of the 3D robot base facing the terrain, M_{vic} to the points of the DEM in the vicinity of the robot and $d(\mathbf{r}, \mathbf{m})$ denotes the Euclidean distance between any two points $\mathbf{r}, \mathbf{m} \in \mathbb{R}^3$. In order to derive an estimate that is not biased to the underlying mesh tessellation, a uniformly distributed set of points representing the 3D mesh of the UGV is used.

3.3.2. Robot orientation

The traversability assessment continues by accounting for the cost induced due to the inclination of the vehicle, with respect to the world frame. Here, we augment the standard formulation by considering not only the roll and pitch, but also the drift in the yaw angle of the vehicle that could be a side effect of the preceding stabilization process (see Section 3.1). We argue that it should be considered mandatory to take into account this effect, since drifting away from the direction of a prescribed path plan could result into the execution of corrective actions and potentially additional planning, overall hampering the success of the initial path plan. In this context, we define the robot's orientation cost O at a given state \mathbf{s} as:

$$O = \frac{o_r(1 - \cos(\phi)) + o_p(1 - \cos(\theta)) + o_y(1 - \cos(\Delta\psi))}{o_r + o_p + o_y} \quad (2)$$

Here, $\Delta\psi$ corresponds to the drift in the yaw angle between the initial angle prescribed by the path planner and the final angle after the completion of the stabilization process. The o_r, o_p and o_y factors denote the significance of each angle and together with certain thresholds that might be applied, they are specific to the robot and the application. The cosine function is used in order to apply a non-linear weighing on the individual angles of the vehicle by increasingly penalizing bigger rotations of the robot and in parallel ensure an equal treatment for negative and positive angles.

3.3.3. Angle stability

We compute a function of the force-angle stability margin [PR96], that penalizes more the tip-over angle as it approaches to zero, instead of using the distance stability margin [MF68] that is not scale invariant. In detail, we consider the robot's angle stability cost A at a state \mathbf{s} as:

$$A = 1 - \sin(\min(\gamma_i)) \quad (3)$$

Here, γ_i is the angle between the gravitational vector \mathbf{g} , emanating from the centre of mass of the robot and the tip-over axis normal \mathbf{n}_i of the i^{th} tip-over axis. Under this formulation, if the minimum tip-over angle becomes negative (i.e. in the event of tip-over) then A will be greater than 1. The magnitude of the force that is exerted along the normal is

omitted here, as our approach is based solely on geometric computations requiring only the 3D surfaces of the robot and the terrain and their relative position.

It should be noted that the approach followed in [PR96] where the convex support polygon is extracted by projecting the contact points onto the horizontal plane does not apply in our case for two main reasons. First, this is due to the fact that we are dealing with *non-planar* terrain and therefore, tipping over could be preceded by collision of the side of the vehicle with the terrain. And second, the contact support points do not in general lie on the same plane, due to the articulating components that are touching the terrain.

The first issue can be considered to render eq. (3) as a pessimistic, worst case measure wherein the robot would not encounter a collision on its side when tipping over. To deal with the second issue we extract the tipping-over axes by sequentially considering as support points the 3D positions of the contact points of the articulating components with the terrain and extract the tip-over axes independently.

4. EXPERIMENTS

A simulated model of a state-of-the-art search and rescue robot (shown in Figure 7) named ΤΑΛΟΣ (TALOS) [†] has been employed [Blu11] for the experiments. Our primary interest in evaluating the proposed methodology concerns its efficiency in order to determine the extent of its applicability in real-time, hence within a real mission.



Figure 7: Urban Search and Rescue robot TALOS.

4.1. Experiment setting

We performed our experiments using a computer equipped with an Intel Core i7 CPU 860 @ 2.8 Ghz and an NVIDIA GeForce GTS 250 graphics card. For the implementation of the physics-based simulation we have used the Open

[†] TALOS is a tracked mobile robot with two passive, tracked bogies at the sides and four active tracked flippers placed at the front and rear. It is equipped with active and passive sensors, an inertia measurement unit, GPS and an on-board computer.

Dynamics Engine (ODE) [Smi] mainly because it is integrated into the Robot Operating System [QCG*09] and allows the connection of the simulated functionality to the various robotic components. The marginal linear and angular velocity for the Algorithm 1 to converge were both set to 0.0025, the world step Δt for advancing the simulation to 0.015, the friction coefficient to $\mu = 50$, surface bouncing to 0.001 and the remaining world parameters were set to their default values.

4.2. Time efficiency

The cumulative results correspond to a total number of 5000 random runs of the complete framework. Since we are using fine meshes of the terrain and the UGV (≈ 4000 polygons) and compute their interaction, an increase in computation time would be reasonable compared to most approaches that perform ordinary convolutions of a 2D polygon-shaped robot footprint with a DEM. However, by measuring the computation time for the stabilization process (Section 3.1, 3.2) as well as for the traversability cost computations (Section 3.3) we obtained that none of these steps appeared to have a dependency on terrain complexity. Instead, relatively constant computation times were attained that are summarized in Table 1. We view this as a positive feature of the proposed framework, since this implies that the computation time only depends on the selected time resolution Δt of the iterative stabilization process, the 3D mesh resolution and the computer hardware.

	Stabilization	G	O	A
Time (msec)	5	0.6	0.0003	0.0008

Table 1. Average computation times for Stabilization and computation of the Ground clearance (G), Orientation (O) and Angle-stability (A) costs.

The reported timings were measured by disabling the rendering part of the physics-based simulation, since for the robot it is not necessary to visualize the stable state but only retrieve the corresponding parameters. Based on the measured, average timings reported on Table 1, the robot could easily assess both the stable state and the traversability costs of hundreds of terrain patches in real-time.

4.3. Mobility regression

In this Section, we investigate the relation of the different optimality criteria that we employ to quantify the robot's mobility, with respect to the terrain complexity.

To generically characterize terrain complexity, we have chosen *terrain slope* and *roughness* as two features/dimensions that can characterize a terrain in question and based on which the physics-based simulation can regress the mobility cost for the robotic vehicle. In Figure 9 we show

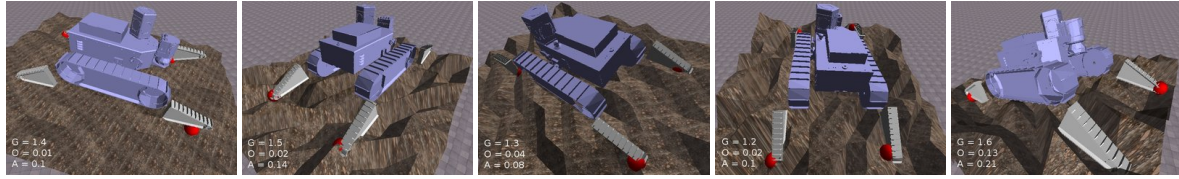


Figure 8: Example stabilization results and traversability cost assessments under different conditions of terrain roughness and slope. The contact points of the active flippers with the terrain are depicted as red spheres.

the traversability cost estimation for G , O and A ranging from a horizontal, even terrain (minimum complexity) to a maximally inclined terrain of 25° in both roll and pitch, with a roughness of 20cm in terms of standard deviation from a best-fit plane (maximum complexity). Here, the threshold limit for the highest allowable terrain complexity has been set according to the mobility capabilities of the *TALOS* robotic vehicle. To facilitate the visualization of the results, in the diagrams of Figure 9 we appoint a single terrain complexity dimension to the horizontal axis, where *slope* and

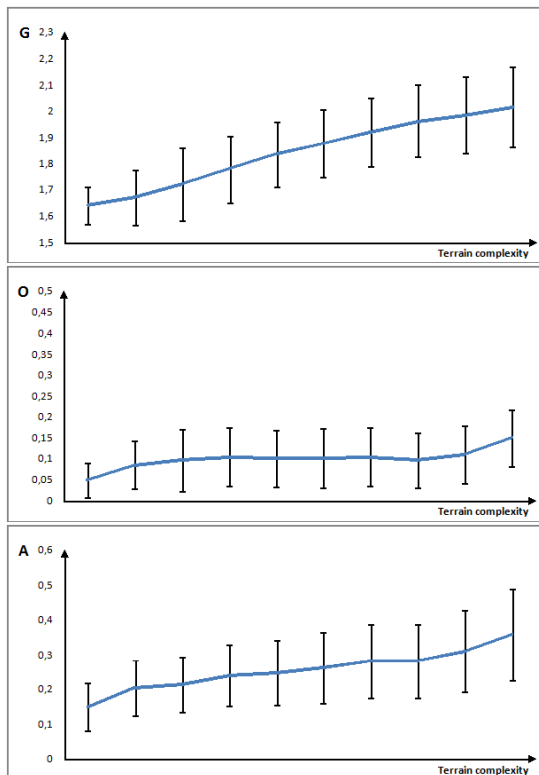


Figure 9: Traversability costs as a function of terrain complexity. From top to bottom, we show the evolution of the Ground clearance G , Orientation O and Angle-stability cost A . The blue lines correspond to the regressed traversability.

roughness have first been normalized to the $[0, 1]$ interval and the final terrain complexity dimension is set as the sum of the two normalized features.

We can observe that the ground clearance cost (top of Figure 9) is regressed almost as a linear function of the terrain complexity and since it is not related to the slope of the terrain, it has a rough linear dependence on the terrain roughness. However, there is significant deviation in the actual value of the G cost, a direct effect of the usage of the directed Hausdorff distance from the UGV to the terrain accounting for the terrain roughness, that would otherwise be transparent if the standard formulation of the ground clearance was used. By looking at the early evolution of the Orientation cost (middle of Figure 9) we can further clearly derive that the deviation of the real data from the regressed trend curve is initially low but as terrain complexity increases, the deviation of the real data increases. This is a direct effect of considering the $\Delta\psi$ factor in eq. (2), i.e. the drift that occurs in the prescribed yaw angle as a result of stabilization. This is indicative of the effect that this drift could have in path planning as terrain complexity increases and could be seen as a gross estimate of track slippage.

In Figure 8 a number of representative examples are given that show the stabilized robot on various terrains, together with the respective traversability costs and finally, in Figure 10, we show the estimated stable state of the UGV for common terrain classes that are encountered in USAR scenarios such as inclined planes and steps.

With respect to the overall utility, the proposed physics-based simulation effectively serves the goal of real-time assessment of stable poses of the UGV and the subsequent computation of heuristic functions (the traversability costs) that guide a high-level, path planning algorithm. The real-time performance requirement, together with constraints on the available computing power on-board the UGV, however, essentially implies that a high-fidelity physics-based simulation of the motion trajectory of the UGV using its tracks along a sequence of poses, still remains a challenge. The proposed framework implements the initiating stage of motion planning, namely, learning the mobility skills of the robot in rough terrain and allowing the regression of robot mobility given two characteristics, namely, terrain slope and roughness. In order to safely and proactively control the actual

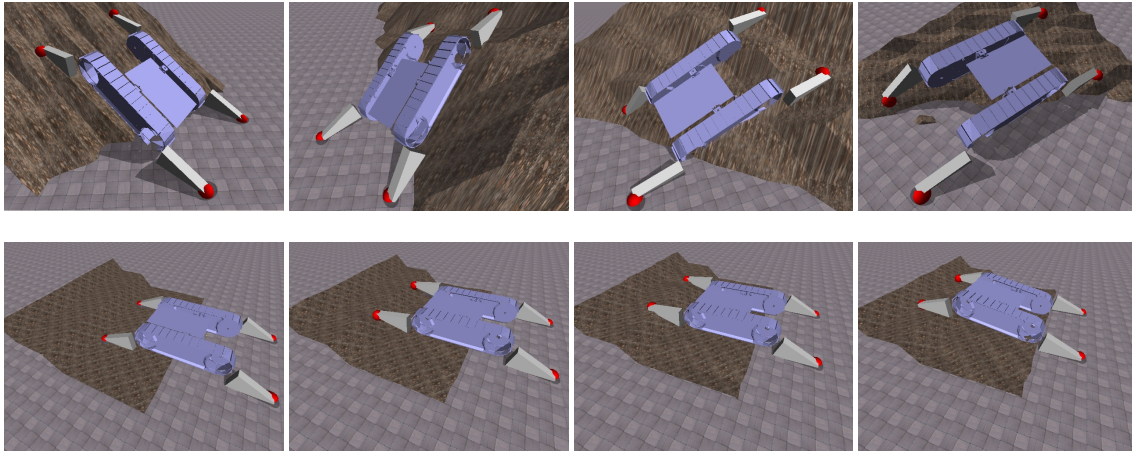


Figure 10: Stabilized robot states for commonly encountered USAR terrains; Top row: Inclined plane, Bottom row: Steps.

robotic vehicle in following a sequence of stable states, more elaborated processing should succeed accounting for execution failures and uncertainty.

5. CONCLUSIONS

We have proposed a methodology to estimate the optimal state of a tracked mobile robot with articulated components in terms of stability, upon a 3D terrain patch. Our approach has been based on a minimal set of perceptual robot capabilities, namely, knowledge of the 3D shape of the terrain surface, the 3D shape of the robot itself and the frame relationship between the two shapes. Based on the estimation of the optimal state of the robot through a physics-based simulation, we compute a number of static 3D traversability cost criteria that we have reformulated to account for fine representations of the robot and the terrain in order to derive more consistent cost estimations.

We have evaluated the applicability of the proposed methodology by using a model of a state-of-the-art search and rescue robot and performed an extensive number of runs, testing its efficiency under varying terrain complexity and proving its applicability in real-time.

6. Acknowledgment

The work presented in this paper describes research performed as part of the EU-FP7 ICT 247870 NIFTI project.

References

- [BLS01] BONNAFOUS D., LACROIX S., SIMEON T.: Motion generation for a rover on rough terrains. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2001), pp. 784–789. 2
- [Blu11] BLUEBOTICS: Mobile robot. No. PCT/EP2011/060937. 2011. 4, 7

- [CB08] CHONNAPARAMUTT W., BIRK A.: Robocup 2007: Robot soccer world cup xi. 2008, ch. A Fuzzy Controller for Autonomous Negotiation of Stairs by a Mobile Robot with Adjustable Tracks, pp. 196–207. 2
- [Gen99] GENNERY D.: Traversability analysis and path planning for a planetary rover. *Autonomous Robots* 6 (1999), 131–146. 2
- [GMM02] GOLDBERG S., MAIMONE M., MATTHIES L.: Stereo vision and rover navigation software for planetary exploration. In *IEEE Aerospace Conference* (2002). 2
- [GPPP11] GIANNI M., PAPADAKIS P., PIRRI F., PIZZOLI M.: Awareness in mixed initiative planning. 2
- [HAM09] HELMICK D., ANGELOVA A., MATTHIES L.: Terrain adaptive navigation for planetary rovers. *Journal of Field Robotics* 26 (2009), 391–410. 2
- [HGF08] HOWARD T., GREEN C., FERGUSON D., KELLY A.: State space sampling of feasible motions for high-performance mobile robot navigation in complex environments. *Journal of Field Robotics* 25, 6-7 (2008), 325–345. 4
- [HJC*08] HUNTSBERGER T., JAIN A., CAMERON J., WOODWARD G., MYERS D., SOHL G.: Characterization of the roams simulation environment for testing rover mobility on sloped terrain. In *International Symposium on Artificial Intelligence, Robotics, and Automation in Space* (2008). 2
- [HK07] HOWARD T., KELLY A.: Optimal rough terrain trajectory generation for wheeled mobile robots. *International Journal of Robotics Research* 26 (2007), 141–166. 2, 4
- [INY11] ISHIGAMI G., NAGATANI K., YOSHIDA K.: Path planning and evaluation for planetary rovers based on dynamic mobility index. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2011). 3
- [KK92] KWEON I. S., KANADE T.: High-resolution terrain map from multiple sensor data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 1 (1992), 278–292. 2
- [KPG*12] KRUIJFF G.-J., PIRRI F., GIANNI M., PAPADAKIS P., PIZZOLI M., SINHA A., PIANESE E., CORRAO S., PRIORI F., FEBRINI S., ANGELETTI S., TRETYAKOV V., LINDER T.: Rescue robots at earthquake-hit mirandola, italy: a field report. In *IEEE International Symposium on Safety, Security, and Rescue Robotics* (2012). 1

- [LF09] LIKHACHEV M., FERGUSON D.: Planning long dynamically feasible maneuvers for autonomous vehicles. *International Journal of Robotics Research* 28 (2009), 933–945. 4
- [LRH94] LANGER D., ROSENBLATT J., HEBERT M.: A behavior-based system for off-road navigation. *IEEE Transactions on Robotics and Automation* 10, 6 (1994), 776–783. 2
- [ME85] MORAVEC H., ELFES A.: High resolution maps from wide angle sonar. In *IEEE International Conference on Robotics and Automation* (1985). 2
- [MF68] MCGHEE R., FRANK A.: On the stability properties of quadruped creeping gaits. *Mathematical Biosciences* 3, 0 (1968), 331 – 351. 7
- [ONY*11] OKADA Y., NAGATANI K., YOSHIDA K., TADOKORO S., YOSHIDA T., KOYANAGI E.: Shared autonomy system for tracked vehicles on rough terrain based on continuous three-dimensional terrain scanning. *Journal of Field Robotics* 28, 6 (2011), 875–893. 2, 3
- [PP11] PAPADAKIS P., PIRRI F.: Consistent pose normalization of non-rigid shapes using one-class support vector machines. In *IEEE Computer Vision and Pattern Recognition, Workshop on Non-rigid Shape Analysis and Deformable Image Alignment* (2011), pp. 23–30. 4
- [PR96] PAPADOPOULOS E., REY D.: A new measure of tipover stability margin for mobile manipulators. In *IEEE International Conference on Robotics and Automation* (1996). 5, 7
- [QCG*09] QUIGLEY M., CONLEY K., GERKEY B., FAUST J., FOOTE T., LEIBS J., WHEELER R., NG A.: Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software* (2009). 8
- [SGS05] SCHOLKOPF B., GIESEN J., SPALINGER S.: Kernel methods for implicit surface modeling. In *Advances in Neural Information Processing Systems* (2005), pp. 1193–1200. 4
- [Smi] SMITH R.: Ode, <http://www.ode.org/>. 8
- [SSS*00] SINGH S., SIMMONS R., SMITH T., STENTZ A., VERMA V., YAHJA A., SCHWEHR K.: Recent progress in local and global traversability for planetary rovers. In *IEEE International Conference on Robotics and Automation* (2000). 2
- [TD99] TAX D., DUIN R.: Support vector domain description. *Pattern Recognition Letters* 20, 11-13 (1999), 1191 – 1199. 4
- [VB04] VUKOBRATOVIC M., BOROVAC N.: Zero-moment point - thirty five years of its life. *International Journal of Humanoid Robotics* 1, 1 (2004), 157–173. 6
- [VDH06] VANDAPEL N., DONAMUKKALA R. R., HEBERT M.: Unmanned ground vehicle navigation using aerial ladar data. *The International Journal of Robotics Research* 25, 1 (2006), 31–51. 2