Design and comparative evaluation of an iterative contact point estimation method for static stability estimation of mobile actively reconfigurable robots

Michael Brunner\textsuperscript{a,}\textsuperscript{*}, Torsten Fiolka\textsuperscript{a}, Dirk Schulz\textsuperscript{a}, Christopher M. Schlick\textsuperscript{a,b}

\textsuperscript{a} Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE, 53343 Wachtberg, Germany
\textsuperscript{b} Institute of Industrial Engineering and Ergonomics (IAW), RWTH Aachen, 52062 Aachen, Germany

HIGHLIGHTS

- We present a new robot pose prediction method for static stability estimation.
- The method approximates the terrain by least-squares planes to reduce the runtime.
- A stochastic version accounts for noise in the robot state and the terrain model.
- We systematically compared it with a physics simulation in many distinct scenarios.
- The new method is significantly faster and competitive in realistic situations.

ARTICLE INFO

Article history:
Received 3 April 2014
Received in revised form
20 August 2014
Accepted 5 September 2014
Available online 16 September 2014

Keywords:
Contact point estimation
Static stability
Reconfigurable robots

ABSTRACT

Due to the advancements of robotic systems, they are able to be employed in more unstructured outdoor environments. In such environments the robot–terrain interaction becomes a highly non-linear function. Several methods were proposed to estimate the robot–terrain interaction: machine learning methods, iterative geometric methods, quasi-static and fully dynamic physics simulations. However, to the best of our knowledge there has been no systematic evaluation comparing those methods.

In this paper, we present a newly developed iterative contact point estimation method for static stability estimation of actively reconfigurable robots. This new method is systematically compared to a physics simulation in a comprehensive evaluation. Both interaction models determine the contact points between robot and terrain and facilitate a subsequent static stability prediction. Hence, they can be used in our state space global planner for rough terrain to evaluate the robot’s pose and stability. The analysis also compares deterministic versions of both methods to stochastic versions which account for uncertainty in the robot configuration and the terrain model. The results of this analysis show that the new iterative method is a valid and fast approximate method. It is significantly faster compared to a physics simulation while providing good results in realistic robotic scenarios.

© 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/).

1. Introduction

As the mobility capabilities of robots continue to evolve, they are employed in more challenging terrains. New applications arise and require the adaption of previous methods and the development of new algorithms to be able to navigate safely through cluttered, uneven and rough environments.

The space robotics community is investigating advancements in rough terrain traversal to enhance the supervisory control of planetary rovers. Avoiding hazardous situations and predicting the robot mobility are of special concern. Failure to do so could jeopardize an entire multi-billion dollar mission. Rescue robotics also depends on the navigation of mobile robots in rough and unstructured environments. The situations at disaster sites usually require robots with enhanced mobility capabilities. The robots are typically deployed to assist first responders in areas where it is too dangerous for humans.

A recent catastrophic event of this kind is the Tohoku earthquake in Japan in 2011 which led to a nuclear meltdown at the...
nuclear power plant of Fukushima Daiichi. Instead of deploying humans, several different reconfigurable robots, among others iRobot Packbots, were teleoperated to assess the situation in contaminated areas [1,2]. In an other incident in May 2012 two major earthquakes in northern Italy caused many casualties and severe damage to a large number of historical buildings. In a human–robot teaming project, reconfigurable mobile robots were used to assess the structural integrity of the buildings [3] since it was too risky for human rescue personnel to enter those sites.

As traversing rough terrain requires robots with enhanced mobility capabilities, usually reconfigurable robots are employed in such applications. Field reports point out that the operators are quickly exhausted by the demanding task of teleoperating reconfigurable robots in rugged environments [4]. A possible solution is to assist operators through suggesting suitable paths in order to increase their situation awareness and reduce mental workload [5].

Safe and efficient motion planning in rough terrain needs a sound analysis of the robot mobility and the terrain traversability. Inclined surfaces increase the risk of tip-over and perturb the motion dynamics. The system’s safety in terms of stability may be estimated by evaluating the geometric shape of the terrain, while identifying roughness and mechanical properties of the terrain may assist in providing dynamically stable paths. In any case, modeling the robot–terrain interaction and determining the 6D robot position and the robot stability given its configuration and the terrain geometry is an essential part in rough terrain navigation. While during navigation sensor readings, e.g. from an inertial measurement unit (IMU), can be used to estimate the robot dynamics as well as its position and attitude, a robot–terrain interaction model is required for an offline global planner in order to provide suitable and feasible plans.

In this paper, we present a new iterative method to estimate the contact points of actively reconfigurable robots (Fig. 1). The method requires the robot’s actuator configuration and an approximate terrain model. It determines the robot’s contact points which are used to estimate the static stability of the system. Further, we present a systematic comparison with a quasi-static physics simulation with respect to accuracy and run-time. This analysis involves a deterministic and a stochastic version of both approaches. The stochastic versions account for the uncertainties in the robot configuration and the terrain model. The evaluation provides insight in the performance of these approaches in a variety of different conditions. The experiment series show that the new method provides competitive estimates compared to a physics simulation in many scenarios, especially and more importantly in realistic situations. It is also significantly faster than a physics simulation. Since, the stability of the robot must be evaluated numerous times during planning, a fast robot–terrain interaction model is essential to provide safe paths in a reasonable amount of time.

This paper is organized as follows: first, a review of related work provides an overview of the previous work in the area of robot–terrain interaction and stability analysis (Section 2). Section 3 describes the quasi-static physics simulation using the Open Dynamics Engine (ODE) and the newly developed iterative geometric pose estimation method for actively reconfigurable robots. Subsequent sections discuss the stability criterion (Section 4) and the stochastic framework (Section 5). The experimental results are shown in Section 6. We conclude in Section 7.

2. Related work

For applications in 2D navigation typically a binary notion of traversability based on height differences is employed [6,7]. An area is either an obstacle and not traversable or it is free and can be traversed. A common approach in 2D navigation is to penalize proximity to obstacles due to the increased chance of collisions [8,9]. However, since in 2D navigation applications the robot’s stability is usually satisfied, it is not evaluated most of the time.

As the terrain becomes more challenging and the navigation involves the traversal of rough areas, a binary traversability assessment based on height differences does not suffice any more. Especially with reconfigurable robots, traversability becomes a function of the terrain shape, the roughness and mechanical properties of the terrain and the robot configuration. Hence, multivariate robot–terrain interaction models to evaluate stability are required for ground-based robots in rough environments. Analyzing the robot–terrain interaction is fundamental for rough terrain navigation. Consequently, this area of research has received and still continues to receive significant attention.

2.1. Robot–terrain interaction

The robot–terrain interaction becomes a non-linear function depending on the type of locomotion, possible actuators and the shape and parameters of the terrain. Inclined surfaces change the dynamics of the robot. Additionally, slip may occur more often in rough terrain. Hence, traction and propulsion are more dominant issues than they are on flat ground. Modeling the robot–terrain interaction is, therefore, important and is addressed in several ways. Due to low speeds when traversing rough terrain, some planners assume sufficient traction and propulsion and concentrate on finding safe and stable paths in a quasi-static examination. On the contrary, physics-based models of the robot and the terrain can be derived and accurate predictions of the state variables can be performed. However, without intimate knowledge about the robot design and the terrain those models are very difficult to generate.

To approximate difficult non-linear functions, machine learning is a popular technique. In machine learning identifying a set of features which encode the necessary information to enable learning is crucial to the performance. Also generating a representative training set allowing generalization without overfitting is important. In [10] a supervised learning technique, called behavior cloning, for a tracked articulated robot traversing rough terrain is introduced. The algorithm learns which action to apply given a set of terrain features and human expert demonstrations. Terrain features are extracted from segmented regions in front of the robot, including the region height, roll and pitch as well as location and magnitude of the largest deviation [11]. [12] extended this technique by learning the value of an action instead of the situation–action mapping. This enables learning also from negative examples and increases the overall performance. In addition, an automated instructor which allows training from simulation rather than from human expert demonstrations reduces the training effort significantly [12,13].

The robot–terrain interaction can also be estimated through detailed mechanic and dynamic models. [14] uses the Open Dynamics Engine (ODE) to include physics and mechanics of the robot, and geometry and material properties of the terrain to perform a forward simulation of the robot’s path. Howard et al. propose a model-predictive trajectory generator which involves several comprehensive models for capturing robot–terrain interaction dynamics [18]. The algorithm models the system dynamics, kinematics, suspension and propulsion with respect to the terrain. Therefore, the kinematic constraints of the robot’s mobility system, the robot’s elevation and attitude as well as wheel slip and actuator dynamics are taken into account when generating trajectories for a wheeled mars rover. In [19] this model-predictive method was also applied to an articulated wheeled mars rover to level out the rover chassis and increase its stability in sloped terrain.

1 The Open Dynamics Engine [15] uses equations of motions which are derived from a Lagrange multiplier velocity-based model according to Stewart/Trinkle [16] and Anitescu/Potra [17].
Approximation of the robots reaction to different terrain conditions through a lookup function is proposed in [20] to enable fast simulation for a real-time application. The kinodynamic robot–terrain interaction is encoded in a lookup function constructed by recording the systems velocity and acceleration bounds for several slopes. On the basis of this approximate model a path through the environment is determined.

Iagnemma et al. present another approach to robot–terrain interaction by improving the prediction of the robot mobility [21]. The online proproceptive algorithm estimates the terrain cohesion and internal friction angles from vertical load and the wheels' torque, sinkage as well as angular and linear velocities. This way the interaction between robot and terrain can be considered during navigation. The estimation algorithm was extended to additional terramechanical parameters, among which are sinkage exponent and shear modulus, using telemetry data [22].

Without detailed knowledge about terrain parameters or mechanical properties of the robot and a forward simulation of the path, some approaches favor an approximation of the robot–terrain interaction through heuristics and evaluations of quasi-static configurations. Hait et al. concentrate on the geometric shapes of the robot and the terrain when they minimize the total energy to estimate the robot's contact points and pose [23]. The placement model considers the uncertainty in the robot's pose and the terrain and performs a worst-case analysis. This approximate model is used by the planner to find paths through rough environments. Inclined areas increase the uncertainty and, thus, are penalized by the algorithm. In [24] an iterative process determines the placement of the robot in rough terrain. Given a geometric terrain model, the algorithm iteratively identifies the placement of each axis and the corresponding wheels for a 3-axis rover with passive suspension. The resulting configuration is associated with a risk value to enable the planner to select safer paths.

Without considering slip and traction or other disturbances related to terrain parameters explicitly, physics-based simulation models are used to estimate the robot's pose given the robot's geometric configuration and a shape model of the terrain. A quasi-static approach is chosen by [25] to determine the orientation and elevation of a tracked robot with four articulated flippers (BlueBotics Absolom robot) in rough terrain. The pose determination is a two-step process: first, the robot's attitude is estimated using the Open Dynamics Engine (ODE). This step only involves the chassis of the robot without the flippers. Robot poses which require the flippers to be stable are discarded due to low mobility. The flippers are adjusted in the second phase to increase stability. Norouzi et al. introduce an algorithm to plan stable paths for an iRobot Packbot robot while maximizing the elevation of a sensor head to increase sensor coverage for exploration [26]. In contrast to [25], the complete robot configuration is used to estimate the contact points between the non-convex shaped robot with the terrain using also an ODE simulation. In an iterative process the gravitational and reaction forces are considered until four contact points are identified. If the pose is based on less than four contacts, or collides with the chassis, or deviates far from the original position, i.e., is slip prone, it will be discarded. To account for unmodeled or unexpected disturbances the uncertainty of the robot pose and the terrain are considered in a stochastic stability analysis [27].

2.2. Stability analysis

While other quantities may result in sub-optimal behavior if falsely estimated, a wrong stability assessment could result in tip-over and damage to the robot. Therefore, the system's stability is one of the most important criteria for rough terrain motion planning. Stability margins can be divided into static and dynamic measures. The appropriate choice of a stability margin depends on the system and its application as there is no optimal stability margin for all applications [28]. Most stability measures originate from walking robots as those are the first who where concerned about stability.

Stability measures such as the popular Zero Moment Point (ZMP) method [29], consider dynamic effects due to inertia, momentum or elasticity. The ZMP is the point on the supporting polygon where the moment due to the terrain–reaction forces and moments is zero. The ZMP is often used for legged robots [30–32]. It assumes the supporting polygon to be confined in a plane. Hence, it is not suited for rough terrain [33]. The Force–Angle Stability Margin (FASM) [34] measures the angle between the resulting force acting from the center of gravity on the ground and the normal to the edges of the supporting polygon. If all angles are positive, the system is considered dynamically stable. The FASM is a valid criterion for rough terrain. It is a often applied to quantify the dynamic stability of robotic machines.

A simple way to measure the static stability of a system is to project the center of gravity onto the supporting polygon [35]. If the projection lies within the supporting polygon, the system will be statically stable. However, this stability criterion does not consider the height of the center of mass. The Energy Stability Margin (ESM) is described as the minimal potential energy necessary to tip the robot around the edges of the supporting polygon [36]. Hirose et al. normalized the ESM to the robot's weight resulting in the Normalized ESM (NESM) [37]. The NESM is the most adequate static stability margin [37,28] and is also widely used in robotic applications.

Norouzi et al. use a quasi-static FASM formulation for a statistical evaluation of the robot's stability [27]. The analysis considers the uncertainty in the robot's configuration, the robot's position and the environment model and estimates the resulting probability distribution. This distribution is used to predict the robot's contact points with the environment, its center of mass and subsequently the robot's stability in the presence of noise.

Many different approaches to stability estimation as well as modeling the robot–terrain interaction are developed and applied in a variety of scenarios. However, besides the work of Garcia et al. [28], which compares stability margins for walking machines, there is no comparison of stability computations or robot–terrain interaction models for mobile ground robots. The systematic comparison of two types of robot–terrain interaction models in this paper is one step to reduce this lack.
3. Robot–terrain interaction models

There exist four major approaches to estimate the robot–terrain interaction: machine learning, full dynamic physics simulation, quasi-static physics simulation and iterative geometric methods. However, to the best of the authors’ knowledge, there is no systematic comparison between such methods in terms of static stability.

Machine learning methods need a training phase and a set of typical terrain features which are hard to determine for a variety of rough terrains. A full dynamics simulation requires sufficiently accurate mechanical models of the robot and the terrain to achieve the desired simulation accuracy. Among other parameters, this involves softness and friction coefficients. It is difficult to estimate these parameters for rough terrain.

Therefore, we decided to concentrate on comparing two methods from the remaining categories: a self-developed iterative geometric procedure and a state-of-the-art quasi-static physics simulation for estimating the contact points of actively reconfigurable robots. Both methods require a geometric model of the robot and a terrain model. Given a state which includes the 2D-position \((x, y)\), the orientation \(\theta\) and the actuator angles \(a_1, \ldots, a_n\) of the robot, the algorithms predict the robot pose which results from the interaction with the terrain. Hence, they determine the height \(z\) and the roll \(\phi\) and the pitch \(\psi\) angle of the robot. The physics simulation is more accurate as it considers the actual shape of the terrain. The iterative geometric method uses an approximate terrain model which makes it less accurate but allows a significantly faster pose estimation.

### 3.1. Iterative geometric method

When planning paths typically flatter and less rougher areas are preferred if possible. Such paths reduce the risk of tip-over or any other damage, thus, generally increase the safety of the system. Consequently, in most robotic applications including rough terrain navigation, the robots mainly drive on fairly smooth surfaces. It is only rarely required to navigate through rougher areas of the terrain. Therefore, we developed a new iterative contact point estimation method for actively reconfigurable robots. It is based on a geometric model of a robot and a simplified environment model.

The terrain below the robot is approximated by a least-squares plane. As long as the residual sum of squares is low, this procedure is very accurate. However, the larger the residuals, the less accurate the method becomes. Our previous work has shown that this approach provides a sufficiently accurate approximation [41–43] for typical rough terrain. Additionally, it reduces the computation time required to estimate the contact points between robot and terrain significantly.

The contact point estimation (Algorithm 1) starts with a leveled robot orientation and determines the rotation which the robot configuration introduces iteratively. The algorithm computes the potential contact points \(CP_r\) and the center of mass \(CM_r\) based on the geometric robot model in the robot coordinate system (lines 5 and 6). Given the robot’s configuration and orientation, a procedure determines for each locomotion component (like wheels, tracks, flippers etc.) the lowest point as a contact candidate. The current implementation of the algorithm assumes four contact candidates. However, this is easily extendable to a larger number.

The robots considered in this work (Section 6.1) require at least three contact points to be stable. Hence, the algorithm determines all possible support triangles \(ST\) from \(CP_r\) and remembers the remaining candidate points (line 7):

\[
ST = \{ (T, C) : T \in P(CP_r) \land |T| = 3 \land C = CP_r \setminus T \},
\]

where \(P(\cdot)\) denotes the power set. If one of the remaining candidate points \(c \in C\) lies below the plane through the points of the triangle \(T\), the three contact candidates cannot result in a valid pose. If the robot assumes this pose on a plane, the candidate point \(c\) will penetrate the ground.

For each valid triangle we evaluate how well the triangle supports the robot. We use barycentric coordinates which are a coordinate system to specify a location of a point with respect to the center of a simplex, in this case a triangle. We project the triangle and the center of mass onto the XY-plane and test whether the center of mass lies within the triangle and determine its distance \(d\) to the center of the triangle (lines 8–13).

Let \(\mathbf{t}_i = (x_i, y_i, z_i)\), \(\mathbf{t}_1 = (x_1, y_1, z_1), \mathbf{t}_2 = (x_2, y_2, z_2)\) be the three corners of the support triangle and \(\mathbf{p} = (x, y)\) the center of mass, all projected onto the XY-plane. Then the barycentric coordinates \(\lambda_1, \lambda_2, \lambda_3\) are given by

\[
\begin{align*}
\lambda_1 &= \frac{(y_2 - y_3)(x - x_2) + (x_3 - x_2)(y - y_3)}{(y_2 - y_3)(x_1 - x_2) + (x_3 - x_2)(y_1 - y_3)} \\
\lambda_2 &= \frac{(y_3 - y_1)(x - x_1) + (x_1 - x_2)(y - y_3)}{(y_3 - y_1)(x_1 - x_2) + (x_1 - x_2)(y_1 - y_3)} \\
\lambda_3 &= 1 - \lambda_1 - \lambda_2.
\end{align*}
\]

It must hold \(\lambda_1 + \lambda_2 + \lambda_3 = 1\). If the point \(\mathbf{p}\) lies in the center of the triangle, it is \(\lambda_1 = \lambda_2 = \lambda_3 = \frac{1}{3}\). The distance to the center of the triangle is given by

\[
d = \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}.
\]

If \(0 \leq \lambda_i \leq 1\) for \(i = 1, 2, 3\) then the point \(\mathbf{p}\) resides within the triangle, otherwise it lies outside.

At this point of the algorithm, we are interested in finding the best support triangle. This does not necessarily require the resulting robot pose to be stable. Nevertheless, we prefer triangles which contain the center of mass according to the barycentric coordinates (inside = 1) and select the triangle with the minimal \(d\). If, however,
This process is performed until the rotation \( \phi \) distance to the triangle center and whether it lies within the triangle (line 14). Given \( \mathbf{s}_{\text{min}} \), the algorithm adjusts the current rotation \( \phi_{\text{config}} \) by the rotation \( \phi_{\text{plane}} \), the triangle plane introduces (line 15). This process is performed until the rotation \( \phi_{\text{config}} \) converges to some limit value (i.e. changes in rotation are below threshold \( \tau_{\phi} \)) or a maximum number of iterations is reached.

The resulting rotation represents the rotation of the robot body which is introduced by the robot’s configuration. For fixed chassis robots no iterative process is required; instead, the rotation can be determined by fitting a plane to the candidate points \( CP_X \) and measuring the induced rotation.

The function \( \text{detTerrainInclination}(\cdot) \) determines the most likely inclination of the terrain \( \phi_{\text{env}} \). Beforehand, several possible surface normals were computed with respect to different footprint sizes of the robot. Given the footprint size and the normal, the size of the inclined surface can be computed. The function selects the surface approximation whose size best fits the dimensions of the current robot configuration (lines 19–22). The robot dimensions are represented by the possible contact points \( CP_X \).

The triangle points \( T \) and the remaining contact candidates which are close to the triangle plane build the corners of the support polygon. Note, the remaining candidates lie above the plane since other constellations where excluded at the beginning. If remaining candidates are too far apart (>\( r \)), they are marked as invalid. Finally, the algorithm determines the contact points \( CP_X \) and center of mass \( CM_X \) in the global coordinate system. They are computed using the plane orientation \( \phi_{\text{env}} \) of the terrain approximation and the given robot state \( \mathbf{x} \) (lines 23–29). Fig. 2 illustrates the different stages of the IG algorithm.

### 3.2. Quasi-static physics simulation

Another way to approach the problem of estimating the contact points between the robot and the terrain is to simulate the objects’ interaction through a physics engine. Using the Open Dynamics Engine (ODE) we developed a simulation module to estimate the robot pose in a quasi-static fashion. The estimated pose and especially the resulting contact points between the robot and the terrain are subsequently used to evaluate the robot’s stability. In contrast to our method, this approach uses a triangular mesh to model the terrain. This is more accurate but also significantly more expensive to compute.

The idea of utilizing a quasi-static simulation to determine the robot pose in rough terrain was previously introduced by [27] as well as [25]. Our iterative procedure (Algorithm 2) uses ODE to perform simulation steps for a specified time step, after which the algorithm tries to find valid contact points. If not enough valid contacts are found, the built-up forces are reset and a new cycle is started continuing with the final pose of the previous iteration. This process is repeated until the robot assumes a stable pose, the simulated pose violates the operating limits, or a maximum number of iterations has been reached. If three or more contact points are identified, the stability of the final pose can be computed by a subsequent algorithm. Otherwise, no valid pose can be determined and the robot configuration is deemed unstable. The iterative procedure reduces the effect of momentum and friction as they are reset every cycle.

In more detail, the ODE-based contact point estimation (Algorithm 2) starts out with the robot being placed above the surface. Iteratively it simulates the robot’s interaction with the terrain for a duration \( \Delta t \). If the simulation results in an invalid state which violates the operating limits of the robot, the simulation is aborted (lines 10–13). The ODE collision points \( P_{\text{collisions}} \) constitute the set of contact candidates. If no collision points are detected, the minimal distance between robot and surface is computed. This distance is used to place the robot directly onto the surface. Before the next iteration starts, the forces of the robot object are reset (lines 15–19). If the simulated robot pose is valid and collision points are present, the algorithm tests the simulated velocities of the robot body. If they are greater than a constant \( \epsilon \), the robot has not assumed a stable pose at this iteration and the simulation continues after resetting the robot body forces. If, however, a stable pose is found, the simulation is done. In this case or if the maximum number of iterations is reached, the contact points \( CP_X \) and the center of gravity \( CM_X \) are computed (lines 4–7 and 23–26) in the global coordinate system. For each element of the locomotion system (tracks, flippers, wheels etc.) which causes collision points, the function \( \text{calcContactPoints}(\cdot) \) determines the outermost to be the associated contact point. \( CP_X \) and \( CM_X \) are used by a subsequent algorithm to estimate the system’s stability as described in Section 4.

### 4. Static stability evaluation

In this paper the static stability of the robot is considered because the robot’s speed is low when traversing hazardous areas. This approach is commonly viewed as appropriate [25,27,38]. It was also supported by our real world experiments. Small grained material like on the hill of rubble may cause slipping even at low speeds. However, tests with our Telemax robot, for example, showed that in the traversable areas of the hill slipping did not prevent path execution.
According to [28], the Normalized Energy Stability Margin (NESM) [37] is the more accurate static stability margin on inclined surfaces compared to the force–angle stability margin (FASM) [34]. Therefore, we use the NESM. In contrast to commonly used margins like the projection of the center of mass onto the supporting polygon, the NESM considers the actual position of the center of mass and directly provides a notion of quality. The NESM basically indicates the amount of kinetic energy required to tip the robot over the “weakest” edge of the supporting polygon. It is derived from the Energy Stability Margin (ESM) [36].

The Energy Stability Margin is depicted in Fig. 3. The rotation axis is given by $F_1$ and $F_2$ as a border of the supporting polygon within the plane $p$. $E$ is the vector from the border to the center of mass ($CM$). $\theta$ depicts the angle between $E$ and the vertical plane and $\psi$ the inclination of the rotation axis with respect to the horizontal plane. $E'$ is obtained by rotating $E$ around the rotation axis until it is contained in plane $p$. The quantity $h = |E|(1 - \cos \theta) \cos \psi$ provides the energy stability level.

$$h = |E|(1 - \cos \theta) \cos \psi. \quad (6)$$

The energy stability level of the $i$th rotation axis is provided by

$$e_i = m_{\text{total}} g h_i, \quad (7)$$

where $m_{\text{total}}$ is the system weight and $g$ the acceleration of gravity. Normalizing these energy levels to the system weight introduces the NESM

$$s = \min_i (h_i), \quad (8)$$

where $h_i$ represents the normalized energy level with respect to the $i$th boundary of the supporting polygon as defined in Eq. (6). Stable states have positive values, unstable states have negative values and the equilibrium is reached at zero. Another way to describe the normalized energy levels $h_i$ is that they represent the height difference the center of gravity overcomes when rotated directly vertical over the $i$th supporting edge (Fig. 4). The computation of the NESM depends on the accuracy of the center of mass. Therefore, rather than using a fixed center of mass position, the distributed center of mass is computed through

$$CM = \frac{1}{m_{\text{total}}} \sum_{i=1}^{n} m_i c_i, \quad (9)$$

where $CM$ is the position of the center of mass of the complete system. $m_{\text{total}}$ is the total mass of the system. $c_i$ and $m_i$ are the centers of mass and the masses of the $n$ body parts, i.e. with respect to this work the chassis and the actuators. Thereby, we can determine the center of mass with respect to the current actuator configuration, which increases the accuracy of the $CM$ estimate. Fig. 4 illustrates the stability measure on the robots considered in this work.

5. Modeling under uncertainty

The deployment of mobile robots in the field involves uncertainties in many aspects. These are the robot’s position and orientation, the actuator configuration for reconfigurable robots and, if used for planning, the terrain model. Modeling these uncertainties requires a stochastic robot–terrain interaction model as proposed in [27]. Norouzi et al. evaluated two methods to estimate the probability density functions of the contact points and the stability. The first one uses a standard Monte Carlo method. Rather than drawing a high number of random samples, the second method uses an unscented transform to determine a sufficient number of samples to represent the distribution. For the considered problem it needs significantly less iterations [27].

For our problem, we have a non-linear function $g$ modeling the robot–terrain interaction, namely either of the two presented algorithms. It determines the robot pose, the contact points and the stability given noisy input data, i.e. the sigma points $x_i$. Each sigma point is transformed using the function $g$. The mean of all transformed sigma points $\mu_x$ represents the predicted robot pose, the contact points and stability value for the input state $x$.

In general, the problem of estimating the model’s output distribution can be formalized as follows. The input is represented by an $n$-dimensional random vector $x$ with expectation $\mu_x$ and covariance $\Sigma_x$. The goal is to predict the expectation $\mu_y$ and covariance $\Sigma_y$ of an $m$-dimensional random vector $y$, where $y$ is given by a non-linear function

$$y = g(x). \quad (10)$$

Using the unscented transform, the mean $\mu_y$ and covariance matrix $\Sigma_y$ of the random variable $x$ is approximated by $2n + 1$
weighted sigma points. For \( i = 0, \ldots, 2n \) the sigma points are determined through
\[
    x_i = \begin{cases} 
    \mu_x & \text{if } i = 0 \\
    \mu_x + \sqrt{(n+k)\Sigma_x} & \text{if } 0 < i \leq n \\
    \mu_x - \sqrt{(n+k)\Sigma_x} & \text{if } n < i \leq 2n
    \end{cases}
\]  
(11)
and the weights are given by
\[
    w_i = \begin{cases} 
    k/(n+k) & \text{if } i = 0 \\
    1/(2(n+k)) & \text{if } i > 0
    \end{cases}
\]  
(12)
where \( k \) is used to preserve the moments of the modeled distribution. If the distribution is assumed to be Gaussian, the value of \( k \) is chosen to be \( k = 3 - n \). However, [27] showed that for the considered problem the model output is not Gaussian and that \( k = 1 \) provides better results. The sigma points are positioned further from the origin and the weights are more balanced.

The sigma points \( x_i \) are transformed using the function \( g \) to yield \( y_i \):
\[
    y_i = g(x_i).
\]  
(13)
The expectation \( \mu_y \) and the covariance \( \Sigma_y \) of the output distribution are computed using the weights \( w_i \) and the transformed sigma points:
\[
    \mu_y = \sum_{i=0}^{2n} w_i y_i
\]  
(14)
and
\[
    \Sigma_y = \sum_{i=0}^{2n} w_i (y_i - \mu_y)(y_i - \mu_y)^T.
\]  
(15)

For the discussed robot–terrain interaction problem, the sigma points are of the form \((x, y, \theta, a_1, \ldots, a_k, s_1, \ldots, s_l)\). They are composed of 2D-robot poses and values \( s_i \) representing the different terrain sections of the robot footprint. See Section 6.2 for a more detailed explanation. The dimension of the sigma points is \( n = 39 \) for the Telerob Telemax robot and \( n = 36 \) for the iRobot Packbot robot. This results in 79 and 73 sigma points for the Telemax and the Packbot robot, respectively. See Section 6.1 for more details about the robot platforms. The prediction of the stability of a state \( x \) requires the transformation of all sigma points. The mean of the transformed sigma points \( \mu_y \) represents the predicted robot pose, the contact points and stability value.

6. Experimental results

This section describes a systematic experimental evaluation of several robot–terrain interaction approaches to determine the robot pose in rough terrain and to subsequently estimate the stability. These approaches are based on the two algorithms described earlier. First, the robot models used for evaluation are explained, followed by the experimental setup and a series of experiments in a number of different environments.

6.1. Robots

We used two robot models for our experiments: a model of the Telerob Telemax and the iRobot Packbot. Both are detailed in this section.

**Telerob Telemax.** The Telerob Telemax robot is a robot originally developed for teleoperated explosive ordnance disposal (EOD) and was modified for autonomous operation (Fig. 1(a)). It is 60 cm long, 40 cm wide and weighs about 70 kg. The Telemax robot is a reconfigurable robot which is able to rotate its four tracks independently of each other by 170°, from entirely folded, −90°, all the way down to 80° lifting the robot about 45 cm up. Completely stretched the robot has a length of about 160 cm. The robot is equipped with a skid-drive, and its maximal translational speed is 1.2 m/s with

---

Fig. 4. Two-dimensional illustration of the stability margin for (a) the iRobot Packbot and (b) the Telerob Telemax robot for different configurations. Side and front views show separate situations, highlighting the pitch and roll angle, respectively.
wheels attached and 1.0 m/s without. The ability to reconfigure allows the robot to overcome steps as high as 50 cm and slopes of up to 45°.

iRobot Packbot. The Packbot 510 is developed by iRobot (Fig. 1(b)). It is a very light, mobile and flexible robot for a variety of tasks and is deployed among others by the US Army. The robot is about 69 cm long and 52 cm wide and weighs about 30 kg. With extended actuators its length is 89 cm. The robot is skid-steered and able to drive up to 2.6 m/s. In contrast to the Telemax, the Packbot robot has two mechanically linked actuators which can be adjusted 360°. The actuators are smaller; hence, the Packbot is able to overcome steps of up to 25 cm of height. However, due to the very low center of mass, it can climb steeper slopes, i.e. up to 60°.

6.2. Experimental setup

We evaluated four different approaches to estimate the contact points and the stability of a system in rough terrain. These approaches are based on the iterative geometric method (IGM) and the quasi-static physics simulation (ODE) algorithms. The first two approaches are deterministic variants of each method which estimate the stability for a given state \( \mathbf{x} \). The second IGM/ODE pair of approaches involves uncertainties about the robot state and the terrain model. It, first, determines the \( 2n + 1 \) sigma points \( \mathbf{x}_i \) as described above and estimates the stability for each of those points. The final prediction is the mean value over all sigma point estimates. As the number of sigma points is \( 2n + 1 \), the time required to provide a single estimate is about \( 2n + 1 \) times higher compared to the deterministic versions. For each estimation the stochastic variants must evaluate 79 states (\( n = 39 \)) for the Telemax robot or 73 states (\( n = 36 \)) for the Packbot robot.

IGM Deterministic (IGM\(_D\)) estimates the contact points for state \( \mathbf{x} \) using the iterative geometric method.

ODE Deterministic (ODE\(_D\)) estimates the contact points for state \( \mathbf{x} \) using the quasi-static physics simulation.

IGM Stochastic (IGMS\(_S\)) involves uncertainties by estimating the contact points for the \( 2n + 1 \) sigma points \( \mathbf{x}_i \), which include state and terrain fluctuations using the iterative geometric method. The mean value is returned as final estimation.

ODE Stochastic (ODE\(_S\)) involves uncertainties by estimating the contact points for the \( 2n + 1 \) sigma points \( \mathbf{x}_i \), which include state and terrain fluctuations using the quasi-static physics simulation. The mean value is returned as final estimation.

Most of the times the iterative geometric method (IGM) requires less than 10 iterations, so the maximum number of iterations is set to 15. The threshold for the rotation changes is 0.2°. The maximum number of iterations for the physics simulation is 2500 with simulation time steps of 0.025 s. If angular and linear velocities are below \( \epsilon = 1e^{-10} \), the pose is assumed to be stable and the simulation is stopped. ODE internal parameters are important for the behavior of the simulation. We use a low object softness value of 0.01. Further, we enforced hard contact constraints and allow an error correction of 80% within one simulation step.

To involve uncertainties into the computation, we assume noise parameters for the robot position and orientation, the actuator configuration and the terrain model. Following the remarks in [27], we assume a position variance of 5 cm, an orientation variance of 2° and a variance of 3° in the actuator angles. To build our real-world maps, we use SICK laser scanners to acquire the data and merge the scans using a standard iterative closest point (ICP) method. The 3D models are inspected afterwards and corrected manually if necessary. Subsequently, missing data is interpolated using a median filter. A Gaussian filter mitigates the noise caused by grass and other vegetation. The iterative geometric method (IGM) uses a 2.5D heightmap with a cell resolution of 0.05 m, ODE uses a triangular mesh built from this 2.5D heightmap. For the resulting models we assume Gaussian noise with zero mean and a variance of 2 cm. However, applying the noise to the entire robot footprint at once would result in a similar surface shape. Thus, [27] proposed to divide the robot’s footprint in several sections to which noise is applied separately. They determined 16 longitudinal and 2 lateral sections for the iRobot Packbot. A finer resolution for the Telerob Telemax affected the output only slightly. Hence, keeping the computational complexity in mind, we also chose 16 × 2 sections for the Telemax robot.

The evaluation aims at comparing the iterative geometric method (IGM) with the quasi-static physics simulation (ODE). IGM is quite fast because it uses an approximate terrain model. ODE is the more accurate but also the computationally more expensive method. To show limitations of both methods, we follow a structured approach by testing the four variants on increasingly complex environments. First, we investigate all approaches on inclined flat surfaces as they pose ideal conditions under which the plane approximation of the IGM algorithm is fully valid (Section 6.3). We continue the evaluation on a series of stairs because they violate the plane assumption but result in the same overall inclination (Section 6.4). These sets of experiments are followed by an evaluation on single steps since they even more increase the fitting error of the regression planes (Section 6.5). To eliminate effects of regular structured environments, the final set of experiments evaluates the approaches on irregular terrain. These terrains are based on smoothed random stepfields [39] of increasing difficulty [40] (Section 6.6). Subsequently the results are discussed with respect to realistic scenarios and the differences in the computational complexity are described (Section 6.7).

The majority of experiments use the Telerob Telemax robot. The four actuators and the higher center of mass compared to the iRobot Packbot make the Telemax the more interesting system. Also the terrain’s effects on the robot’s stability are more apparent and are easier to illustrate using the Telemax robot. We selected three representative actuator configurations of the Telerob Telemax: a rather compact configuration of \( e^{−30} = (−30°, −30°, −30°, −30°) \), the configuration which yields the maximum ground contact on flat surfaces \( e^{21} = (21°, 21°, 21°, 21°) \), and a configuration which raises the chassis and the center of mass of the robot \( e^{45} = (45°, 45°, 45°, 45°) \). These configurations are illustrated in Fig. 5.

At this point it is important to stress that the experimental scenarios are selected to provide more insight into the different approaches. They are part of a systematic evaluation to show the influence of the actuator configuration, the orientation and the terrain on the static stability of the robot. As such, they do not all represent realistic scenarios. For instance, inclined terrain is usually approached in the direction of its gradient. Offsets to this direction are avoided such that the slope affects the often stronger pitch-stability of the robot. This holds for inclined flat surfaces and even more so for stairs and single steps. Additionally, inclinations and stairs are traversed with a high ground contact configuration. Folded configurations provide too little support. Elevated configurations raise the center of gravity too much.

6.3. Inclined flat surfaces

In this section, we start our evaluation with a series of experiments on inclined flat surfaces to study the four different approaches under conditions where the plane assumption of the IGM is perfectly valid.

The experiments involve flat surfaces of ten different inclinations; starting a 0° and increasing the slope gradually in steps of
5° up to a maximum inclination of 45°. We test the three actuator configurations $C_{-30°}, C_{21°}$, and $C_{45°}$. With respect to the system’s stability, it is important at which angle to the gradient of the surface inclination the robot travels. Therefore, we evaluate three different offsets to the surface gradient, i.e. 0°, 45° and 90° (Fig. 6). Since the Telemax robot is assumed to be symmetrical in terms of its geometry and mass distribution, other offsets being multiples of 45° are equivalent to one of the three.

The results are shown in Fig. 7. The x-axis shows the inclination of the surface in degrees, the y-axis the stability estimations in centimeter. The stability value is the minimal difference in height the center of mass must overcome when being rotated over an edge of the supporting polygon (see Fig. 3). Hence, the higher this value, the more stable the robot pose. A value of zero indicates that the center of mass is directly above an edge of the supporting polygon. Thus, such situations are regarded as marginally stable.

If the robot faces the gradient of the inclined surface (offset 0°), a higher inclination will worsen the pitch-stability of the system. Since the folded configuration $C_{-30°}$ has no additional support through the actuators, its roll and pitch-stability are almost the same and its overall stability decreases rapidly with increasing terrain slope (Fig. 7(a)). As the most support is granted by $C_{21°}$, it is the most stable configuration for an offset of 0° (Fig. 7(d)). The stability values of $C_{45°}$ are dominated by the small roll-stability of the elevated configuration. However, at a surface inclination of 40° the pitch stability becomes smaller than the roll-stability and the overall value decreases (Fig. 7(g)). For the Telerob Telemax, like for most robotic systems, the pitch-stability is better than the roll-stability. As expected, if the offset to the gradient of the surface inclination increases, the roll angle of the system also increases and the stability declines quickly. However, what is interesting is the fact that the most folded configuration $C_{-30°}$ is more stable than the maximum contact configuration $C_{21°}$. This is explained by the buckling of the Telemax actuators. This buckling causes a more elevated chassis for the $C_{21°}$ configuration (Fig. 7(b), (c), (e) and (f)). As the center of mass is the highest for $C_{45°}$, it is the least stable configuration in these situations (Fig. 7(h) and (i)).

IGM$_D$ and ODE$_D$ predict in almost every situation very similar stability. In contrast, IGM$_S$ and ODE$_S$ show larger differences. In general, the estimated stability using IGM$_S$ is closer to the other approaches than the predictions of ODE$_S$. This is due to a combination of factors. First, the least-squares plane of the robot’s footprint has a normalizing effect reducing the influence of the terrain distortions. Second, if the terrain distortions influence the overall surface slope as is the case in the 45° and 90° offset scenarios, then the pairs of opposing sigma points cancel themselves out; one reduces the surface slope, the other increases it. As the 0° offset scenarios show, this effect only comes into play if the weakest support edge is influenced by the general surface inclination. This is not the case for $C_{21°}$ (Fig. 7(d)) and not for $C_{45°}$ until an inclination of 40° (Fig. 7(g)). And of course for 0° surface slope across all scenarios. The differences between ODE$_S$ and IGM$_S$ are due to the more accurate terrain model. Since the terrain variations are not smoothed, they in general cause less stable poses. Consequently, ODE$_S$ estimates a lower stability in all scenarios. However, this effect diminishes for $C_{45°}$ at an offset of 0°. The very small contact areas with the surface make the robot less prone to the terrain variations. Hence, IGM$_S$ and ODE$_S$ estimate very similar values.

The bars show the standard deviation of the sigma points predictions. Since IGM$_S$ smooths the terrain variations of the sigma point, its standard deviation is generally smaller compared to ODE$_S$. If the terrain becomes steeper and the stability smaller, the variation in the stability values of the sigma points decrease. An increasing number of sigma points become unstable, i.e. 0 cm, which shortens the range. Since the weaker roll-stability is not affected by the inclination the scenarios of Fig. 7(d) and (g) (offset: 0° with configurations $C_{21°}$ and $C_{45°}$), the standard deviation is large even for steeper inclinations. The sigma point predictions vary stronger for a configuration of $C_{21°}$ than for $C_{45°}$. Due to the larger ground contact of $C_{21°}$, the robot stability is more affected by terrain variations. $C_{45°}$ has only four contact points usually at the tip of the actuators and is less affected.

Comparing the different approaches we find the deterministic versions, IGM$_D$ and ODE$_D$, agree in almost every scenario with only a few exceptions in which ODE$_D$ estimates a slightly lower stability. The predictions of IGM$_S$ and ODE$_S$ are further apart due to the aforementioned reasons.

6.4. Flight of stairs

The next set of experiments evaluates the different robot-terrain interaction models on stairs of increasing steepness. Stairs
are chosen since they violate the plane assumption of the IGM models to some extent but still result in a constant overall inclination as assumed by those models. This is especially the case for tracked robots as they usually interact solely with the edges of the stairs. Table 1 shows the root-mean-squared errors (RMSE) of the regression planes for the stair scenarios.

The experimental setup is similar to the one on inclined flat surfaces. By setting the step depth to 25 cm and adjusting the height of the risers accordingly, we again achieve inclinations of 0°, . . . , 45° in 5° steps. We consider the same three offsets to the direction of the stairs gradients, i.e. 0°, 45° and 90°, as well as the same three configurations e−30, e21, and e45. Fig. 8 illustrates the setup.

The results on the flights of stairs are shown in Fig. 9. In contrast to the previous series of experiments the IGM and ODE estimates differ to a greater extent on the flights of stairs. This is due to the more distinct geometry of stairs which is considered by the ODE methods only. Like on flat surfaces, the IGM estimates are smoothly declining in most instances if the slope increases. The ODE methods consider the actual geometry of the stairs and can catch contact points which do not lie on the actuators’ lowest points. This can result in larger supporting polygons and is the reason why the ODE methods estimate higher stability values than the IGM versions for configuration e−30 and steeper slopes at an offset of 0° (Figs. 9(a) and 10(a)). For 0° offset e21 and e45 behave almost as before.

![Figure 7](image1.png)

Fig. 7. Comparison of the stability estimates of the four approaches on inclined flat surfaces. The top row shows the results for the Telemax robot with a folded configuration of e−30, the second row for the maximum ground contact configuration e21 and the third row for an elevated configuration e45. The columns correspond to the offset to the surface gradients, i.e. 0°, 45° and 90°. The higher the stability value, the more stable the pose. Zero represents the equilibrium and is regarded as unstable. The bars indicate the standard deviation across the sigma point estimates.

<table>
<thead>
<tr>
<th>Stairs slope</th>
<th>5°</th>
<th>10°</th>
<th>15°</th>
<th>20°</th>
<th>25°</th>
<th>30°</th>
<th>35°</th>
<th>40°</th>
<th>45°</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plane RMSE</td>
<td>0.63</td>
<td>1.23</td>
<td>1.82</td>
<td>2.41</td>
<td>2.97</td>
<td>3.49</td>
<td>4.00</td>
<td>4.47</td>
<td>4.91</td>
</tr>
</tbody>
</table>

Table 1
RMSE of the least-squares planes on the stairs maps in cm.
Fig. 8. The different evaluated offsets to the gradient of the stairs.

(a) Offset: 0°. (b) Offset: 45°. (c) Offset: 90°.

Fig. 9. Comparison of the stability estimates of the four approaches on flights of stairs of varying inclination. The top row shows the results for the Telemax robot with a folded configuration of $C_{-30}$, the second row for the maximum ground contact configuration $C_{21}$ and the third row for an elevated configuration $C_{45}$. The columns correspond to the offset to the stairs gradients, i.e. 0°, 45° and 90°. The higher the stability value, the more stable the pose. Zero represents the equilibrium and is regarded as unstable. The bars indicate the standard deviation across the sigma point estimates.

(a) Offset: 0°, Config: $C_{-30}$. (b) Offset: 45°, Config: $C_{-30}$. (c) Offset: 90°, Config: $C_{-30}$.

(d) Offset: 0°, Config: $C_{21}$. (e) Offset: 45°, Config: $C_{21}$. (f) Offset: 90°, Config: $C_{21}$.

(g) Offset: 0°, Config: $C_{45}$. (h) Offset: 45°, Config: $C_{45}$. (i) Offset: 90°, Config: $C_{45}$.

However, since contact with the stairs is not always with the tip of the actuators or on the step rather than on the risers' edges, the support polygon is smaller or the robot's pitch more severe. This effect results in the rapid decrease in ODE stability estimates at higher inclinations for $C_{21}$ and $C_{45}$ (Fig. 9(d) and (g)). These situations are visualized by Fig. 10(c) and (d).

At an offset of 45° the ODE stability estimates are generally lower than the IGM estimates. For $C_{-30}$ and $C_{21}$ the ODE and IGM
predictions decrease in a similar manner for an increasing inclination (Fig. 9(b) and (e)). The normalizing effect of the least-squares plane lessens the slope which is introduced by the actual geometry of the stairs. Thus, the IGM versions overestimate the stability. For $C^45$ the ODE variants already start at low slopes to estimate relatively small stability values since the simulation finds only three contact points compared to the four of the IGM approaches (Figs. 9(h) and 10(e)). This highlights another difference between the two concepts. Due to the approximated terrain model, the IGM is always able to determine four contact points for the considered configurations even if they result in unstable poses. In contrast, the ODE simulation accounts for the mass distribution and gravity force and determines the actual number of valid contact points. This usually results in more realistic estimates. Nevertheless, it points to a problem with the ODE simulation. During parameter evaluation, we found the simulation to be quite volatile to ODE internal parameters.

Finally, at 90° offset to the stairs’ inclinations the ODE methods predict higher stability values than their IGM counterparts. The actual geometries of the stairs work in favor of the robot’s stability, by slightly decreasing the inclination of the terrain (Fig. 9(c), (f) and (i)). Additionally, Fig. 10(b) illustrates these situations.

When comparing the different IGM and ODE versions, we find that the IGM versions behave much like before. Also IGM$^h$ and ODE$^h$ tend to estimate the lowest stability as usual. However, the differences between IGM and ODE are larger as on inclined flat surfaces but still quite small in most cases. This is especially the case in the more realistic scenarios. The general behavior of the standard deviation of the sigma point predictions is the same as before. ODE$^h$ stability estimates deviate stronger from the mean than IGM$^h$ estimates. The more unstable the final predictions are, the smaller the variations become. Also, the more ground contact the robot has, the larger the stability estimates of the sigma points vary.

### 6.5. Single steps

The flat surfaces do not violate the IGM assumption at all while the flights of stairs do moderately. This set of experiments is conducted on single steps of different heights. They cause an even higher error of the least-squares planes, see Table 2.

As before, we evaluate the stability estimates of the four approaches for the same three configurations $C^{-30}$, $C^{21}$, and $C^{45}$. We consider step heights between 20 cm and 45 cm with a gradual increase of 5 cm. The ground-step proportion of the robot’s footprint influences the orientation of the least-squares planes. Therefore, we test three positions with different proportions: a little in front of the steps, directly on the step edges and almost on top of the steps, i.e. 25%, 50% and 75% step proportion of the footprint (Fig. 11). Previous experiments showed that the influence of steps is less obvious if they degrade only the pitch-stability. The roll-stability is actually lower and, thus, represents the overall stability. Therefore, we present experiments with an offset of 45° to the steps, such that the roll-stability is also affected by the steps. However, in real applications robots of this size would approach steps of more than 20 cm in height with a little angular offset to the steps as possible. Hence, the tested situations involving an offset of 45° are artificial and unlikely to occur in real applications as any navigation algorithm would avoid such offsets to reduce the danger of tip-over.

The results are shown in Fig. 12. All IGM and ODE approaches estimate very low stability values for configuration $C^{-30}$ in the three scenarios (Fig. 12(a)–(c), different scale). A folded configuration like $C^{-30}$ provides too little support for the robot in order to overcome steps of the tested height.

With the configuration $C^{21}$ the situation is different. If only a small portion of the robot resides on the step, the ODE methods estimate a low stability. One of the actuators rests on the steps edge and increases the inclination through its buckling (Fig. 12(d)). Directly on the step, the influence of the steps is reduced as the edges

### Table 2

<table>
<thead>
<tr>
<th>Step height</th>
<th>Plane RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 cm</td>
<td>4.86</td>
</tr>
<tr>
<td>25 cm</td>
<td>5.97</td>
</tr>
<tr>
<td>30 cm</td>
<td>7.04</td>
</tr>
<tr>
<td>35 cm</td>
<td>8.03</td>
</tr>
<tr>
<td>40 cm</td>
<td>8.96</td>
</tr>
<tr>
<td>45 cm</td>
<td>9.57</td>
</tr>
</tbody>
</table>

The results are shown in Fig. 12. All IGM and ODE approaches estimate very low stability values for configuration $C^{-30}$ in the three scenarios (Fig. 12(a)–(c), different scale). A folded configuration like $C^{-30}$ provides too little support for the robot in order to overcome steps of the tested height.

With the configuration $C^{21}$ the situation is different. If only a small portion of the robot resides on the step, the ODE methods estimate a low stability. One of the actuators rests on the steps edge and increases the inclination through its buckling (Fig. 12(d)). Directly on the step, the influence of the steps is reduced as the edges

![Fig. 10. Selection of simulated situations on flights of stairs. The center of mass is the black sphere, the green spheres are the determined contact points and the green lines are the edges of the supporting polygon.](image-url)
reach between two actuators without touching the chassis. As a result, the simulations predict a better stability compared to the IGM approaches (Fig. 12(e)). Almost on top of the steps, the ODE methods are unable to find a stable pose for $e^{21}$ and predict marginal stability (Fig. 12(f)). The situation is illustrated in Fig. 13(a). The IGM methods overestimate the stability if the footprint includes

Fig. 11. The different evaluated offsets to the edge of the step.

Fig. 12. Comparison of the stability estimates of the four approaches on single steps of varying heights. The top row shows the results for the Telemax robot with a folded configuration of $e^{-30}$, the second row for the maximum ground contact configuration $e^{21}$ and the third row for an elevated configuration $e^{45}$. The columns correspond to the ground–step proportion, namely 25%, 50% and 75% of the step within the robot’s footprint. The offset to the step is 45°. The higher the stability value, the more stable the pose. Zero represents the equilibrium and is regarded as unstable. The bars indicate the standard deviation across the sigma point estimates.
75% of one level (Fig. 12(d) and (f)). However, the stability predictions decrease for higher steps.

Fig. 12(g) also shows unexpected results for configuration $C_{45}$ in front of the steps. The stability estimated by ODE actually increases slightly with the height of the step. This effect is caused through a single contact point on top of the step and two on the ground. With an increasing step height the center of mass moves further and further to the broader area of the support triangle, making the pose more stable (Fig. 13(b)). As an elevated configuration, $C_{45}$ provides quite a lot of space between the actuators. When placed directly on top of the step, the robot contacts with the ground and the step level while the edge resides in the space between the actuators (Figs. 12(h) and 13(c)). This leads to a less inclined robot orientation and, hence, to a more stable pose compared to the IGM approaches. Just on top of the steps, ODE predicts unstable poses as before (Fig. 13(a)). Like for $C_{21}$, IGM estimates higher stability compared to ODE due to the same reasons.

The experiments on top of the steps revealed a minor issue with the ODE simulation. Despite testing several parameter settings of the ODE simulation, there remain isolated situations in which the simulation does not converge. This being said, these situations are usually borderline cases with very low stability and would be regarded as unstable for real-world applications. Thus, an “unstable” prediction due to insufficient contact points does not hurt much.

In summary, the IGM approaches overestimate the stability in some of these scenarios. The IGM variants always find valid contact points on the fitted planes. The ODE methods also detect invalid states due to contact with the robot chassis. Furthermore, if the edge of a step resides between the actuators of the robot, the robot pose will be less inclined and the stability will increase, a situation only the ODE methods are able to identify. However, as mentioned earlier, these scenarios are chosen to provide more insight into the properties of the algorithms and are less likely to occur in real applications.

6.6. Random terrain

The final set of experiments is performed on random terrain modules to eliminate effects of manually constructed environments and to evaluate the methods on irregular terrain. To generate the maps for this experiment series, we used a method from the literature. First, we determined a field of small terrain patches at random similar to random step fields [39] and smoothed the resulting map with a Gaussian blur. Second, to increase the difficulty of the terrain incrementally, we used the method proposed by [40]. The terrain slopes are increased by scaling the height values

$$\forall v \in V : h'(v) = h(v)(1 + S \cdot s),$$

where we chose $S = 2.5$ and $s \in [0, 10]$ for our experiments. This results in a total of 11 maps, which allows us to evaluate the robot–terrain interaction approaches on terrain of increasing difficulty. Fig. 14 illustrates these terrain modifications. The RMSE of the least-squares planes is shown in Fig. 15. The RMSE increases with the magnitude of the slopes. The iRobot Packbot is smaller than the Telerob Telemax. Hence, the smaller footprint plane can be fitted better to terrain profiles and the RMSE is smaller in general. This is also the reason why the RMSE of the Telemax footprint increases more strongly.

We again test the performance of the two IGM approaches and the two ODE methods. Besides the Telerob Telemax robot, we also used the iRobot Packbot for these experiments. 200 random positions $(x, y, \theta)$, i.e. 2D position and heading, with the maximum ground contact configuration are evaluated on each map. This configuration is $C_{21}$ for the Telemax and $C_{17} = (17^\circ)$ for the Packbot (Fig. 5).

The static stability of the Telemax and the Packbot robot are evaluated on the random terrains. Fig. 16(a) and (b) shows the average stability values on the increasingly steeper environments. The bars represent the standard deviation of the stability values. The boxes depict the standard deviation of the sigma point predictions for $\text{IGM}_1$ and $\text{ODE}_0$. The IGM and ODE$_0$ stability estimates for the Telemax robot start at similar values on the flattest terrain module. The ODE predictions decrease faster for higher slopes since the occasions in which ODE estimates a low stability due to just three valid contact points. With increasing slopes the IGM stability estimates of all approaches vary stronger due to more and
more emerging terrain characteristics. The ODE estimates vary less in steeper environments since the values approach zero stability. The variations within the set of sigma point predictions are almost constant for IGM. This is caused by the normalization of the regression planes. For ODE, the fluctuations are relatively large at low slopes and decrease for higher slopes since greater numbers of sigma points are predicted unstable.

Similarly, the ODE stability estimates for the Packbot robot decrease faster than the IGM estimates. Again ODE₆ predicts the lowest values. In contrast to the Telemax robot, the stability estimates vary more strongly. The standard deviation of the IGM₆ sigma point predictions has a similar size as those for the Telemax. For ODE₆, the standard deviation of the sigma point estimates is considerably larger for the Packbot.

These sets of experiments underline the fact that in general the Packbot is the more stable system compared to the Telemax. Compared to the ODE methods, the IGM approaches estimate higher stability values due to the normalizing effect of the regression planes. The estimates of the IGM variants are quite close. The ODE₆ stability predictions are noticeable lower than the ODE₀.

### 6.7. Discussion

With only few exceptions the curves of the IGM methods are smoothly decreasing with increasing difficulty of the terrain. Due to the normalizing effect of the plane fittings, edges are less influential. Therefore, the IGM methods tend to overestimate the stability in general but is quite accurate in realistic situations as discussed later. IGM₀ and IGM₅ usually produce very similar estimates. Only if the noise in the robot state or the terrain model affects the roll-stability while the terrain does not, or if the robot state is just stable/unstable, IGM₆ predicts different stability values. In these cases the IGM₆ estimates usually tend toward the ODE predictions. However, since the differences between IGM₀ and IGM₅ are very small, the stochastic version results in too little improvement to justify the higher computational effort.

This leads to another important point not discussed so far, namely the computational complexity and the run time. Table 3 depicts the run times for all compared approaches split up for the two robots and the different terrains. It shows that the huge advantage of the IGM approaches is their very low run time. The computing time of the different approaches differ significantly. IGM₆ is about 3 orders slower than IGM₀.

The ODE-based simulation methods estimate lower stability values in general. While the predictions of the ODE methods and the IGM methods are very close for flat inclined surfaces, they are more apart on stairs and even more on single steps and the random terrains. Since the ODE methods consider the actual shape of the terrain, their predictions are more accurate. Also the curves are less smooth compared to the IGM curves and not necessarily decreasing in more difficult or inclined terrain. Especially on the single steps it becomes apparent that considering the actual terrain shape can increase the stability if the robot state diminishes the influence of the step edges. In general, because the shape of the terrain is accounted for by the ODE methods, the stability estimates of ODE₆ are lower compared to the other approaches. The run times of the ODE methods, even of ODE₀, are significantly higher compared to the IGM approaches. ODE₆ is by an order of 4 slower than IGM₆. ODE₆ can require more than 30 s depending on the complexity of the terrain to make a prediction. Therefore, the ODE run times are prohibitively high for global planning algorithms which have to evaluate several 100,000 configurations.

If time is not an issue, ODE₆ represents the best choice as it is the most accurate and most valid method. However, for real applications the high computational efforts of the ODE methods become a factor. This begs the question whether the IGM approaches and due to the aforementioned reasons especially IGM₀ can be used in realistic real-world scenarios. To answer this question, first, we discuss the performance of IGM₀ in the previous experiments which do represent realistic scenarios. Second, we take a look at two real-world environments (Fig. 17).

As mentioned earlier, inclined terrain is approached best in the direction of its gradient. This is more important on stairs or steps with discrete contact points. Even if not all of the inclined flat surface scenarios (Section 6.3) are realistic, IGM₀ and ODE₀ predict almost identical stability for these scenarios. Driving on inclined terrain requires an actuator configuration with a large area of ground contact. It provides high support for the robot while maintaining a low center of gravity. Therefore, a good way to traverse stairs is with configuration C¹ and 0° offset to the stairs gradient (Fig. 9(d)). The predictions of IGM₀ and ODE₀ are again almost the same up to an inclination of 30°. The manufacturer specified operating limit for the Telemax is 45° slope. Traversing stairs of inclinations so close to the physical possible limit autonomously is very risky and, hence, less realistic. The artificial step scenarios (Section 6.5) do not represent realistic scenarios. Such challenging terrain cannot be approached at an 45° angle. The experiments are meant to provide more insight about the methods and to show their limitations. It is difficult to comment on how realistic the randomly chosen positions in the final experiment series are. IGM estimates higher stability than ODE which is problematic. However, the general shape of the curves is similar enough such that an appropriate safety margin is able to adapt the IGM stability estimates to the ODE estimates.

The robot–terrain interaction model is intended to be used in our global path planner. Provided a model of the robot and a map of the terrain, it evaluates the robot configuration with respect to the terrain. For this purpose the terrain models are built offline in advance and provided to the planner. Without the global information of a map, the probability of exposing the robot to unnecessary

![Fig. 15. The RMSE of the least-squares planes on the random terrain modules. The values are averaged over 200 random positions. The bars indicate the standard deviation. Depicted are the TeleMax Telemax and the iRobot Packbot.](image-url)
Fig. 16. Comparison of the stability estimates of the four approaches on smoothed random steps fields. The results are on a series of maps with increasing slopes for the Telemax robot with configuration $C^{21}$ (left) and the Packbot robot with configuration $C^{17}$ (right). For each map 200 random positions are evaluated and the average stability is determined. The higher the stability value, the more stable the pose. Zero represents the equilibrium and is regarded as unstable. The boxes indicate the standard deviation of the sigma point stability predictions of the stochastic approaches. The bars show the standard deviation of the stability estimates for the randomly sampled positions.

Fig. 17. The testing hill (left) and the hill of rubble (right).

Table 4
RMSE values in cm for the two natural terrains “Hill” and “Rubble” and the two considered robots, the Telerob Telemax and the iRobot Packbot.

<table>
<thead>
<tr>
<th>Robot</th>
<th>Terrain</th>
<th>Min</th>
<th>Median</th>
<th>75%-Quantile</th>
<th>90%-Quantile</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telemax</td>
<td>Hill</td>
<td>0.108</td>
<td>1.382</td>
<td>2.543</td>
<td>4.387</td>
<td>36.054</td>
</tr>
<tr>
<td></td>
<td>Rubble</td>
<td>0.112</td>
<td>0.790</td>
<td>2.013</td>
<td>4.216</td>
<td>36.038</td>
</tr>
<tr>
<td>Packbot</td>
<td>Hill</td>
<td>0.083</td>
<td>1.063</td>
<td>1.935</td>
<td>3.367</td>
<td>27.383</td>
</tr>
<tr>
<td></td>
<td>Rubble</td>
<td>0.059</td>
<td>0.641</td>
<td>1.413</td>
<td>2.733</td>
<td>27.364</td>
</tr>
</tbody>
</table>

risk is large, especially in rough terrain. Consider the usage of the robot–terrain interaction models during execution, e.g. to plan local paths for a limited look-ahead. In this case the models must deal with incomplete information. Due to the low-positioned sensors on the robot and the structure of the terrain, small terrain areas are often occluded. The IGM approach can handle the missing data easily without any adjustment since the least-squares planes may simply be based on less data points. However, this involves the assumption that the occluded areas generally follow the shape of the visible parts. Depending on the conformity of the terrain, this assumption can be violated arbitrarily strong. This being said, the ODE approach additionally requires some sort of data augmentation to allow the generation of the triangle mesh for which no simple solution exists. The assumptions about the missing terrain areas generally increase the risk during rough terrain traversal.

Fig. 18 and Table 4 show the root-mean-squared error of the least-squares planes on two real-world maps (Fig. 17): an artificial testing hill and a hill of rubble. The graphs show the actual values and the colored maps the distribution of these values across the terrain. Since the iRobot Packbot is smaller than the Telerob Telemax, its footprints and the fitting errors are smaller. The extreme values are caused by fences, lamp posts or trees. The median and the quantiles indicate that the RMSE for most of the environments lies within the range considered in the previous experiments. Additionally, Fig. 19 shows the testing hill and the hill of rubble colored with the IGM$_D$ stability values for the Telemax and the Packbot robot. The values are the average over four orientations, i.e. $0^\circ$, $45^\circ$, $90^\circ$ and $135^\circ$. The stability values in all figures show that IGM$_D$ is able to provide a valid basis for a relative comparison of the stability within a global planning application. Therefore, the performance of IGM$_D$ in realistic scenarios is sufficiently high to be used in our planning framework. Since it is also very fast, IGM$_D$ supports reasonably fast planning in rough terrain.

7. Conclusion and future work

In this paper we presented a novel geometry-based method to estimate the robot–terrain interaction of actively reconfigurable robots, like the Telerob Telemax or the iRobot Packbot. The major part of this paper presents a systematic evaluation of implementations of two main approaches to robot–terrain interaction: a quasi-static simulation and the aforementioned geometric method. We
also included stochastic variants of these methods in our evaluation to involve uncertainties about the robot state and the environment. The analysis in this paper provides the first systematic comparison of some of the major concepts to robot–terrain interaction and provides insight in the performance of those methods in different conditions.

Altogether, we evaluated the four approaches on more than 40 different maps. We started on inclined flat surfaces of multiple slopes. We continued with stairs of multiple slopes and single steps of several heights. We ended with random terrains of increasing degrees of slopes. On all maps, the Telemax robot was used. Additionally, the Packbot robot was used on the random terrains.

IGM\(_D\) and IGM\(_S\) usually estimate very similar stability values. IGM\(_S\) only produces different estimates if the robot state is just stable/unstable or if the noise in the robot state or the terrain model affects the roll-stability while the terrain does not. However, due to the very small differences between IGM\(_D\) and IGM\(_S\), the higher computational effort of IGM\(_D\) cannot be justified. Through the normalizing regression planes, IGM\(_D\) tends to overestimate the stability in general. However, in realistic scenarios IGM\(_D\) is quite accurate. The ODE methods are more accurate as they account for the actual terrain shape. However, this accuracy increases the time required for a single stability prediction by a factor of 2.7\(\times 10^3\) on average depending on the terrain complexity. These are prohibitively high values for many applications. Considering the performance of IGM\(_D\) in realistic situations, we can conclude that the method provides good estimates and is a valid and practical alternative to a physics simulation. Especially, since IGM is several orders faster.

Future work will concentrate on further integrating these robot–terrain interaction models in our global rough terrain motion planner to assess the risk of a robot pose. The risk value is composed of a traversability value, which accounts for uncertainty, a traction estimate and the stability of the robot. The robot–terrain interaction model is intended to provide the stability estimation within this composition. It must be fast since robot poses are evaluated frequently along the path segments to ensure a safe path.
Besides more simulation experiments we will perform additional real-world experiments to study the influence of the different robot–terrain interaction models on the planning results. Another direction of future work is to investigate possibilities to use IGMMo when appropriate and ODEo in very rough areas.

Acknowledgment
The work of the fourth author was supported by the German Research Foundation DFG under the Cluster of Excellence “Integrative Production Technology for High-Wage Countries”.

References
Dirk Schulz received his Doctorate degree in Computer Science from the University of Bonn in 2002. From 2002 to 2007 he was a post doctoral researcher at the Computer Science department of the University of Bonn. Since 2008 he is a senior researcher at the Fraunhofer Institute for Communications, Information Processing and Ergonomics FKIE in Wachtberg, Germany. His primary research interests include networked multi-robot systems as well as state estimation and sensor fusion techniques with applications in robotics and intelligent environments.

Christopher M. Schlick was born in Germany on July 1, 1967. He received the M.S. degree (Dipl.-Ing.) in Electrical Engineering from Berlin University of Technology in 1992, Ph.D. degree (Dr.-Ing.) in Mechanical Engineering from Aachen University of Technology in 1999, and the Habilitation degree (Dr.-Ing. habil) also in Mechanical Engineering from Aachen University of Technology in 2004. He worked for the computer networks industry in 1992 and 1993 as a design engineer. From 1994 to 2000, he joined the Institute of Industrial Engineering and Ergonomics at Aachen University of Technology. From 2000 to 2004 he was the head of department of human–machine systems at the Research Institute for Communication, Information Processing and Ergonomics, Wachtberg, Germany. He is now a full professor of industrial engineering and ergonomics at RWTH Aachen University and deputy director of the Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE. His current interests are human–machine and human–robot systems.