Whole-Body Planning for Obstacle Traversal with Autonomous Mobile Ground Robots

Martin Oehler, Stefan Kohlbrecher and Oskar von Stryk

Technische Universität Darmstadt, Germany
{oehler,kohlbrecher,stryk}@sim.tu-darmstadt.de

Abstract. A common challenge for autonomous mobile ground robots in unstructured environments is the traversal of obstacles without risking to tip over. Previous research on prevention of vehicle tip-over is mostly limited to basic mobility systems with only few degrees of freedom (DOF). In this paper, a novel whole-body motion planning approach is presented. Based on a 3D world model and a given planned path, the trajectories of all joints are optimized to maximize robot stability. The resulting motion plan allows the robot to cross obstacles without tipping over. Compared to existing approaches, the proposed approach considers environment- and self-collisions during planning. Few assumptions about the robot configuration are made which enables the adoption to different mobile platforms. This approach is evaluated for a simulated and a real robot. The platform is a tracked vehicle with adjustable flippers and a five DOF manipulator arm. In several test scenarios, it is shown that the proposed approach effectively prevents tip-over and increases robot stability.

Keywords: Stability Optimization, Unstructured Environment

1 Introduction

A common challenge for autonomous robots operating in unstructured environments is the traversal of obstacles. To effectively increase its stability the robot can use arm joints to shift the center of mass (COM) and flipper joints to keep contact with the ground.

We propose a division into two sub-problems: While *pre-planning* computes a trajectory for all joints based on a map of the environment, *reactive behavior* corrects for disturbances during execution, such as unstable ground or slippage.

This work focuses on *pre-planning* and presents a novel whole-body planning approach that utilizes an online-generated 3D map of the environment to optimize joint trajectories along a given path. Vehicle tip-over is prevented by assessing joint configurations with a stability metric. The resulting motion plan allows the robot to safely cross generic obstacles. The proposed method is applicable to a wide range of wheeled or tracked robot platforms. Performance is evaluated in various scenarios in simulation and on the real robot.

The evaluation platform is the *Hector Tracker* robot. The mobile platform features adjustable flippers, a five DOF arm and multiple sensors. An environment map is generated in real-time using a rotating Velodyne VLP-16 Lidar located at the rear.

2 Related Work

2.1 Stability Margin

Postural stability is critical for a vehicle's ability to traverse uneven terrain. Various analytical stability margins have been proposed that predict the probability of tip-over given the vehicle location. Commonly, the calculation is based on the location of the COM and the support polygon which is defined as the convex hull of all ground contact points.

For the analysis of quadruped walking gaits McGhee and Frank [6] proposed a stability margin defined as the shortest distance from the COM, projected onto the plane of the support polygon, to any point on the boundary of the support polygon. An energy-based formulation was proposed by Messuri *et al.* [7]. Their stability margin is defined as the minimum impact energy which can be sustained by the vehicle without tipping over. Unlike the work by McGhee *et al.*, this formulation considers COM-height changes. The force-angle stability margin was proposed by Papadopoulos *et al.* [11] and has been used in numerous works [2–4, 8]. It combines the distance of the COM to the support polygon edges with the angles of the gravity vector to the edges. The least stable axis determines the stability of the whole system. A positive value indicates a stable position. Similar to the energy-based formulation by Messuri *et al.* [7], the measure is sensitive to height changes of the COM. Furthermore, the evaluation is computationally cheap. For these reasons, we utilize the force-angle stability margin.

2.2 Stability Control

We categorized the existing work on the prevention of vehicle tip-over on uneven terrain into two classes: *Pre-planning* and *reactive behavior.*

Reactive Behavior Grand *et al.* [4] optimized stability and traction of the wheel-legged robot *Hylos* by adjusting its posture. A different approach for the same robot was proposed by Besseron *et al.* [3]. They exploit existing redundancies of robot kinematics by decoupling control of posture and trajectory. Stability is optimized with a potential field formulation. In contrast, Ohno *et al.* [9] performed online reconfigurations of the flipper on a tracked vehicle to prevent tip-over around the roll axis. Ground contacts are approximated dependent on sub-track contact.

Pre-Planning Norouzi *et al.* [8] proposed a path planning method that also generates optimal configurations of the flipper and 1-DOF arm of the tracked *iRobot Packbot.* The optimization is embedded into the *A** search algorithm and considers visibility, traction, energy consumption and stability. The physics engine Open Dynamics Engine (ODE) is used together with a 3D model of the environment to predict contact points. The work of Beck *et al.* [2] was implemented on the same platform. Given a path, the approach optimizes flipper and arm position by considering stability, equal distribution of contact forces, low energy consumption and operation within nominal joint positions. The prediction of contact points is idealized and depends on the flipper position and terrain slope.

The presented methods of stability control are unsuitable for adoption on the Hector Tracker. With five DOF its arm kinematics is much more complex than previous evaluation platforms. Furthermore, these works did not address important aspects like environment- and self-collision avoidance.

3 Whole-Body Planning

The whole-body planner generates a motion plan to safely cross obstacles. The planner takes a path $\mathbf{s}$, defined by N_s equidistant waypoints, as input. Each waypoint $s_i = (x, y, \psi)^T, i = 1, ..., N_s$ is defined by a 2D-position (x, y) and a heading ψ in the xy-plane. The remaining pose components height z, roll ϕ and pitch θ are constrained by the ground geometry and estimated as part of the approach.

3.1 Cost Function

The goal of optimization is the maximization of robot stability by finding an optimal robot posture $\mathbf{p}_i^*$ at each point on the path s_i. The parameter vector $\mathbf{p} = \left(q_1, ..., q_{N_p}\right)$ specifies the joint angles of the robot, where N_p is the number of available DOF. It is evaluated using an appropriate objective function $\varphi(\mathbf{p})$, which has to be minimized:

$$\mathbf{p}^* = \arg\min_{\mathbf{p}} \varphi(\mathbf{p}) \tag{1}$$

Stability Criterion The main component of the objective function is the stability criterion. It is evaluated by applying a cost function $w(\cdot)$ to the force-angle stability margin β_i of every axis of the support polygon:

$$\varphi(\mathbf{p}) = \frac{1}{N_s} \sum_{i}^{N_s} w(\beta_i(\mathbf{p})) \tag{2}$$

where N_s specifies the number of support polygon edges. The ground contact points depend on $\mathbf{p}$ and are predicted by an appropriate contact estimation using a model of the world.

We propose to use an exponential weighting function:

$$w(x) = ae^{-bx+c} \tag{3}$$

Compared to a quadratic-inverse formulation as for example used in [4], it also punishes negative stabilities as would be the case for unstable postures.

Movement Penalty To prevent large movements between waypoints that increase the stability only slightly, a penalty term $r(\cdot)$ on joint movement is added to the objective:

$$\varphi(\mathbf{p}) = \frac{\sum_i^{N_s} w(\beta_j(\mathbf{p}))}{N_s} + K_m \sum_j^{N_p} r(p_j - q_j) \tag{4}$$

where $\mathbf{q}$ is the previous joint configuration and K_m is a factor that determines the trade-off between stability and trajectory execution time. It is reasonable to do large motions if it is coupled with a meaningful stability increase. The Lorentzian function (Equation 5) fulfills this requirement because the applied penalty is capped.

$$r_{lor}(x) = \ln\left(1 + \frac{x^2}{2\sigma^2}\right) \tag{5}$$

3.2 Optimization Constraints

To prevent damage to the robot, collisions with the environment and itself have to be avoided. This is achieved by modeling collisions as constraints of the optimization problem.

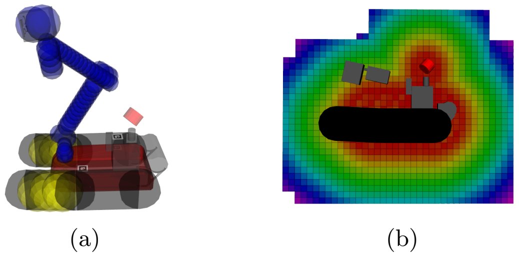
Environment Collision Avoidance The world representation used to formulate environment collision constraints is an Euclidean Signed Distance Field (ESDF) [10]. It consists of a 3-dimensional uniform grid with each cell containing the distance to the closest surface. Environmental collisions are only modeled for the manipulator arm since the flipper is intended to be in contact with the ground. Each link of the arm is approximated with multiple spheres (Figure 1a). One constraint per sphere is added to the optimization problem:

$$b_{i,j}(\mathbf{p}) = \varPhi({}^W c_{i,j}) - r_{i,j} > 0 \tag{6}$$

The function $\varPhi(\cdot)$ evaluates the ESDF at the sphere center ${}^W c_{i,j}$ with link index i and sphere index j to get the distance to the closest obstacle. Subsequently, the radius $r_{i,j}$ of the respective sphere is subtracted to check for a collision.

Self-Collision Avoidance To check for self-collisions efficiently, we distinguish between dynamic and static links. The position of dynamic links depends on the optimization parameter $\mathbf{p}$ whereas the static part is rigidly attached to the robot base. Analogous to the environment collision avoidance, the dynamic links of the robot are sampled with spheres (Figure 1a). One constraint per unique pair of spheres is added to the problem. The number of constraints is reduced by using the Allowed Collision Matrix (ACM). To prevent a collision, the distance between the sphere centers minus their radii must be greater than zero:

$$b_{i,j;m,n}(\mathbf{p}) = \|{}^B c_{i,j} -{}^B c_{m,n}\|_2 - r_{i,j} - r_{m,n} > 0; \quad i \neq m \tag{7}$$



(a) (b)

Fig. 1. Models used for collision avoidance. 1a shows the sphere-decomposition. Blue spheres model the arm and are used for environment and self-collision. The flippers, visualized with yellow spheres, are only considered for self-collision. 1b shows a slice of the robot ESDF used for self-collision checking against static links.

The first link is indexed with i and its spheres with j, the second link is indexed with m and its spheres with n.

To reduce computational cost, static links are represented by an ESDF. It is generated by computing the distances to the links in a grid around each link (Figure 1b). Collision checks are formulated analogous to environment collision checks (Equation 6).

3.3 Optimization Process

As gradient-based optimization techniques can be faster and more efficient than gradient-free ones, a differentiable objective function is desired. In the original definition of the force-angle stability margin [11], the sign of the angle θ_i between force vector $\hat{f}_i$ and axis normal $\hat{l}_i$ is given by the piecewise-defined function σ_i. As the discontinuity is not differentiable, a signed-angle formulation is used instead:

$$\theta_i = \operatorname{atan2}((\hat{f}_i \times \hat{l}_i) \cdot \hat{a}_i,\ \hat{l}_i \cdot \hat{f}_i) \tag{8}$$

Computation of the stability margin relies on the contact estimation to determine robot pose and support polygon. This function space is highly complex and discontinuous. To find feasible solutions, the optimization process is separated into two consecutive phases. In the first phase, only the contact geometry is optimized by adjusting the flipper position, eliminating the need for collision constraints. As the contact estimation is non-differentiable, only gradient-free solvers are suitable. Because we do not optimize directly for stability in this phase, an adjusted objective function is used that maximizes the area A of the support polygon:

$$\varphi(p_f) = -A - K_t \frac{C_t}{N_t} + K_c \frac{C_c}{N_c} \tag{9}$$

Additionally, two terms for assessment of ground geometry are introduced. The first term rewards contact points C_t with the tracks, the second term punishes contact points C_c with the chassis. The total number of sampled points on the tracks is labeled N_t and N_c for the chassis respectively.

In the next phase, the COM is shifted by varying the arm joint angles. The contact geometry is fixed and given by the first phase. Since the contact estimation is no longer involved, efficient gradient-based optimization algorithms can be applied that also include the presented collision constraints.

If the final optimization solution $\mathbf{p}^*$ is instable, the procedure stops as a safe traversal is not possible.

4 Evaluation

The proposed whole-body planning method was implemented using ROS[1] as middleware. The platform used for evaluation in simulation and reality is the *Hector Tracker* robot. To assess the effectiveness of the approach, the robot has to cross different obstacles without tip-over. The commanded path is a straight line and defined with a resolution of 0.02 m. The optimized robot postures $\mathbf{p}_i^*, i = 1, ..., N_s$ at each waypoint are connected by a joint trajectory which is executed in synchronization with vehicle movement. The environment map is represented by an ESDF. It was generated online with *Voxblox*[2] and captured using the VLP-16 Lidar and a depth camera. The contact estimation operates directly on the ESDF and is based on iterative optimization. The parameters used for evaluation were determined experimentally and are listed in Table 1. Derivatives of objective and constraint functions were determined with automatic differentiation using RBDL [5] and Ceres [1]. During trajectory execution, the actual stability margin is measured and compared to the predicted stability during pre-planning. In simulation, the ground truth is used as robot pose. On the real robot, a SLAM estimate based on fusing track odometry, IMU and Lidar data is used.

Table 1. Parameters used for evaluation in simulation and on the real robot.

Exponential		**Flipper Objective**		**Penalty Flipper**		**Penalty Arm**	
a	14	K_t	4	K_m	0.2	K_m	0.0025
b	-2.6	K_c	3	σ	0.2	σ	0.2
c	0.3						

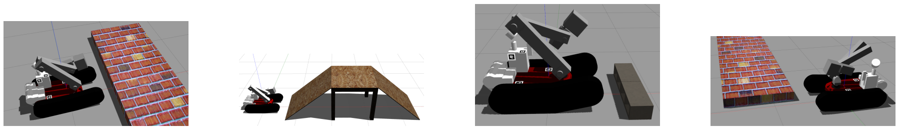
Simulation Experiments in simulation were performed in the open-source robotics simulator Gazebo[3]. Four different testing scenarios were chosen to cover a variety of obstacles: A step with a height of 0.15 m, a double ramp with an inclination angle of 40°, a cinder block with a height of 0.14 m and an asymmetrical step with the same height as the normal step (Figure 2).

The results are compared to a scripted behavior which switches hand-tuned joint configurations based on IMU feedback.

[1] http://www.ros.org/

[2] https://github.com/ethz-asl/voxblox

[3] http://gazebosim.org/



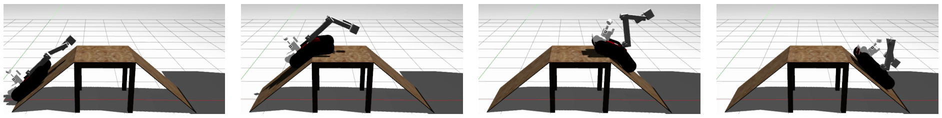
(a) Step (b) Ramp (c) Cinder Block (d) Asym. Step

Fig. 2. The four evaluation scenarios in Gazebo.

The minimum and average stabilities for each scenario are summarized in Table 2. Using whole-body planning, the robot managed to cross all obstacles successfully, as can be seen by the positive minimum stability. The reference behavior failed to traverse the ramp and fell during descend on the downwards slope. Overall, the proposed approach significantly outperformed the reference behavior in all scenarios in terms of average stability. Comparing the minimum stability, the reference behavior performed only slightly better in the cinder block task. Differences between measured and predicted stabilities are explained by approximation effects in map data and contact estimation. A detailed visualization of the generated motion plan in the ramp scenario can be seen in Figure 3.

Table 2. Comparison of stability margin results. *Plan* labels the predicted stability during planning, *Measured* refers to the actual stability during execution. Measured values during execution of the reference behavior are given in the last column.

	Plan		**Measured**		**Reference**	
	avg	min	avg	min	avg	min
Step	2.26	1.8	2.23	0.96	1.7	0.28
Ramp	1.98	0.63	1.75	0.04	1.16	-1.65
Cinder Block	2.3	1.78	1.92	0.02	1.68	0.13
Asymmetrical Step	1.68	0.87	1.57	0.07	0.57	0.05
Step (Real Robot)	2.26	1.8	2.23	0.96	-	-



(a) $x = 1\,\mathrm{m}$ (b) $x = 2\,\mathrm{m}$ (c) $x = 3.25\,\mathrm{m}$ (d) $x = 3.8\,\mathrm{m}$

Fig. 3. Visualization of the whole-body motion plan in the ramp scenario.

Real Robot To demonstrate, that the simulation results transfer to reality, we also evaluated our method on the real robot. Compared to simulation, the planner has to deal with inaccuracies of the robot model. The test was conducted using a single step with a height of 0.18 m. The robot successfully managed to climb the step as can be seen in Figure 4 and Table 2. Even with model inaccuracies, the difference between average expected stability and measured stability is only 0.03, showing that the method successfully extends to real robots. A full video of the experiment is available online[4].

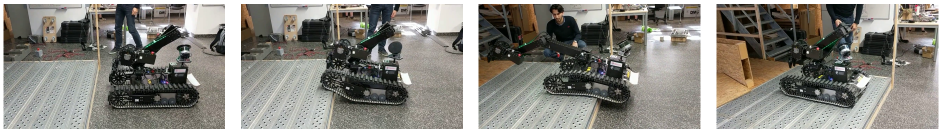


Fig. 4. Visualization of the whole-body motion plan on the real robot.

5 Conclusion

In this paper, we present a novel whole-body planning approach for traversal of obstacles with autonomous mobile ground robots. Based on a 3D map of the environment and a given path, optimal joint trajectories for manipulator arm and flippers are computed to maximize stability and safely cross obstacles. The method generalizes to arbitrary robot platforms using wheeled or tracked locomotion, as long as basic sensor requirements are satisfied. The effectiveness of the approach was shown by successfully crossing diverse obstacles in simulation and with a real robot. The proposed method consistently outperformed the hand-tuned reference in terms of average stability.

References

1. Agarwal, S., Mierle, K., Others: Ceres solver. http://ceres-solver.org
2. Beck, C., Miró, J.V., Dissanayake, G.: Trajectory optimisation for increased stability of mobile robots operating in uneven terrains. In: 2009 IEEE International Conference on Control and Automation, pp. 1913–1919. IEEE (2009)
3. Besseron, G., Grand, C., Amar, F.B., Bidaud, P.: Decoupled control of the high mobility robot hylos based on a dynamic stability margin. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2435–2440. IEEE (2008)
4. Grand, C., Benamar, F., Plumet, F., Bidaud, P.: Stability and traction optimization of a reconfigurable wheel-legged robot. The International Journal of Robotics Research **23**(10-11), 1041–1058 (2004)

[4] https://youtu.be/dVLi2w4l3Lg

5. Kudruss, M., Manns, P., Kirches, C.: Efficient derivative evaluation for rigid-body dynamics based on recursive algorithms subject to kinematic and loop constraints. IEEE Control Systems Letters **3**(3), 619–624 (2019)
6. McGhee, R.B., Frank, A.A.: On the stability properties of quadruped creeping gaits. Mathematical Biosciences **3**, 331–351 (1968)
7. Messuri, D., Klein, C.: Automatic body regulation for maintaining stability of a legged vehicle during rough-terrain locomotion. IEEE Journal on Robotics and Automation **1**(3), 132–141 (1985)
8. Norouzi, M., Miro, J.V., Dissanayake, G.: Planning stable and efficient paths for reconfigurable robots on uneven terrain. Journal of Intelligent & Robotic Systems **87**(2), 291–312 (2017)
9. Ohno, K., Takeuchi, E., Chun, V., Tadokoro, S., Yuzawa, T., Yoshida, T., Koyanagi, E.: Rollover avoidance using a stability margin for a tracked vehicle with sub-tracks. In: 2009 IEEE International Workshop on Safety, Security & Rescue Robotics (SSRR 2009), pp. 1–6. IEEE (2009)
10. Oleynikova, H., Taylor, Z., Fehr, M., Siegwart, R., Nieto, J.: Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1366–1373. IEEE (2017)
11. Papadopoulos, E., Rey, D.A.: The force-angle measure of tipover stability margin for mobile manipulators. Vehicle System Dynamics **33**(1), 29–48 (2000)