

Optimization-Based Planning for Autonomous Traversal of Obstacles with Mobile Ground Robots

Martin Oehler*

Stefan Kohlbrecher*

Oskar von Stryk*

* Simulation, Systems Optimization and Robotics Group
Department of Computer Science
Technische Universität Darmstadt, Germany

ABSTRACT

Mobile robotic platforms which are traversing unstructured environments with challenging uneven terrain are permanently endangered of falling over. Previous research on trajectory planning methods for the prevention of vehicle tip-over is mostly limited to basic mobility systems with only few degrees of freedom (DOF). This paper proposes a novel optimization-based planning approach that enables mobile robots to autonomously traverse obstacles and rough terrain more safely. A 3D world model as provided from external sensors like Lidar is used to compute a whole-body motion plan in advance by optimizing the trajectories of each joint. Active flipper tracks maximize ground contact for improved traction and, if available, manipulator arm joints are used to further improve stability metrics. Additional constraints prevent collisions with the environment and the robot itself. The presented approach makes only few assumptions about the robot's configuration and is applicable to a wide range of wheeled or tracked platforms. This is demonstrated by experimental evaluation for two different robots in simulation and for one physical robot. In four different test scenarios it is shown, that the proposed approach effectively prevents vehicle tip-over during traversal of uneven ground.

Keywords: Vehicle stability, Obstacle traversal, Tip-over prevention, Flipper control

1 Introduction

In recent years, demand has been growing for mobile robotic systems that can autonomously navigate and traverse unstructured environments, e.g. a partially collapsed building after an earthquake [9]. These environments have high requirements on the mobility of the robotic platform. There is a high risk of the robot tipping over while traversing rough terrain or climbing obstacles. Falling over can damage the robot and typically leads to the end of the mission. In the worst case, the robot's hardware may be lost.

In many situations, tipping can be prevented by reconfiguring the robot kinematics while driving. The robot can stay stable by using the arm joints to shift the center of mass (COM) and flipper joints to keep close contact with the

ground. Teleoperation of such challenging maneuvers is possible. However, it requires expert skills by the operator and becomes increasingly difficult with a higher number of DOF of the robot. Therefore, only slow operation is possible under high stress for the human operator. Also the risk of erroneous movements that lead to tip-over of the robot is high.

Autonomous capabilities can reduce operator stress by controlling manipulator arm and flipper joints to automatically balance the robot based on ground geometry information and stability assessment. We propose a division of this task 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

Contact author: Martin Oehler¹

¹E-mail: oehler@sim.tu-darmstadt.de

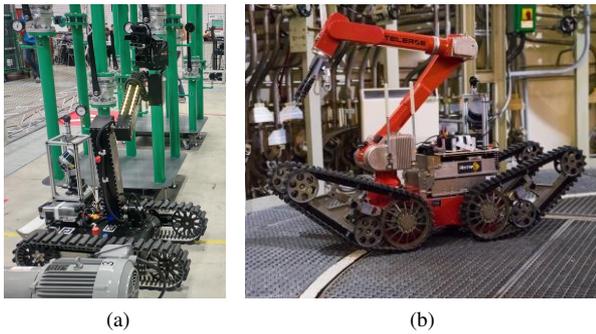


Figure 1 Evaluation platforms *Hector Tracker* (a) and *DRZ Telemax* (b)

resulting motion plan allows the robot to safely traverse difficult terrain. The proposed method is applicable to a wide range of wheeled or tracked robot platforms. This is demonstrated by evaluating the performance on two different robot platforms in various scenarios in simulation and on the real robot.

The first evaluation platform *Hector Tracker* (Figure 1a) features one flipper track on each side of the robot, which can be adjusted in the front part and a five DOF manipulator arm. It is equipped with multiple external sensors to perceive the environment.

The second robot *DRZ Telemax* (Figure 1b) is a highly mobile platform with four independent flippers. The manipulator arm features six DOF. On the back, multiple sensors are mounted similar to the *Hector Tracker*.

Both robots are capable of generating a 3D environment map in real-time using a rotating *Velodyne VLP-16* lidar.

This paper is an extension of work originally presented in the 28th International Conference on Robotics in the Alpe-Adria-Danube Region (RAAD 2019) [11]. For the first time, the extended flipper objective function is described in Section 3.3. New investigations and experimental evaluations for a second robot platform, the *DRZ Telemax*, are performed and reported. A detailed analysis and discussion of computational requirements in Section 4.3 gains further insights.

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.

A comprehensive overview and analysis of existing stability margins has been given by Garcia *et al.* [4]. For the analysis

of quadruped walking gaits, McGhee and Frank [7] 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.* [8]. 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.* [15] and has been used in numerous works [2, 3, 5, 10]. 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.* [8], the measure is sensitive to height changes of the COM. Furthermore, the evaluation is computationally cheap and therefore well suited for frequent evaluations within robot onboard trajectory planning. For these reasons, we utilize the force-angle stability margin in the proposed approach.

2.2 Stability Control

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

Reactive Behavior Grand *et al.* [5] 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 redundancies of robot kinematics by decoupling control of posture and trajectory. Stability is optimized with a potential field formulation. In contrast, Ohno *et al.* [12] performed online reconfigurations of the flipper on a tracked vehicle to prevent tip-over around the roll axis. Ground contacts are approximated depending on flipper track contact. A different approach to flipper control has been proposed by Okada *et al.* [13]. They equipped their robot with two lidars that scan the ground on each side of the robot in a line. The robot posture is adjusted to match the least-squares plane estimate of the ground surface. The desired pose is evaluated with a stability margin and adjusted until a stable pose is found.

Pre-Planning Norouzi *et al.* [10] 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 not well suited for adoption on the *Hector Tracker* or *DRZ Telemax*. With five and six DOF respectively, the arm kinematics of each of these platforms is very complex. Therefore, further important aspects like environment- and self-collision avoidance have to be taken into account which have not been considered by previous works.

3 Whole-Body Planning

The presented approach is a generic optimization-based whole-body planner that generates motion plans to safely cross obstacles and difficult terrain for a wide range of mobile ground robots. The planner takes a path s , defined by N_s equidistant waypoints, as input. Each waypoint $s_i = (x, y, \psi)^T, i = 1, \dots, N_s$ 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. The desired path has to be generated by an external planner that is outside of the scope of this work.

3.1 Cost Function

The optimization goal is the maximization of robot stability by finding an optimal robot configuration \mathbf{p}_i^* at each pose s_i on the path. The parameter vector $\mathbf{p} = (q_1, \dots, q_{N_p})$ specifies the flipper and manipulator arm 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:

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

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 a single edge of the support polygon. A differentiable objective function is obtained by computing the mean cost:

$$\varphi(\mathbf{p}) = \frac{1}{N_s} \sum_i^{N_s} w(\beta_i(\mathbf{p})) \quad (2)$$

where N_s specifies the number of support polygon edges. The computation of the force-angle stability margin requires the ground contact points of the robot. As these are unknown beforehand, they are predicted by a contact estimation based on a model of the world captured by the robot.

We propose to use an exponential weighting function for the stability margin:

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

with the parameters $a, b, c > 0$. Compared to a quadratic-inverse formulation as for example used in [5], the function also punishes a negative stability margin as would be the case for unstable postures.

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

$$\varphi(\mathbf{p}) = \frac{1}{N_s} \sum_i^{N_s} w(\beta_i(\mathbf{p})) + \frac{K_m}{N_p} \sum_j^{N_p} r(p_j - q_j) \quad (4)$$

where q_j is the previous configuration of joint j 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 margin increase. The Lorentzian function (Equation 5) fulfills this requirement because the applied penalty is capped.

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

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) [14]. 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. Similar to [16], each link of the arm is approximated with multiple spheres (Figure 2a). One constraint per sphere is added to the optimization problem:

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

The function $\Phi(\cdot)$ 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 We distinguish between dynamic and static links to check for self-collisions efficiently. The position of dynamic links depends on the optimization parameter \mathbf{p} whereas the static part is fixed to the robot base. Analogous to the environment collision avoidance, the dynamic links of the robot are sampled with spheres (Figure 2a). One constraint per unique pair of spheres is added to the problem. The number of constraints is reduced by using the

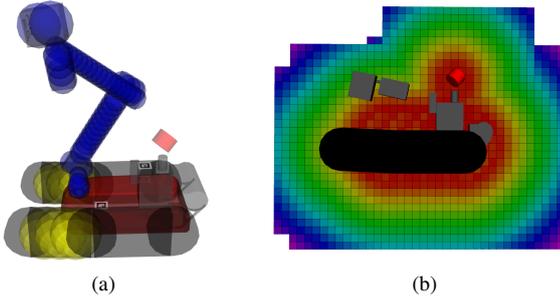


Figure 2 Models used for collision avoidance. 2a shows the sphere-decomposition. Blue spheres model the arm and are used for environment and self-collision. The flippers, visualized with yellow spheres, are only considered for self-collision. 2b shows a slice of the robot ESDF used for self-collision checking against static links. The color illustrates the distance to the collision geometry.

Allowed Collision Matrix (ACM). To prevent a collision, the distance between the sphere centers minus their radii must be greater than zero:

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

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 2b). Collision checks are formulated analogous to environment collision checks (Equation 6).

3.3 Optimization Process

In the previous Sections 3.1 and 3.2, objective function (Equation 4) and constraints (Equation 6 and 7) of the optimization have been introduced. In the following, it is presented how the function is optimized.

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 [15], 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:

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

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 (see Figure 3). In the first

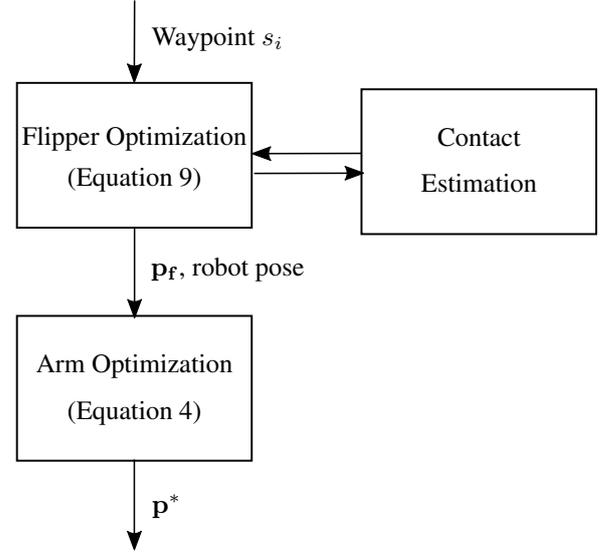


Figure 3 The optimization process is separated into two consecutive phases. The first phase determines the flipper position and, therefore, the robot pose. The second phase determines the final solution by optimization the arm configuration.

phase, only the contact geometry is optimized by adjusting the flipper position p_f , 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 $\varphi_F(p_f)$ is used that maximizes the area A of the support polygon:

$$\varphi_F(p_f) = -A + K_t \varphi_t(p_f) + K_c \varphi_c(p_f) + K_p \varphi_p(p_f) \quad (9)$$

Additionally, two terms for assessment of ground contact are introduced with the weights K_t and K_c and a position term $\varphi_p(p_f)$ with the weight K_p .

The first term φ_t rewards traction with the ground.

$$\varphi_t(p_f) = -\frac{1}{N_t} \sum_i^{N_t} \left(\max\left(\frac{-2}{\pi} \theta_i + 1, 0\right) \right) \quad (10)$$

The term sums up a reward for each contact point of the tracks with the ground, where 0 indicates no traction and 1 full traction. The reward is normalized by the total number of sampled points on the track N_t . θ_i is the angle between the negative gravity vector and the surface normal \vec{n}_i at contact point i :

$$\theta_i = \text{acos} \left((0, 0, 1)^T \vec{n}_i \right) \quad (11)$$

The second term φ_c punishes contact points with the chassis in order to avoid such undesired collisions.

$$\varphi_c(p_f) = \frac{C_c}{N_c} \quad (12)$$

where C_c is the number of chassis contacts and N_c the total number of sampled points on the chassis.

The third term linearly penalizes the position difference between the pose estimate ${}^W T_E$ of the contact estimation and the homogeneous transformation ${}^W T_S$ to the original waypoint s_i on the path:

$${}^S T_E = ({}^W T_S)^{-1} {}^W T_E \quad (13)$$

$$\varphi_p(p_f) = \max(0, \|{}^S \hat{r}_E\| - c_p) \quad (14)$$

where ${}^S \hat{r}_E \in \mathbb{R}^2$ is the xy-translatory component of the transformation ${}^S T_E$. The constant c_p specifies the allowed deviation.

In the second phase, the COM is shifted by adjusting the manipulator arm joint angles by optimizing the original objective function (Equation 4). The ground contact geometry is assumed 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 consider the collision constraints presented in Section 3.2.

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

4 Evaluation

The proposed whole-body planning method was implemented using ROS* as middleware. The *Hector Tracker* robot has been used for evaluation in simulation and on the real robot. Tests on the *DRZ Telemax* were performed in simulation only. While the robot features four independent flippers, front and rear flippers are only moved pair-wise in these tests.

Effectiveness and versatility of the approach are assessed by crossing four different types of obstacles without tip-over with both robots. The commanded path is a straight line and defined with a resolution of 0.02 m. The optimized robot configurations $\mathbf{p}_i^*, i = 1, \dots, N_s$ 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 is generated in real-time with *Voxblox*† and captured using a rotating VLP-16 lidar. Sensor noise is smoothed out by incrementally integrating laser scans into the map representation. The

*<http://www.ros.org/>

†<https://github.com/ethz-asl/voxblox>

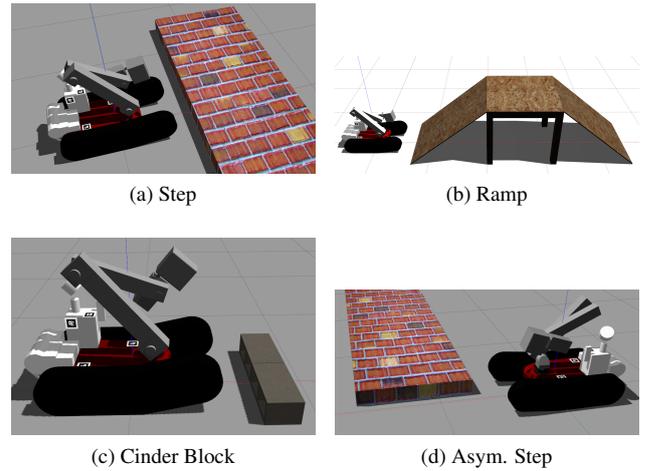


Figure 4 The four evaluation scenarios in Gazebo.

contact estimation operates directly on the ESDF and is based on iterative optimization. Derivatives of objective and constraint functions were determined with automatic differentiation using RBDL [6] and Ceres [1].

In Section 4.1 and 4.2, the predicted stability margin during planning is compared to the actual stability margin which is measured during trajectory execution. 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. This is followed by an analysis and discussion of the required computation time of a whole-body motion plan on the *DRZ Telemax* in Section 4.3.

4.1 Simulation

Experiments in simulation were performed in the open-source robotics simulator Gazebo‡. 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 4).

In case of the *Hector Tracker*, the results are compared to a scripted behavior which switches hand-tuned joint configurations based on IMU feedback. No such behavior exists for the *DRZ Telemax*. Comparing against a default configuration is not possible as the robot is unable to traverse the obstacles without reconfiguration. Therefore, no reference is available.

The results of the experiments with the *Hector Tracker* robot are summarized in Table I. The robot managed to traverse all four obstacles using the proposed whole-body planner, as can be seen by the positive minimum stability margin. The reference behavior failed to traverse the ramp

‡<http://gazebo.org/>

Table I - Evaluation results on the *Hector Tracker*. *Plan* labels the predicted stability margin during planning, *Measured* refers to the actual stability margin 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
Block	2.3	1.78	1.92	0.02	1.68	0.13
Asym.	1.68	0.87	1.57	0.07	0.57	0.05
Real robot	2.26	1.8	2.23	0.96	-	-

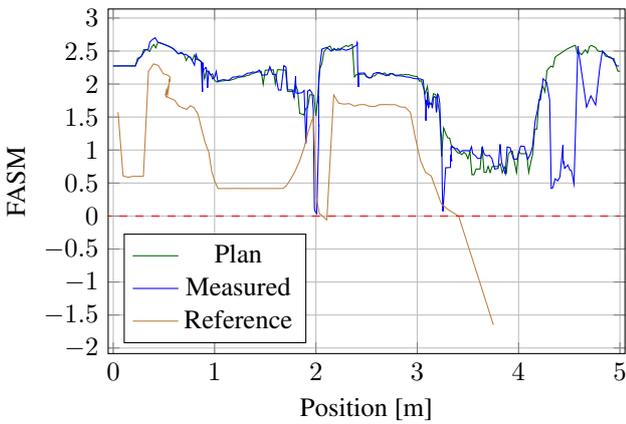


Figure 5 Stability graph of *Hector Tracker* in the ramp scenario.

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 margin. Differences between measured and predicted stability margin are explained by approximation effects in map data and contact estimation. Even though the robot crossed all obstacles, the minimum measured stability margin is close to zero in the ramp, cinder block and asymmetrical step scenarios. To explain this behavior, we take a closer look at the stability graph in the ramp scenario (Figure 5). A detailed visualization of the generated motion plan can be seen in Figure 6. The instabilities occur during the two transitions between slope and elevated floor at $x = 2$ and $x = 3.25$. The robot cannot keep contact to the ground with the flippers because only the front flippers can be adjusted. Due to this kinematic constraint, the robot is unable to traverse the obstacle without tipping in movement direction. As the robot tips into a stable state, this behavior is desired. Likewise, the *DRZ Telemax* crossed all obstacles successfully. The detailed results can be seen in Table II. In all scenarios, average and minimum measured stability are close to the predicted ones. Compared to the *Hector Tracker*, the *DRZ Telemax* features independently adjustable front and rear flippers. These allow the robot to cross the obstacles

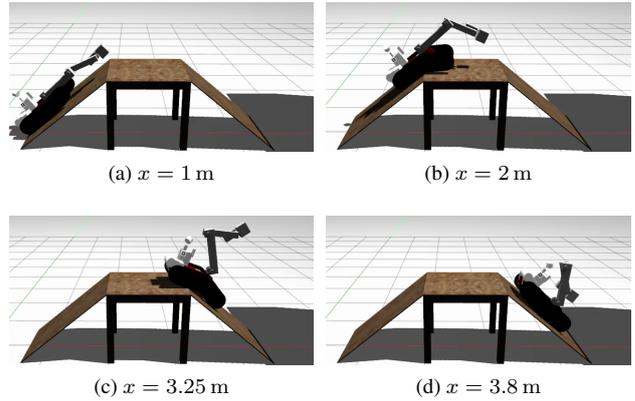


Figure 6 Visualization of the whole-body motion plan with *Hector Tracker* in the ramp scenario.

Table II - Evaluation results on the *DRZ Telemax*. *Plan* labels the predicted stability margin during planning, *Measured* refers to the actual stability margin during execution.

	Plan		Measured	
	avg	min	avg	min
Step	1.42	1.04	1.35	0.5
Ramp	1.35	0.58	1.23	0.64
Cinder Block	1.4	0.9	1.35	0.88
Asym. Step	1.1	0.2	1.05	0.35

without tipping motions. Figure 7 shows the motion plan on the asymmetrical step. It can be seen how the manipulator arm is shifted to the side to keep balance. The corresponding stability graph can be seen in Figure 8.

4.2 Real Robot

The evaluation on the real robot demonstrates, that the simulation results transfer to reality. 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 the motion plan visualization in Figure 9 and the measured stability margin in the last row of Table I. Even with model inaccuracies, the difference between average expected stability and measured stability is only 0.03, demonstrating that the method successfully extends to the real robot. A full video of the experiment is available online[§].

4.3 Computation Time

An important metric of the whole-body planner is the computation time to generate a plan. In this section, the computational requirements of the proposed approach are

[§]<https://youtu.be/dVLi2w4l3Lg>

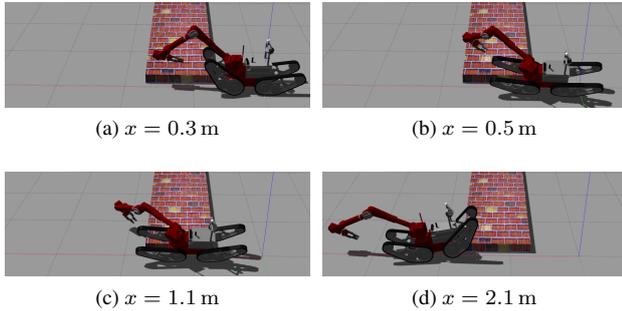


Figure 7 Visualization of the whole-body motion plan with *DRZ Telemax* on the asymmetrical step. The arm is moved to the side to balance the robot.

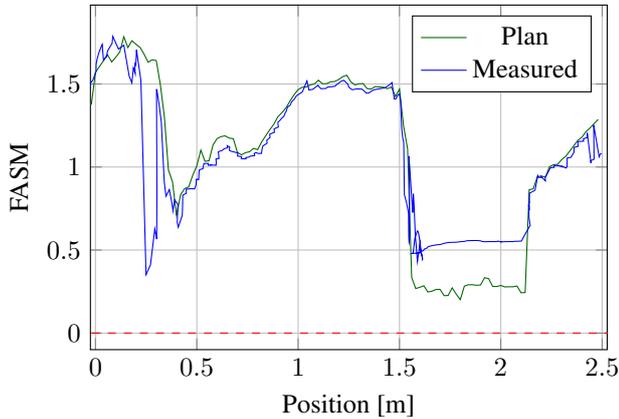


Figure 8 Stability graph of *DRZ Telemax* on the asymmetrical step.



Figure 9 Visualization of the whole-body motion plan on the real *Hector Tracker* robot

Table III - Computation time to generate the whole-body motion plan in each simulation scenario with the *DRZ Telemax* on the notebook CPU *Intel i7-4710HQ* @ 2.5 GHz.

Scenario	Traveled Distance [m]	Planning Time [s]	Time per Meter [s/m]
Step	2.8	142.59	50.93
Ramp	5	272.56	54.51
Block	2.1	110.64	52.69
Asym. Step	2.8	168.8	60.29

Table IV - Trajectory execution duration and average robot speed in each simulation scenario with the *DRZ Telemax*.

Scenario	Execution Duration [s]	Average Speed [m/s]
Step	83.7	0.033
Ramp	151.7	0.033
Cinder Block	66.4	0.032
Asym. Step	86.1	0.033

analyzed and discussed. Basis of the analysis is the *DRZ Telemax* in the four simulation scenarios. The results in each scenario can be seen in Table III. Computation was performed on the notebook CPU *Intel i7-4710HQ* at 2.5 GHz on a single core. Notably, the planning time is higher in the asymmetrical step scenario because the robot has to move the manipulator arm to the side to stay stable (see Figure 7) which requires more optimization steps. On average, the required planning time per meter is 54.6s. This equals a maximum speed of 0.018 m s^{-1} if the plan is generated just in time. We compare this to the trajectory execution time in Table IV. On average, the robot moved with a speed of 0.033 m s^{-1} . Therefore, to achieve real-time performance, a speed-up of the computation time of 1.78 is required. With further tuning to the algorithm and updated computing hardware, this seems achievable.

5 Conclusion

In this paper, we presented a novel whole-body planning approach for the autonomous traversal of challenging terrain with 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 a stability margin and traction. The method generalizes to arbitrary robot platforms using wheeled or tracked locomotion, as long as basic sensor requirements are satisfied. The versatility of the approach has been shown by successfully crossing diverse obstacles with two different robot platforms in simulation and a real robot. The proposed method consistently outperformed a hand-tuned reference in terms of average stability.

6 Acknowledgment

Research presented in this paper has been supported in parts by the German Federal Ministry of Education and Research (BMBF) within the subproject "Autonomous Assistance Functions for Ground Robots" of the collaborative A-DRZ project (grant no. 13N14861) and by the German Federal Ministry of Economics & Technology's "EXIST Forschungstransfer" (grant no. 03EFLHE061). This work has been performed in the context of the LOEWE center *emergenCITY*.

REFERENCES

- [1] S. Agarwal, K. Mierle, and Others. Ceres solver. <http://ceres-solver.org>.
- [2] C. Beck, J. V. Miró, and G. Dissanayake. Trajectory optimisation for increased stability of mobile robots operating in uneven terrains. In *2009 IEEE International Conference on Control and Automation*, pages 1913–1919. IEEE, 2009.
- [3] G. Besseron, C. Grand, F. B. Amar, and P. Bidaud. Decoupled control of the high mobility robot hylas based on a dynamic stability margin. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2435–2440. IEEE, 2008.
- [4] E. Garcia, J. Estremera, and P. G. De Santos. A classification of stability margins for walking robots. *Robotica*, 20(6):595–606, 2002.
- [5] C. Grand, F. Benamar, F. Plumet, and P. Bidaud. Stability and traction optimization of a reconfigurable wheel-legged robot. *The International Journal of Robotics Research*, 23(10-11):1041–1058, 2004.
- [6] M. Kudruss, P. Manns, and C. Kirches. 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.
- [7] R. B. McGhee and A. A. Frank. On the stability properties of quadruped creeping gaits. *Mathematical Biosciences*, 3:331–351, 1968.
- [8] D. Messuri and C. Klein. 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.
- [9] R. R. Murphy, S. Tadokoro, D. Nardi, A. Jacoff, P. Fiorini, H. Choset, and A. M. Erkmen. Search and rescue robotics. *Springer handbook of robotics*, pages 1151–1173, 2008.
- [10] M. Norouzi, J. V. Miro, and G. Dissanayake. Planning stable and efficient paths for reconfigurable robots on uneven terrain. *Journal of Intelligent & Robotic Systems*, 87(2):291–312, 2017.
- [11] M. Oehler, S. Kohlbrecher, and O. von Stryk. Whole-body planning for obstacle traversal with autonomous mobile ground robots. In *International Conference on Robotics in Alpe-Adria Danube Region*, pages 250–258. Springer, 2019.
- [12] K. Ohno, E. Takeuchi, V. Chun, S. Tadokoro, T. Yuzawa, T. Yoshida, and E. Koyanagi. 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)*, pages 1–6. IEEE, 2009.
- [13] Y. Okada, K. Nagatani, and K. Yoshida. Semi-autonomous operation of tracked vehicles on rough terrain using autonomous control of active flippers. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2815–2820. IEEE, 2009.
- [14] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto. Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1366–1373. IEEE, 2017.
- [15] E. Papadopoulos and D. A. Rey. The force-angle measure of tipover stability margin for mobile manipulators. *Vehicle System Dynamics*, 33(1):29–48, 2000.
- [16] D. Pavlichenko and S. Behnke. Efficient stochastic multicriteria arm trajectory optimization. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4018–4025. IEEE, 2017.