2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Local Behavior-Based Navigation in Rough Off-Road Scenarios based on Vehicle Kinematics

Patrick Wolf[1], Thorsten Ropertz[1], Moritz Oswald[1] and Karsten Berns[1]

***Abstract*— This paper describes a novel behavior-based local navigation approach for rough off-road scenarios. Trajectory candidates are generated based on vehicle kinematics and dynamics as well as the desired global trajectory. In contrast to on-road local navigation approaches, the work at hand proposes the use of a shiftable elevation grid map instead of occupancy maps since traversability in rough terrains does not only depend on location, but also on the robot's orientation. The traversability is evaluated by determining tire contact points with the terrain to take various different safety and efficiency aspects like underbody collisions and rollover risk into account. By exploiting the behavior-based control paradigm, the navigation approach can be easily extended and its robustness is shown in experimental evaluations using an Unimog U5023.**

***Keywords*— robotics, off-road navigation, behavior-based control, tentacles**

I. INTRODUCTION

In the area of on-road traffic, driver assistant systems are more and more evolving to autonomous driving systems. Thereby, the navigation is of special importance: Pathways as well as obstacles have to be detected and suitable trajectories to realize a safe and efficient ride calculated. In on-road scenarios the navigation is rather easy since the vehicles have to stay on the flat, drivable, strictly defined pathway while avoiding previously known obstacle classes, and following given traffic rules. Navigation in off-road scenarios is quite more challenging due to missing predefined pathways, which are known to be drivable. Instead, the local navigation approach has to judge the traversability based on noisy sensor data. Thereby, different aspects have to be regarded like collisions and roll-over risk.

This paper presents a new approach for behavior-based navigation in off-road scenarios. Raw distance data perceived by 3D range sensors is processed and quality assessment information is generated using the behavior-based perception approach described in [1]. A brief introduction to the applied behavior-based control architecture is given in section III. In order to increase the sensor data quality and the robustness of the navigation approach, the processed distance data is collected in a map. While for other approaches, especially in on-road scenarios, simple occupancy grid maps are sufficient, off-road-scenarios require more information since the traversability does not only depend on the grid cell and its surrounding, but also on the vehicle state like the orientation. Therefore, a shiftable elevation grid map (section IV) serves as a data base for the trajectory evaluation. In section V, the actual navigation approach is described. Possible trajectory candidates are generated based on the vehicle's kinematics and current orientation exploiting clothoids. The candidates are evaluated with respect to possible collisions (front and underbody) and rollover risk. By incorporating an adaptive weighting of the evaluation results, the selection process can respect various dynamically changing preferences like shortest or flattest path. The described approach has been experimentally evaluated using an Unimog U5023 in the V-Rep simulation environment as described in section VI. It has shown to correctly predict the vehicle state on the trajectory candidates and efficiently select suitable trajectories based on given preferences. Finally, a short summary and an outlook to ongoing work is given in section VII.

II. RELATED WORK

There are plenty of approaches existing which handle local on-road navigation but few of them are able to cope with off-road conditions as well. In [2], [3], the *TerraMax* and its navigation approach is presented which participated in the DARPA Grand Challenge from 2005. For localization, two different GPS receivers as well as a forward-looking vision system is used. Based on single and multi-plane laser scanner sensor data, the local path planner adapts the given route considering dynamic limits, road edges, terrain slope, local obstacles and more. The system creates a tree of trajectory candidates starting at the current position as root node and branches based on the dynamics and steering model. Candidate selection is based on a scoring algorithm considering the distance between trajectory candidate and base frame, resulting curvature, obstacle avoidance, and vehicle boundaries. In addition to the adapted trajectory, the local planner provides a speed limit based on the path geometry and current track conditions. While the scoring resembles the approach of the work at hand, the traversability assessment takes only obstacles into account and does not regard the ground surface geometry.

Another local navigation approach for off-road scenarios is presented in [4]. Instead of assessing the terrain traversability using predefined characteristics, a two stage learning algorithm is proposed. First, a short range, geometry-based local terrain classifier is trained using proprioceptive sensor data like terrain geometry from range sensors, appearance from cameras, and accelerations from IMUs. Due to the modularity of the behavior-based approach proposed in this paper, additional learned classifiers could also be added to

*This work is promoted by the European Union from the "European Regional Development Fund" (ERDF/EFRE).

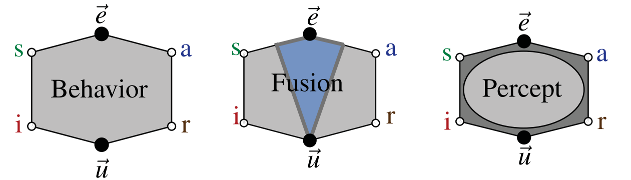
[1] Robotics Research Lab, Dep. of Computer Science, University of Kaiserslautern, 67663 Kaiserslautern, Germany, {patrick.wolf, ropertz, m_oswald11, berns}@cs.uni-kl.de

978-1-5386-3081-5/18/$31.00 ©2018 IEEE

extend the predefined, validated classifiers with high performance but possibly not predictable classifiers. The local classifier is then used to provide sample data for the second learning algorithm which implements a long range, image-based classifier that continuously generalizes geometry to appearance. This two stage learning approach allows for updating the long-range assessment during runtime and avoid myopic behavior. In contrast, the work at hand prevents myopic behavior by considering previous decisions.

III. Integrated Behavior-Based Control (iB2C)

The basic concept of behavior-based control architectures is that the overall system behavior emerges from the local interaction of simple behavior components which realize only little functionality. They have shown to be robust against environmental changes due to the partially overlapping functionality of the individual components and the dynamic arbitration process. One instance of the architecture class is the integrated behavior-based control (iB2C) architecture [5], [1] developed at the Robotics Research Lab of the University of Kaiserslautern. Complex control systems can be developed using three basic component types: behavior modules are used for control tasks, percept behaviors are suited for robot perception, and fusion behaviors coordinate the interaction of network components (see figure 1).



(a) Standard iB2C behavior $B = (f_a, F)$.

(b) Fusion iB2C behavior $F = (f_a, F)$.

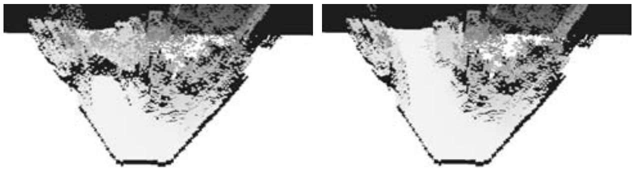
(c) Percept iB2C behavior $P = (f_a, f_{p_u}, F)$.

Fig. 1: iB2C network components [1].

All components provide the same unified interface consisting of stimulation s and inhibition i, which allow for adjusting the maximum relevance in the current system state. The target rating output r indicates the contentment of the behavior with the current system state and is defined by the activity function $f_a(\vec{e})$. The behavior's activity $a = \min(s \cdot (1-i), r)$ reflects the actual relevance of the behavior in the current system state and is used by fusion behaviors to perform the arbitration process. In addition, each behavior component provides an application specific interface consisting of the input vector $\vec{e}$ and output vector $\vec{u}$ containing arbitrary control and sense data. Thereby, the output vector is defined by the transfer function $F(\vec{e})$. For coordination purposes, there are different fusion approaches predefined.

IV. SHIFTABLE ELEVATION GRID MAP

In order to provide a robust environment representation for the navigation, a shiftable elevation grid map is proposed. It allows for decoupling the perception and control development processes and arbitrary data exchange in a generalized representation [6]. The map content shifts in discrete steps with respect to the vehicle's translations. Distance data from various sensor systems like laser scanners and stereo cameras are inserted into the map, which averages the height within regular grid cells to calculate an abstract representation of the 3D surface as shown in figure 2.



(a) Without height filter.

(b) With height filter.

Fig. 2: Shiftable elevation map of a 3D surface containing ground, obstacle and overhanging objects.

Unfortunately, noisy sensor data and negative obstacles insert artifacts and gaps into the map. More convenient navigation results and the avoidance of false positive obstacle interaction can be achieved through applying cell content filters:

A. Height Filter

The elevation map is a 2.5D map and cannot consider overhanging points without blocking the current pathway. Therefore, a height filter is applied. A perceived 3D point corresponds either to the ground, an obstacle, or an overhang. The latter case requires special attention since averaging of such points results in false obstacles. Therefore, an overhang point is rejected if it does not intersect with the working space of the robot. The point e_{min} represents the lowest height of all points within one grid cell and is used as lower limit. A point e_i is rejected if its relative height exceeds a threshold given by the vehicle height h_{vehicle} and a safety margin $h_{\text{tolerance}}$.

$$e_i \leq e_{\text{min}} + h_{\text{vehicle}} + h_{\text{tolerance}} \tag{1}$$

B. Gap Filter

In off-road environments, negative obstacles like holes can usually only be detected if they are close to the vehicle. Gaps in point clouds located farther away usually origin in occultation due to the perspective and can be only detected correctly when the sensors field of view covers the overall area [2].

A gap filter can be applied to prevent unnecessary slow downs caused by the low level safety system and reduces the number of holes within the map. The robot's tire diameter d_{tire} is considered and maximum immersion i_{max} derived

$$i_{\text{max}} = \frac{1}{2} d_{\text{tire}} - d_{\text{cell}} \tag{2}$$

with d_{cell} the size of a grid cell. The immersion distance i_{max} is used to virtually close gaps and to extended map borders by inserting additional cells.

V. TRAJECTORY GENERATION

Vehicles based on Ackermann kinematics can drive circular paths. Since the steering angle cannot be changed instantaneously, but continuously over time, possibly drivable trajectories are formed like *tentacles* starting with the current curvature and changing it continuously. While tentacle based approaches often utilize a occupancy grid map for planning [7], [8], the work at hand proposes the usage of clothoids for trajectory candidates in combination with elevation maps for trajectory planning. Occupancy maps provide insufficient information about the environment to enable a safe and robust navigation in extremely rough areas. Therefore, the surface's shape is additionally considered to retrieve more convenient trajectories. The best tentacle candidate is provided to a low level controller which executes the path.

A. Clothoids

Tentacles are defined by first order clothoids [9]. For a discrete curvature κ can be described as

$$\kappa(s) = as + b \tag{3}$$

where the angle θ changes over the distance s and time t

$$\theta(s) = \int_0^s \kappa(t)dt + \theta_0 \tag{4}$$

Clothoids are defined by Fresnel integrals

$$\gamma(s) = \left(\int_0^s \cos\theta(t)dt + x_0, \int_0^s \sin\theta(t)dt + y_0 \right), \tag{5}$$

which can be approximated by Taylor expansion for efficient computation [10].

B. Virtual Tentacle Sensor

Initially, a set of clothoids is generated based on the robot's current velocity and steering angle, and fitted onto the elevation map presented in section IV. Next, normals of the vehicle frame are calculated for the respective shape of the environment. Therefore, contact points of the wheels with the terrain are determined. Afterwards, the frame normal is calculated considering different vehicle properties as axle shrinking, mass distribution and hovering wheels. Based on the normals along the trajectory candidates, the vehicle state can be predicted and exploited for optimal path selection.

a) Pendulum Axes: The wheel contact points are calculated in 3D to avoid offsets during the determination of corresponding grid cells. The exact wheel positions $\vec{w_i}_{,j}$ with $i \in \{\text{front, rear}\}$ and $j \in \{\text{right, left}\}$ are calculated based on the previous normal $\vec{n}$ and current direction $\vec{d}$

$$\vec{w}_{\text{rear},j} = \vec{\mathrm{k}} \pm \frac{s_{\text{wheelbase}}\vec{n} \times \vec{d}}{2} \tag{6}$$

$$\vec{w}_{\text{front},j} = \vec{w}_{\text{rear},j} + s_{\text{axlebase}}\vec{d} \tag{7}$$

where $s_{\text{wheelbase}}$ and s_{axlebase} describe the lengths of wheel- and axle base. $\vec{\mathrm{k}}$ represents the kinematic center of the vehicle relative to the origin of the shiftable elevation map M. x- and y-values of the wheel points $w_{i,j}$ are used to determine corresponding cells of M and retrieve the height z.

Next, axle orientations $\vec{a}_i$ are determined, based on the 3D contact points $\vec{w}_{i,j}$ of tire pairs

$$\vec{a}_i = \vec{w}_{i,\text{right}} - \vec{w}_{i,\text{left}} \tag{8}$$

For the normal computation two cases have to be distinguished: Either all wheels touch the ground, or one wheel hovers. A two wheel case is neglected since the vehicle's frame always stabilizes on three wheels. Nonetheless, the event is detected by sensing normal shifts in successive three wheel contact detections.

b) 4 Wheel Contact: In a four wheel contact scenario, the front axle angle stays within a given threshold ϕ, which is defined by the vehicle's kinematic. For instance, the Unimog U5023's axis shrinks with $\phi = 30°$. All wheels touch the ground as long as the axis angles are within the limit, otherwise one wheel starts hovering.

The four wheel contact case normal $\vec{n}$ and support $\vec{s}$ are given by

$$\vec{n} = \vec{a}_{\text{rear}} \times \vec{c}_{\text{rear2front}} \tag{9}$$

$$\vec{s} = \vec{w}_{\text{rear, left}} + \frac{\vec{a}_{\text{rear}}}{2} \tag{10}$$

where $\vec{c}_{\text{rear2front}}$ is the vector from the center of the rear axle to the center of the front axle, with

$$\vec{w}_{\text{rl2fl}} = \vec{w}_{\text{front, left}} - \vec{w}_{\text{rear, left}} \tag{11}$$

$$\vec{c}_{\text{rear2front}} = -\frac{\vec{a}_{\text{rear}}}{2} + \vec{w}_{\text{rl2fl}} + \frac{\vec{a}_{\text{front}}}{2} \tag{12}$$

and $\vec{w}_{\text{rl2fl}}$ pointing from the left rear wheel $\vec{w}_{\text{rear, left}}$ to the left front wheel $\vec{w}_{\text{front, left}}$.

c) 3 Wheel Contact: In the 3 wheel contact case, one of two diagonal wheels can hover. Therefore, four vehicle frames are calculated based on respectively a single hovering wheel. Based on the frame orientation, the position of the center of gravity (CoG) is determined considering current axle weights and mass distribution. The vehicle's normal $\vec{n}$ is calculated for a hovering front right wheel $w_{\text{front, right}}$ as follows

$$\vec{w}_{\text{fr2rr}} = \vec{w}_{\text{rear, right}} - \vec{w}_{\text{front, right}} \tag{13}$$

$$\vec{c}_{\text{front}} = \vec{w}_{\text{front, right}} + \frac{\vec{a}_{\text{hover}}}{2} \tag{14}$$

$$\vec{c}_{\text{rear}} = \vec{w}_{\text{rear, right}} + \frac{\vec{a}_{\text{ground}}}{2} \tag{15}$$

$$\vec{c}_{\text{front2rear}} = \vec{c}_{\text{front}} - \vec{c}_{\text{rear}} \tag{16}$$

$$\vec{n} = \vec{d}_{\text{front2rear}} \times \vec{a}_{\text{ground}} \tag{17}$$

with $w_{\text{front, right}}$ lifted into the air. $\vec{a}_{\text{hover}}$ matches the front axle $\vec{a}_{\text{front}}$, as well as $\vec{a}_{\text{ground}}$ is equal to $\vec{a}_{\text{rear}}$. $\vec{w}_{\text{fr2rr}}$ points from the front right to the rear right wheel, $\vec{c}_{\text{front}}$ and $\vec{c}_{\text{rear}}$ are the centers of front and rear axes. The vector $\vec{c}_{\text{front2rear}}$ directs from front to the rear axle center. $\vec{a}_{\text{hover}}$ is calculated by rotating $\vec{a}_{\text{ground}}$ around $\vec{w}_{\text{fr2rr}}$ by the amount of the angle threshold ϕ. All other wheel calculations follow the presented scheme. All cases where the Euclidean distance from CoG to the hovering wheel position is larger than the distance to the diagonally opposite wheel are tagged as valid.

Finally, the angles between up to four valid normal candidates $\vec{\eta}_i$ at time i and the previous normal $\vec{n}_{i-1}$ are compared. The candidate with smallest angle difference is chosen as normal $\vec{n}_i$. The selection process prevents the normal from oscillating between multiple three wheel contacts during CoG transitions over the ground wheels' diagonal. Therefore, the normal vector $\vec{n}$ tilts consistently and smoothly.

C. Tentacle Rejection and Limitation

The extracted tentacles provide normals of the vehicle frame on discrete positions on each clothoid of the tentacle set. The normals on the vehicle's path are evaluated for constraints as collisions and rollovers. In order to limit the computational costs of the approach, the tentacle's length is limited to a safe operation space in the presence of a detected event.

a) Rollover Prediction: Normal projections are used to check for vehicle rollovers and cut the length of the clothoid before the event can occur. The normal on the clothoid is projected onto an ellipse. The area of the described ellipse describes maximal acceptable roll φ and pitch θ angles. The maximum components $x_{\max}, y_{\max}$ of the ellipse are defined as

$$(x_{\max}, y_{\max})^T = (\sin(\varphi), \sin(\theta))^T \tag{18}$$

The clothoid length is limited to the previous sample point P_{n-1} of the projected normal point $P_n = (x_n, y_n)$ which exceeds the limit

$$\left(\frac{x_n}{x_{\max}}\right)^2 + \left(\frac{y_n}{y_{\max}}\right)^2 \leq 1 \tag{19}$$

b) Underbody Collision Prediction: Normals and support vectors are used for collisions prediction with the vehicle's body. Hereby, the ground clearance g is compared to the difference of a plane defined by the wheels and the perceived environment represented by elevation map cells.

The support $\vec{s}$ and normal vector $\vec{n}$ define a plane E corresponding to the underbody of the vehicle

$$E : 0 \geq \vec{n} \cdot \vec{x} - (\vec{s} + g \cdot \vec{n}) \tag{20}$$

where $\vec{x}$ represents a grid cell of the elevation map with coordinates x, y and height z. All cells within the area defined by the wheel contact points are considered for collision prediction. Clothoids are shortened if the distance of a cell height z_i to the plane E is larger than clearance g.

D. Tentacle Selection

The best tentacle candidate is selected after clothoid generation and evaluation. A tentacle's relevance is weighted based on multiple factors as distance to target, flatness and consistency. Costs are derived by combining a set of cost factors c_i with a significance weight s_i for n tentacles. The tentacle with lowest total cost c_{total} wins and is selected for navigation

$$c_{\text{total}} = c_0 \cdot s_0 + c_1 \cdot s_1 + \ldots + c_n \cdot s_n \tag{21}$$

All cost values are normalized $c_i \in [0, 1]$. The significance s_i can be used to adapt the desired navigation behavior by granting larger or lower importance to certain properties.

a) Distance to target: Navigation costs c_{dist} increase for tentacles which do not point towards the robot's target

$$c_{\text{dist}} = \frac{d - d_{\min}}{d_{\max} - d_{\min}} \tag{22}$$

where d is the minimum distance of the selected tentacle to a goal point. $d_{\min}$ and $d_{\max}$ represent the minimum and maximum target distance of the overall tentacle set.

b) Flatness: Costs increase for rough surfaces and it is preferred to navigate on flat areas. Constant hill climbing and parallel inclination drives are challenging but uncritical in contrast to jumpy orientation changes. Therefore, the cost c_{flat} is defined as

$$c_{\text{flat}} = \sum_{i=0}^{n-1} |\alpha_i| \tag{23}$$

with α_i as angle between the normal n_i and n_{i+1}.

c) Consistency: Consistency costs prevent sudden tentacle changes during navigation. Therefore, the cost c_{cons} increase for switching tentacles.

$$c_{\text{cons}} = \sum_{i=0}^{n} |\vec{p}_i - p_{\vec{\text{current}}_i}| \tag{24}$$

with p_i the i-th sample point of the current tentacle and $p_{\vec{\text{current}}}$ the corresponding point of the currently selected tentacle.

E. Implementation

The navigation approach is implemented using iB2C as shown in figure 3. The percept GRID MAP holds and maintains the elevation map, which provides data to other percepts. Possible trajectories are generated by the TENT. GENERATOR. Since the whole navigation approach depends on the map quality, the map percept stimulates other percepts (green). For each tentacle, there are corresponding rejection (grey) and weighting percepts (yellow) instantiated. If a trajectory is rejected, the fusion inhibits the weighting (TENT. WEIGHTING). Finally, all trajectory candidates (their weighting) are connected to a maximum fusion (TENT. SELECTION), which forwards the trajectory with highest activity.

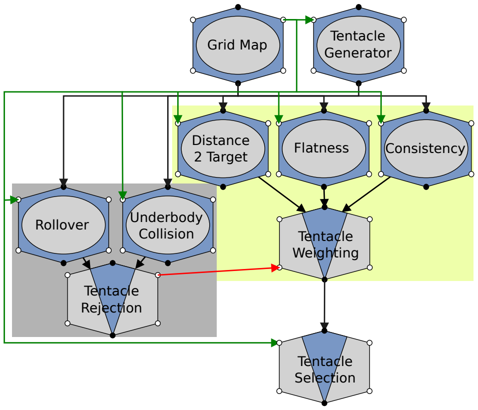
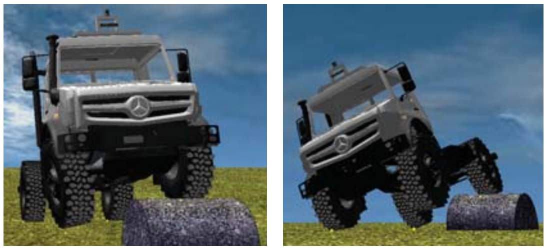


Fig. 3: iB2C network: grid map and tentacle generator (top), tentacle rejection (grey) and tentacle weighting (green)



(a) Front axle on obstacle. (b) Rear axle on obstacle.

Fig. 4: Pendulum axes of Unimog U5023. The rear axle stays in alignment with the vehicle frame both times, as the springs' ranges are smaller. This results in a straight vehicle frame (a) and a rolling frame (b).

VI. EXPERIMENTS

The elevation map and trajectory generation was tested in simulated and real world tests.

a) Implementation: The system is implemented using Finroc, a C++ and Java robot control framework consisting of a highly modular structure. It is real-time capable and offers lock-free, zero-copy implementations [11].

b) Simulation Environment: The simulation tool V-Rep was used which features C/C++, Python, Java, Lua, Matlab, Octave and Urbi controllers. It is a versatile simulation engine for physic computations and sensor processing [12]. Finroc accesses the engine via a C++ API.

c) Simulated Unimog U5023: The V-Rep model of the Unimog U5023 is depicted in figure 4. It considers mass distribution, pendulum axes, differential locks, working group and standard gears, as well as tire pressure. The simulation model implements all relevant vehicle specification and characteristics [13].

The Unimog U5023 uses several sensors to perceive environment and vehicle state. A Bumblebee XB3 stereo vision system is mounted on top of the cabin. Supplementary, a u-blox NEO-7P GNSS, a Mircostrain 3DM-GX3-25 IMU and vehicle CAN data are provided.

A. Normal Generation and Rejection

The subsystems of the off-road navigation framework are tested for component validation. The Unimog U5023 passed an cylindrical obstacle (see figure 4) subsequently with both left wheels. Hereby, body and tires were lifted depending on the robot's position, center of mass and pendulum axes.

The corresponding clothoids, elevation map and normals are shown in figure 5. Normals are depicted in green and three wheel contacts in blue. The perceived bump is visualized using gray tiles. The tentacle sensor predicts a series of three wheel contacts in front of the obstacle and during rear axle bump contact. The roll orientation is predicted to change significantly during the second three wheel contact sequence. Measured orientations are shown in figure 6. The vehicle's axle interleaves to a maximum of 30° while the main body normal changes in pitch ((-6°)) and roll ((2°)). During the second three wheel phase, roll reaches a maximum ((15°)) and pitch changes slightly ((3°)). Roll decreases ((-2°)) after passing the obstacle which is caused by the Unimog U5023's springs. The measured results match the normal prediction.

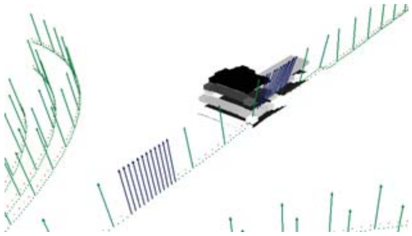


Fig. 5: Path candidates and normals of the cylindrical obstacle. Predicted three wheel case normals are shown in blue.

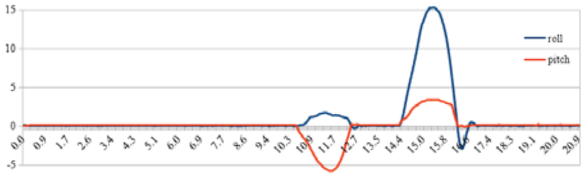
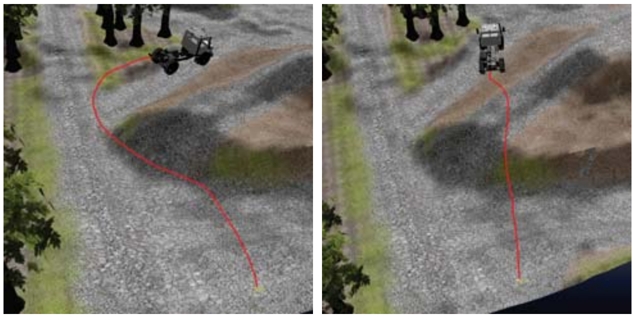


Fig. 6: Actual roll (blue) and pitch (orange) angles correspond to predicted normals.

B. Simulated Off-Road Navigation

The restriction of normals and weighting was tested in a simulated, 3D reconstructed Daimler testing area Ötigheim. The Unimog U5023 repeated a target approach with different normal restrictions. In a first run, normals were allowed to reach a maximum inclination of 10° and in a second test 30°. The traveled trajectories are visualized in figure 7.



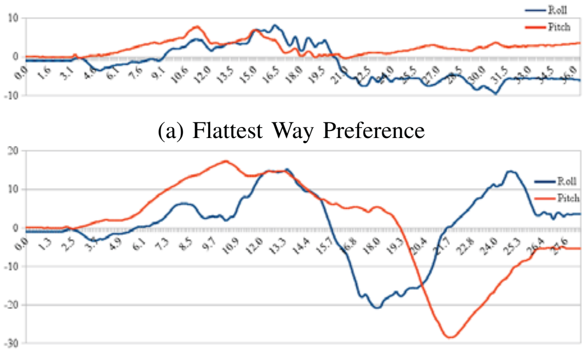
(a) Flattest Way Preference (b) Shortest Way Preference

Fig. 7: The vehicle approaches a target point. Tentacles are restricted to flat pathways with normal maximums of 10° (a) and rough surfaces with normal maximums of 30° (b).

The measured orientation angles are depicted in figure 8. It can be seen that the Unimog U5023 fulfilled all restrictions in both tests. First, a flat trajectory was selected. In the second test, a rough maneuver with high pitch an roll angles was performed using a short but difficult pathway.

C. Off-Road Navigation

The approach was also tested in a real world environment. The GatorX855D robot of the Robotics Research Lab

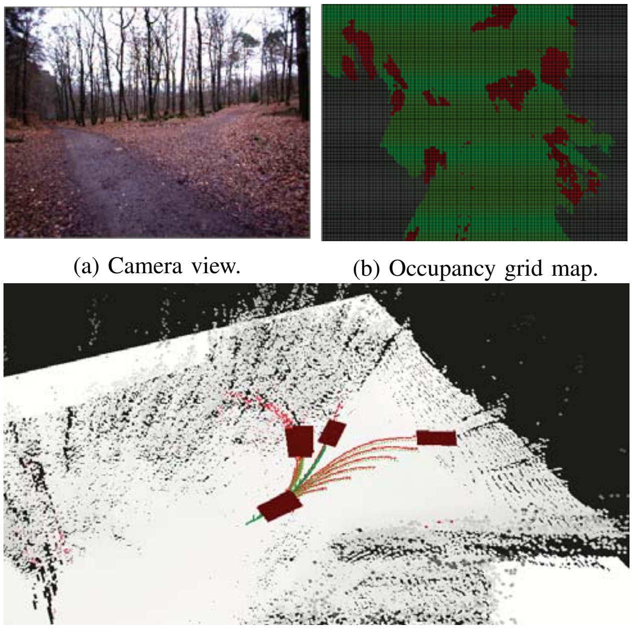


(a) Flattest Way Preference

(b) Shortest Way Preference

Fig. 8: Roll and pitch angles during point approach. Angles of flat trajectory are within 10° limit (a). Large and multiple changes in the vehicle frame are limited to 30° (b).

followed a generated trajectory over a distance of 750 m through the Rhineland-Palatinate forest. The sensor layout was similar to previous tests. Exemplary results are displayed in figure 9 and can be compared to an occupancy grid map.



(a) Camera view.

(b) Occupancy grid map.

(c) Elevation grid map, tentacles and predicted collisions.

Fig. 9: Elevation map and evaluated tentacles. (a) shows the camera view, (b) occupancy map, (c) the elevation map, clothoids and the predicted collisions (red box).

VII. CONCLUSION

This paper presents a new approach for local off-road navigation with respect to off-road vehicle kinematics. Clothoid-based trajectory candidates are generated based on the vehicle's capabilities. While in the area of on-road navigation, occupancy grid maps are usually sufficient for traversability evaluation, in rough off-road terrains a suitable assessment is not possible based on this strong abstraction. In order to assess the pathway candidates under the consideration of vehicle kinematics, pendulum axis, CoG, and tire contact points, a shiftable elevation map is introduced for 3D surface representation. A dynamic selection process is presented, which evaluates different aspects of the trajectories in form of cost functions and allows for online adaptation by adjusting the weighting process. Thereby, especially safety relevant events as collisions or rollovers are robustly predicted. Experimental evaluation of the proposed navigation approach proved a robust robot behavior during testing and correct prediction of the vehicle's state.

Future work will extend the approach by considering additional surface properties. Currently, only geometrical information is regarded, while other properties are also highly relevant. Adding environmental features to the map as compression, cohesion, or friction will improve the system and can be considered during tentacle evaluation.

REFERENCES

[1] T. Ropertz, P. Wolf, and K. Berns, "Quality-based behavior-based control for autonomous robots in rough environments," in *Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2017)*, O. Gusikhin and K. Madani, Eds., vol. 1. Madrid, Spain: SCITEPRESS – Science and Technology Publications, Lda, July 26–28 2017, pp. 513–524, iSBN: 978-989-758-263-9.

[2] D. Braid, A. Broggi, and G. Schmiedel, "The terramax autonomous vehicle concludes the 2005 darpa grand challenge," in *Intelligent Vehicles Symposium, 2006 IEEE*. IEEE, 2006, pp. 534–539.

[3] ——, "The terramax autonomous vehicle," *Journal of Field Robotics*, vol. 23, no. 9, pp. 693–708, 2006.

[4] M. Bajracharya, A. Howard, L. H. Matthies, B. Tang, and M. Turmon, "Autonomous off-road navigation with end-to-end learning for the lagr program," *Journal of Field Robotics*, vol. 26, no. 1, pp. 3–25, 2009. [Online]. Available: http://dx.doi.org/10.1002/rob.20269

[5] M. Proetzsch, "Development process for complex behavior-based robot control systems," Ph.D. dissertation, Robotics Research Lab, University of Kaiserslautern, München, Germany, June 2010.

[6] B.-H. Schäfer, M. Proetzsch, and K. Berns, "Action/perception-oriented robot software design: An application in off-road terrain," in *IEEE 10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Hanoi, Vietnam, December 17–20 2008.

[7] F. v. Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller, and H.-J. Wuensche, *Driving with Tentacles - Integral Structures for Sensing and Motion*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 393–440.

[8] M. Himmelsbach, T. Luettel, F. Hecker, F. von Hundelshausen, and H.-J. Wuensche, "Autonomous off-road navigation for mucar-3," *KI - Künstliche Intelligenz*, vol. 25, no. 2, pp. 145–149, 2011.

[9] D. Carroll, E. Hankins, E. Köse, and I. Sterling, "A survey of the differential geometry of discrete curves," *arXiv preprint arXiv:1311.5862*, 2013.

[10] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*. Courier Corporation, 1964, vol. 55.

[11] M. Reichardt, T. Föhst, and K. Berns, "Introducing finroc: A convenient real-time framework for robotics based on a systematic design approach," Robotics Research Lab, Department of Computer Science, University of Kaiserslautern, Kaiserslautern, Germany, Technical Report, 2012.

[12] M. Freese, S. Singh, F. Ozaki, and N. Matsuhira, "Virtual robot experimentation platform v-rep: A versatile 3d robot simulator." in *International Conference on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR)*, ser. Lecture Notes in Computer Science, N. Ando, S. Balakirsky, T. Hemker, M. Reggiani, and O. von Stryk, Eds., vol. 6472. Springer, 2010, pp. 51–62.

[13] A. Rödler, "Unimog. body and implement mounting directives. U4023/U5023 Euro VI," Daimler AG, Mercedes-Benz Special Trucks, Wörth, Germany, Manual, August 2014.