Verti-Wheelers: Wheeled Mobility on Vertically Challenging Terrain

Aniket Datar, Chenhui Pan, and Xuesu Xiao

***Abstract*— Most autonomous navigation systems assume wheeled robots are rigid bodies and their 2D planar workspaces can be divided into free spaces and obstacles. However, recent wheeled mobility research, showing that wheeled platforms have the potential of moving over vertically challenging terrain (e.g., rocky outcroppings, rugged boulders, and fallen tree trunks), invalidate both assumptions. Navigating off-road vehicle chassis with long suspension travel and low tire pressure in places where the boundary between obstacles and free spaces is blurry requires precise 3D modeling of the interaction between the chassis and the terrain, which is complicated by suspension and tire deformation, varying tire-terrain friction, vehicle weight distribution and momentum, etc. In this paper, we present a learning approach to model wheeled mobility, i.e., in terms of vehicle-terrain forward dynamics, and plan feasible, stable, and efficient motion to drive over vertically challenging terrain without rolling over or getting stuck. We present physical experiments on two wheeled robots and show that planning using our learned model can achieve up to 60% improvement in navigation success rate and 46% reduction in unstable chassis roll and pitch angles.**

I. INTRODUCTION

Wheeled robots, arguably the most commonly used mobile robot type, have autonomously moved from one point to another in a collision-free and efficient manner in the real world, e.g., transporting materials in factories or warehouses, vacuuming our homes or offices, and delivering food or packages on sidewalks. Thanks to their simple motion mechanism, most wheeled robots are treated as rigid bodies moving through planar workspaces. After tessellating their 2D workspaces into obstacles and free spaces, classical planning algorithms plan feasible paths in the free spaces that are free of collisions with the obstacles.

However, recent advances in wheeled mobility have shown that even conventional wheeled robots (i.e., without extensive hardware modification such as active suspensions or adhesive materials) have previously unrealized potential to move over vertically challenging terrain (e.g., in mountain passes with large boulders or dense forests with fallen trees), where vehicle motion is no longer constrained to a 2D plane (Fig. 1). In those environments, neither assumptions of rigid vehicle chassis and clear delineation between obstacles and free spaces in a simple 2D plane are valid. Thanks to the long suspension travel and reduced tire pressure, off-road vehicle chassis are able to drive *over* obstacles (rather than to *avoid* them) and experience significant deformation to conform with the irregular terrain underneath the robot, which will be otherwise deemed as non-traversable according to conventional navigation systems. Therefore, autonomously navigating wheeled robots in vertically challenging terrain without rolling over or getting stuck requires a precise understanding of the 3D vehicle-terrain interaction.

All authors are with the Department of Computer Science, George Mason University {adatar, cpan7, xiao}@gmu.edu

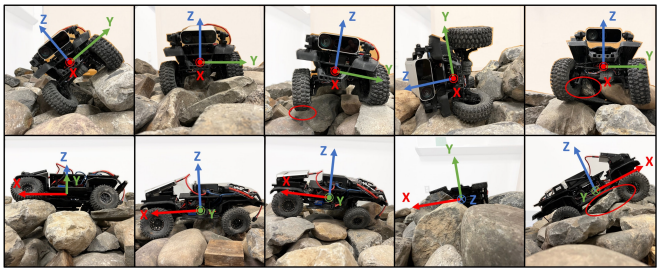

Fig. 1: Front and side view (1st and 2nd row) of a wheeled robot navigating vertically challenging terrain: (from left to right) large roll angle, stable chassis, suspended wheel, roll-over, and get-stuck.

In this paper, we develop a learning approach to model 3D vehicle-terrain interactions and plan vehicle trajectories to drive wheeled robots on vertically challenging terrain. Considering the difficulty in analytically modeling and computing vehicle poses using complex vehicle dynamics in real time, we adopt a data-driven approach to model the forward vehicle-terrain dynamics based on terrain elevation maps along potential future trajectories. We develop a Wheeled Mobility on Vertically Challenging Terrain (WM-VCT) planner, which uses our learned model's output in a novel cost function in 3D and produces feasible, stable, and efficient motion plans to autonomously navigate wheeled robots on vertically challenging terrain. We present extensive physical experiment results on two wheeled robot platforms and compare our learning approach against four existing baselines and show that our learned model can achieve up to 60% improvement in navigation success rate and 46% reduction in unstable chassis roll and pitch angles.

II. APPROACH

The difficulties in navigating a wheeled mobile robot on vertically challenging terrain are two fold: (1) the high variability of vehicle poses due to the irregular terrain underneath the robot may overturn the vehicle (rolling-over, 4th column in Fig. 1); (2) not being able to identify that a certain terrain patch is beyond the robot's mechanical limit and therefore needs to be circumvented may get the robot stuck (immobilization, 5th column in Fig. 1). Therefore, this work takes a structured learning approach to address both challenges by learning a vehicle-terrain forward dynamics model based on the vertically challenging terrain underneath

the vehicle, using it to rollout sampled receding-horizon trajectories, and minimizing a cost function to reduce the chance of rolling-over and immobilization and to move the vehicle toward the goal.

A. Motion Planning Problem Formulation

Consider a discrete vehicle dynamics model of the form $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t)$, where $\mathbf{x}_t \in X$ and $\mathbf{u}_t \in U$ denote the state and input space respectively. In the normal case of 2D navigation planning (Fig. 2 left), $X \subset \mathbb{SE}(2)$ and $X = X_{\text{free}} \cup X_{\text{obs}}$, where X_{free} and X_{obs} denote free spaces and obstacle regions. $\mathbf{x}_t$ includes the translations along the $\mathbf{x}$ and $\mathbf{y}$ axis (x and y) and the rotation along the $\mathbf{z} = \mathbf{x} \times \mathbf{y}$ axis (yaw) of a fixed global coordinate system. For input, $\mathbf{u}_t = (v_t, \omega_t) \in U \subset \mathbb{R}^2$, where v_t and ω_t are the linear and angular velocity. Finally, let $X_{\text{goal}} \subset X$ denote the goal region. The motion planning problem for the conventional 2D navigation case is to find a control function $u : \{t\}_{t=0}^{T-1} \rightarrow U$ that produces an optimal path $\mathbf{x}_t \in X_{\text{free}}, \forall t \in \{t\}_{t=0}^{T}$ from an initial state $\mathbf{x}_0 = \mathbf{x}_{\text{init}}$ to the goal region $\mathbf{x}_T \in X_{\text{goal}}$ that follows the system dynamics $f(\cdot, \cdot)$ and minimizes a given cost function $c(x)$, which maps from a state trajectory $x : \{t\}_{t=0}^{T} \rightarrow X$ to a positive real number. In many cases, $c(x)$ is simply the total time step T to reach the goal. Considering the difficulty in finding the absolute minimal-cost state trajectory, many mobile robots use sampling-based motion planners to find near-optimal solutions.

Conversely, in our case of wheeled mobility on vertically challenging terrain, vehicle state $X \subset \mathbb{SE}(3)$ (i.e., translations and rotations along the $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ axis) with the same input $\mathbf{u}_t = (v_t, \omega_t) \in U \subset \mathbb{R}^2$. The system dynamics enforces that $\mathbf{x}_t$ is always "on top of" a subset of X_{obs} (i.e., vertically challenging terrain underneath and supporting the robot) or some boundary of X (i.e., on a flat ground) due to gravity, requiring a 3D, 6-DoF vehicle-terrain dynamics model in $\mathbb{SE}(3)$ (Fig. 2 right).

B. Vehicle-Terrain Dynamics Model Learning

Compared to the simple 2D vehicle dynamics in $\mathbb{SE}(2)$, our non-rigid vehicle-terrain dynamics on vertically challenging terrain in $\mathbb{SE}(3)$ becomes more difficult to model, considering the complex interaction between the terrain and chassis via the long suspension travel and deflated tire pressure of off-road vehicles to assure adaptivity and traction (Fig. 1). Therefore, this work adopts a data-driven approach to learn the vehicle-terrain dynamics model, which can be used to rollout trajectories for subsequent planning.

To be specific, $\mathbf{x}_t = (x_t, y_t, z_t, r_t, p_t, \phi_t)$, where the first and last three denote the translational (x, y, z) and rotational (roll, pitch, yaw) component respectively along the $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ axis. Note that unlike most 2D navigation problems in which the next vehicle state $\mathbf{x}_{t+1}$ only relies on the current vehicle state $\mathbf{x}_t$ and input $\mathbf{u}_t$ alone, our next vehicle state is additionally affected by the vertically challenging terrain underneath and in front of the vehicle in the current time step, denoted as $\mathbf{m}_t$. Therefore, the forward dynamics on vertically challenging terrain can be formulated as

$$\mathbf{x}_{t+1} = f_\theta(\mathbf{x}_t, \mathbf{u}_t, \mathbf{m}_t), \tag{1}$$

which is parameterized by θ and will be learned in a data-driven manner. Training data of size N can be collected by driving a wheeled robot on different vertically challenging terrain and recording the current and next state, current terrain, and current input: $\mathcal{D} = \{\langle \mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{m}_t, \mathbf{u}_t \rangle_{t=1}^{N}\}$. Then we learn θ by minimizing a supervised loss function:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{(\mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{m}_t, \mathbf{u}_t) \in \mathcal{D}} \| f_\theta(\mathbf{x}_t, \mathbf{u}_t, \mathbf{m}_t) - \mathbf{x}_{t+1} \|_H, \tag{2}$$

where $||\mathbf{v}||_H = \mathbf{v}^T H \mathbf{v}$ is the norm induced by a positive definite matrix H, used to weigh the learning loss of the different dimensions of the vehicle state $\mathbf{x}_t$. The learned vehicle-terrain forward dynamics model $f_\theta(\cdot, \cdot, \cdot)$ can then be used to rollout future trajectories for minimal-cost planning.

C. Sampling-Based Receding-Horizon Planning

We adopt a sampling-based receding-horizon planning paradigm, in which the planner first uniformly samples input sequences up until a short horizon H, uses the learned model f_θ to rollout state trajectories, evaluates their cost based on a pre-defined cost function, finds the minimal-cost trajectory, executes the first input, replans, and thus gradually moves the horizon closer to the final goal. In this way, the modeling error can be corrected by frequent replanning. However, an under-actuated wheeled robot, i.e., using $\mathbf{u}_t = (v_t, \omega_t) \in U \subset \mathbb{R}^2$ to actuate $\mathbf{x}_t = (x_t, y_t, z_t, r_t, p_t, \phi_t) \in X \subset \mathbb{SE}(3)$ subject to f_θ, may easily end up in many terminal states outside of X_{goal}, which the vehicle cannot escape and recover from, i.e., rolling over or immobilization (getting stuck) due to excessive roll and pitch angles, irregular terrain geometry, and large height change, e.g., on a large rock. Therefore, while our goal is still to minimize the traversal time T leading to X_{goal}, for our receding-horizon planner, we seek to optimize five cost terms on a state trajectory $\mathbf{x}_{0:H} = \{\mathbf{x}_t\}_{t=0}^{H}, \text{s.t.}, \mathbf{x}_{t+1} = f_\theta(\mathbf{x}_t, \mathbf{u}_t, \mathbf{m}_t), \forall t < H$, which starts at the current time 0 and ends at the horizon H, to avoid these two types of terminal states on vertical challenging terrain and also move the robot towards the goal:

$$\begin{aligned} c(\mathbf{x}_{0:H}) = w_1 c_{\text{rp}}(\mathbf{x}_{0:H}) + w_2 c_{\text{tg}}(\mathbf{x}_{0:H}) + w_3 c_{\text{hc}}(\mathbf{x}_{0:H}) \\ + w_4 c_{\text{mb}}(\mathbf{x}_{0:H}) + w_5 c_{\text{est}}(x_H), \end{aligned} \tag{3}$$

where $c_{\text{rp}}(\cdot)$, $c_{\text{tg}}(\cdot)$, and $c_{\text{hc}}(\cdot)$ denote the cost corresponding to the robot's (extensive) roll and pitch angle, (irregular) underneath terrain geometry, and (large) terrain height change respectively; $c_{\text{mb}}(\cdot)$ is the cost of moving out of the observable map boundary; $c_{\text{est}}(\cdot)$ is the estimated cost to reach the final goal region X_{goal} from the state on the horizon x_H, which can be computed by the Euclidean distance $c_{\text{est}}(x_H) = ||x_H - x_G||_2$, where x_G is any state inside X_{goal}. w_1 to w_5 are corresponding weights for the cost terms.

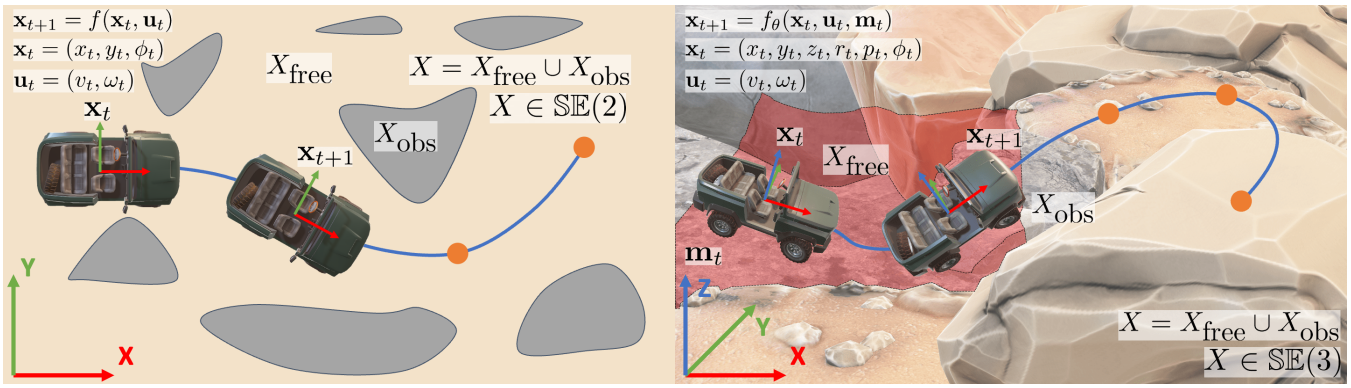

Fig. 2: 2D navigation in $\mathbb{SE}(2)$ vs. 3D, 6-DoF navigation on vertically challenging terrain in $\mathbb{SE}(3)$.

D. Modeling Rolling-Over and Immobilization

Vehicle roll-over is often associated with large roll and pitch angles, which we therefore seek to minimize along the state trajectory. Note that roll r_t and pitch p_t are part of the vehicle state $\mathbf{x}_t$. Therefore, we design a cost term that considers the absolute values of roll and pitch:

$$c_{\text{rp}}(\mathbf{x}_{0:H}) = w_{1,1} \sum_{t=0}^{H} |r_t| + w_{1,2} \sum_{t=0}^{H} |p_t|, \tag{4}$$

where $w_{1,1}$ and $w_{1,2}$ weigh the effect of the absolute value of roll and pitch.

Similarly, vehicle immobilization often happens when the vehicle state does not change from time to time due to irregular underneath terrain geometry. Therefore, trajectories on which vehicle state significantly changes, especially along the translational dimension x and y, are encouraged:

$$c_{\text{tg}}(\mathbf{x}_{0:H}) = -w_{2,1} \sum_{t=1}^{H} |x_t - x_{t-1}| - w_{2,2} \sum_{t=1}^{H} |y_t - y_{t-1}|, \tag{5}$$

where $w_{2,1}$ and $w_{2,2}$ weigh the effect of the displacement along x and y direction.

Furthermore, the vehicle should prefer gentle slope rather than large height change to avoid immobilization, so we encourage small displacement in the z direction along the trajectory:

$$c_{\text{hc}}(\mathbf{x}_{0:H}) = \sum_{t=1}^{H} |z_t - z_{t-1}|. \tag{6}$$

With the cost function (Equation (3)) and cost terms (Equation (4), (5), and (6)) defined, we can use any motion planner to find the minimal-cost 6-DoF state trajectory $\mathbf{x}_{0:H}$ (see details in Section III).

III. IMPLEMENTATION

We present implementation details of our WM-VCT navigation planner onboard two physical wheeled vehicles.

A. Physical Robots and Vertically Challenging Testbed

We implement our WM-VCT planner on two open-source wheeled robot platforms, the Verti-Wheelers (VWs), one with six wheels (V6W, $0.863\text{m} \times 0.249\text{m} \times 0.2\text{m}$) and the other with four wheels (V4W, $0.523\text{m} \times 0.249\text{m} \times 0.2\text{m}$). Both robots are equipped with a Microsoft Azure Kinect RGB-D camera with a 1-DoF gimbal actuated by a servo to fixate the field of view on the terrain in front of the vehicle regardless of chassis pose. NVIDIA Jetson computers (ORIN and Xavier for V6W and V4W respectively) provide onboard computation. We use low-gear and lock both front and rear differentials to improve mobility on vertically challenging terrain. Both robots are tested on a $3.1\text{m} \times 1.3\text{m}$ rock testbed (with the highest vertical point of X_{obs} reaching 0.6m) composed of hundreds of rocks and boulders of an average size of 30cm, similar to the size of the robots (Fig. 1). The rocks and boulders on the testbed are shuffled many times during experiments. The vehicle state estimation for $\mathbf{x}_t$ is provided by an online Visual Inertia Odometry system provided by the `rtabmap_ros` package. In order to represent $\mathbf{m}_t$, we process the RGB-D input into an elevation map, a 2D grid where each pixel (8mm resolution) indicates the height of the terrain at that point.

B. WM-VCT Planner Implementation

1) Dynamics Model Decomposition: We decompose the full 6-DoF vehicle-terrain dynamics model $\mathbf{x}_{t+1} = (x_{t+1}, y_{t+1}, z_{t+1}, r_{t+1}, p_{t+1}, \phi_{t+1}) = f_\theta(\mathbf{x}_t, \mathbf{u}_t, \mathbf{m}_t)$ into three parts: We utilize a planar Ackermann-steering model to obtain approximate $(x_{t+1}, y_{t+1}, \phi_{t+1})$ based on (x_t, y_t, ϕ_t) and $\mathbf{u}_t$; z_{t+1} is determined by the value of the elevation map $\mathbf{m}_t$ at (x_{t+1}, y_{t+1}) (line 9); We use a neural network to predict the roll r_{t+1} and pitch p_{t+1} angles based on $\mathbf{m}_t$, r_t, and p_t (line 11, potentially with additional history values). In practice, we find such a decomposition very efficient and sufficiently accurate in capturing the 6-DoF vehicle dynamics with a very small amount of training data (approximately 30 minutes) and leave learning the full dynamics model as future work. We use two 40×100 ($0.32\text{m} \times 0.8\text{m}$) elevation maps centered at both the robot's current and next position and aligned with the current and next yaw angle, i.e., underneath $(x_t, y_t)/(x_{t+1}, y_{t+1})$ and aligned with ϕ_t/ϕ_{t+1} (line 10). In our neural network model, a fully connected sequential head (8000-64-32-8 neurons) processes the $2 \times 40 \times 100$ elevation maps into a 8-dimensional embedding, which is then concatenated with the 8-dimensional embedding from

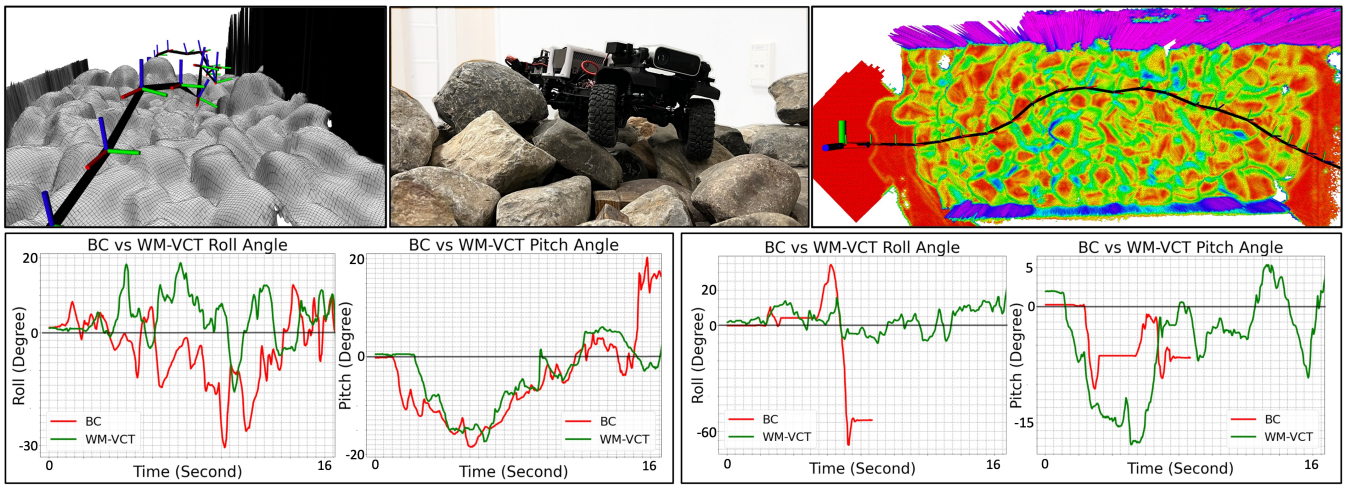

Fig. 3: Physical Experiments: The V6W (top middle) navigates through a vertical challenging environment (front and top view of the elevation map shown in top left and top right with the planned 6-DoF vehicle poses and trajectory in black); roll and pitch values of two successful WM-VCT trials (green lines at the bottom) are shown, while BC suffers from larger values in the first (red lines bottom left) and fails the second (red lines bottom right).

the last and current roll and pitch angles, r_{t-1}, r_t, p_{t-1}, and p_t, and fed into two more fully connected layers (16-8-2), before finally producing the next roll r_{t+1} and pitch p_{t+1} values. To train the model for each vehicle, we use 43161 data frames for V6W and 41367 for V4W in the open-source Verti-Wheelers datasets, roughly 30 minutes of data each demonstrating both VWs crawling over different vertically challenging testbeds.

2) Sampling and Roll-Out: For the sampling-based receding-horizon planner, we keep the linear velocity v constant (0.1m/s) and sample 11 angular velocities ω, or in our case, steering curvatures evenly from $[-0.78\text{rad}, 0.78\text{rad}]$ (line 1). With a step size of 1 second, we roll out our vehicle dynamics model with the 11 (v, ω) pairs five times (roll-out horizon H_{r}, lines 7-17), evaluate the trajectory costs (Equation (3) to (6), line 18), and expand the search tree again from the 3rd state (update horizon H_{u}, lines 21-23) on the lowest-cost trajectory using the same 11 velocity pairs. We repeat this process ten times (max iteration I) and find the overall minimal-cost trajectory of horizon 30 ($\mathbf{x}_{0:30}$).

3) Motion Controller: We implement a low-level controller operating at a frequency of 30Hz, with the objective of tracking the trajectory that incurs the lowest cost. Specifically, aiming to reach the next state $\mathbf{x}_{t+1}$ from the current state $\mathbf{x}_t$, the linear velocity controller determines the throttle command (ranging from $[-1.0, 1.0]$) by considering the current pitch angle p_t: the controller switches throttle command among three intervals, i.e., 0.15 for pitch less than -5°, 0.20 for pitch from -5° to 5°, and 0.30 for pitch greater than 5°. Such a mechanism approximately maintains constant velocity with respect to changing terrain. The steering angle is calculated by measuring the angle between the current yaw angle ϕ_t and the line connecting the current 2D position (x_t, y_t) to the next position (x_{t+1}, y_{t+1}). Meanwhile, while the receding-horizon planner constantly replans at 2Hz, the controller will also trigger instant replanning when the distance between the robot and the planned trajectory exceeds a predefined threshold (0.4m).

IV. EXPERIMENTS

Our proposed WM-VCT navigation planner is compared against four baselines. The three baseline algorithms developed with the open-source Verti-Wheelers project, i.e., Open-Loop (OL), Rule-Based (RB), and Behavior Cloning (BC), are implemented on our robots and compared against WM-VCT. Please visit our project website for the experimental results and videos `https://cs.gmu.edu/~xiao/Research/Verti-Wheelers/`

V. CONCLUSIONS

We present a learning approach to enable wheeled mobility on vertically challenging terrain. Going beyond the current 2D motion planning assumptions of rigid vehicle bodies and 2D planar workspaces which can be divided into free spaces and obstacles, our WM-VCT planner first learns to model the non-rigid vehicle-terrain forward dynamics in $\mathbb{SE}(3)$ based on the current vehicle state, input, and underlying terrain. Leveraging the trajectory roll-outs under a sampling-based receding-horizon planning paradigm using the learned vehicle-terrain forward dynamics, WM-VCT constructs a novel cost function in 3D to prevent the vehicle from rolling-over and immobilization when facing previously non-traversable obstacles. We show that our WM-VCT planner can produce feasible, stable, and efficient motion plans to drive robots over vertically challenging terrain toward their goal and outperform several state-of-the-art baselines on two physical wheeled robot platforms.