Differentiable Wheel Terrain Interaction Model and Pose Predictions on Uneven Terrain.

Amith Manoharan[1], Aditya Sharma[2], Himani Belsare[2], Kaustab Pal[2], K. Madhava Krishna[2], Arun Kumar Singh[1]

***Abstract*— Off-road navigation of autonomous vehicles requires the use of $6dof$ models for planning safe paths that satisfy stability constraints. The existing approaches pose prediction approaches mostly rely on training neural networks from vehicle motion data. However, these approaches are data intensive, struggle to generalize to novel scenes and require navigating the vehicle over potentially dangerous terrains for data collection.**

In this paper, we present a model-based approach that only requires an elevation map of the terrain (such as pointcloud obtained from a LiDAR). We formulate the wheel-terrain interaction and the resulting $6dof$ pose prediction of the vehicle as a non-linear least squares (NLS) problem. Importantly, we can leverage implicit differentiation rules to compute the gradient of the predicted pose with respect to the input parameters. We also briefly discuss how such differentiable models can be leveraged for gradient-based planning over uneven terrains.

I. Introduction

Off-road navigation of autonomous wheeled vehicles forms an important component in applications such as forestry, construction, and search and rescue. Trajectory planning over uneven terrains requires predicting the $6dof$ pose of the vehicle considering the wheel-terrain interaction. This can be achieved in two-ways. First, we can employ complex physics-engines [1], [2], which are computationally too expensive to be used for online planning. Second, we can learn the interaction and pose prediction model by fitting neural networks to vehicle motion data on different terrains [3]. However, purely learning based approaches are data-hungry, struggle to generalize to novel scenarios and may require vehicle to navigate over potentially dangerous terrains for data collection.

Contributions: We derive a set of coupled non-linear equations that views the wheel-terrain interaction as a closed-loop kinematic chain. These equations are solved as a non-linear least squares (NLS) problem and provides an implicit mapping between the yaw-plane motion of the vehicle and pitch, roll, height and contact-points variation over the terrains. Importantly, we show implicit gradients from NLS can be used in downstream trajectory planning. We also demonstrate that the prediction from our NLS-based approach closely matches the output from a high-fidelity physics engine. As a result, our NLS can be viewed as parallelizable and differentiable world-model, albiet just at the kinematic level.

[1] are with the Institute of Technology, University of Tartu, Tartu, Estonia.
[2] are with RRC, IIIT Hyderabad, India.

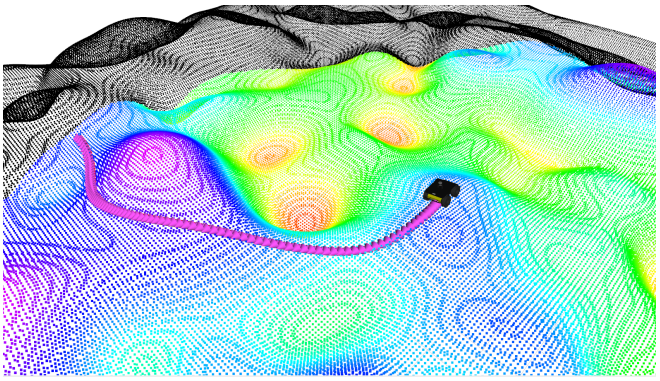


Fig. 1: A Husky robot is shown navigating in an uneven terrain. The black points represent the complete terrain and the multi-color points represent a local patch. The trajectory (pink) successfully avoids the high-cost regions to reach the goal.

II. Main Algorithmic Results

In this section, we first present a simple approach for fitting a functional form to a terrain elevation map and subsequently leveraging it for pose prediction. We also briefly discuss how the latter can be used in downstream planning tasks.

A. Modeling the Terrain in a Functional Form

We assume that the elevation data, which maps the height z^j of the terrain to the (x^j, y^j), can be obtained online as point clouds with an on-board LiDAR. We derive an analytical relationships of the form:

$$z^j = f(x^j, y^j), \tag{1}$$

by approximating f in terms of Fourier basis functions:

$$\begin{aligned} f(x^j, y^j) = \sum_{n=1}^{N} & a_n \cos\left(\omega_{1,n} x^j + \omega_{2,n} y^j\right) \\ & + b_n \sin\left(\omega_{3,n} x^j + \omega_{4,n} y^j\right), \end{aligned} \tag{2}$$

where $\omega_{1,n}, \omega_{2,n}, \omega_{3,n}, \omega_{4,n}$ are the frequencies, N is the total frequencies, and a_n, b_n are the weights affiliated with each function, which can be obtained by solving the regression problem given below.

$$\sum_{j=1}^{M} \| f(x^j, y^j) - z^j \|_2^2, \tag{3}$$

where M is the number of points in the given point-cloud. An example fit acquired over terrain patches of radius $7m$ is shown in Fig.3.

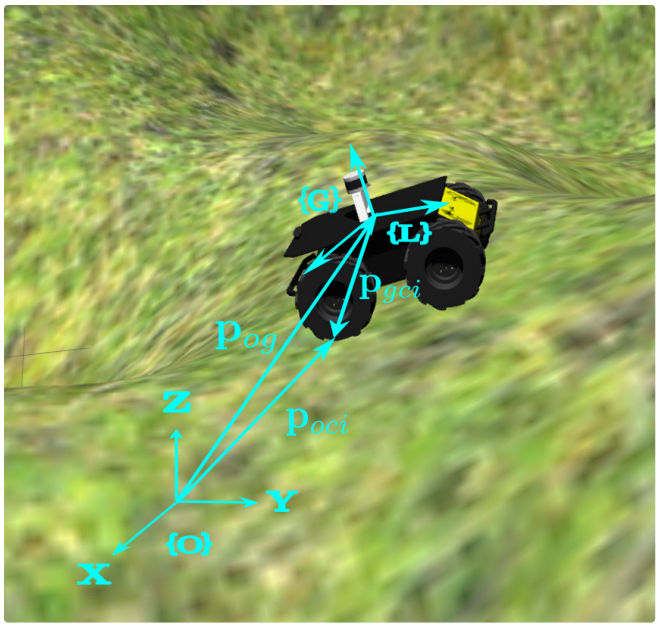
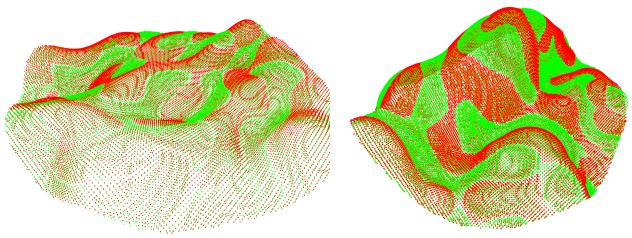


Fig. 2: The Husky robot and the geometry vectors describing the holonomic constraints.



(a) (b)

Fig. 3: The Red color represents the ground truth, and the green color represents the prediction.

B. Differentiable Wheel-Terrain Interaction

We define the yaw plane states of the vehicle at time k using the vector $\mathbf{x}_k = \begin{bmatrix} x_k & y_k & \alpha_k \end{bmatrix}^T$, where (x_k, y_k) is the position, and α_k is the yaw angle of the vehicle. For vehicles with no-active suspension, the evolution of the yaw plane configuration, i.e., $\mathbf{x}_k$, can be directly controlled, whereas the rest of the attitude variables, i.e., z_k coordinate, roll β_k, and pitch γ_k, is a function of $\mathbf{x}_k$ and the terrain geometry, which can be represented as:

$$\begin{aligned} z_k &= s_1(x_k, y_k, \alpha_k), \\ \beta_k &= s_2(x_k, y_k, \alpha_k), \\ \gamma_k &= s_3(x_k, y_k, \alpha_k). \end{aligned} \tag{4}$$

We define the following reference frames as shown in Fig. 2. The global reference frame is $\{O\}$, and the reference frame $\{G\}$ moves along with the vehicle, attached to its center. A similar frame $\{L\}$ is defined with an orientation the same as the vehicle. The vector defining the wheel-ground contact point $\mathbf{p}_{oc,i}$ can be obtained straight from the terrain equation. Then, the contact points can be devised by first following through $\mathbf{p}_{og}$ that identifies the center of the vehicle and then moving along $\mathbf{p}_{gc,i}$. This understanding is encoded using the following equations, similar to the loop closure equations for parallel manipulators [4] [5].

$$\mathbf{p}_{og} + \mathbf{p}_{gc,i} = \mathbf{p}_{oc,i}, \tag{5}$$

$$\mathbf{p}_{gc,i} = \mathbf{R} \begin{bmatrix} \delta_i h & r_i w & -(l_i) \end{bmatrix}, \forall i = 1,2,3,4, \tag{6}$$

$$\delta_i = \begin{cases} 1, & i = 1,4, \\ -1, & i = 2,3, \end{cases}$$

$$r_i = \frac{2.5 - i}{|2.5 - i|},$$

$$\mathbf{p}_{og} = \begin{bmatrix} x_k & y_k & z_k \end{bmatrix}, \tag{7}$$

$$\mathbf{p}_{oc,i} = \begin{bmatrix} x_{c,i_k} & y_{c,i_k} & z_{c,i_k} \end{bmatrix}. \tag{8}$$

The vector $\mathbf{p}_{gc,i}$ describes the contact points in the vehicle local frame $\{L\}$ multiplied with the rotation matrix between the $\{O\}$ and $\{L\}$ (refer Fig. 2). The constants h and w are the half-width and half-breadth of the chassis, r_i and δ_i are variables used to ensure the proper sign of w and h corresponding to each vertex of the chassis, and l_i are the equivalent leg lengths, which also include the radius of the wheels. Since $\mathbf{p}_{oc,i}$ also satisfy the terrain equation (1), it is possible to compose:

$$z_{c,i_k} = f(x_{c,i_k}, y_{c,i_k}). \tag{9}$$

Expanding (5) and stacking alongside (9), it is possible to obtain the following set of coupled non-linear equations.

$$g_i(\mathbf{x}_k, \mathbf{u}_k) = 0, \tag{10}$$

where $\mathbf{u}_k(\mathbf{x}_k) = \begin{bmatrix} z_k & \beta_k & \gamma_k & x_{c,i_k} & y_{c,i_k} & z_{c,i_k} \end{bmatrix}^T$. We define the pose prediction as a non-linear least squares (NLS) problem written as follows:

$$\mathbf{u}_k^*(\mathbf{x}_k) = \min_{\mathbf{u}_k} \|g_i(\mathbf{x}_k, \mathbf{u}_k)\|_2^2. \tag{11}$$

Implicit Differentiation For an efficient solution of the trajectory optimization problem introduced later, it is important to compute the Jacobian of $\mathbf{u}_k^*(\mathbf{x}_k)$ with respect to its input parameter $\mathbf{x}_k$. However, the relationship between $\mathbf{u}_k^*$ and $\mathbf{x}_k$ does not have an analytical form and thus requires tools from implicit differentiation. The derivation is based on Dini's implicit function theorem [6] applied to the first-order optimality condition. We define the following proposition that has been adapted from [7] for our pose prediction NLS, where $\mathrm{D}(.)$ represents the differential operator [7].

Proposition 1: Consider the NLS problem (11). We can define $\mathbf{H} = \mathrm{D}^2_{\mathbf{u}_k\mathbf{u}_k}\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k(\mathbf{x}_k)) \in \mathbb{R}^{m \times m}$, where $\mathbf{g}(.)$ is obtained by stacking $g_i(.)$ and $\mathbf{B} = \mathrm{D}^2_{\mathbf{x}_k\mathbf{u}_k}\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k(\mathbf{x}_k)) \in \mathbb{R}^{m \times n}$. Then, the Jacobian of the optimal pose is

$$\mathrm{D}\mathbf{u}_k^*(\mathbf{x}_k) = -\mathbf{H}^{-1}\mathbf{B},$$

with the assumption that $\mathbf{H}$ is non-singular.

Proof: The first-order optimality condition of our NLS is given by $\mathrm{D}_{\mathbf{u}_k}\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{0}_{1 \times m}$. Differentiating both

sides of this optimality condition with respect to $\mathbf{x}_k$ provides us the following equation

$$\begin{aligned}\mathbf{0}_{m\times n} &= \mathrm{D}(\mathrm{D}_{\mathbf{u}_k}\mathbf{g}(\mathbf{x}_k,\mathbf{u}_k))^T \\ &= \mathrm{D}^2_{\mathbf{x}_k\mathbf{u}_k}\mathbf{g}(\mathbf{x}_k,\mathbf{u}_k) + \mathrm{D}^2_{\mathbf{u}_k\mathbf{u}_k}\mathbf{g}(\mathbf{x}_k,\mathbf{u}_k)\mathrm{D}\mathbf{u}_k(\mathbf{x}_k),\end{aligned}$$

which can be rearranged to

$$\mathrm{D}\mathbf{u}_k(\mathbf{x}_k) = -(\mathrm{D}^2_{\mathbf{u}_k\mathbf{u}_k}\mathbf{g}(\mathbf{x}_k,\mathbf{u}_k))^{-1}\mathrm{D}^2_{\mathbf{x}_k\mathbf{u}_k}\mathbf{g}(\mathbf{x}_k,\mathbf{u}_k), \quad (12)$$

when $\mathrm{D}^2_{\mathbf{u}_k\mathbf{u}_k}\mathbf{g}(\mathbf{x}_k,\mathbf{u}_k)$ is non-singular. ■

C. Bi-Level Optimization Based Trajectory Planning

The differentiability of our NLS based wheel-terrain interaction and pose prediction can be fully exploited by formulating trajectory planning as the following bi-level problem :

$$\min \sum_k \left(c_r(\ddot{x}_k,\ddot{y}_k,\dot{x}_k,\dot{y}_k) + c_s(\mathbf{u}^*_k(\mathbf{x}_k))\right), \quad (13a)$$

$$(x_0,y_0,\dot{x}_0,\dot{y}_0,\ddot{x}_0,\ddot{y}_0) = \mathbf{b}_0, \quad (13b)$$

$$(x_n,y_n,\dot{x}_n,\dot{y}_n,\ddot{x}_n,\ddot{y}_n) = \mathbf{b}_n, \quad (13c)$$

$$\underline{x} \le x_k \le \overline{x}, \quad (13d)$$

$$\underline{y} \le y_k \le \overline{y}, \quad (13e)$$

$$\mathbf{u}^*_k(\mathbf{x}_k) = \min_{\mathbf{u}_k} \|g_i(\mathbf{x}_k,\mathbf{u}_k)\|_2^2. \quad (13f)$$

$$c_a(\ddot{x}_k,\ddot{y}_k) = \ddot{x}_k^2 + \ddot{y}_k^2, \quad (13g)$$

$$c_c(\ddot{x}_k,\ddot{y}_k,\dot{x}_k,\dot{y}_k) = \ddot{y}\dot{x} - \ddot{x}\dot{y}/(\dot{x}^2+\dot{y}^2+\epsilon)^{\frac{3}{2}}, \quad (13h)$$

$$c_r(.) = c_a(.) + c_c(.). \quad (13i)$$

The term $(c_r(.))$ in the cost function (13a) ensures smoothness in the trajectory and $(c_s(.))$ is the stability cost, which in our work has been modeled based on tip-over stability criterion [8]. The vectors $\mathbf{b}_0, \mathbf{b}_n$ are the stacked initial and final positions, velocities, and accelerations. The bounds on the positions (x_k, y_k) are given by $\underline{x}, \underline{y}, \overline{x}, \overline{y}$.

We can solve the above bi-level problem using projected gradient descent. The main complexity lies in computing $\nabla_{\mathbf{x}_k} c_s(\mathbf{u}^*_k(\mathbf{x}_k))$ but which can be done by using the Jacobian defined in Proposition 1.

III. Results

In this section, we validate our proposed wheel-terrain interaction and pose predictor along with the bi-level optimizer based stable trajectory planning on uneven terrain.

A. Validation through the Gazebo physics simulator

We used the Husky robot in Gazebo with a synthetic terrain model and collected the ground truth attitude data, z_{gt}, β_{gt}, and γ_{gt}. It was then compared with the predicted values z_{pred}, β_{pred}, and γ_{pred} obtained by solving the NLS. The comparison results are plotted in Figures 4b and 4c. It can be seen that the accuracy of the prediction is very high since both the values align closely. Next, we present the quantitative results of the minimum, maximum, and median error acquired across several simulations in Figures 4d and 4e. We can see that the error values are low, meaning that the NLS optimizer can successfully predict the vehicle's attitude on different terrains.

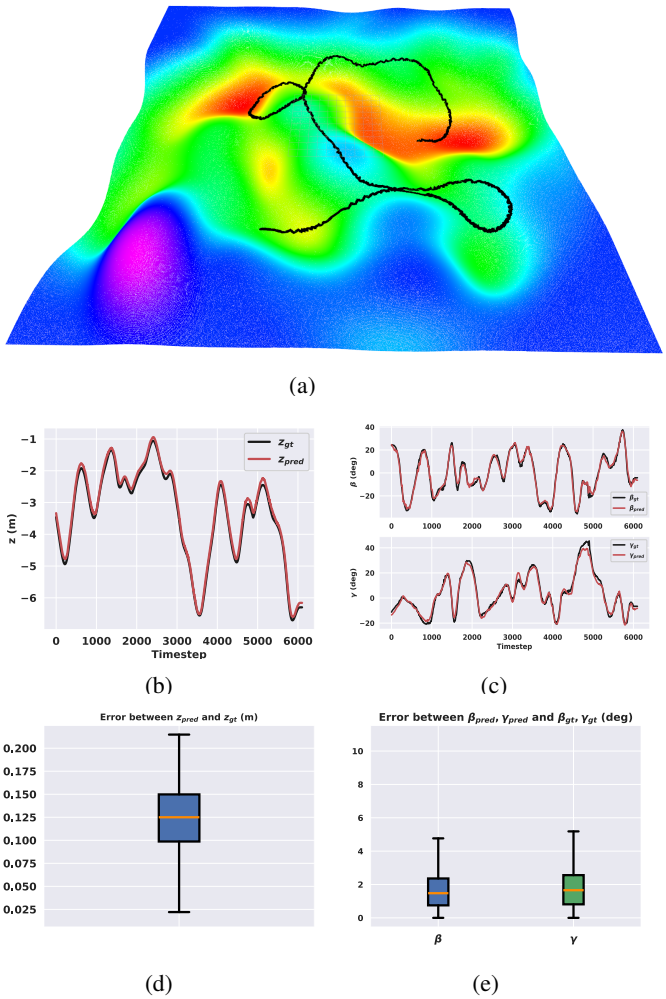


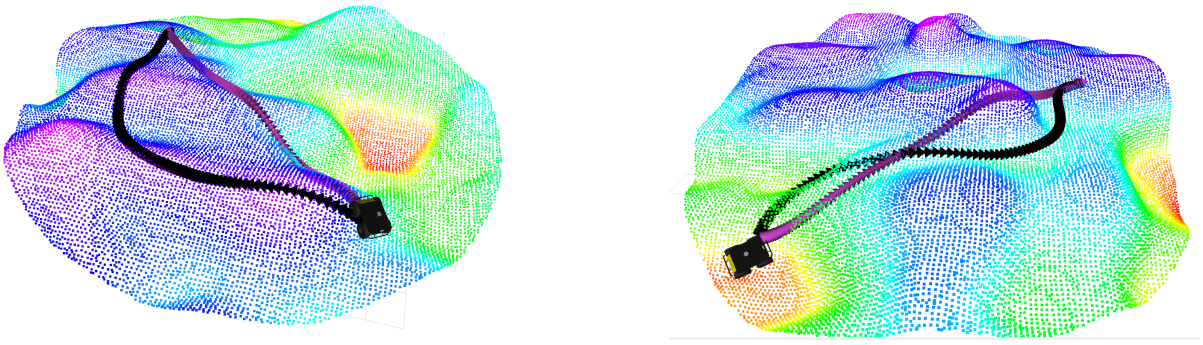
Fig. 4: (a) Trajectory obtained from manually driving Husky on a synthetic terrain in Gazebo. (b) z Ground truth and predicted values. (c) β and γ Ground truth vs predicted. (d) Error statistics for z. (e) Error statistics for β and γ.

B. Validation of the Safe Trajectory Planner

Here, the efficacy of the trajectory planner is validated with the help of tip-over stability criteria as a metric for the safety of the trajectories. Two example trajectories generated by the planner with and without the stability cost $c_s(.)$ (based on tip-over [8]) are shown in Fig 5. It can be seen that the stability cost helps in generating trajectories that take into account the terrain gradients while the trajectories without the stability cost run through ditches and hills. Clearly, a vehicle going through the former trajectory would be safer than the latter. We also show the values for the roll angle β and the pitch angle γ obtained for both the trajectories in Fig. 6. The stability cost helps reduce the magnitude and sudden changes in the angles.

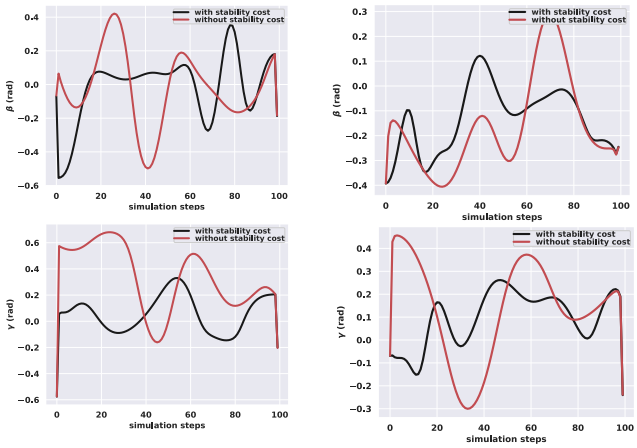
IV. Conclusions

A differentiable wheel-terrain interaction model to predict safe trajectories in uneven terrains was presented, which performs equal to that of a high-fidelity physics engine. The proposed approach has a wide variety of applications in machine learning and reinforcement learning, since it can be used for data collection and in learning pipelines due to the



(a) Example 1 (b) Example 2

Fig. 5: Trajectories with (black) and without stability cost(pink). Trajectories without the stability cost cuts across the terrains and move through hills and ditches since it is primarily motivated by minimizing the acceleration cost. In contrast, the stability cost based on tip-over criterion of [8] aligns the vehicle motion with the terrain gradient and minimizes sudden changes in pitch, roll and height.



(a) Example 1 (b) Example 2

Fig. 6: Evolution of Roll and pitch angles with and without the stability cost for the two example trajectories shown in Fig.5

ability to leverage implicit gradients. Our immediate efforts are to extend the NLS approach to consider the underlying dynamics of the vehicle-terrain system.

REFERENCES

[1] V. Wiberg, E. Wallin, T. Nordfjell, and M. Servin, "Control of rough terrain vehicles using deep reinforcement learning," *IEEE robotics and automation letters*, vol. 7, no. 1, pp. 390–397, 2021.

[2] A. Gattupalli, V. P. Eathakota, A. K. Singh, and K. Madhava Krishna, "A simulation framework for evolution on uneven terrains for synchronous drive robot," *Advanced Robotics*, vol. 27, 2013.

[3] R. Agishev, T. Petříček, and K. Zimmermann, "Trajectory optimization using learned robot-terrain interaction model in exploration of large subterranean environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3365–3371, 2022.

[4] N. Chakraborty and A. Ghosal, "Kinematics of wheeled mobile robots on uneven terrain," *Mechanism and machine theory*, vol. 39, no. 12, pp. 1273–1287, 2004.

[5] A. K. Singh and K. M. Krishna, "Feasible acceleration count: A novel dynamic stability metric and its use in incremental motion planning on uneven terrain," *Robotics and Autonomous Systems*, vol. 79, pp. 156–171, 2016.

[6] A. L. Dontchev and R. T. Rockafellar, *Implicit functions and solution mappings*. Springer, 2009, vol. 543.

[7] S. Gould, R. Hartley, and D. Campbell, "Deep declarative networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 8, pp. 3988–4004, 2021.

[8] E. Papadopoulos and D. A. Rey, "The force-angle measure of tipover stability margin for mobile manipulators," *Vehicle System Dynamics*, vol. 33, no. 1, pp. 29–48, 2000.