arXiv:2102.12667v1 [cs.RO] 25 Feb 2021

Learning Inverse Kinodynamics for Accurate High-Speed Off-Road Navigation on Unstructured Terrain

Xuesu Xiao[1], Joydeep Biswas[1], and Peter Stone[1,2]

***Abstract*— This paper presents a learning-based approach to consider the effect of unobservable world states in kinodynamic motion planning in order to enable accurate high-speed off-road navigation on unstructured terrain. Existing kinodynamic motion planners either operate in structured and homogeneous environments and thus do not need to explicitly account for terrain-vehicle interaction, or assume a set of discrete terrain classes. However, when operating on unstructured terrain, especially at high speeds, even small variations in the environment will be magnified and cause inaccurate plan execution. In this paper, to capture the complex kinodynamic model and mathematically unknown world state, we learn a kinodynamic planner in a data-driven manner with onboard inertial observations. Our approach is tested on a physical robot in different indoor and outdoor environments, enables fast and accurate off-road navigation, and outperforms environment-independent alternatives, demonstrating 52.4% to 86.9% improvement in terms of plan execution success rate while traveling at high speeds.**

I. Introduction

Current mobile robot navigation methods can navigate a robot from one point to another safely and reliably in structured and homogeneous environments [1], [2], such as indoor hallways or outdoor paved surfaces. These consistent environments allow the robots to use simple kinodynamic motion planners independent of the environment, thanks to the limited environment disturbances and stochasticity.

Dating back at least to DARPA's Grand Challenge [3] and LAGR (Learning Applied to Ground Vehicles) [4] program, researchers have also looked into applying autonomous navigation in unstructured outdoor environments. Challenges arise from multiple fronts in those natural spaces, but most off-road navigation work focused on perception, e.g. detecting natural obstacles [4], classifying underlying terrain types [5]–[7], or building semantic maps [8]–[10]. For motion control, most off-road robots simply travel at low speeds to minimize uncertainty and to maximize safety [11]–[14].

While recent advances in deep learning provide roboticists with a different avenue to investigate those perception problems in off-road navigation, researchers also started to combine perception, planning, and motion control using end-to-end learning in unstructured environments [15], [16]. These systems do not require a heavily-engineered navigation pipeline, and can react to natural environments in a data-driven approach. Although these methods can enable successful navigation, they are data-intensive and and generally do not lead to better navigation than their classical counterparts.

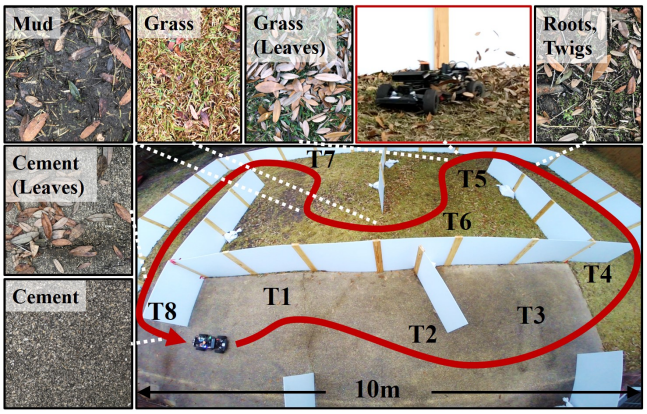


Fig. 1: The UT Automata scale 1/10th autonomous vehicle drives an eight-turn (T1–T8 on the red path) outdoor race track with unstructured terrain. Close-ups of some terrain (black inserts) are shown, including cement, mud, grass, which are covered by leaves, stalks, and/or twigs at different densities. For high-speed terrain-aware navigation, the robot has to interact with these unstructured terrain (red insert).

In this paper, we focus on the motion control side of off-road navigation on unstructured terrain, and use learning to capture the effect of complex and unknown environmental factors on the robot's low-level kinodynamic model. We use onboard inertial observation to encode environmental factors and learn an inverse kinodynamic model to produce fast and accurate low-level control inputs. Using our method, even with extensive disturbances caused by high-speed terrain-vehicle interaction, the robot is still able to perform fast and accurate off-road navigation (Fig. 1), compared to a kinodynamic model based on ideal assumptions and a learned model that does not consider environmental factors.

II. Related Work

In this section, we review related off-road navigation work in terms of perception and motion control.

A. Off-road Perception

The first challenge arises from off-road navigation is perception. In unstructured off-road environments, perception is no longer simply in the geometric sense (e.g. free vs. obstacle), but also requires semantic information (e.g. grass vs. mud). A plethora of research in terrain classification

[1]Xuesu Xiao, Joydeep Biswas, and Peter Stone are with Department of Computer Science, The University of Texas at Austin, Austin, TX 78712. {xiao, joydeepb, pstone}@cs.utexas.edu
[2]Peter Stone is with Sony AI.

has leveraged vibration-based signals to classify terrain types [5]–[7]. Vision-based sensors, e.g. LiDAR and camera, combined with current deep learning methods, have also been used to build semantic maps [8]–[10]. These perception methods assign costs to discrete terrain classes for planning, but do not consider robot's kinodynamic model when moving on these terrain. Our work does not distinguish among discrete terrain classes and uses observations collected during interactions with different terrain to enable fast and accurate kinodynamic planning.

B. Off-road Motion Control

Although research thrust for off-road navigation has been primarily focused on perception, roboticists have also investigated off-road navigation from the motion control side. Many wheel slip models [11]–[14] have been developed and used to design controllers to minimize slip. Most of these models treat slip only as a function of the vehicle kinematics. But to achieve high-speed off-road navigation, slip is inevitable and also highly dependent on the underlying terrain.

Researchers have also used machine learning for motion control in off-road navigation. A recent survey [17] pointed out that learning is most efficient when targeting at navigation components, e.g., learning local planners [18]–[21], costmaps [22], [23], or planner parameters [24]–[26]. Research on using learning for motion control in off-road scenarios is scarce. Pan, et al. [27] enabled high-speed navigation with end-to-end imitation learning from RGB input in a closed circular dirt track. The expert demonstrator is a model predictive controller with access to high-precision sensors including GPS-RTK. The end-to-end learning approach most likely does not generalize well to other terrain and tracks. Aiming at a variety of terrain, Siva, et al. [28] used imitation learning from human demonstration to navigate five discrete terrain types (concrete, grass, mud, pebble, rock). In contrast, our method only targets at learning a kinodynamic model and can navigate any global path. We also do not intentionally separate terrain into discrete types, and treat different terrain characteristics in a continuous manner.

III. Learning Inverse Kinodynamic Models

Most navigation systems either assume kindodynamic models to be independent of the environment, or that there exists a discrete set of environment classes [28], e.g., one model for paved terrain and another for grass. In this work, we relax these assumptions by learning a single continuous inverse kinodynamic model that accounts for environmental factors across different terrain types without having to perform discrete terrain classification, or analytic modelling. The learned model takes as input inertial observations that make the impact of environmental factor on kinodynamic motion observable (e.g. how bumpiness from gravel will result in understeer at high speeds). This inverse kinodynamic model is learned in a data-driven manner.

A. Problem Formulation

Given vehicle state x, control input u, and world state w, the state dynamics and observation y are given by

$$\dot{x} = f(x, u, w), \qquad y = g(x, w), \tag{1}$$

where $f(\cdot,\cdot,\cdot)$ is the system's forward kinodynamic function, while $g(\cdot,\cdot)$ is the observation function. Note that in most cases, w is not directly observable and cannot be easily modeled. A navigation planner generates a global plan $\Pi : [0,1] \rightarrow X$ mapping from a unitless progress variable $s \in [0,1]$ to planned vehicle state $x \in X$, incorporating both global (*e.g.,* traversable map) and local (*e.g.,* sensed obstacles) information to take the robot from the start state $\Pi(0)$ to the goal state $\Pi(1)$. A projection operator $\rho : X \rightarrow [0,1]$ maps the robot state x (e.g., from localization) to infer the progress variable s (i.e., the robot's progress along the global plan so far), such that the closest state in the plan to a robot state x is $\Pi(\rho(x))$. For simplicity of notation, we represent the projected state at any time as $x_\Pi = \Pi(\rho(x))$. We also omit the explicit time-dependence of variables $x(t)$, $u(t)$, and $y(t)$, denoting them simply as x, u, and y. The objective of our controller u is thus to minimize the total navigation time T while following the plan precisely, as represented by the joint cost function

$$J = T + \gamma \int_0^T ||x(t) - x_\Pi(t)||^2 dt. \tag{2}$$

Here, γ is a hyperparameter that trades total navigation time for execution accuracy.

We formulate the solution to this optimal control problem as a receding-horizon controller u^* over a unitless progress horizon (along the global plan) Δ and corresponding time-horizon Δt such that the control input drives the robot state from x to the receding horizon plan state $\Pi(\rho(x) + \Delta)$ over time-period Δt:

$$u^* = \arg\min_u \left(\Delta t + \gamma \left\| \Delta x_\Pi - \int_0^{\Delta t} f(x, u, w) dt \right\|^2 \right),$$
$$\Delta x_\Pi = \Pi(\rho(x) + \Delta) - x, \tag{3}$$

where Δx_Π is the state change between the receding horizon plan and current vehicle state.[1] The optimal control u^* can be solved using the receding horizon inverse kinodynamic model f^{-1} as

$$u^* = f^{-1}(\Delta x, x, w), \tag{4}$$

that takes as input the desired relative state change Δx, the current robot state x, and world state w. Unfortunately, it is hard to express f^{-1} accurately via analytical models, and even if it could be expressed accurately, computing u^* is error-prone since the world state w is not directly observable.

[1] In general, the minus signs in Eqn. 3 is the generalized difference operator $\ominus$ over Special Euclidean Group $\mathrm{SE}(n)$ and the corresponding Lie Algebra.

B. Learning Inverse Kinodynamics

In this work, in order to enable fast and accurate navigation under the influence of different terrain interactions from the world state w, we adopt a data-driven approach to capture the effect of w. Specifically, we introduce the function

$$f_\theta^+(\Delta x, x, y) \approx f^{-1}(\Delta x, x, w), \tag{5}$$

parameterized by θ, as an approximation for the original receding horizon inverse kinodynamic function (superscript $^+$ denotes pseudo inverse). The key insight in this approximation is that the impact of w on u^* becomes predictable given observations y related to high-speed terrain-aware navigation—in our case we use onboard inertial sensing to capture speed-dependent terrain interaction. To learn f_θ^+, a training dataset $\mathcal{T}$ with N samples

$$\mathcal{T} = \{\langle \Delta x^i, x^i, u^i, y^i \rangle_{i=1}^N\}$$

is desired, using the optimal but unknown receding horizon inverse kinodynamic function $u^i = f^{-1}(\Delta x^i, x^i, w^i)$ and observation function $y^i = g(x^i, w^i)$ from Eqns. 1 and 4. Unfortunately, we neither know f^{-1}, nor do we know the world states w^i. However, we do have access to f as a black-box function via real-world execution: we can simply pick arbitrary sample controls u^i at corresponding starting states x^i, and observe the resulting state change Δx^i after the chosen receding horizon Δt, including the impact of the unknown w^i:

$$\Delta x^i = \int_0^{\Delta t} f(x^i, u^i, w^i) dt.$$

Thus, the original chosen control u^i is the control[2] for the resulting state change Δx^i from the original state x^i, for the corresponding but unknown (and hence unrecorded) world state w^i. Along with the corresponding sensor observation y^i, we generate each sample i for dataset $\mathcal{T}$. To ensure that the learned parameters θ of f_θ^+ approximate f^{-1} accurately at states that the robot will encounter during execution, $\mathcal{T}$ must include representative samples for x, u, and y. With the collected dataset $\mathcal{T}$, we formulate deriving $f_\theta^+(\cdot,\cdot,\cdot)$ as a learning problem by minimizing a supervised loss:

$$\begin{aligned}\theta^* &= \arg\min_\theta \sum_{(\Delta x^i, x^i, y^i)\in\mathcal{T}} \|f^{-1}(\cdot,\cdot,\cdot) - f_\theta^+(\Delta x^i, x^i, y^i)\|_H \\ &= \arg\min_\theta \sum_{(u^i, \Delta x^i, x^i, y^i)\in\mathcal{T}} \|u^i - f_\theta^+(\Delta x^i, x^i, y^i)\|_H,\end{aligned} \tag{6}$$

where $||v||_H = v^T H v$ is the norm induced by positive definite matrix H, used to weigh the learning loss between the different dimensions of the control input u^i. We represent $f_\theta^+(\cdot,\cdot,\cdot)$ as a neural network and can therefore use gradient decent to find an approximately optimal θ^*. In this work, we collect raw 6-DoF readings from an onboard IMU, and construct y by feeding inertial data through an autoencoder to encode relevant terrain-vehicle interaction at different driving speeds. More details are provided in Sec. IV.

[2]Here, we assume the optimal control that follows the global path (second term in Eqn. 2) also implicitly minimizes navigation time (first term).

C. Online Execution

The learned inverse kinodynamic model $f_{\theta^*}^+(\cdot,\cdot,\cdot)$ provides a means to approximately account for w using the onboard observation y for fast, terrain-aware, and precise navigation. At each time step t during online execution, we compute the desired change of state $\Delta x_\Pi = \Pi(\rho(x)+\Delta) - x$ with x from localization, the projection operator $\rho(\cdot)$, projection horizon Δ, and global plan $\Pi(\cdot)$. Along with onboard observation y and current vehicle state x, we use the learned inverse kinodynamic model $f_{\theta^*}^+$ to produce system control input:

$$u(t) = f_{\theta^*}^+(\Delta x_\Pi, x, y), \tag{7}$$

and repeat this process for every time step.

IV. Experiments

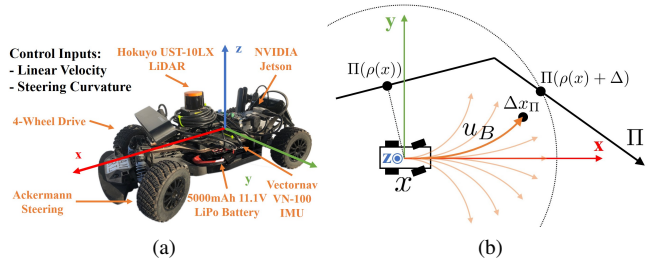
In this section, we present experimental results using a learned inverse kinodynamic model $f_{\theta^*}^+(\Delta x, x, y)$, which considers unobservable world state w by taking y as input, and can precisely track different global plans by different Π.

We denote the baseline forward and inverse kinodynamic functions, which do not consider world state w, as $f_B(x,u)$ and $f_B^+(\Delta x, x)$, respectively. As an ablation study to test the effectiveness of capturing the world state w with y, we also learn an ablated inverse kinodynamic model $f_{A\phi^*}^+(\Delta x, x)$, which is parameterized by ϕ^* and does not take observation y as input to represent w. We show that the learned $f_{A\phi^*}^+$ can outperform the baseline f_B^+. Adding the learned observation y as another input, $f_{\theta^*}^+(\Delta x, x, y)$ can outperform both $f_{A\phi^*}^+(\Delta x, x)$ and $f_B^+(\Delta x, x)$.

The learned inverse kinodynamic model for online execution is also agnostic to different online plans and unseen terrain. We show that the model learned through a global planner Π_1 (a randomly exploring policy driven by a human operator) can also generalize well to a complicated global planner Π_2 on an outdoor race track, and a simple global planner Π_3 on an indoor track. We also test the generalizability with respect to unseen world states through the encoded y and $f_{\theta^*}^+(\Delta x, x, y)$ on unseen terrain.

A. Implementation

1) Robot Platform: Our learned inverse kinodynamic model is implemented on a UT Automata robot, a scale 1/10th autonomous vehicle platform (Fig. 2a). The Ackermann-steering four-wheel drive vehicle is equipped with a 2D Hokuyo UST-10LX LiDAR for localization, a Vectornav VN-100 IMU for inertial sensing (200Hz), a Flipsky VESC 4.12 50A motor controller, and a Traxxas Titan 550 Motor. Although the platform has individual suspensions, the relatively short-travel suspensions are not specifically designed for off-road navigation. We specifically pick this platform because its small size and weight and the lack of designed off-road capability can maximize the difference in accuracy between navigating with and without the learned terrain-aware inverse kinodynamic model. The robot has a NVIDIA Jetson onboard, but only the CPU is used during deployment.



(a) (b)

Fig. 2: (a): A UT Automata robot, scale 1/10th autonomous vehicle platform used in the experiments. (b): The sampling-based baseline approximates Δx_Π by rolling out the optimal u_B among 100 samples with f_B.

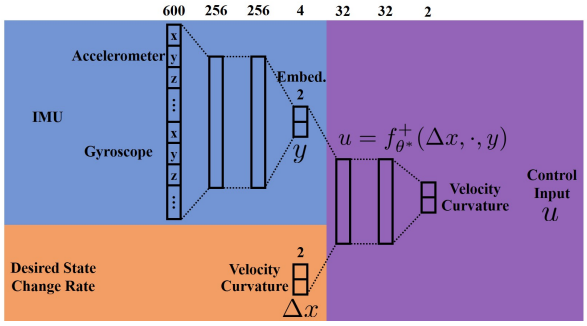


Fig. 3: Neural Network Architecture. Input: blue IMU encoder and orange desired state change (desired velocity and curvature in practice); Output: purple learned function approximator $f^+_{\theta^*}$ as the inverse kinodynamic model.

2) Environment: The environment comprises of cement, grass, and mud; and some patches are covered by different artifacts, such as leaves, stalks, and/or twigs, with different densities (Fig. 1). For a small vehicle like the UT Automata robot, these artifacts can cause significantly different world states (Fig. 1 red insert). Note that the terrain also changes due to environmental factors such as sunlight, wind, and moisture, and is also affected by the robot's wheel and chassis. To minimize the effect of artifacts being pushed off the course during extended experiments, we frequently shuffle and redistribute the artifacts. We do not specify discrete terrain classes and treat the terrain characteristics in a continuous manner.

3) Model Implementation: During autonomous navigation, the robot uses Episodic Non-Markov Localization (ENML) [29] with a pre-built map of the environment to derive vehicle state x. A global planner Π includes a pre-generated global path for the robot to follow and uses line-of-sight control (similar to [30], [31]) to generate desired receding horizon plan state $\Pi(\rho(x)+\Delta)$ on the global path 1m away from the robot. In a model predictive control manner, the robot uses the baseline forward kinodynamic function f_B and samples candidate velocity and curvature control inputs $u \in U$ evenly distributed within a physically-feasible window to jointly find the desired state change Δx_Π and control input u_B (shown in Fig. 2b). More specifically, to compute control input, the baseline inverse kinodynamic model f^+_B produces the curvature input, which results in the desired Δx_Π and drives the robot as close to $\Pi(\rho(x)+\Delta)$ as possible:

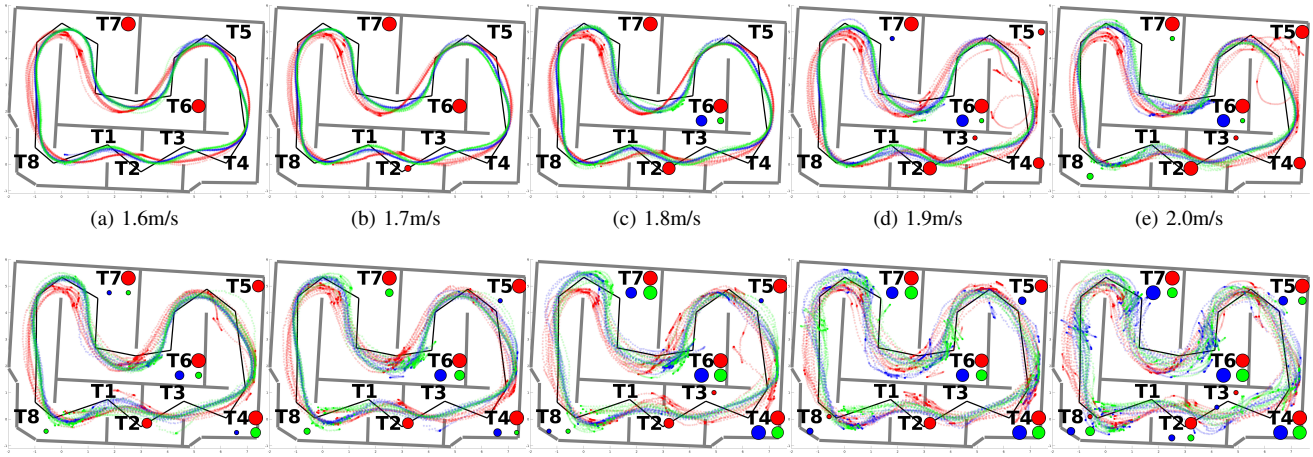
$$\begin{aligned} u_B &= f^+_B(\cdot,\cdot) \\ &= \arg\min_u \|\Pi(\rho(x)+\Delta) - \int_{t=0}^{\Delta t} f_B(x,u)dt\|, \end{aligned} \tag{8}$$

for the second term in Eqn. 2, and selects the fastest possible velocity for the first term T, considering the robot's acceleration limit and a safety distance to decelerate in case of obstacles.

For our learned ablated and final model, $f^+_{A\phi^*}$ and $f^+_{\theta^*}$, we utilize the Δx_Π from the baseline kinodynamic model (corresponds to u_B), but instead of using the baseline's control input, we query our learned models to produce $u = f^+_{A\phi^*}(\Delta x_\Pi, x)$ or $u = f^+_{\theta^*}(\Delta x_\Pi, x, y)$. In practice, we use the baseline control input $u_B = \{v, c\}$ (linear velocity and steering curvature) to represent the desired state change rate Δx_Π.

4) Data Collection: To collect training data, the robot is teleoperated with a joystick in an open environment with linear velocity $v \in [0, 3]\text{m/s}$ and steering curvature $c \in [-1.35, 1.35]\text{m}^{-1}$ for 30 minutes (24418 data points). The teleoperator randomly varies both linear velocity and steering curvature (Π_1). In our specific implementation, the robot reasons in the robot frame and therefore the state x^i in the training trajectory $\mathcal{T} = \{\langle \Delta x^i, x^i, u^i, y^i \rangle_{i=1}^N\}$ becomes the origin in the robot frame. The ground truth Δx^i is represented as real $\{v_r^i, c_r^i\}$, where v_r^i is from vehicle odometry and $c_r^i = \omega_r^i / v_r^i$ (ω_r^i is the sensed angular velocity around the vertical z axis from the IMU). Currently, we take v_r^i from wheel odometry only, which can be further improved by adding visual, point cloud, and/or inertial information in future work. The commanded control input $u^i = \{v_c^i, c_c^i\}$ is recorded from joystick input. For y, we collect the 6-DoF raw IMU signal, including 3-DoF accelerometer and 3-DoF gyroscope, as a sliding history window.

5) Network Architecture: As a function approximator for $f^+_{\theta^*}$, we use a two-layer neural network with 32 neurons each layer (shown in purple in Fig. 3). The neural network takes realistic/desired $\{v_r^i, c_r^i\}$ (as a proxy for Δx^i, Fig. 3, orange) and observation y as input, and outputs to-be-commanded control input $u^i = \{v_c^i, c_c^i\}$. For observation y, we concatenate the last 100 IMU readings (0.5s) into a 600-dimensional vector, and feed it into two 256-neuron layers as an autoencoder (Fig. 3, blue). The final embedding for y is a two dimensional vector, then concatenated with $\{v_r^i, c_r^i\}$, and finally trained in an end-to-end fashion. The entire network architecture is shown in Fig. 3. For the ablated model $f^+_{A\phi^*}$, the two-dimensional y embedding is removed (only the orange and purple components remain). Training both models takes less than five minutes on a NVIDIA GeForce GTX 1650 laptop GPU. During runtime, the trained model is used onboard the robot's Jetson CPU with `libtorch`.



(a) 1.6m/s (b) 1.7m/s (c) 1.8m/s (d) 1.9m/s (e) 2.0m/s

(f) 2.1m/s (g) 2.2m/s (h) 2.3m/s (i) 2.4m/s (j) 2.5m/s

Fig. 4: Results of Outdoor Experiments on Seen Terrain: Localized robot positions of all 300 laps are plotted around the pre-defined global path (black line segments) on the map (grey lines). The size of the circles at each turn denotes the number of failures at that turn. Red: baseline $f_B^+(\Delta x, x)$, blue: ablated model $f_{A\phi^*}^+(\Delta x, x)$, green: learned model $f_{\theta^*}^+(\Delta x, x, y)$.

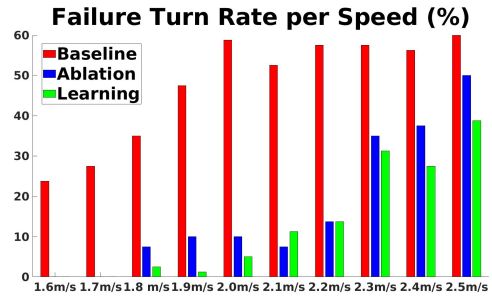


Fig. 5: Failure Rate Per Target Speed

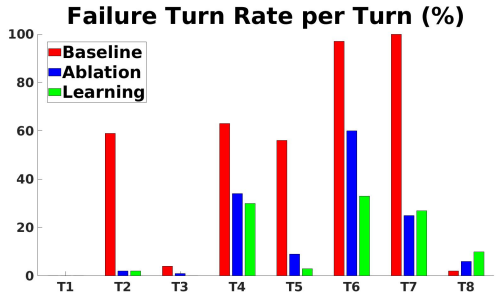


Fig. 6: Failure Rate per Turn

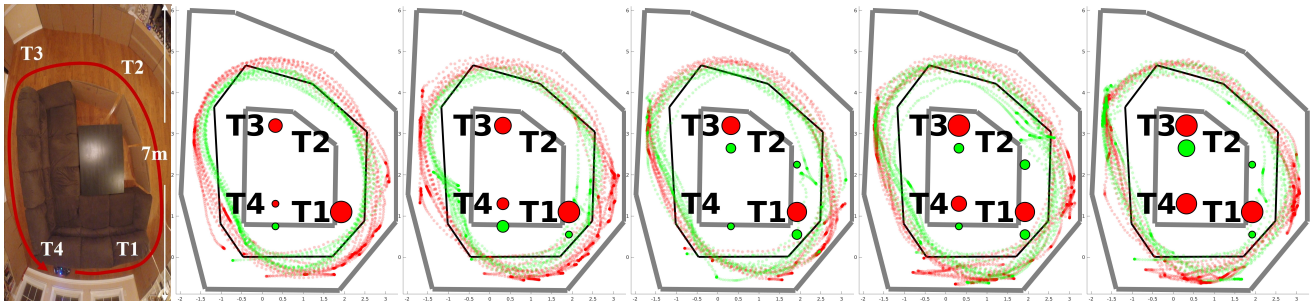
B. Navigation on Seen Terrain

We first test the inverse kinodynamic model's performance on the same terrain where the training data is collected, but with a different global planner Π_2. After collecting the training data with Π_1, an outdoor race track is constructed using plastic panels and wooden posts (Fig. 1). Starting from the origin (robot location in Fig. 1), eight turns are created (T1–T8, Π_2). While Turn 1, 2, and 3 are relatively gentle left-, right-, and left-hand turns, Turn 4 and Turn 8 are roughly 90° left-hand turns. Turn 5, 6, and 7 are sharp 180° left-, right-, and left-hand turns.

Ten different target speeds (the maximum speed the robot targets at reaching while maintaining safety tolerance to decelerate and avoid potential collisions) are tested, ranging from 1.6m/s to 2.5m/s with 0.1m/s intervals. For the three models, the baseline $f_B^+(\Delta x, x)$, the ablated $f_{A\phi^*}^+(\Delta x, x)$, and learned $f_{\theta^*}^+(\Delta x, x, y)$, we repeat ten trials/laps each for statistical significance. A total 300 laps are executed. The localized robot position from ENML are shown in Fig. 4 in the subplots corresponding to the target speeds.

At lower target speeds, the green trajectories by the learned model $f_{\theta^*}^+(\Delta x, x, y)$ are much closer to the pre-defined global path, compared to the baseline $f_B^+(\Delta x, x)$, because the latter model fails to consider the world state caused by the unstructured terrain. With increasing speed, the robot trajectory becomes more scattered around the global path due to increased stochasticity from vehicle-terrain interaction. But overall speaking, the green trajectories are always closer to the global path than other alternatives. The blue trajectory is generated by the ablated model $f_{A\phi^*}^+(\Delta x, x)$. Like $f_{\theta^*}^+$, it learns from actual terrain interactions, but it does not consider the current observation y. So $f_{A\phi^*}^+$ is roughly an averaged model over the continuous spectrum of terrain. Therefore, $f_{A\phi^*}^+$ outperforms the baseline f_B^+, but underperforms $f_{\theta^*}^+$ because it fails to consider the current world state.

At each turn, the size of the red, blue, and green circles represents the number of failed turns (collision or getting stuck) in the ten attempted turns. Turn 6 and 7 cause a lot of trouble for the baseline even at lower speeds. With increasing speed, more turns cause failure for other models as well, but in general, the baseline fails more frequently at most turns than the ablated and learned models. Fig. 5 and Fig. 6 show the percentage of failed turns per target speed and per turn, respectively. In Fig. 5, failure rate increases with faster speed,



(a) Unseen Test (b) 2.4m/s (c) 2.5m/s (d) 2.6m/s (e) 2.7m/s (f) 2.8m/s

Fig. 7: Experiment Results in Unseen Environment

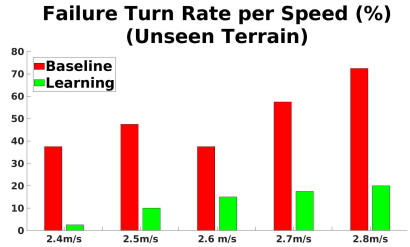


Fig. 8: Failure Rate per Target Speed on Unseen Terrain

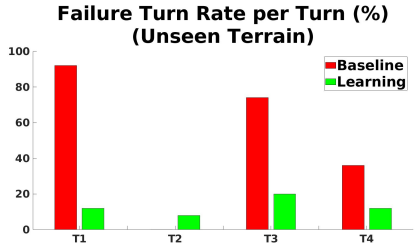


Fig. 9: Failure Rate per Turn on Unseen Terrain

while within each speed, the learned model achieves the lowest failure rate, while the baseline fails most frequently. In Fig. 6, Turn 6 is the most difficult for all three alternatives, and at most turns, the learned model outperforms the ablation and the baseline. Since Turn 8 is immediately after the terrain change from grass to cement, the robot sometimes oversteers (to compensate for slip on grass) and does not react quickly enough to understeer (for higher friction on cement), causing it to get stuck in a few laps with the learned model. This problem can be addressed by adding forward looking camera to predict future wheel-terrain interaction in future work. The overall success rates of the three models for all turns are shown in Tab. I.

TABLE I: Overall Success Rates of All Speeds and Turns

	Baseline	Ablation	**Learning**
Seen	52.4%	82.9%	**86.9%**
Unseen	49.5%	–	**87.0%**

C. Navigation on Unseen Terrain

To test that the learned model generalizes to different global planners Π and also to unseen terrain, we further conduct an indoor experiment with a different track and global path (Π_3) on an unseen wooden floor (Fig. 7a). Note that $f^+_{\theta^*}$ has only seen training data from random exploration on the outdoor terrain (Fig. 1). Since the unseen wooden floor is relatively more consistent and therefore easier to navigate than the outdoor unstructured terrain,[3] we increase the navigation target speed to 2.4m/s - 2.8m/s, also with 0.1m/s intervals. The baseline and the learned model are applied with these five different target speeds, ten repetitions each. Fig. 7 shows the results from the 100 laps on the unseen terrain with a different global path. Similar to the results on seen terrain, the learned model produces more concentrated and also closer robot trajectories to the global path to be tracked. As shown in Fig. 8 and 9, the learned model also outperforms the baseline in terms of failure rate at all target speeds and in most turns (except Turn 2). The overall success rates of the baseline and learned model are shown in Tab. I.

V. Conclusions

In this paper, we present a data-driven approach to learn an inverse kinodynamic model for accurate high-speed navigation on unstructured terrain. To capture the elusive and stochastic world state caused by vehicle-terrain interaction at different high speeds, we use an inertia-based observation embedding as an input to the learned inverse kinodynamic function. This approach is tested on a physical robot on seen and unseen terrain with different global plans at different high speeds. The experimental results show that the learned model can significantly outperform an ideal baseline model without consideration of world state. Our ablation study also shows our observation embedding is useful to enable fast and accurate off-road navigation on unstructured terrain. For future work, better ground truth linear velocity estimation needs to be investigated: in addition to wheel odometry alone, other sources of perception, e.g. vision, point cloud, and/or inertia, can be leveraged. Better linear velocity estimation can account for significant wheel slippage on more

[3]We speculate that the generalization would not be as good were the model trained indoors (on easy terrain) but applied outdoors.

challenging terrain, e.g. on ice, and enable even faster navigation. Adding vision-based observation also has the potential to enable the robot to prepare for future interactions, e.g. to reduce the failures at Turn 8. Another interesting direction to investigate in the future is generalization from easier to harder environments.

ACKNOWLEDGMENTS

This work has taken place in the Learning Agents Research Group (LARG) and Autonomous Mobile Robotics Laboratory (AMRL) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by NSF (CPS-1739964, IIS-1724157, NRI-1925082), ONR (N00014-18-2243), FLI (RFP2-000), ARO (W911NF-19-2-0333), DARPA, Lockheed Martin, GM, and Bosch. AMRL research is supported in part by NSF (IIS-1954778, SHF-2006404), DARPA (HR001120C0031), Amazon, JP Morgan, and Northrop Grumman Mission Systems. The views and conclusions contained in this document are those of the authors alone. Peter Stone serves as the Executive Director of Sony AI America and receives financial compensation for this work. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research. The authors would like to thank Sadegh Rabiee, Tongrui Li, and Kavan Sikand for the help with UT Automata.

REFERENCES

[1] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*. IEEE, 1993, pp. 802–807.

[2] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[3] G. Seetharaman, A. Lakhotia, and E. P. Blasch, "Unmanned vehicles come of age: The darpa grand challenge," *Computer*, vol. 39, no. 12, pp. 26–29, 2006.

[4] L. D. Jackel, E. Krotkov, M. Perschbacher, J. Pippine, and C. Sullivan, "The darpa lagr program: Goals, challenges, methodology, and phase i results," *Journal of Field robotics*, vol. 23, no. 11-12, pp. 945–973, 2006.

[5] C. Bai, J. Guo, and H. Zheng, "Three-dimensional vibration-based terrain classification for mobile robots," *IEEE Access*, vol. 7, pp. 63 485–63 492, 2019.

[6] W. Shi, Z. Li, W. Lv, Y. Wu, J. Chang, and X. Li, "Laplacian support vector machine for vibration-based robotic terrain classification," *Electronics*, vol. 9, no. 3, p. 513, 2020.

[7] M. Mei, J. Chang, Y. Li, Z. Li, X. Li, and W. Lv, "Comparative study of different methods in vibration-based terrain classification for wheeled robots with shock absorbers," *Sensors*, vol. 19, no. 5, p. 1137, 2019.

[8] D. Maturana, P.-W. Chou, M. Uenoyama, and S. Scherer, "Real-time semantic mapping for autonomous off-road navigation," in *Field and Service Robotics*. Springer, 2018, pp. 335–350.

[9] P. Wolf, A. Vierling, T. Ropertz, and K. Berns, "Advanced scene aware navigation for the heavy duty off-road vehicle unimog," in *IOP Conference Series: Materials Science and Engineering*, vol. 997, no. 1. IOP Publishing, 2020, p. 012093.

[10] S. Sharma, J. E. Ball, B. Tang, D. W. Carruth, M. Doude, and M. A. Islam, "Semantic segmentation with transfer learning for off-road autonomous driving," *Sensors*, vol. 19, no. 11, p. 2577, 2019.

[11] S. Rabiee and J. Biswas, "A friction-based kinematic model for skid-steer wheeled mobile robots," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8563–8569.

[12] Y. Tian and N. Sarkar, "Control of a mobile robot subject to wheel slip," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 3, pp. 915–929, 2014.

[13] F. Rogers-Marcovitz, M. George, N. Seegmiller, and A. Kelly, "Aiding off-road inertial navigation with high performance models of wheel slip," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 215–222.

[14] M. Seyr and S. Jakubek, "Proprioceptive navigation, slip estimation and slip control for autonomous wheeled mobile robots," in *2006 IEEE Conference on Robotics, Automation and Mechatronics*. IEEE, 2006, pp. 1–6.

[15] U. Muller, J. Ben, E. Cosatto, B. Flepp, and Y. L. Cun, "Off-road obstacle avoidance through end-to-end learning," in *Advances in neural information processing systems*. Citeseer, 2006, pp. 739–746.

[16] A. Giusti, J. Guzzi, D. C. Cireşan, F.-L. He, J. P. Rodríguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. Di Caro *et al.*, "A machine learning approach to visual perception of forest trails for mobile robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 661–667, 2015.

[17] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Motion control for mobile robot navigation using machine learning: a survey," *arXiv preprint arXiv:2011.13112*, 2020.

[18] ——, "Toward agile maneuvers in highly constrained spaces: Learning from hallucination," *arXiv preprint arXiv:2007.14479*, 2020.

[19] X. Xiao, B. Liu, and P. Stone, "Agile robot navigation through hallucinated learning and sober deployment," *arXiv preprint arXiv:2010.08098*, 2020.

[20] B. Liu, X. Xiao, and P. Stone, "Lifelong navigation," *arXiv preprint arXiv:2007.14486*, 2020.

[21] H.-T. L. Chiang, A. Faust, M. Fiser, and A. Francis, "Learning navigation behaviors end-to-end with autorl," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2007–2014, 2019.

[22] M. Wigness, J. G. Rogers, and L. E. Navarro-Serment, "Robot navigation from human demonstration: Learning control behaviors," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1150–1157.

[23] C. Richter and N. Roy, "Safe visual navigation via deep learning and novelty detection," 2017.

[24] Z. Xu, G. Dhamankar, A. Nair, X. Xiao, G. Warnell, B. Liu, Z. Wang, and P. Stone, "Applr: Adaptive planner parameter learning from reinforcement," *arXiv preprint arXiv:2011.00397*, 2020.

[25] Z. Wang, X. Xiao, B. Liu, G. Warnell, and P. Stone, "Appli: Adaptive planner parameter learning from interventions," *arXiv preprint arXiv:2011.00400*, 2020.

[26] X. Xiao, B. Liu, G. Warnell, J. Fink, and P. Stone, "Appld: Adaptive planner parameter learning from demonstration," *arXiv preprint arXiv:2004.00116*, 2020.

[27] Y. Pan, C.-A. Cheng, K. Saigol, K. Lee, X. Yan, E. A. Theodorou, and B. Boots, "Imitation learning for agile autonomous driving," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 286–302, 2020.

[28] S. Siva, M. Wigness, J. Rogers, and H. Zhang, "Robot adaptation to unstructured terrains by joint representation and apprenticeship learning," in *Robotics: science and systems*, 2019.

[29] J. Biswas and M. M. Veloso, "Episodic non-markov localization," *Robotics and Autonomous Systems*, vol. 87, pp. 162–176, 2017.

[30] X. Xiao, J. Dufek, T. Woodbury, and R. Murphy, "Uav assisted usv visual navigation for marine mass casualty incident response," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6105–6110.

[31] X. Xiao, E. Cappo, W. Zhen, J. Dai, K. Sun, C. Gong, M. J. Travers, and H. Choset, "Locomotive reduction for snake robots," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3735–3740.