NAUTS: Negotiation for Adaptation to Unstructured Terrain Surfaces

Sriram Siva[1], Maggie Wigness[2], John G. Rogers[2], Long Quang[2], and Hao Zhang[1,3]

***Abstract*— When robots operate in real-world off-road environments with unstructured terrains, the ability to adapt their navigational policy is critical for effective and safe navigation. However, off-road terrains introduce several challenges to robot navigation, including dynamic obstacles and terrain uncertainty, leading to inefficient traversal or navigation failures. To address these challenges, we introduce a novel approach for adaptation by negotiation that enables a ground robot to adjust its navigational behaviors through a negotiation process. Our approach first learns prediction models for various navigational policies to function as a terrain-aware joint local controller and planner. Then, through a new negotiation process, our approach learns from various policies' interactions with the environment to agree on the optimal combination of policies in an online fashion to adapt robot navigation to unstructured off-road terrains on the fly. Additionally, we implement a new optimization algorithm that offers the optimal solution for robot negotiation in real-time during execution. Experimental results have validated that our method for adaptation by negotiation outperforms previous methods for robot navigation, especially over unseen and uncertain dynamic terrains.**

I. Introduction

In recent years, autonomous mobile robots have been increasingly deployed in off-road field environments to carry out tasks related to disaster response, infrastructure inspection, and subterranean and planetary exploration [1], [2], [3]. When operating in such environments, mobile robots encounter dynamic, unstructured terrains that offer a wide variety of challenges (as seen in Fig. 1), including dynamic obstacles and varying terrain characteristics like slope and softness. In these environments, terrain adaptation is an essential capability that allows ground robots to perform successful maneuvers by adjusting their navigational behaviors to best traverse the changing unstructured off-road terrain characteristics [4], [5].

Given its importance, the problem of robot adaptation over unstructured terrains has been extensively investigated in recent years. In general, terrain adaptation has been addressed using three broad categories of methods. The first category, classic control-based methods, use mathematical tools from control theory [6], [7], [8] to design robot models that achieve the desired robot behavior and perform robust ground maneuvers in various environments. The second category, learning-based methods, use data-driven formulations to either imitate an expert demonstrator [5], [9], [10], learn from trial-and-error in a reinforcement learning setting [11], [12], [13], or use online learning to continuously learn and adapt in an environment [14], [15], [16]. Finally, the third category, machine-learning-based control, exploits the advantage of integrating machine learning into control theory to learn accurate robot dynamics and accordingly adapt navigational behaviors [17], [18], [19].

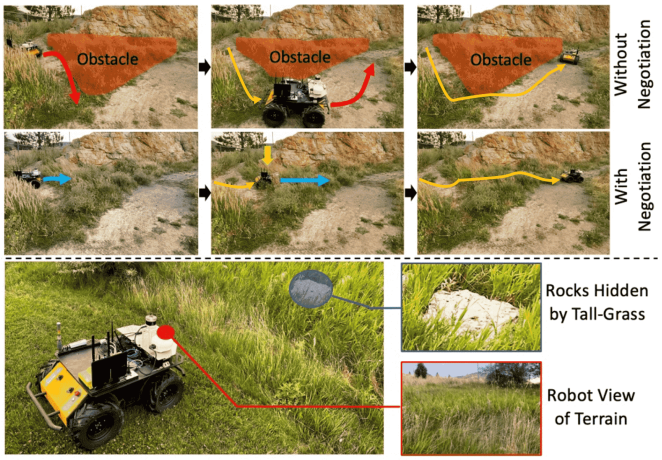


Fig. 1. Robots operating in dynamic, unstructured environments often generate sub-optimal behaviors leading to inefficient robot traversal or even navigation failure. For example, robots may consider tall grass terrain as an obstacle. Terrain negotiation allows robots to explore different navigation policies to determine the optimal combination for successful and efficient navigation in unknown terrains. In this example, the robot initially treats tall grass as an obstacle but simultaneously explores a max speed policy. The robot then quickly observes that the max speed policy improves efficiency by traversing across tall grass, and thus, learns to give more importance to the max speed policy compared to obstacle avoidance.

However, unstructured terrains often have dynamic obstacles that change their state as the robot traverses over them, such as tall grass. Additionally, these terrains can occlude future obstacles and ground cover, leading to traversal uncertainty (e.g., grass occluding a rock as seen in Fig. 1). These challenges can also be observed in commonly traversed unstructured environments such as sand, snow, mud, and forest terrains. As characteristics of such terrains cannot be modeled beforehand, robots cannot be trained for all possible terrain variations and must therefore adapt as these variations are encountered. Existing methods for robot navigation generally lack robustness to address these challenges as they are designed as a local controller to execute a single robot navigation policy, causing inefficient (e.g., longer traversal time and distance) or even failed navigation. In addition,

*This work was partially supported by NSF CAREER Award IIS-1942056 and U.S. DEVCOM ARL SARA Program W911NF-20-2-0107.

[1]Sriram Siva is with the Computer Science Department, Colorado School of Mines (Mines), Golden, CO 80401, USA. Hao Zhang is partially affiliated with Mines. Email: sivasriram@mines.edu.

[2]Maggie Wigness, John G. Rogers, and Long Quang are with the DEVCOM Army Research Laboratory (ARL), Adelphi, MD 20783, USA. Email: {maggie.b.wigness.civ, john.g.rogers59.civ, long.p.quang.civ}@army.mil.

[3]Hao Zhang is with the Manning College of Information and Computer Sciences (CICS), University of Massachusetts Amherst, Amherst, MA 01002, USA. Email: hao.zhang@cs.umass.edu.

current methods such as [9], [10] require significant amounts of training data to learn optimal navigational behaviors. The challenge of quickly learning a joint local controller and planner to enable adaptive behaviors has not been addressed.

In this paper, we introduce our novel approach to robot navigation: *Negotiation for Adaptation to Unstructured Terrain Surfaces* (NAUTS). Instead of generating terrain-aware behaviors for only the current time steps, NAUTS learns a non-linear prediction model to estimate future robot behaviors and states for several different policies. Each policy represents a series of navigational behaviors that can be learned either using imitation learning [5] or self-supervised learning [10] according to a specific goal (e.g., obstacle avoidance, maximum speed, etc.). NAUTS then learns from the continuous interaction of these different policies with the terrain to generate optimal behaviors for successful and efficient navigation. We define *negotiation* as the process of learning robot navigation behaviors from online interactions between a library of policies with the terrain in order to agree on an optimal combination of these policies. The learning of both the non-linear prediction models and policy negotiation are integrated into a unified mathematical formulation under a regularized optimization paradigm.

There are three main contributions of this paper:

- We introduce a novel non-linear prediction model to estimate goal-driven future robot behaviors and states according to various navigational policies and address the challenge of learning a terrain-aware joint local controller and planner.
- We propose one of the first formulations on negotiation for robot adaptation under a regularized optimization framework. Our approach allows a robot to continuously form agreements between various navigational policies and optimally combines them to i) improve the efficiency of navigation in known environments and ii) learn new navigation policies quickly in unknown and uncertain environments.
- We design a new optimization algorithm that allows for fast, real-time convergence to execute robot negotiation during deployment.

As an experimental contribution, we provide a comprehensive performance evaluation of learning-based navigation methods over challenging dynamic unstructured terrains.

II. Related Work

The related research in robot terrain adaptation can be classified under methods based on classical control theory, learning-based, and machine-learning-based control.

The methods developed under the classical control theory use pre-defined models to generate robust navigational behaviors and reach the desired goal position in an outdoor field environment. Earlier methods used a fuzzy logic implementation to perform navigation [20], [21], without using the knowledge of a robot's dynamics. This led to the development of system identification [22], where methods learn robot dynamics using transfer functions to model linear robotic systems and perform navigation [23], [24]. More recently, trajectory optimization models such as differential dynamic programming (DDP), specifically iterative linear quadratic regulator (iLQR), used knowledge of non-linear robot dynamics to solve navigation tasks [25], [26]. Model predictive control (MPC) learns to be robust to robot model errors and terrain noise by implementing a closed-loop feedback system during terrain navigation [27], [28], [29]. However, these methods can approximate robot dynamics to a limited extent as these methods cannot learn from high-dimensional robot data and lack the ability to adapt as terrain changes.

Learning-based methods use data-driven formulations to generate navigational behaviors in various environments. Early methods used Koopman operator theory [30] to model non-linear robot systems using an infinite-dimensional robot observable space [31], [32] to perform terrain navigation. Subsequent learning-based methods mainly used learning from demonstration (LfD) [33] to transfer human expertise of robot driving to mobile robots [9], [34]. One method to perform terrain-aware navigation combined representation learning for terrain classification with apprenticeship learning to perform terrain adaptation [5]. Kahn and Levine [10] learned navigational affordance from experts over various terrains for carrying out off-road navigation. Recently, consistent behavior generation was achieved [35] to match actuation behaviors with a robot's expected behaviors. Unlike learning from demonstration, reinforcement learning based methods learn purely from a robot's own experience in an unknown environment in a trial-and-error fashion [11], [12]. Rapid motor adaptation was achieved by updating learned policies via inferring key environmental parameters to successfully adapt in various terrains [13]. Life-long learning methods, similar to reinforcement learning, sequentially improve the performance of robot navigation by continuously optimizing learned models [16], [36]. Rather than just learning a robot model, learning-based methods also learn robot interactions with the terrain, thus being terrain-aware. However, these methods fail in unstructured environments [37] as they cannot adapt on the fly with the terrain or exhibit catastrophic forgetting [38], which is the tendency to forget previously learned data upon learning from new data.

Machine-learning-based control methods learn robot behaviors by combining data-driven formulations into predefined robot models [39], [40]. Early methods used Dynamics Mode Decomposition (DMD) [41] and Sparse Identification of Non-Linear Dynamics (SINDy) [42] to learn data-driven models based on system identification and performed terrain navigation [43], [44]. Later, evolutionary algorithms were developed to optimize parameters of a robot model in an online learning fashion for robust navigation [45], [46]. For robots with multiple degrees of freedom, methods were developed that use a combination of iterative Linear Quadratic Regulators (iLQR) and machine learning search to explore multiple robot configurations and plan self-adaptive navigation [47]. Similar approaches were designed using a neural network based functional approximator to learn a robot dynamics model and adapt this model with online learning [48]. Robust path planning was performed for safe navigation of autonomous

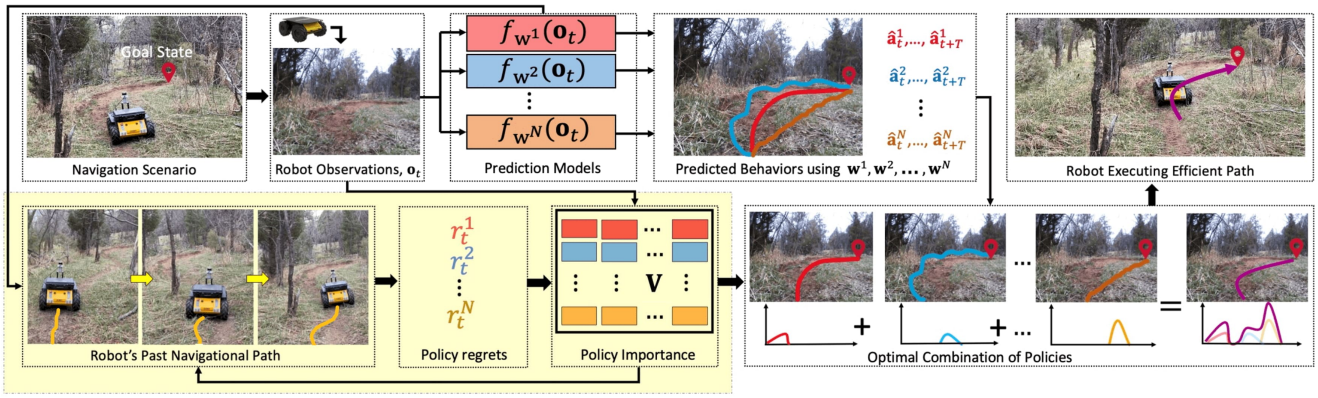


Fig. 2. Overview of our proposed NAUTS approach for robot negotiation to adapt over unstructured terrains. Illustrated is the learning performed by our approach during the training phase. The module in the yellow box illustrates robot negotiation during the execution stage.

vehicles under perception uncertainty [49]. However, these methods do not address adaptation to previously unseen, unstructured terrains, and are unable to address the dynamic nature of the terrain, which often leads to ineffective terrain traversal.

III. Approach

In this section, we discuss our proposed method, NAUTS, for robot traversal adaptation by negotiation. An overview of the approach is illustrated in Fig. 2.

A. Learning Policy Prediction Models

Our approach first learns a non-linear prediction model to estimate future robot states and behaviors for each policy in a previously trained library. Navigational policies describe various goals of navigation, e.g., obstacle avoidance, adaptive maneuvers or max speed. This model enables our approach to predict how a policy works without the requirement of knowing its implementation (i.e., the policy can be treated as a black box). Formally, at time t, we denote the robot terrain observations (e.g., RGB images) input to the i-th policy as $\mathbf{o}_t^i \in \mathbb{R}^q$, where q is the dimensionality of the terrain observations. The robot behavior controls, i.e, navigational behaviors (e.g., linear and angular velocity), and states (e.g., robot's body pose and position) output from the policy are denoted as $\mathbf{a}_t^i \in \mathbb{R}^c$ and $\mathbf{s}_t^i \in \mathbb{R}^m$, with c and m denote the dimensionality of robot behaviors and states respectively. Then the i-th policy can be represented as $\pi^i : (\mathbf{s}_t^i, \mathbf{o}_t^i) \rightarrow \mathbf{a}_t^i$.

Let $\mathbf{g}$ denote the relative goal state (with respect to $\mathbf{s}_t^i$) that the robot needs to reach at time $t+T$. For every policy π^i, we propose to learn a prediction model $f_{\mathbf{w}^i} : (\mathbf{o}_t^i, \mathbf{g}) \rightarrow (\hat{\mathbf{a}}_{t:t+T}^i, \hat{\mathbf{s}}_{t:t+T}^i)$ that predicts a sequence of goal driven T-future robot behaviors $\hat{\mathbf{a}}_{t:t+T}^i$ and states $\hat{\mathbf{s}}_{t:t+T}^i$. The prediction model estimates behaviors for the present time and functions like a local controller, and by estimating robot behaviors and states for future T-steps, it functions as a local planner. We introduce a shallow Gaussian Process (GP) [50] to implement $f_{\mathbf{w}^i}$ that is parameterized by $\mathbf{w}^i$, as shown in Fig. 3. This shallow Gaussian Process with a recursive kernel has been shown in [50] to be equivalent to, but more data-efficient than, a deep Bayesian CNN with infinitely many filters. In addition, as this Gaussian Process assumes that each weight of the network is a distribution instead of scalar values, it allows for uncertainty modeling and thus, is robust to environmental variations. We then learn the prediction model $f_{\mathbf{w}^i}$ by solving the following regularized optimization problem:

$$\min_{\mathbf{w}^i} \; \lambda_1 \mathcal{L}\big((\pi^i(\mathbf{s}_{t:t+T}^i, \mathbf{o}_{t:t+T}^i), \mathbf{s}_{t:t+T}^i), f_{\mathbf{w}^i}(\mathbf{o}_t^i, \mathbf{g})\big) + \lambda_2 \|\mathbf{g}^i - (\hat{\mathbf{s}}_{t+T}^i - \hat{\mathbf{s}}_t^i)\|_2^2 \tag{1}$$

where $\mathcal{L}(\cdot)$ is the cross-entropy loss [51], mathematically expressed as $\mathcal{L}(p, q) = -\mathbb{E}_p[\log(q)]$. This loss helps the prediction model to be insensitive to noisy observations in unstructured environments due to the logarithmic scale. The first part of Eq. (1) models the error of predicting T-future robot behaviors and states from actual navigational behaviors and states. The second part of Eq. (1) models the error of the robot failing to reach its relative goal state. The hyper-parameters λ_1 and λ_2 model the trade-off between the losses.

Following Eq. (1), the robot learns prediction models for N-different policies. However, when navigating over unstructured terrains, a single policy may not always prove to be effective for all scenarios. For example, the policy of obstacle avoidance may lead to longer traversal time in grass terrain, and the policy of max speed may cause collisions with occluded obstacles.

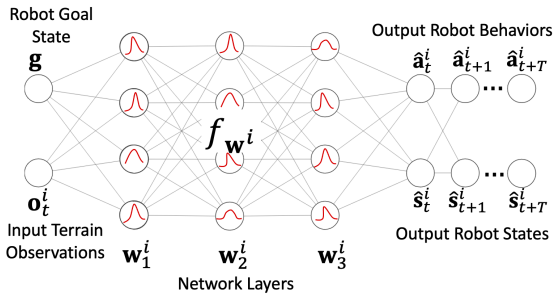


Fig. 3. A shallow GP is designed to implement our prediction model $f_{\mathbf{w}^i}$.

B. Robot Negotiation for Terrain Adaptation

The key novelty of NAUTS is its capability of negotiating between different policies to perform successful and efficient navigation, especially in unstructured off-road terrains. Given

N-policies in the library, NAUTS formulates robot adaptation by negotiation under the mathematical framework of multi-arm bandit (MAB) optimization [52]. MAB comes from the hypothetical experiment where the robot must choose between multiple policies, each of which has an unknown regret with the goal of determining the best (or least regretted) outcome on the fly. We define regret, $r_t^i : \big(\mathbf{o}_{t-T}^i, \mathbf{g}\big)\big) \rightarrow \mathbb{R}^+$, of the i-th policy at time t as the error of not reaching i) the goal position and ii) the goal position in minimum time and effort. We calculate the regret for each policy as:

$$r_t^i = \Big(\frac{\|\mathbf{g}\|_2 \|\hat{\mathbf{s}}_t^i\|_2}{(\mathbf{g})^\top (\hat{\mathbf{s}}_t^i)} - 1 \Big) + \sum_{k=t-T}^{t} (t-k)(\hat{\mathbf{a}}_k^i)^\top \hat{\mathbf{a}}_k^i \quad (2)$$

where the first part of Eq. (2) models the error of not reaching the goal position, with zero regret if the robot reached its goal position. This error grows exponentially if the robot has deviated from the goal position. The second part of Eq. (2) models the error of not reaching the goal in minimum time and effort. Specifically, the regret is smaller when the robot uses fewer values of navigational behaviors to reach the same goal and also if the robot reaches the goal in minimum time due to the scaling term $(t-k)$.

Unstructured terrain-aware negotiation can be achieved using the best subset of policies that minimize the overall regret in the present terrain as:

$$\begin{aligned} \min_{\mathbf{V}} \quad & \lambda_3 \sum_{i=1}^{N} \mathcal{R}(\mathbf{o}_t^i, r_t^i; \mathbf{v}^i) + \lambda_4 \|\mathbf{V}\|_E \quad (3) \\ \text{s.t.} \quad & \sum_{i=1}^{N} (\mathbf{o}_t^i)^\top \mathbf{v}^i = 1 \end{aligned}$$

where $\mathcal{R}(\cdot)$, parameterized by $\mathbf{v}^i \in \mathbb{R}^q$, is the terrain-aware regret of choosing policy π^i in the present terrain and $\mathbf{V} = [\mathbf{v}^1, \ldots, \mathbf{v}^N] \in \mathbb{R}^{N \times q}$. Mathematically, $\mathcal{R}(\mathbf{o}_t^i, r_t^i; \mathbf{v}^i) = \sum_{k=t}^{t+T} \|r_k^* - (\mathbf{o}_t^i)^\top \mathbf{v}^i r_k^i\|_2^2$, with $r_k^* = \min r_k^i; i = 1, \ldots, N$. The use of a linear model enables real-time convergence for terrain-aware policy negotiation. The column sum of $\mathbf{V}$ indicates the weights of each policy towards minimizing the overall regret of robot navigation. In doing so, the robot recognizes the important policies and exploits these policies to maintain efficient navigation. However, we also need to explore the various policies to improve navigation efficiency or even learn in an unknown environment, which is achieved by the regularization term in Eq. (3), called the exploration norm. Mathematically, $\|\mathbf{V}\|_E = \sum_{i=1}^{N} \frac{\|\mathbf{V}\|_F}{\|\mathbf{v}^i\|_2}$, where the operator $\|\cdot\|_F$ is the Frobenius norm with $\|\mathbf{V}\|_F = \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{q} (v_j^i)^2}$. The exploration norm enables NAUTS to continuously explore all navigational policies in any terrain. Specifically, the exploration norm enables NAUTS to explore sub-optimal policies by ensuring $\mathbf{v}^i \neq \mathbf{0}$. If $\mathbf{v}^i = \mathbf{0}$, i.e., if the i-th policy is given zero importance, then the value of objective in Eq. (3) would be very high. The hyper-parameters λ_3 and λ_4 control the trade-off between exploration and exploitation during negotiation. The constraints in Eq. (3) normalize the various combination of navigational policies.

Algorithm 1: Optimization algorithm for solving the robot negotiation problem during execution in Eq. (3).

Input : Policies $\mathbf{W}^*$ and Weights $\mathbf{V}^* \in \mathbb{R}^{N \times q}$
Output : Optimized Weights for Negotiation $\mathbf{V}^* \in \mathbb{R}^{N \times q}$

```
while goal is not reached do
    for i = 1, ..., N do
        Obtain predicted behavior â^i_{t:t+T} and states ŝ^i_{t:t+T}
          from f_{w^i*}(o_{t0}, g);
        Calculate regret of i-th policy r^i from Eq. (2);
    Calculate r*_{t0} = min r^i_{t0};  i = 1, ..., N;
    while not converge do
        Calculate diagonal matrix Q with the i-th diagonal
          block given as I / (2||V||_E);
        Compute the columns of the distribution V
          according to Eq. (7);
return: V* ∈ R^{N×q}
```

Integrating prediction model learning and policy negotiation under a unified mathematical framework, robot adaptation by negotiation can be formulated as the following regularized optimization problem:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{V}} & \sum_{i=1}^{N} \Big(\lambda_1 \mathcal{L}\big((\pi^i(\mathbf{s}_{t:t+T}^i, \mathbf{o}_{t:t+T}^i), \mathbf{s}_{t:t+T}^i), f_{\mathbf{w}^i}(\mathbf{o}_t^i, \mathbf{g})\big) \\ & + \lambda_2 \|\mathbf{g}^i - (\hat{\mathbf{s}}_{t+T}^i - \hat{\mathbf{s}}_t^i)\|_2^2 + \lambda_3 \mathcal{R}(\mathbf{o}_t^i, r_t^i; \mathbf{v}^i) \Big) + \lambda_4 \|\mathbf{V}\|_E \\ \text{s.t.} \quad & \sum_{i=1}^{N} (\mathbf{o}_t^i)^\top \mathbf{v}^i = 1 \quad (4) \end{aligned}$$

where $\mathbf{W} = [\mathbf{w}^1, \ldots, \mathbf{w}^N]$. During the training phase, we compute the optimal $\mathbf{W}^*$ and $\mathbf{V}^*$.

During execution, we fix $\mathbf{W}^*$, meaning the prediction models do not update during execution. However, our approach continuously updates $\mathbf{V}^*$ in an online fashion, which allows for negotiation at each step. At every time step t_0, we acquire observations $\mathbf{o}_{t_0}$. For a given robot goal state $\mathbf{g}$, we dynamically choose the best combination of policies as:

$$\mathbf{a}_{t_0:t_0+T} = \sum_{i=1}^{N} (\mathbf{o}_{t_0})^\top \mathbf{v}^{i*} f_{\mathbf{w}^{i*}}(\mathbf{o}_{t_0}, \mathbf{g}) \quad (5)$$

where $\mathbf{a}_{t_0}$ is the behavior executed by the robot following policy negotiation at time t_0 and the behaviors $\mathbf{a}_{t_0:t_0+T}$ make up the local plan for the robot.

C. Optimization Algorithm

During training, we reduce Eq. (4) to simultaneously optimize $\mathbf{W}^*$ and $\mathbf{V}^*$. As the first term is non-linear, reducing Eq. (4) amounts to optimizing a non-linear objective function. We use the zeroth order non-convex stochastic optimizer from [53]. This optimizer has been proven to avoid saddle points and avoids local minima during optimization [53], and is specifically designed for constrained optimization problems like in Eq. (4). Additionally due to its weaker dependence on input data dimensionality [53], $\mathbf{W}$ and $\mathbf{V}$ can be computed faster despite using high dimensional terrain observations.

To perform robot adaptation by negotiation, we optimize $\mathbf{V}$ in an online fashion during the execution phase by solving the MAB optimization problem in Eq. (3), which has a convex objective with non-smooth regularization term. To perform fast online learning for negotiation, we introduce a novel iterative optimization algorithm that is tailored to solve the regularized optimization in Eq. (3), which at each time step performs fast iterations and converges in real-time to a global optimal value of $\mathbf{V}$. This optimization algorithm is provided in Alg. 1. Specifically, to solve for the optimal $\mathbf{V}$, we minimize Eq. (3) with respect to $\mathbf{v}^i$, resulting in:

$$\sum_{k=t}^{t+T} \lambda_3 \big(2(r_k^i)^2 (\mathbf{o}_t^i)^\top (\mathbf{o}_t^i)\mathbf{v}^i - 2r_k^* r_k^i \mathbf{o}_t^i\big) + \lambda_4 \mathbf{Q}\mathbf{v}^i = 0 \quad (6)$$

where $\mathbf{Q}$ is a block diagonal matrix expressed as $\mathbf{Q} = \frac{\mathbf{I}}{2\|\mathbf{V}\|_E}$ and $\mathbf{I} \in \mathbb{R}^{N\times N}$ is an identity matrix. Then, we compute $\mathbf{v}^i$ in a closed-form solution as:

$$\mathbf{v}^i = (\lambda_4 \mathbf{Q} + 2\sum_{k=t}^{t+T} \lambda_3 (r_k^i)^2 (\mathbf{o}^i)^\top \mathbf{o}^i)^{-1} \lambda_3 \sum_{k=t}^{t+T} (2r_k^* r_k^i \mathbf{o}^i) \quad (7)$$

Because $\mathbf{Q}$ and $\mathbf{V}$ are interdependent, we are able to derive an iterative algorithm to compute them as described in Algorithm 1.

Convergence. Algorithm 1 is guaranteed to converge to the optimal solution for the optimization problem in Eq. (3)[1].

Complexity. For each iteration of Algorithm 1, computing Steps 3, 4, and 7 is trivial, and Step 8 is computed by solving a system of linear equations with quadratic complexity.

IV. Experiments

This section presents the experimental setup and implementation details of our NAUTS approach, and provides a comparison of our approach with several previous state-of-the-art methods.

A. Experimental Setup

We use a Clearpath Husky ground mobile robot for our field experiments. The robot is equipped with an Intel Realsense D435 color camera, an Ouster OS1-64 LiDAR, a Global Positioning System (GPS), and an array of sensors including a Microstrain 3DM-GX5-25 Inertial Measurement Unit (IMU) and wheel odometers. The robot states, i.e., robot pose, are estimated using an Extended Kalman Filter (EKF) [54], applied on sensory observations from LiDAR, IMU, GPS, and wheel odometers. The RGB images and the estimated robot states are used as our inputs. The robot runs a 4.3 GHz i7 CPU with 16GB RAM and Nvidia 1660Ti GPU with 6GB of VRAM, which runs non-linear behavior prediction models at 5Hz and policy negotiation at 0.25 Hz.

We evaluate our approach on navigation tasks that require traversing from the robot's initial position to a goal position, and provide a performance comparison against state-of-the-art robot navigation techniques including Model Predictive Path Integral (MPPI) [7] control, Terrain Representation and Apprenticeship Learning (TRAL) [5], Berkley Autonomous Driving Ground Robot (BADGR) [10], and Learning to Navigate from Disengagements (LaND) [9]. To quantitatively evaluate and compare these approaches to NAUTS, we use the following evaluation metrics:

- *Failure Rate (FR)*: This metric is defined as the number of times the robot fails to complete the navigation task across a set of experimental trials. If a robot flips or is stopped by a terrain obstacle, it is considered a failure. Lower values of FR indicate better performance.
- *Traversal Time (TT)*: This metric is defined as the time taken to complete the navigation task over given terrain. Smaller values of TT indicate better performance.
- *Distance traveled (DT)*: This metric is defined as the total distance traveled by the robot when completing a navigational task. A smaller DT value may indicate better performance.
- *Adaptation time (AT)*: This metric is defined as the time taken by the robot to regain half its linear velocity when introduced to an unseen unstructured environment. A lower value of AT may indicate better performance.

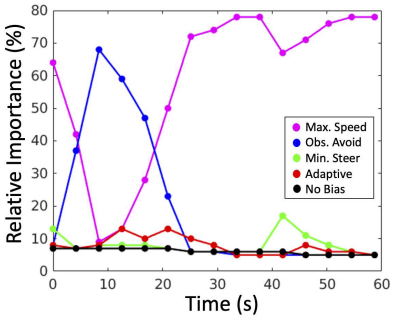
To collect the training data, a human expert demonstrates robot driving over simple terrains of concrete, short grass, gravel, medium-sized rocks, large-sized rocks and forest terrain. Each of these terrain were used to learn one specific aspect of robot navigation such as adjusting traversal speeds over large-sized rocks, or obstacle avoidance using the forest terrain. Specifically, we used these terrains to learn from a library of five distinct navigational policies:

- *Maximum Speed:* When following this navigational policy, the human expert drives with the maximum traversal speed irrespective of the terrain the robot traverses upon. The aim when following the maximum speed navigational policy is to teach the robot to cover as much distance as possible in the least amount of time. Thus, while collecting training data with this policy the expert demonstrator uses straight line traversal without steering the robot.
- *Obstacle Avoidance:* While following this policy, the expert demonstrates how to maneuver by driving around obstacles to avoid collision. To learn this policy, expert demonstrations in forest terrains are used where humans navigate the forest by avoiding trees and logs while moving the robot through the terrain. The underlying goal with this policy is to teach the robot to steer around obstacles.
- *Minimum Steering:* For this policy, the expert drives the robot with limited steering. During navigation, linear velocity is fixed to 0.75 m/s and obstacle avoidance is performed by beginning to turn the robot when it is further away from obstacles instead of making short, acute turns. The policy differs from obstacle avoidance by maintaining a fixed speed while taking a smooth and long maneuver around obstacles.
- *Adaptive Maneuvers:* While following this policy, the expert varies the robot's speed across different terrain

[1]Derivation is presented in the supplementary material at: https://arxiv.org/pdf/2207.13647.pdf

(a) Tall grass terrain

(b) Paths taken by different methods



(c) Importance of different policies

Fig. 4. The tall grass terrain used in our experiments and the qualitative results over this terrain.

to reduce traversal bumpiness. Specifically, with terrains that are relatively less rugged such as concrete or short-grass, the expert demonstrator uses high speed maneuvers. On the other hand, over terrains with high ruggedness such as gravel or medium sized rocks, the expert demonstrator uses slower speeds, with the slowest traversal speed across the large rocks terrain.

- *No Navigational Bias:* When following this policy, multiple expert demonstrators navigate the robot across the different terrains without particular policy bias, i.e., without following any specific navigational policy. The underlying goal behind using such policy is to cover most of the common navigational scenarios encountered by the robot, and include the navigational bias from multiple expert demonstrators.

For each policy, the robot is driven on each of the different terrains, resulting in approximately 108000 distinctive terrain observations with the corresponding sequence of robot navigational behaviors and states for each navigational policy. No further pre-processing is performed on the collected data. We use this data to learn optimal π^i, $i = 1, \ldots, N$ and $\mathbf{V}$ during training. We learn these parameters for different values of hyper-parameters of the NAUTS approach, i.e., λ_1, λ_2, λ_3, λ_4 and T. The combination of these hyper-parameters that results in the best performance of NAUTS during validation are then used for our experiments. In our case, the optimal performance of NAUTS is obtained at $\lambda_1 = 0.1, \lambda_2 = 10$, $\lambda_3 = 1$ and $\lambda_4 = 0.1$ for $T = 9$.

TABLE I

QUANTITATIVE RESULTS FOR SCENARIOS WHEN THE ROBOT TRAVERSES OVER DYNAMIC, UNCERTAIN GRASS TERRAIN.

Metrics	MPPI [7]	TRAL [5]	BADGR [10]	LaND [9]	**NAUTS**
FR (/10)	3	3	**1**	5	**1**
TT (s)	88.72	72.99	64.47	90.18	**58.79**
DT (m)	68.58	56.69	50.29	64.93	**36.57**
AT (s)	14.23	10.92	–	–	**6.24**

B. Navigating over Dynamic Uncertain Grass Terrain

In this set of experiments, we evaluate robot traversal performance over the tall grass terrain environment, as shown in Fig. 4(a). This is one of the most commonly found terrains in off-road environments and is characterized by deformable dynamic obstacles added with the terrain uncertainty of occluded obstacles. The process of negotiation is continuously performed throughout the execution phase. The evaluation metrics for each of the methods are computed across ten trial runs over the tall grass terrain environment.

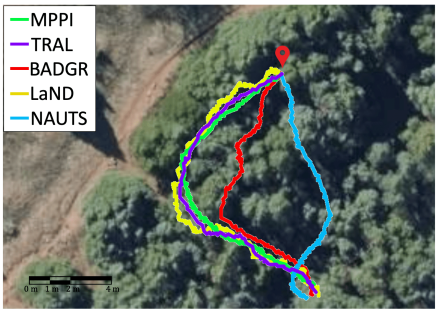
The quantitative results obtained by our approach and its comparison with other methods are presented in Table I. In terms of the FR metric, BADGR and NAUTS obtain the lowest values, whereas MPPI, TRAL and LaND have high FR values. Navigation failure for MPPI, TRAL and LaND generally occurred as the robot transitioned into the tall grass terrain where it would get stuck after determining the tall grass was an obstacle. Failure cases for NAUTS and BADGR occurred when the robot was stuck in the tall grass terrain due to wheel slip. Both NAUTS and BADGR obtain significantly fewer failures than MPPI and LaND methods due to their ability to adapt to different terrains.

When comparing the traversal time and the distance traversed by the different methods, we observe that NAUTS obtains the best performance followed by BADGR and TRAL. The LaND and MPPI approaches obtain higher TT and DT metrics, with MPPI performing the poorest in terms of DT and LaND performing poorest in terms of TT. A qualitative comparison, from a single trial, of the path traversed by these methods is provided in Fig. 4(b). Notice, MPPI, LaND, and TRAL all consider tall grass as obstacles and avoid this terrain while traversing. We observe that BADGR and NAUTS explore tall grass terrain and the shortest path is taken with our NAUTS approach resulting in the lowest DT and TT values.

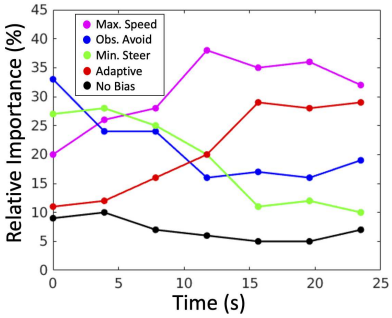
NAUTS also performs better than the TRAL and MPPI approaches in terms of the AT metric. The AT metric is observed when robots encounter an unseen terrain and require adaptation. In this environment, that happens when the robot transitions into the tall grass terrain. We do not provide AT values for BADGR and LaND as both approaches have a fixed linear velocity without adaptation. Overall, we observe that our approach obtains successful navigation (from FR metric) and better efficiency (from TT and DT metrics) over previous methods.

Fig. 4(c) illustrates the NAUTS negotiation process between the five policies in the tall grass terrain. NAUTS learns optimal combinations of policies in real-time during execution

(a) Forest terrain



(b) Path taken by different methods



(c) Importance of different policies

Fig. 5. The forest terrain used in our experiments and the qualitative results over this terrain.

(each update is marked by dots in the figure). Initially, max speed has higher importance over other policies. However, as the robot enters tall grass, obstacle avoidance becomes more important. While traversing further, the robot learns to give more importance to the max speed policy again and obstacle avoidance becomes less important. All other policies have relatively low importance, but they never reach zero, as NAUTS regularly evaluates the different policies.

TABLE II

QUANTITATIVE RESULTS FOR SCENARIOS WHEN THE ROBOT TRAVERSES OVER UNSEEN DYNAMIC, UNSTRUCTURED OFF-ROAD FOREST TERRAIN.

Metrics	MPPI [7]	TRAL [5]	BADGR [10]	LaND [9]	**NAUTS**
FR (/10)	5	5	4	7	**2**
TT (s)	34.28	33.95	26.17	33.98	**24.21**
DT (m)	24.68	23.77	20.94	26.51	**16.45**
AT (s)	10.04	11.93	–	–	**7.32**

C. Navigating on Unseen Unstructured Forest Terrain

In this set of experiments, we evaluate navigation across forest terrains. Apart from high uncertainty and dynamic obstacles, this terrain has different characteristics that the robot has not previously seen during training, e.g, terrain covered with wood chips, dried leaves, rocks, and tree branches. Similar to the previous set of experiments, the evaluation metrics in the forest terrain are computed across ten runs for each of the methods.

The quantitative results over off-road forest terrain are presented in Table II. In terms of the FR metric, we observe a similar trend seen in the tall grass terrain experiments. Specifically, MPPI and TRAL have similar performance in terms of FR metrics. Our NAUTS approach obtains the lowest FR value followed by the BADGR approach, and the LaND approach obtains the highest value. Common failures in the forest terrain occur when tree branches occluding the terrain are classified as obstacles or traversing over large rocks, wooden tree barks, or mud in the terrain cause the robot to get stuck. NAUTS also obtains better efficiency in both the TT and DT metrics, followed by the BADGR approach. Again, MPPI and TRAL both obtain similar TT and DT values, and LaND exhibits the worst performance.

Fig. 5(b) illustrates qualitatively how MPPI, TRAL, and LaND avoid uncertain and unseen paths and follow an existing trail. However, BADGR explores unknown paths, reaching the goal faster than these methods, and NAUTS outperforms all methods by exploring different policies in this unseen terrain. In this set of experiments, the AT metric is observed throughout navigation as each section of the terrain is not previously seen by the robot and requires the robot to adapt. NAUTS obtains better AT values than MPPI and TRAL, indicating a better adaptation capability.

Fig. 5(c) illustrates the negotiation process by NAUTS in the forest terrain. At the start of the navigation task, each policy has different importance, with obstacle avoidance being the most significant. As the robot continues with the navigation task, it learns to use the optimal combination of policies, which results in the most efficient navigation. Thus, the max speed and adaptive navigational policies become more significant than other policies. It is important to note that there is no single optimal policy throughout navigation due to i) the highly unstructured nature of this terrain and ii) the continuous exploration of the NAUTS approach.

V. CONCLUSION

In this paper, we introduce the novel NAUTS approach for robot adaptation by negotiation for navigating in unstructured terrains, that enables ground robots to adapt their navigation policies using a negotiation process. Our approach learns a non-linear prediction model to function as a terrain-aware joint local controller and planner corresponding to various policies, and then uses the negotiation process to form agreements between these policies in order to improve robot navigation efficiency. Moreover, our approach explores different policies to improve navigation efficiency in a given environment continuously. We also developed a novel optimization algorithm that solves the global optimal solution to the robot negotiation problem in real-time. Experimental results have shown that our approach enables a robot to negotiate its behaviors with the terrain and delivers more successful and efficient navigation compared to the previous methods.

REFERENCES

[1] D. Lattanzi and G. Miller, "Review of Robotic Infrastructure Inspection Systems," *JIS*, vol. 23, no. 3, p. 04017004, 2017.

[2] M. J. Schuster, S. G. Brunner, K. Bussmann, S. Büttner, A. Dömel, M. Hellerer, H. Lehner, P. Lehner, Porges, *et al.*, "Towards Autonomous Planetary Exploration," *JINT*, vol. 93, no. 3, pp. 461–494, 2019.

[3] H.-T. L. Chiang, B. HomChaudhuri, L. Smith, and L. Tapia, "Safety, Challenges, and Performance of Motion Planners in Dynamic Environments," in *Robotics Research*. Springer, 2020, pp. 793–808.

[4] G. Sartoretti, S. Shaw, K. Lam, N. Fan, M. Travers, and H. Choset, "Central Pattern Generator with Inertial Feedback for Stable Locomotion and Climbing in Unstructured Terrain," in *ICRA*, 2018.

[5] S. Siva, M. Wigness, J. Rogers, and H. Zhang, "Robot Adaptation to Unstructured Terrains by Joint Representation and Apprenticeship Learning," in *RSS*, 2019.

[6] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces," *T-RO*, vol. 12, no. 4, pp. 566–580, 1996.

[7] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive Driving with Model Predictive Path Integral Control," in *ICRA*, 2016.

[8] L. Moysis, E. Petavratzis, C. Volos, H. Nistazakis, and I. Stouboulos, "A Chaotic Path Planning Generator Based on Logistic Map and Modulo Tactics," *RAS*, vol. 124, p. 103377, 2020.

[9] G. Kahn, P. Abbeel, and S. Levine, "LaND: Learning to Navigate from Disengagements," *RAL*, vol. 6, no. 2, pp. 1872–1879, 2021.

[10] P. A. Gregory Kahn and S. Levine, "BADGR: An Autonomous Self-Supervised Learning-based Navigation System," vol. 6, no. 2, 2021, pp. 1312–1319.

[11] G. Kahn, A. Villaflor, B. Ding, P. Abbeel, and S. Levine, "Self-supervised Deep Reinforcement Learning with Generalized Computation Graphs for Robot Navigation," in *ICRA*, 2018.

[12] S.-H. Han, H.-J. Choi, P. Benz, and J. Loaiciga, "Sensor-Based Mobile Robot Navigation via Deep Reinforcement Learning," in *BIGCOMP*, 2018.

[13] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "RMA: Rapid Motor Adaptation for Legged Robots," *RSS*, 2021.

[14] B. Liu, X. Xiao, and P. Stone, "A Lifelong Learning Approach to Mobile Robot Navigation," *RAL*, 2021.

[15] F. Zenke, B. Poole, and S. Ganguli, "Continual Learning through Synaptic Intelligence," in *ICML*, 2017.

[16] G. Kahn, A. Villaflor, P. Abbeel, and S. Levine, "Composable Action-Conditioned Predictors: Flexible Off-policy Learning for Robot Navigation," in *CoRL*, 2018.

[17] J.-Y. Jhang, C.-J. Lin, C.-T. Lin, and K.-Y. Young, "Navigation Control of Mobile Robots Using an Interval Type-2 Fuzzy Controller Based on Dynamic-group Particle Swarm Optimization," *IJCAS*, vol. 16, no. 5, pp. 2446–2457, 2018.

[18] A. Sinha, M. O'Kelly, R. Tedrake, and J. C. Duchi, "Neural Bridge Sampling for Evaluating Safety-Critical Autonomous Systems," *NIPS*, 2020.

[19] A. Sinha, M. O'Kelly, H. Zheng, R. Mangharam, J. Duchi, and R. Tedrake, "Formulazero: Distributionally Robust Online Adaptation via Offline Population Synthesis," in *ICML*, 2020.

[20] A. Saffiotti, "The uses of Fuzzy Logic in Autonomous Robot Navigation," *IJSC*, vol. 1, no. 4, pp. 180–197, 1997.

[21] M. Wang and J. N. Liu, "Fuzzy Logic-Based Real-Time Robot Navigation in Unknown Environment with Dead Ends," *RAS*, vol. 56, no. 7, pp. 625–643, 2008.

[22] L. Rabiner, R. Crochiere, and J. Allen, "FIR System Modeling and Identification in the Presence of Noise and with Band-Limited Inputs," *ICASSP*, vol. 26, no. 4, pp. 319–333, 1978.

[23] Y. Bolea, A. Grau, and A. Sanfeliu, "Non-speech Sound Feature Extraction Based on Model Identification for Robot Navigation," in *CIARP*, 2003.

[24] D. Pebrianti, Y. H. Hao, N. A. S. Suarin, L. Bayuaji, Z. Musa, M. Syafrullah, and I. Riyanto, "Motion Tracker Based Wheeled Mobile Robot System Identification and Controller Design," in *Intelligent Manufacturing & Mechatronics*, 2018.

[25] J. Van Den Berg, S. Patil, and R. Alterovitz, "Motion Planning under Uncertainty using differential Dynamic Programming in Belief Space," in *Robotics Research*. Springer, 2017, pp. 473–490.

[26] H.-j. Zhang, J.-w. Gong, Y. Jiang, G.-m. Xiong, and H.-y. Chen, "An Iterative Linear Quadratic Regulator based Trajectory Tracking Controller for Wheeled Mobile Robot," *JZUS-C*, vol. 13, no. 8, pp. 593–600, 2012.

[27] T. M. Howard, C. J. Green, and A. Kelly, "Receding Horizon Model-Predictive Control for Mobile Robot Navigation of Intricate Paths," in *FSR*, 2010.

[28] O. A. Hafez, G. D. Arana, and M. Spenko, "Integrity Risk-Based Model Predictive Control for Mobile Robots," in *ICRA*, 2019.

[29] A. Tahirovic and G. Magnani, "General Framework for Mobile Robot Navigation using Passivity-based MPC," *TACON*, vol. 56, no. 1, pp. 184–190, 2010.

[30] B. O. Koopman, "Hamiltonian Systems and Transformation in Hilbert Space," *PNAS*, vol. 17, no. 5, pp. 315–318, 1931.

[31] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Generalizing Koopman Theory to Allow for Inputs and Control," *SIADS*, vol. 17, no. 1, pp. 909–930, 2018.

[32] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition," *JNS*, vol. 25, no. 6, pp. 1307–1346, 2015.

[33] C. G. Atkeson and S. Schaal, "Robot Learning from Demonstration," in *ICML*, 1997.

[34] M. Wigness, J. G. Rogers, and L. E. Navarro-Serment, "Robot Navigation from Human Demonstration: Learning Control Behaviors," in *ICRA*, 2018.

[35] S. Siva, M. Wigness, J. Rogers, and H. Zhang, "Enhancing Consistent Ground Maneuverability by Robot Adaptation to Complex Off-Road Terrains," in *CoRL*, 2021.

[36] Z. Wang, X. Xiao, B. Liu, G. Warnell, and P. Stone, "APPLI: Adaptive Planner Parameter Learning from Interventions," in *ICRA*, 2021.

[37] M. H. Nampoothiri, B. Vinayakumar, Y. Sunny, and R. Antony, "Recent Developments in Terrain Identification, Classification, Parameter Estimation for the Navigation of Autonomous Robots," *SNAS*, vol. 3, no. 4, pp. 1–14, 2021.

[38] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, "Overcoming Catastrophic Forgetting with Hard Attention to the Task," in *ICML*, 2018.

[39] T. Duriez, S. L. Brunton, and B. R. Noack, *Machine Learning Control-Taming Nonlinear Dynamics and Turbulence*. Springer, 2017.

[40] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.

[41] P. J. Schmid, "Dynamic Mode Decomposition of Numerical and Experimental Data," *JFM*, vol. 656, pp. 5–28, 2010.

[42] G. Mamakoukas, M. Castano, X. Tan, and T. Murphey, "Local Koopman Operators for Data-Driven Control of Robotic Systems," in *RSS*, 2019.

[43] H. Wang and N. Noguchi, "Real-time States Estimation of a Farm Tractor using Dynamic Mode Decomposition," *GPS Solutions*, vol. 25, no. 1, pp. 1–12, 2021.

[44] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-driven Modeling of Complex Systems*. SIAM, 2016.

[45] C. Cáceres, J. M. Rosário, and D. Amaya, "Approach of Kinematic Control for a Non-Holonomic Wheeled Robot using Artificial Neural Networks and Genetic Algorithms," in *IWOBI*, 2017.

[46] D. R. Ramírez, D. Limón, J. Gomez-Ortega, and E. F. Camacho, "Nonlinear MBPC for mobile robot navigation using genetic algorithms," in *ICRA*, 1999.

[47] M. T. Gillespie, C. M. Best, E. C. Townsend, D. Wingate, and M. D. Killpack, "Learning Nonlinear Dynamic Models of Soft Robots for Model Predictive Control with Neural Networks," in *RoboSoft*, 2018.

[48] A. Nagariya and S. Saripalli, "An Iterative LQR Controller for Off-road and On-road Vehicles using a Neural Network Dynamics Model," in *IV*, 2020, pp. 1740–1745.

[49] M. Alharbi and H. A. Karimi, "A Global Path Planner for Safe Navigation of Autonomous Vehicles in Uncertain Environments," *Sensors*, vol. 20, no. 21, p. 6103, 2020.

[50] A. Garriga-Alonso, C. E. Rasmussen, and L. Aitchison, "Deep Convolutional Networks as Shallow Gaussian Processes," *ICLR*, 2019.

[51] Z. Zhang and M. R. Sabuncu, "Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels," in *NIPS*, 2018.

[52] L. Chan, D. Hadfield-Menell, S. Srinivasa, and A. Dragan, "The Assistive Multi-Armed Bandit," in *HRI*, 2019.

[53] K. Balasubramanian and S. Ghadimi, "Zeroth-Order Nonconvex Stochastic Optimization: Handling Constraints, High Dimensionality, and Saddle Points," *FoCM*, vol. 22, no. 1, pp. 35–76, 2022.

[54] G. G. Rigatos, "Extended Kalman and Particle Filtering for Sensor Fusion in Motion Control of Mobile Robots," *IMACS*, vol. 81, no. 3, pp. 590–607, 2010.