

Interaction-Aware Control for Vegetation Override in Robotic Off-road Environments

Charles Noren^{a,*}, Bhaskar Vundurthy^a, Sebastian Scherer^a and Matthew Travers^a

^aThe Robotics Institute, School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, 15213-3890, Pennsylvania, United States of America

ARTICLE INFO

Keywords:

Trajectory Optimization
Vegetation Override
Off-road Driving
Robotics

ABSTRACT

Robotic systems tasked with completing off-road economic, military, or humanitarian missions often encounter environmental objects when traversing unstructured terrains. Certain objects (e.g. safety cones) must be avoided to ensure operational integrity, but others (e.g. small vegetation) can be interacted with (e.g. overridden/pushed) safely. Pure object-avoidance assumptions in conventional robotic system navigation policies may lead to inefficient (slow) or overly-cautious (immobilized) traversal behaviors in off-road terrains. To address this gap in system performance, we draw inspiration from existing hybrid dynamic system control literature. We have designed a nonlinear trajectory optimization controller that utilizes vegetation-interaction models as a jump map in the dynamics constraint. In contrast to purely vision-based navigation policies which classify the traversability of obstacles, the allowable subset of objects with which the vehicle can safely interact is now characterized by a data-driven collision model and the existence of a dynamically-feasible trajectory which satisfies the contact constraints. The controller's capabilities are demonstrated on a full-sized autonomous utility task vehicle where objects including posts and trees of up to 25.4 [mm] and 81.8 [mm] diameter are overridden.

1. Introduction

When [I am] going through vegetation and driving through or over it, [I] always have it in the back of my head how I am going to hit it. Typically that is head on with the front bumper as that will be the strongest point on the vehicle and give me the best leverage to run over [the vegetation] and go through it.

Ryan Arciero, Professional Off-road Driver

The execution of off-road vehicle operations is influenced by the decision of whether to take an action to “avoid” or “strike” environmental objects (Rybansky, 2017). As seen in the quote above, when striking vegetation, professional off-road racing drivers carefully internally model and allow certain collisions for safe travel off-road. It stands to reason that in order to emulate the high performance of professional drivers in off-road conditions, robotic vehicles will also need to consider the same decisions about collisions. However, there is a great imbalance of work in the development of the “avoid” and “strike” actions for robotic platforms, with a majority of work centering on the “avoid” action. We believe that this imbalance has led to a capability gap in robotic off-road operations that require vegetation override. Specifically, we claim that robot control policies which rely solely on avoiding environmental objects exist within a paradigm that both inadequately represents and models the challenges regarding off-road terrains and environmental objects.

For tasks such as on-road (Paden et al., 2016) and on-trail robotic operations, robotic vehicles generally only demonstrate the need for avoidance capabilities. Requirements reaching back to the Defense Advanced Research Projects Agency (DARPA) Grand and Urban Challenge(s) (Thurn et al., 2007; Urmson et al., 2009) reflect the need for certain areal or object avoidance behaviors. Thus, many methods have evolved in the field robotics, planning, and control domains to address such requirements. Expressing object avoidance as a general constraint in the state space for both state-based planning (Lavelle, 2006) and for optimal-control-based approaches (Jianyu et al., 2017; Howell et al., 2019) are well-established (Paden et al., 2016). Furthermore, the use of penalty-based methods for object avoidance (Williams et al., 2018) and avoidance-guaranteed “proof-by-construction” techniques have seen renewed attention in recent years (Liu and Tomizuka, 2014; Noren et al., 2021). However, the quote above speaks to the need to traverse objects, and thus, only taking avoidance actions or forbidding contact may not reflect expert driving behavior.

Although works developing the “strike” action set are in fewer numbers, the field robotics, planning, and control communities have all evolved different means to reason about collisions. The approaches from the field robotics community, such as classifying objects as “strike-able” via visual measures of object geometry or through the use of virtual and physical bumpers (Kelly et al., 2006) draw from the practical needs of robotic platform operation. High-level planning methods, such as those described by Rybansky (2017), develop collision rules dependent on the vehicle configuration and environmental structure to govern expectations (for example, maximum traversal speed through an area) on vehicle operations. Outside of treating contacts as

✉ cnoren@andrew.cmu.edu (C. Noren); pvundurthy@andrew.cmu.edu (B. Vundurthy); basti@andrew.cmu.edu (S. Scherer); mtravers@andrew.cmu.edu (M. Travers)

ORCID(s): 0000-0002-5684-5446 (C. Noren); 0000-0002-6876-4424 (B. Vundurthy); 0000-0002-8373-4688 (S. Scherer)

disturbances, control approaches generally fall into two categories: through-contact models and hybrid-contact models. Through-contact models attempt to capture the nonsmooth physics of the interaction and directly evaluate those physics during the determination of the control (Posa et al., 2013; Howell et al., 2023). Hybrid-contact models develop rules-based functional mappings to model the changes of state associated with a discrete event (Hargraves and Paris, 1987) such as a collision. These methods are commonly used in direct collocation trajectory optimization problems for legged locomotion (Kelly, 2017). Unfortunately, many of the methods that allow contact for a “strike” action solely deal with geometric representations of the terrain or object and do not capture all aspects needed for traversing or “overriding” an object.

Furthermore, an additional complicating factor in utilizing contact models for robotics applications lies in their ability to be utilized for online reasoning. Such constraints on real-time operations are common in the planning domain, where no assurances exist on an object-interaction-free path. These types of problems are known as the “navigation among moveable obstacles” (NAMO) problem (Wilfong, 1991). In this particular problem, the goal is to enable the planning entity (e.g., robotic platform) to restructure, interact with, or rearrange the environment in order to meet the planning entity’s goals (Ellis et al., 2022). Early approaches to the NAMO problem performed a state-space decomposition to find manipulation points, and then performed a heuristic search over the freespace configuration components in order to avoid the complexity of multi-object planning (Stilman and Kuffner, 2004). Recently, Saxena has shown that connecting the problem to the multi-agent pathfinding domain allows for a decomposition of the problem into a configuration-search step and a physics simulation step (Saxena et al., 2021; Saxena and Likhachev, 2023a,b). This online simulation of physical interaction has proven computationally costly and difficult to run in real-time, though attempts at learning dynamical constraints (Scholz et al., 2016) and massively parallelized simulation of simple models (Abraham et al., 2020) have shown improvements to plan computation speed. Unfortunately, these methods often do not leverage existing terramechanical modeling studies or rely on compute-expensive processes which may not be available on a robotic platform.

Due to the prevalence and need to interact with vegetation in the common area of operations of interest, the off-road mobility and cross-country movement communities have a long history of modeling such nongeometric requirements for vegetation with low computational requirements. For completeness, we first provide a brief overview of classical and contemporary vegetation-interaction modeling techniques before we state the claimed *contributions of this work*.

The U.S. Army Engineer Waterways Experiment Station (Blackmon and Randolph, 1968) and the U.S. Army Engineer Research and Development Center (Mason et al., 2012) have developed models for the override of different

post and vegetation classes. These studies contribute to larger mobility models, such as the North Atlantic Treaty Organization (NATO) Reference Mobility Model (NRMM) (Bradbury et al., 2018), which are used to analyze the capabilities of both crewed vehicles and robotic platforms (Vong et al., 1999). The NRMM continues to see advancements and refinements to develop higher-fidelity representations of vegetative objects, with recent work focusing upon improving override-force modeling utilizing a robotic test platform (Moore et al., 2024). Yet, advancements in the vegetation-interaction modeling domain are not solely captured within the NRMM. Rybansky (2020) performed a significant study that conducted several vegetation overrides with different classes of vehicles. However, Rybansky’s mobility models remain confined to the domain of vehicle mobility analysis and were not used in an online capacity for reasoning in robotic platform operations.

Finally, work in the traversability-prediction domain for off-road robotic driving has also demonstrated the capability to implicitly represent vegetative objects to a robotic system. While many off-road driving datasets include multi-modal sensory data depicting vegetative objects (Jiang et al., 2020), other datasets may include vegetation interaction data itself (Sivaprakasam et al., 2024) or label the traversability of vegetative objects by considering the vegetation models from the NRMM or Mason et al. (2012) in the labeling process (Sharma et al., 2022). Implicit representations of vegetative objects can then be captured in the learned off-road mobility and traversability policies for off-road operations (Castro et al., 2023; Frey et al., 2023; Chen et al., 2024; Frey et al., 2024). In particular, online learning methodologies can take advantage of observed proprioceptive data (Castro et al., 2023) or human demonstration (Frey et al., 2023) to adapt the system behavior online in response to environmental stimuli, but these works do not directly model collisions.

In particular, the current state-of-the-art for off-road robotic systems that perform traversal with collisions utilizes haptic feedback to update the robotic system’s confidence of its environmental obstacle model online (Prágr et al., 2023). Undoubtedly, when model confidence is low, such a learning-based approach provides a robust foundation for constructing a model representation online. However, the approach does not take advantage of pre-existing collision models which allow for aggressive maneuvers that exploit awareness of the model during online operation.

We propose an interaction-aware control system that models the effects of non-prehensile interactions on a robotic platform’s motion. This system not only provides a principled methodology for reasoning about interactions with vegetative objectives (like “striking”) but also more accurately captures the system’s motion when such interactions occur. As far as the authors are aware, this is one of the first works that pairs vegetation collision models with online trajectory optimization to override vegetation. In particular, this work accomplishes the above research objective by:

1. **Efficiently modeling vegetation interactions** through the use of existing low-computation cost vegetation override models,
2. **Generating dynamically-feasible interaction trajectories** through the imposition of collocation constraints on the vehicle's motion model, and
3. **Operating in real-time** to enable the maneuver of a nonholonomic wheeled robotic platform through a cluttered off-road environment.

This work begins with a description of the proposed control structure in the second section, and controller simulations are shown in the third. Results from hardware experimentation follow the simulation results, and the paper concludes with reflections and a discussion of future work.

2. Method (Algorithm) Overview

This section begins with a description of direct collocation methods for trajectory optimization and then discusses vegetation and vehicle modeling. The proposed trajectory optimization control technique is then motivated from the description and models.

2.1. Direct Trajectory Optimization Techniques

Trajectory optimization problems determine an optimal state and control sequence that minimize an objective function (Kelly, 2017). Equation (1) shows an objective function of the trajectory optimization problem where the boundary term is not explicitly dependent on time

$$J(t_0, t_f, \mathbf{x}(t), \mathbf{u}(t)) = J_f(\mathbf{x}(t_0), \mathbf{x}(t_f)) + \int_{t_0}^{t_f} w(\mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau, \quad (1)$$

and where decision variables $t_0, t_f, \mathbf{x}(t), \mathbf{u}(t)$ are, in order: the initial time, the final time, and the state and control trajectories. Terms $J_f(\cdot)$ and $w(\cdot)$ are the boundary and integral cost terms. The objective function is then minimized in a mathematical program

$$\min_{t_0, t_f, \mathbf{x}(t), \mathbf{u}(t)} J(t_0, t_f, \mathbf{x}(t), \mathbf{u}(t)), \quad (2a)$$

$$\text{subject to } \dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)), \quad (2b)$$

$$\mathbf{h}(t, \mathbf{x}(t), \mathbf{u}(t)) \leq 0, \quad (2c)$$

$$\mathbf{g}(t_0, t_f, \mathbf{x}(t_0), \mathbf{x}(t_f)) \leq 0, \quad (2d)$$

where (2b) are the system dynamics and (2c) and (2d) represent general path and boundary constraints, respectively.

Direct trajectory optimization techniques approximately solve the mathematical program for the trajectory optimization proposed in (2) by discretizing and transcribing the problem into a more general nonlinear program. A standard-form nonlinear program is

$$\min_{\mathbf{z}} R(\mathbf{z}), \quad (3a)$$

$$\text{subject to } c(\mathbf{z}) = \mathbf{0}, \quad (3b)$$

$$d(\mathbf{z}) \leq \mathbf{0}, \quad (3c)$$

where all functions are assumed to be at least C^2 smooth. Direct collocation methods represent continuous-time trajectories with a spline of N time-parameterized polynomial segments; and thus the methods discretize the continuous-time trajectory via the $N + 1$ knot points. To ensure that the solution to the program posed in (2) is feasible with respect to the system dynamics as described in (2b), an integral form of the dynamics is approximated through numerical quadrature. Through the use of different quadrature rules, classically trapezoidal or Simpson quadrature, different approximating polynomials are recovered. These approximate integrals are thus posed as collocation constraints in (3b). Additional integral (2c) and boundary constraints (2d) are posed as constraints in (3b) and (3c). Finally, a discrete-time representation of the objective function in (1) must be posed for the nonlinear program (through approximations such as quadrature) and via an augmented state variable \mathbf{z} containing all the decision variables at the knot points.

2.2. Vegetation Override Models

The vegetation override models developed by the off-road mobility and cross-country movement communities, including those by Blackmon and Randolph (1968) and Mason et al. (2012), abstract complex collision interactions into useful low-computational-cost approximations. These approximations generally characterize the required force, work, or velocity a vehicle needs to override a subset of vegetation given some parameterization of the vehicle or environment.

2.2.1. 2012 Mason Override Model

Mason et al. (2012) presents a model for vertically embedded objects in the ground. These objects consist of posts and small trees. Equations are introduced in Mason et al. (2012) to capture the necessary override force for post-like objects, which were then validated primarily through pull tests. The model in Mason et al. (2012) is mainly characterized by vehicle mass and geometry (pushbar height), vegetation and emplacement geometry, soil parameters, and a series of regression coefficients. Mason et al. (2012) then relates a series of energy expenditure and traversal velocity equations from collected data and provides a force model. Equation 10 from Mason et al. (2012) describes the minimum velocity at which the vehicle is required to travel in order to override a post, v_{over} , and is given as

$$v_{over} = \sqrt{\frac{2k\alpha\gamma_d D L_t}{m(h + 0.5 * L_t)}}, \quad (4)$$

where L_t is the burial depth, m is the vehicle mass, h is the height from the ground surface at which the override force

is applied, D is the post diameter, γ_d is the dry density of soil, and α and k are empirical factors derived as described in [Mason et al. \(2012\)](#). As seen in (4), weather conditions directly influence required minimum velocity for override given that a decrease in the dry density of soil yields a lower override velocity.

2.2.2. 1968 Blackmon Override Model

The model presented in [Blackmon and Randolph \(1968\)](#) was regressed from a series of vegetation override tests of different vegetation types in different environments. [Blackmon and Randolph \(1968\)](#) provide unique regressions for force and energy expenditure for these different vegetation types, including: singular coniferous and hardwood trees, arrays of multiple trees struck in unison, and “clumps” of bamboo. From continuous measurements of pushbar force, drivetrain metrics, distance traveled, time, measurements of the impacted trees, and characterizations of the aftermath of the collision, [Blackmon and Randolph \(1968\)](#) construct a model primarily parameterized by the geometry of individual or multiple trees (for example, the radius of a tree) or the clump diameter.

In this study, the authors consider only experiments which require the override of a single tree or post. While [Blackmon and Randolph \(1968\)](#) provide additional override models, this simplification to a single class of vegetation was drawn from limitations in the perception of the necessary characterizing features for arrays of trees. Thus the methodologies presented herein are not limited in scope to single standing trees, aside from the limits discussed in the original [Blackmon and Randolph \(1968\)](#) manuscript itself. Given this simplification, equations B10-B12 from [Blackmon and Randolph \(1968\)](#) which describe the force and work required to override a single standing tree are of particular interest. These equations largely take the form

$$F_h = K_f d_s^3, \quad (5)$$

and

$$W = K_w d_s^3, \quad (6)$$

where F_h horizontal pushbar force, W is the work required to fail a single standing tree, d_s is the stem diameter, and K_f, K_w are constants that are dependent on vehicle geometry (e.g. pushbar height).

Note that one advantage of Blackmon’s model over Mason’s model for use onboard robotic systems is that the model is dependent solely on a visibly measurable quantity (the stem diameter). This single parameter dependency yields a much lower requirement on robotic sensing capabilities and *a priori* soil characterization likely at the expense of model fidelity.

The measure of work produced from Blackmon’s models may then be combined with additional vehicle information (e.g., the operating mass) in order to generate a suitable v_{over} for a sensed piece of vegetation. The equivalent relations to determine this v_{over} may be calculated in the manner discussed in [Mason et al. \(2012\)](#). This allows for either

model to be used in the trajectory optimization techniques discussed later in the manuscript.

2.3. Vehicle Modeling



Figure 1: The modified Polaris RZR UTV

Although the mathematical outline provided above can be applied to multiple dynamical systems, the platform used in this work is shown in Figure 1. The platform is a modified Polaris RZR utility task vehicle (UTV) that can travel up to 20 [$\frac{m}{s}$] in cluttered off-road terrains. The vehicle is ruggedized to collisions and is equipped with an onboard sensor suite that contains monocular and stereo cameras, as well as multiple Light Detection and Ranging (LiDAR) sensors. A nonlinear bicycle model was used to represent the vehicle dynamics. The vehicle state was modeled as: $\mathbf{x} = [p_x, p_y, \psi, \delta, v]$. In order, the elements of this state vector are the vehicle: x-position, y-position, heading, steering angle, and velocity. The model of the vehicle controls include acceleration and steering rate: $\mathbf{u} = [a, \dot{\delta}]$. The continuous-time vehicle dynamics are

$$\begin{aligned} \dot{p}_x &= v * \cos(\psi + \text{atan}(\frac{L_f}{L * \delta})), \\ \dot{p}_y &= v * \sin(\psi + \text{atan}(\frac{L_f}{L * \delta})), \\ \dot{\psi} &= \frac{v}{L} \cos(\text{atan}(\frac{L_f}{L * \delta})) * \tan(\delta), \\ \dot{\delta} &= \dot{\delta}, \\ \dot{v} &= a, \end{aligned} \quad (7)$$

and the parameters may be found in Table 1. The vehicle’s total length is L_T , wheelbase length is L , front axle to center of mass distance is L_f , front axle to nose distance is L_{nose} , lower bull-bar strut height is h_{nose} , width is w , and the mass of the vehicle is m .

Property	Unit	Magnitude
L_T	[m]	3.785
L	[m]	2.972
L_f	[m]	1.412
L_{nose}	[m]	0.915
h_{nose}	[m]	0.533
w	[m]	1.828
m	[kg]	901.0

Table 1
Platform Parameters

2.4. Direct Collocation for Vegetation Override

One central aspect of the collision models presented in the previous section is that each model captures a loss of kinetic energy due to the collision corresponding to the failing of the vegetation or post. However, the structure of the trajectory optimization problem and the nonlinear program in (2) and (3) require at least C^2 smooth functions to accurately represent the dynamics in the collocation constraints. Thus, to incorporate the previously mentioned vegetation models in the trajectory optimization, the impact dynamics must be accounted for. From the formulation, the collocation constraints are evaluated at specific collocation points, which need not be the same as the knot points described earlier. This realization gives rise to the idea of introducing a pointwise-in-time discontinuity in the form of a functional map, commonly described as a “jump” map, between the two states which represents the collision. This concept exists at the center of hybrid-contact representations of collisions and multi-phase direct collocation methods, and is discussed in more detail in [Hargraves and Paris \(1987\)](#) and [Kelly \(2017\)](#). These approaches have not been implemented using the aforementioned vegetation models.

To capture the effects of the collision at a knot point, a mapping must be defined to transition the state at the time of collision, $\mathbf{x}(t_{col})$, to a new post-collision state. That knot point representing the pre-collision state, $\mathbf{x}_{col} = \mathbf{x}(t_{col})$, is mapped to the next knot point via the mapping

$$\mathbf{x}_{k+1} = \mathbf{f}_{col}(\mathbf{x}_k), \quad (8)$$

where subscript k is a general indexing of the knot points, instead of via the selected quadrature rule. While the technique is compatible with the different collision models as described in the vegetation modeling section, this work represented the collision as a loss of velocity at the point of collision. More specifically,

$$\mathbf{x}_{k+1} = \mathbf{f}_{col}(\mathbf{x}_k) = \mathbf{x}_k - [0; 0; 0; 0; v_{over}], \quad (9)$$

where v_{over} could be defined (for example, via [Mason et al. \(2012\)](#)’s model) in (4), where the last element of \mathbf{x}_k corresponds to the velocity in (7). However, enforcing such a mapping at the point of collision does not solely account for the effects of the collision. For the simple jump map in (9), collisions could propel the system in reverse at low speeds, so either a guard function or a lower-bound on

the minimal allowable velocity must be enforced to ensure compliance with physical laws. A natural requirement to ensure the vegetation is overridden is to enforce that the velocity reaches at least v_{over} at the time of collision. Just as the velocity constraint is imposed at the point of collision, additional constraints or allowances may be associated with the knot point \mathbf{x}_{col} .

Unfortunately, specifying the time of contact or the contact sequence is a non-trivial matter. For the posed collision problem, while the time of collision is not known, the point in space at which the contact occurs is known. Fixed time-stepping methods that discretize the problem posed in (2) with a constant time step are challenging to use, as specifying which particular knot point (if any) will represent \mathbf{x}_{col} is equivalent to knowing the specified contact sequence at best and may result in an unsolvable problem at worst. However, by adding the time step used during the determination of the collocation constraints into the decision variables and by allowing the solver to determine a unique time step for each segment, the knot point which specifies the collision takes new meaning. Instead of representing a specific time of collision, the knot point \mathbf{x}_{col} can now be enforced as a specific state at the point of collision through an equality constraint

$$\mathbf{x}_{col} = \mathbf{x}_{obj}, \quad (10)$$

where \mathbf{x}_{obj} is a positional representation of the object in the state space. Equation (10) is then added as a general equality constraint in (3b). Selecting the specific index of \mathbf{x}_{col} is a method hyperparameter. Associated with this additional time decision variable includes the need to provide bounds on sizes of the time steps. Simple inequality bounds may be posed as $h_{low} \leq h \leq h_{high}$ and $h_{low} > 0$. This constraint on the time step lower bound ensures that the solver does not take unphysical actions.

Finally, if no feasible trajectories are found during the solution, such as if the needed velocity to override given a particular model was not obtainable before collision, the platform is commanded to remain stationary or to take an emergency avoidance action. While no uncertainty in the position of the object was considered in this work, uncertainty in the measurements of the vegetation was addressed by taking the maximum v_{over} as prescribed by the vegetation model during the time the vegetation was observed.

2.5. Algorithm Overview

For a given vegetation model (such as [Mason et al.’s 2012](#) post-override model), a set of vehicle dynamics, and the interaction model capturing the “jump map” in the prior section, an operational logic can be implemented on a robotic platform to override vegetation-like objects in the environment. The approach is outlined in Figure 2. Note that this approach assumes that any vegetation that is considered for override may be overridden if the override velocity can be reached from the initial state and given the dynamics (2b).

This logic is demonstrated in an example case motivated by the presented experiments. Given the vehicle’s starting

location (position “A” in Figure 2), a target (goal) location (position “B” in Figure 2) and a single intervening piece of vegetation (located at x_t in Figure 2), the vehicle computes the override velocity (v_{over}) from the selected override model. Different characteristics of the observed piece of vegetation may imply different override velocities are required at the point of collision (x_{col}). For the example figure, assume that the vegetation located at x_t has an override velocity that scales with trunk size. For the given trunk size, the required override velocity is v_2 , outlined in purple. In order to be successfully traversed, the vehicle must reach a velocity of at least v_2 at the point of collision. For vegetation with smaller trunk sizes (for example, vegetation that requires an override velocity v_1 corresponding to the green line), a smaller velocity may be achievable given the initial velocity of the vehicle. For vegetation that requires a large override velocity, the required velocity ($v_3 > v_{col}$, in red) may not be overridden safely as the required velocity can never be reached. The vehicle then executes the corresponding action set determined by the trajectory optimization technique that corresponds to the velocity profile that achieves a velocity equal to or higher than the override velocity (v_2 in this example instance). In the example, either velocity trajectory that takes a velocity higher than the override velocity is thus valid, and a further selection of which velocity is taken depends on other design choices. For example, an emphasis for an added margin of safety may prefer the velocity trajectory corresponding to v_{col} while a preferred minimum-kinetic-energy approach may take the velocity trajectory corresponding to v_2 . In this work, we select the trajectory that minimizes the point-wise ℓ_2 -distance evaluated at the knot points between a reference velocity (nominal travel speed) and the trajectory.

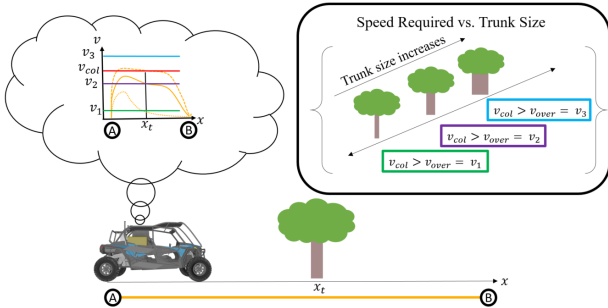


Figure 2: Control logic diagram of a vehicle performing a vegetation override in the presented control framework

3. Simulated Vegetation Override

Before running on the vehicle hardware, a series of tests were conducted in simulation. For all simulation tests, the [Mason et al. \(2012\)](#) model was utilized by modifying only the vehicle and post parameters (L_t, m, h, D).

An example override for a 31.75 [mm] post that is embedded 0.3048 [m] into the ground is shown in Figure 3.

The post is placed 20 [m] in front of the robot in the x-direction. In the figure, the top plot represents distance from the goal and the bottom represents the simulated velocity. Of particular interest in this simulation is the collision that occurs at the 20 [m] mark, where the velocity drops substantially due to the collision with the simulated post. In the velocity graph included in Figure 3, this drop in velocity is labeled v_{over} . To improve the times of solving the trajectory optimization problem, the problem was initialized by dissecting the trajectory into m segments, where m is the number of expected collisions plus one. For a scenario with a singular object that must be overridden, such as in Figure 3, the entire trajectory that is generated is coupled together at the collision points. This is shown by the two colored areas in Figure 3. The bifurcation of the trajectory requires that an additional boundary be generated for each section. For the presented scenario, the initial candidate solution was then constructed by linearly interpolating between the start position and \mathbf{x}_{obj} , and then between \mathbf{x}_{obj} and the end position. While additional optimization could be performed, for the problems considered in this work, the initialization dropped the number of solver iterations by around a third. Note the one exception to this statement is in the initialization of the control trajectory, in which the initialization held constant values.

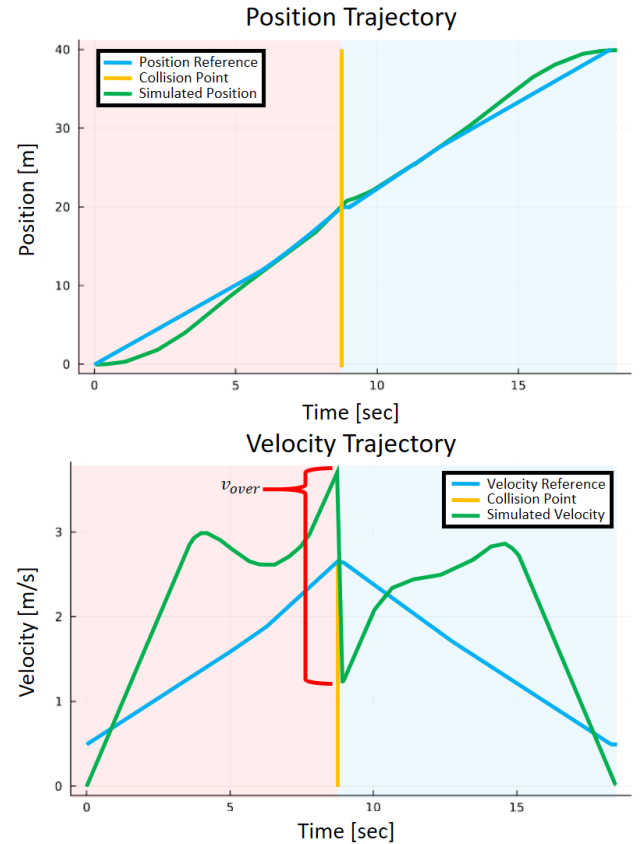


Figure 3: Simulated vegetation override for 31.75 [mm] post. The controller achieves an actual velocity (green) above the required threshold.

4. Vegetation Override Hardware Results

Three hardware results are presented in this section. The first two results use post override models from [Mason et al. \(2012\)](#) and the third result uses the longitudinal override model for a single standing tree described in [Blackmon and Randolph \(1968\)](#). In the experiments, the vehicle was commanded to travel at a nominal speed of 5 [m/s] for the first two experiments and 3 [m/s] in the final experiment. A model predictive controller, similar to the unconstrained version of the work provided in [Jianyu et al. \(2017\)](#), was implemented as a tracking controller. The results provided in Figure 4, Figure 5, and Figure 6 reflect this tracking controller following the trajectory as determined by the trajectory optimization controller.

The reported velocity for each run was calculated using a featured-based Simultaneous Localization and Mapping (SLAM) methodology known as “Super Odometry” ([Zhao et al., 2021](#)). Super Odometry fuses multiple sensing modalities, including LiDAR, Inertial Measurement Unit(s) (IMU), and Global Navigation Satellite System (GNSS) to simultaneously provide a register pointcloud map of the environment and an estimate of the system odometry. An XSens MTI-630 AHRS IMU (“XSens”, one onboard) and Carnegie Robotics Duro GNSS (“Duro”, one onboard) were fused in the SLAM setup to generate a state estimate with a position accuracy greater than 0.4 [m] and a velocity accuracy greater than 0.08 [m/s]. While operations in Experiment 1 and Experiment 2 were in environments where GPS could sufficiently localize the robotic system, in regions with heavier canopy cover such as in Experiment 3, localization from GPS data alone increasingly difficult. However, the large number of unique environmental features (e.g. tree trunks) captured by the onboard LiDAR provided sufficient environmental characterization to localize the robotic system. Initial concerns of matched features existing primarily on the overridden object proved to be unfounded in sufficiently dense forest environments.

4.1. Experiment 1 - Straight Line Test

The first experiment that was conducted on hardware was a straight-line test. The object that the platform collided with was a 31.75 [mm] pine dowel rod that stood 0.914 [m] above the ground and was embedded around 0.305 [m] below the ground. The soil surrounding the dowel rod was compacted by hand. It had not rained for more than a week, and soil conditions were dry, even at a depth of 0.305 [m]. See [Mason et al. \(2012\)](#) for further discussion on the influence of weather conditions on required minimum override velocity. The minimum required override velocity computed using [Mason et al.’s 2012](#) model was 2.7118 [m/s]. The post was placed around 11.70 [m] from the front of the robotic platform. The red arrow in Figure 4 marks the position of the object as seen by the robotic platform’s perception system. In Figure 4, the vehicle began its trajectory at viewpoint zero. GPS had the vehicle localized at the centroid of the blue circle in Figure 4 at the time of collision. The vehicle’s end goal is depicted as a red square.

In order to constrain the platform into overriding the vegetation, keep-out zones were enforced around the vehicle. These zones are all areas shown in bright pink in Figure 4. The gray areas in Figure 4 indicate all the surrounding objects that are above a height limit of 0.5 [m]. The planned vehicle trajectory is shown in green. Platform viewpoints are provided in Figure 4, which are captured along the executed trajectory. A subset of key viewpoints correspond to the numbers located on the obstacle and collision map subfigure in Figure 4. Note that the vegetation in each photo is highlighted with a bounding box for better visibility of the extremities of the vegetation.

The vehicle reached a speed greater than the necessary threshold in order to impact the vegetation. The collision with the pine dowel rod occurred at around 4.5 [m/s]. The registered collision time was earlier than expected, but this is likely due to the fact that the point of time of collision is calculated when the vehicle first makes contact with the vegetation and the vehicle models used in the collision planning did not include the length of the nose of the vehicle. The vehicle suffered no damage during the test, but the post was almost completely failed. After collision, the post remained at an angle, with the end of the post that was suspended in mid-air sitting around 0.20 [m] over the ground surface. The post had displaced some soil, as evident in the post-collision subfigure in Figure 4. While the soils were compacted by the experimenters before the tests, this displaced soil may be evident of a lack of strong compaction at the surface. The post failed roughly 0.271 [m] from its bottom-most point, a little below the ground plane.

4.2. Experiment 2 - Post Override Test

The second experiment used the model from [Mason et al. \(2012\)](#) to represent the post for override. This post is 25.4 [mm] in diameter, stands 0.914 [m] above the ground, and is embedded around 0.305 [m] below the ground. The parameterization of the soil was assumed consistent with the results found in [Mason et al. \(2012\)](#). This yielded an override minimum velocity of 2.4255 [m/s]. The post’s position is marked with indicator “B” in Figure 5. Additionally, the electrical pole in Figure 5 that may be seen in viewpoint zero is indicated with an “A” in the collision map for localization. The vehicle began its trajectory at viewpoint zero. GPS had the vehicle localized at the centroid of the blue circle in Figure 5 at the time of collision.

The vehicle reached the target speed and x-coordinate position at the correct collision time. Again, note the vehicle’s nose is making contact with the pole at the time of the collision. Another thing to note is vehicle did strike the post on the passenger-side of the front bull-bar. This is a near-head-on collision with the post, but the alignment was meant to be towards the center of the bull-bar not the side.

After the collision, the post was completely failed by the platform. Additionally, the post had been slightly pulled out of the ground by around 50 [mm]. Figure 5 contains an image of the failed post. As in Experiment 1, the platform was not harmed.

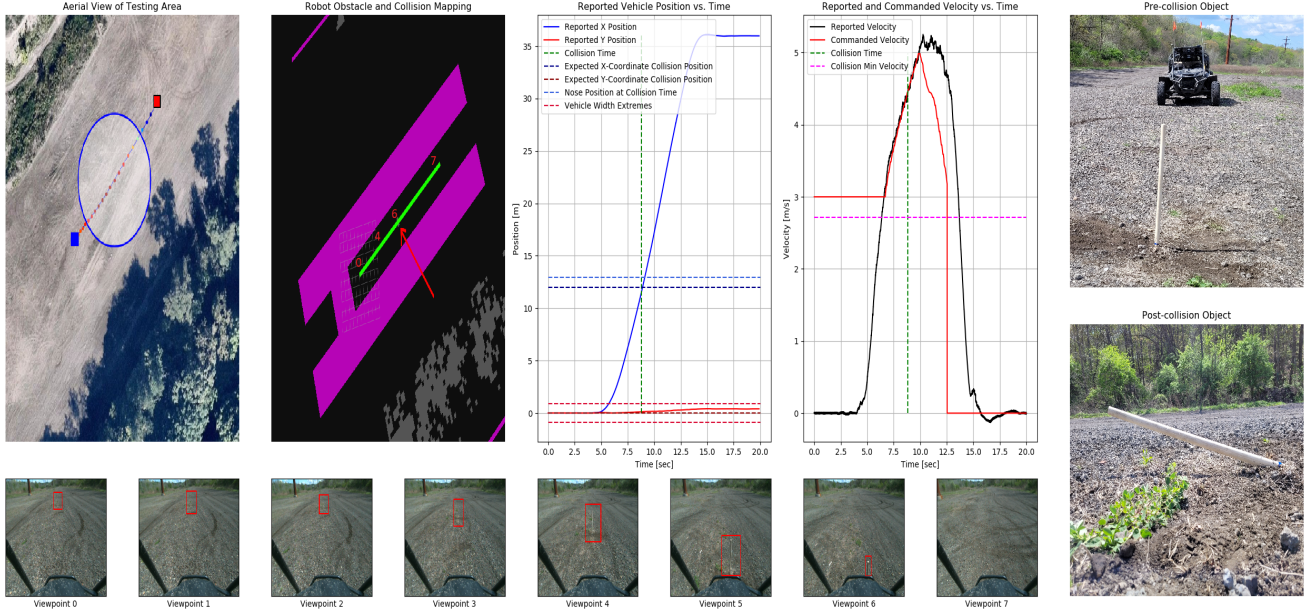


Figure 4: The first hardware experiment: a straight line trajectory through an embedded 31.75 [mm] post.

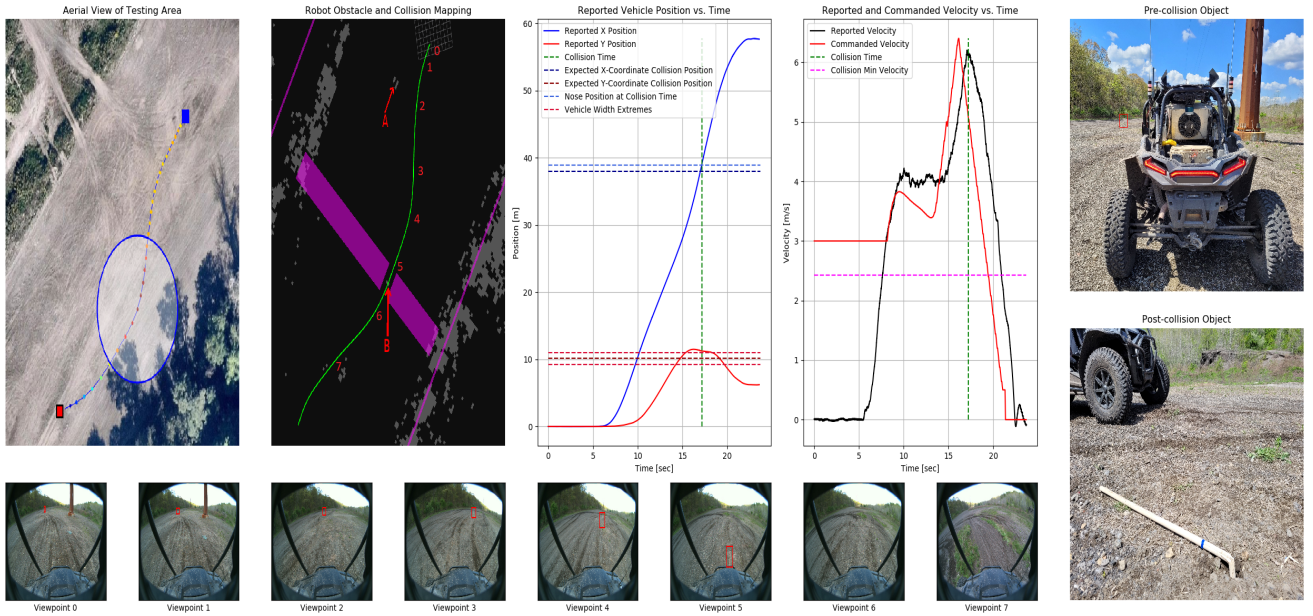


Figure 5: The second hardware experiment: a turning trajectory through an embedded 25.4 [mm] post.

4.3. Experiment 3 - Small Tree Override Test

The final experiment used a simplified model from [Blackmon and Randolph \(1968\)](#) to compute the required override velocity for the small piece of vegetation (tree). In Blackmon's model, the expression for required work needed to fell a single standing tree relies on the diameter of the stem. This information, combined with vehicle inertial information, may then be used to calculate an override velocity in the manner presented in [Mason et al. \(2012\)](#). As the tree does not maintain a uniform radius, the average width of the tree at the point of impact (81.8 [mm]) was used

as an approximation. This measurement yielded a minimum override velocity of 0.758 [m/s]. The tree's position is approximately at the centroid of the yellow box in Figure 6. The vehicle began its trajectory at viewpoint zero and GPS had the vehicle located within the blue circle at the time of collision.

The vehicle reached the target speed and x-coordinate position slightly before the collision time (0.48 [sec]) which was likely due to the poor velocity tracking exhibited by the longitudinal controller at times 6-10 [sec]. At around 12 [sec], the vehicle's nose made contact with the stem of

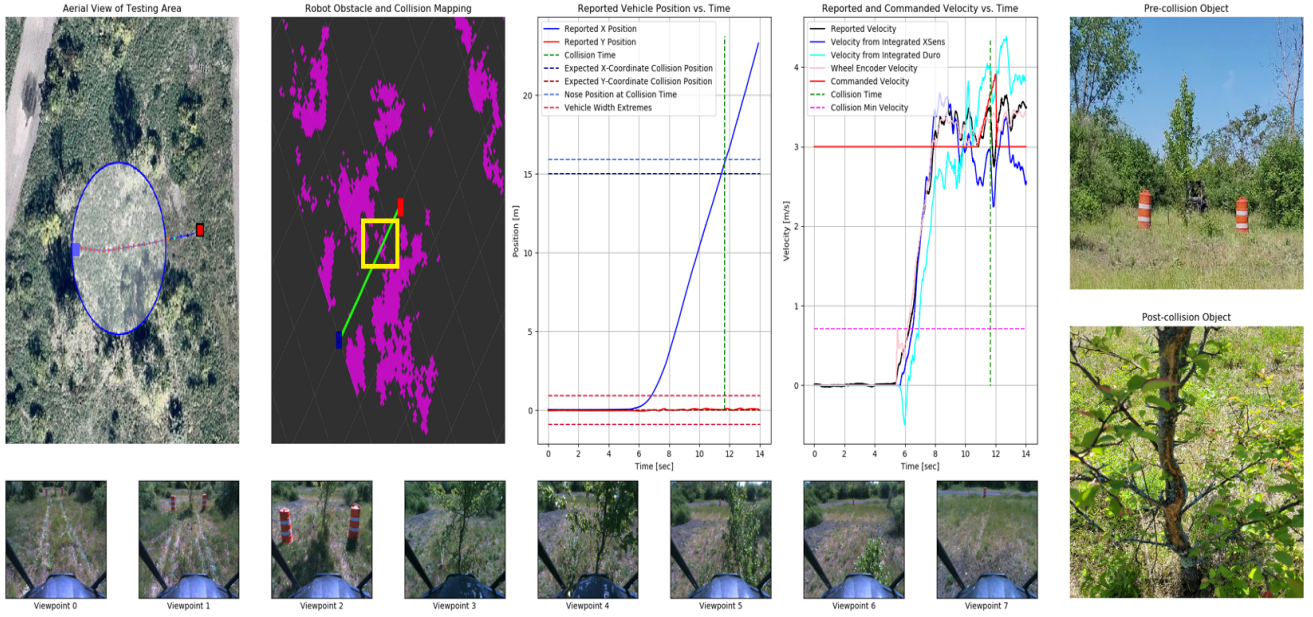


Figure 6: The third hardware experiment: a straight line trajectory through an embedded 81.8 [mm] tree.

the tree, yielding an immediate slowdown that is reflected in the vehicle's reported velocity. The reported velocity loss from Super Odometry was 0.732 [m/s], which is lower than the expected loss of 0.758 [m/s]. Expectantly, integrating the IMU and GNSS output also directly reflects this slowdown in velocity. After colliding with the tree, the vehicle continued to roll until it came to a complete stop.

During the point of collision, the tree was overridden as shown in the point-of-view camera angles in Figure 6. However, after the vehicle traversed over the tree, the root structure of the tree returned it to an upright position. The final resting position and orientation of the tree after collision is shown in Figure 7.

4.4. Discussion

The approach presented in this manuscript utilizes existing low-computation cost collision models to capture the effect of interactions with vegetative objects in the environment and design a trajectory capable of overriding that vegetation. The approach leverages the robotic platform's onboard perception system to estimate the parameters of the vegetation models (e.g., tree diameter), which is then used to fit the collision model. While there is inherent uncertainty in the measurement made by the onboard sensor suite, a conservative approximation (e.g., taking the largest-observed diameter in a time window) is utilized in this work to estimate the parameters of the vegetation model. Such approximations could lead to overly conservative behaviors when overriding vegetation of significant size but allow for the system to model vegetation interactions solely through its onboard sensors. The generated trajectory captured the environmental interaction by modeling the expected loss of velocity experienced by the vehicle due to the collision without requiring expense-to-evaluate simulations. This allowed



Figure 7: A final location and orientation of the tree which participated in Experiment 3.

the vehicle to decide in real-time whether to override post-like environment objects and how to maneuver the wheeled robotic platform through a cluttered off-road environment.

While the hardware results are expected to show losses in kinetic energy due to the collision, surprisingly, no large loss of kinetic energy is experienced during the collision with posts. While the vehicle engine was in use during the impacts presented in this work and the bull-bar has a compliant mount, the loss of energy associated with both experiments was expected to have resulted in a larger drop in

the kinetic energy than was observed. The only experiment that reflects an observable drop in kinetic energy results in an equivalent loss of velocity of 0.732 [m/s] on collision, which is lower than the predicted value of 0.758 [m/s]. This final experiment reflects that a loss of kinetic energy could be experienced during the interaction with the post and that a dynamically-feasible interaction trajectory should account for this loss to more accurately represent the real-world effects of the vegetation override interaction chosen by the robotic system.

5. Conclusions

The off-road operation of robotic systems continues to be a challenging area. It is not only clear from robotic-operated off-road vehicles, but also from human-operated off-road vehicles, that the unstructured nature of the terrain and any associated uncertainty yields a problem that is drastically different from on-road or trail driving. The presented work advances the state-of-the-art in off-road navigation by addressing vegetation interactions through the use of online hybrid dynamic optimization-based vehicle controllers with classical vegetation override models. This understudied vital area of off-road driving has a wide range of applications, and many areas of improvement to bring robotic platform operations more inline with expert human operators.

This work presents a trajectory optimization method that combines hybrid-dynamics, a free-time formulation, and existing parameterized vegetation models from the off-road mobility and cross-country movement literature in order to override environmental vegetation. The method works by enforcing minimum velocity constraints such that the collisions with the vegetation occur at a minimum override speed. These collisions occur at a designer-specified index of collision, but the designer need not select the time of collision, as that is handled directly by the solver.

While the collisions modeled in this work occur at a unique instance in time, future extensions should also look into addressing extended collisions with distributed objects, such as dense foliage. General improvements to the algorithm include changing the vehicle model to capture the frontal geometry, better representations of vegetative objects (the representation of a bush-like object as a singular point such as the center of a tree showed difficulty in determining collision time). Furthermore, exploring the implications of the designer-selected collision index hyperparameter and developing automated approaches to how best to select that index should be made a priority. Work in adaptive gridding or meshing could assist in selecting the collision index.

In particular, we postulate that extending the study presented in this manuscript to directly account for environmental aleatoric and epistemic uncertainty could provide a means for safer off-road behaviors. Of larger concern is that the provided vegetation models are themselves approximations of the vehicle-vegetation interaction, with factors that can be non-observable (e.g. root depth) or difficult to monitor (e.g. soil moisture) in real time. The author's

believe that incorporating information-seeking behaviors to investigate model quality or including a measure of vehicle-vegetation interaction model confidence with respect to previously overridden vegetation should be a priority for the wider community.

Finally, we implore the off-road mobility community to continue to construct and improve on low-computational cost vegetation interaction models and datasets for use by robotic systems. The algorithm presented in this paper takes advantage of low-computational cost representations of vegetation-interactions to perform overrides, but its extensibility to other types of vegetative and commonly encountered non-vegetative objects is dependent on these models. As an alternative to developing explicit vegetation-interaction models utilized in this paper, the implicit models developed in learning-based approaches continue to show powerful advances in off-road vehicle control. However, many of these approaches are limited by availability of high-fidelity off-road interaction data or representative simulated data.

6. Acknowledgements

This research was sponsored by the Defense Advanced Research Projects Agency (DARPA) (#HR001121S0004). The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

Distribution Statement "A" (Approved for Public Release, Distribution Unlimited)

The authors of this publication would like to acknowledge the contributions of Scott Hurley and Jacob Lammott, for their assistance in the field experimentation. Additional thanks to Robert Bittner, Norman Papernick, and Dr. Herman Herman. We would also like to give our thanks to Ryan Arciero for providing his thoughts on vegetation override. This work would not have been possible without the efforts of Team DEAD Fast.

CRedit authorship contribution statement

Charles Noren: Conceptualization, Methodology, Software, Writing. **Bhaskar Vundurthy:** Conceptualization, Investigation, Methodology, Supervision. **Sebastian Scherer:** Conceptualization, Resources. **Matthew Travers:** Conceptualization, Funding acquisition.

References

- Abraham, I., Handa, A., Ratliff, N., Lowrey, K., Murphey, T.D., Fox, D., 2020. Model-based generalization under parameter uncertainty using path integral control. *IEEE Robotics and Automation Letters* 5, 2864–2871.
- Blackmon, C.A., Randolph, D.D., 1968. An analytical model for predicting cross-country vehicle performance, Vol. II: Longitudinal Obstacles. Appendix B: Vehicle Performance in Lateral and Longitudinal Obstacles, Vegetation. Technical Report. U.S. Army Engineering Waterways Experiment Station.

- Bradbury, M., Dasch, J., Gonzales-Sanchez, R., Hodges, H., Iagnemma, K., Jain, A., Jayakumar, P., Letherwood, M., McCullough, M., Priddy, J., Wojtysiak, B., 2018. Next Generation NATO Reference Mobility Model (NRMM) Development. Technical Report. North Atlantic Treaty Organization Science and Technology Organization.
- Castro, M.G., Triest, S., Wang, W., Gregory, J.M., Sanchez, F., Rogers, J.G., Scherer, S., 2023. How does it feel? self-supervised costmap learning for off-road vehicle traversability, in: 2023 IEEE International Conference on Robotics and Automation (ICRA), pp. 931–938. doi:[10.1109/ICRA48891.2023.10160856](https://doi.org/10.1109/ICRA48891.2023.10160856).
- Chen, E., Ho, C., Maulimov, M., Wang, C., Scherer, S., 2024. Learning-on-the-drive: Self-supervised adaptation of visual offroad traversability models. URL: <https://arxiv.org/abs/2306.15226>, arXiv:2306.15226.
- Ellis, K., Zhang, H., Stoyanov, D., Kanoulas, D., 2022. Navigation among movable obstacles with object localization using photorealistic simulation, in: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1711–1716.
- Frey, J., Mattamala, M., Chebrolu, N., Cadena, C., Fallon, M., Hutter, M., 2023. Fast Traversability Estimation for Wild Visual Navigation, in: Proceedings of Robotics: Science and Systems, Daegu, Republic of Korea. doi:[10.15607/RSS.2023.XIX.054](https://doi.org/10.15607/RSS.2023.XIX.054).
- Frey, J., Patel, M., Atha, D., Nubert, J., Fan, D., Agha, A., Padgett, C., Spieler, P., Hutter, M., Khattak, S., 2024. Roadrunner – learning traversability estimation for autonomous off-road driving. URL: <https://arxiv.org/abs/2402.19341>, arXiv:2402.19341.
- Hargraves, C.R., Paris, S.W., 1987. Direct trajectory optimization using nonlinear programming and collocation. Journal of Guidance, Control, and Dynamics 10, 338–342.
- Howell, T., Jackson, B., Mancheseter, Z., 2019. Altro: A fast solver for constrained trajectory optimization. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 7674–7679.
- Howell, T., Tracy, K., Le Cleach, S., Manchester, Z., 2023. Calipso: A differentiable solver for trajectory optimization with conic and complementarity constraints. In: Billard, A., Asfour, T., Khatib, O. (eds) Robotics Research ISRR 2022. Springer Proceedings in Advanced Robotics 27, 1–25.
- Jiang, P., Osteen, P., Wigness, M., Saripalli, S., 2020. Rellis-3d dataset: Data, benchmarks and analysis. arXiv:2011.12954.
- Jianyu, C., Zhan, W., Tomizuka, M., 2017. Constrained iterative lqr for on-road autonomous driving motion planning. IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 1–7.
- Kelly, A., Stentz, A., Amidi, O., Bode, M., Bradley, D., Diaz-Calderon, A., Happold, M., Herman, H., Mandelbaum, R., Pilarski, T., Rander, P., Thayer, S., Vallidis, N., Warner, R., 2006. Towards reliable off road autonomous vehicles operating in challenging environments. The International Journal of Robotics Research 25, 449–483.
- Kelly, M., 2017. An introduction to trajectory optimization: How to do your own direct collocation. SIAM Review 59, 849–904.
- Lavalle, S.M., 2006. Planning Algorithms. University of Illinois, Cambridge University Press. (Online, accessed 05/15/2023).
- Liu, C., Tomizuka, M., 2014. Control in a safe set: Addressing safety in human-robot interactions. Proceedings of the ASME 2014 Dynamic Systems and Control Conference 3, 22–24.
- Mason, G.L., Gates, B.Q., Moore, V.D., 2012. Determining forces required to override obstacles for ground vehicles. Journal of Terramechanics 49, 191–196.
- Moore, M.N., Goodin, C., Salmon, E., Cole, M.P., Jayakumar, P., English, B., 2024. Override forces through clumps of small vegetation. Journal of Terramechanics 116, 100988. URL: <https://www.sciencedirect.com/science/article/pii/S0022489824000302>, doi:<https://doi.org/10.1016/j.jterra.2024.100988>.
- Noren, C., Zhao, W., Liu, C., 2021. Safe adaptation with multiplicative uncertainties using robust safe set algorithm. Proceedings of the 2021 Modeling Estimation and Control Conference (MECC) 54, 360–365.
- Paden, B., Cap, M., Young, S.Z., Yershov, D., Frazzoli, E., 2016. A survey of motion planning and control techniques for self-driving urban vehicles. IEEE Transactions on Intelligent Vehicles 1, 33–55.
- Posa, M., Cantu, C., Tedrake, R., 2013. A direct method for trajectory optimization of rigid bodies through contact. The International Journal of Robotics Research 33, 69–81.
- Prágr, M., Bayer, J., Faigl, J., 2023. Autonomous exploration with online learning of traversable yet visually rigid obstacles. Autonomous Robots doi:[10.1007/s10514-022-10075-4](https://doi.org/10.1007/s10514-022-10075-4).
- Rybansky, M., 2017. Trafficability analysis through vegetation. Proceedings of the International Conference on Military Technologies (ICMT), 207–210.
- Rybansky, M., 2020. Determination the ability of military vehicles to override vegetation. Journal of Terramechanics 91.
- Saxena, D., Likhachev, M., 2023a. Planning for manipulation among movable objects: Deciding which objects go where, in what order, and how. Proceedings of the International Conference on Automated Planning and Scheduling 33, 668–676.
- Saxena, D., Saleem, M.S., Likhachev, M., 2021. Manipulation planning among movable obstacles using physics-based adaptive motion primitives, in: Proceedings of the 2021 International Conference on Robotics and Automation (ICRA), pp. 6570 – 6576.
- Saxena, D.M., Likhachev, M., 2023b. Planning for complex non-prehensile manipulation among movable objects by interleaving multi-agent pathfinding and physics-based simulation, in: 2023 IEEE International Conference on Robotics and Automation (ICRA), pp. 8141–8147. doi:[10.1109/ICRA48891.2023.10161006](https://doi.org/10.1109/ICRA48891.2023.10161006).
- Scholz, J., Jindal, N., Levihn, M., Isbell, C.L., Christensen, H.I., 2016. Navigation among movable obstacles with learned dynamic constraints, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3706–3713.
- Sharma, S., Dabbiru, L., Hannis, T., Mason, G., Carruth, D.W., Doude, M., Goodin, C., Hudson, C., Ozier, S., Ball, J.E., Tang, B., 2022. Cat: Cava traversability dataset for off-road autonomous driving. IEEE Access 10, 24759–24768. doi:[10.1109/ACCESS.2022.3154419](https://doi.org/10.1109/ACCESS.2022.3154419).
- Sivaprakasam, M., Maheshwari, P., Castro, M.G., Triest, S., Nye, M., Willits, S., Saba, A., Wang, W., Scherer, S., 2024. Tartandrive 2.0: More modalities and better infrastructure to further self-supervised learning research in off-road driving tasks. URL: <https://arxiv.org/abs/2402.01913>, arXiv:2402.01913.
- Stilman, M., Kuffner, J., 2004. Navigation among movable obstacles: real-time reasoning in complex environments, in: 4th IEEE/RAS International Conference on Humanoid Robots, 2004., pp. 322–341 Vol. 1. doi:[10.1109/ICHR.2004.1442130](https://doi.org/10.1109/ICHR.2004.1442130).
- Thurn, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.E., Koelen, C., Markey, C., Rummel, C., Niekirk, J., Jensen, E., Alessandrini, P., Bradski, G., Nefian, A., Mahoney, P., 2007. Stanley: The robot that won the darpa grand challenge. Springer Tracts in Advanced Robotics 36, 1–44.
- Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M.N., Dolan, J., Duggins, D., Galatali, T., Geyer, C., Gittleman, M., Harbaugh, S., Herbert, M., Howard, T., Kolski, S., Kelly, A., Likhachev, M., McNaughton, M., Miller, N., Peterson, K., Pilnick, B., Rajikumar, R., Rybski, P., Salesky, B., Seo, Y.W., Singh, S., Snider, J., Stentz, A., Whittaker, W., Wolkowicki, Z., Zigar, J., Bae, H., Brown, T., Demitrish, D., Litkouhi, B., Nickolaou, J., Sadekar, V., Zhang, W., Struble, J., Taylor, M., Darms, M., Ferguson, D., 2009. Autonomous driving in urban environments: Boss and the urban challenge. Springer Tracts in Advanced Robotics 56.
- Vong, T.T., Hass, G.A., Henry, C.L., 1999. NATO Reference Mobility Model (NRMM) Modeling of the DEMO III Experimental Unmanned Ground Vehicle (XUV). Technical Report. Army Research Laboratory, Aberdeen Proving Ground.
- Wilfong, G., 1991. Motion planning in the presence of movable obstacles. Annals of Mathematics and Artificial Intelligence.
- Williams, G., Drews, P., Goldfain, B., Rehg, J.M., Theodorou, E.A., 2018. Information-theoretic model predictive control: Theory and applications to autonomous driving. IEEE Transactions on Robotics 6, 1603–1622.

Zhao, S., Zhang, H., Wang, P., Nogueira, L., Scherer, S., 2021. Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 8729–8736.