

PERFORMANCE ANALYSIS OF CONSTANT SPEED LOCAL OBSTACLE AVOIDANCE CONTROLLER USING AN MPC ALGORITHM ON GRANULAR TERRAIN

Nicholas Haraus

Dept. of Mechanical Engineering
Marquette University
Milwaukee, WI

A Model Predictive Control (MPC) LIDAR-based constant speed local obstacle avoidance algorithm has been implemented on rigid terrain and granular terrain in **Chrono** to examine the robustness of this control method. Provided LIDAR data as well as a target location, a vehicle can route itself around obstacles as it encounters them and arrive at an end goal via an optimal route. Using **Chrono**, a multibody physics API, this controller has been tested on a complex multibody physics HMMWV model representing the plant in this study. A penalty-based DEM approach is used to model contacts on both rigid ground and granular terrain. We draw conclusions regarding the MPC algorithm performance based on its ability to navigate the **Chrono** HMMWV on rigid and granular terrain. A novel simulation framework has been developed to efficiently simulate granular terrain for this application.

1 ACKNOWLEDGEMENTS

s:Acknowledgements

2 TABLE OF CONTENTS

s:TableofContents

Contents

1 ACKNOWLEDGEMENTS	i
2 TABLE OF CONTENTS	ii
3 LIST OF TABLES	iv
4 LIST OF FIGURES	v
5 LIST OF EQUATIONS	vi
6 LIST OF SYMBOLS	vii
7 INTRODUCTION	1
7.1 ProblemSummary	2
7.2 Literature Review	2
7.3 Objectives and Methodology	7
8 THEORETICAL BACKGROUND	7
8.1 Chrono Multibody Physics Package	7
8.1.1 Vehicle Modeling	8
8.2 MPC Based Local Obstacle Avoidance : History and Current Algorithm . .	8
8.3 Multi-DOF Vehicle Dynamics Models	12
8.3.1 2-DOF Vehicle Model	12
8.3.2 14-DOF Vehicle Model	14
8.4 Pacejka Magic Formula Tire Model	15
8.5 Optimization Problem	15
9 Experimental Setup	15
9.1 Evaluation Metrics	15
9.2 Validating Chrono Simulation with Previously Published Results	17
9.3 Simulation Parameters	17
9.4 Rigid Terrain Simulation	18
9.5 Granular Terrain Simulation	18
10 EXPERIMENTAL RESULTS AND DISCUSSION	20
10.1 Influence of model fidelity	20
10.2 Influence of terrain type	23
11 FINAL EXPERIMENT WITH NEW VEHICLE MODEL ON GRANULAR MATERIAL	24
12 CONCLUSIONS	24
12.1 Future Work	25

13 BIBLIOGRAPHY	26
14 APPENDICES	29

3 LIST OF TABLES

s:tables

List of Tables

1	Individual Simulation Test Information	17
2	Obstacle Field Parameters	18
3	Test Evaluation Metrics Summary on Obstacle Field 1	20
4	Test Evaluation Metrics Summary on Obstacle Field 2.	23

4 LIST OF FIGURES

s:figures

List of Figures

1	Schematic of MPC LIDAR-Based Constant Speed Local Obstacle Avoidance Controller	9
2	Sample obstacle field and LIDAR output	10
3	Potential Paths predicted by 2-DOF Internal Vehicle Controller Model	13
4	2-DOF Vehicle Model	14
5	Full-vehicle HMMWV multibody model.	16
6	Obstacle fields.	17
7	Relocating Granular Patch that follows the vehicle	19
8	Test results on obstacle field 1.	21
9	Test results on obstacle field 2.	22

5 LIST OF EQUATIONS

s:equations

6 LIST OF SYMBOLS

s:symbols

7 INTRODUCTION

s:introduction

Obstacle avoidance is a crucial capability for Autonomous Ground Vehicles (AGVs) of the future. This refers to a ground vehicle's ability to sense its surrounding environment, develop an optimal path around the obstacles in the environment, generate optimal control commands to satisfy that path, and physically navigate around the obstacles safely and to a desired endpoint. Safety is defined as avoiding collisions as well as enforcing limitations on excessive sideslip or tire lift-off. An ideal control algorithm is one that is capable of pushing a vehicle to its performance limits by using knowledge of its dynamic capabilities and surrounding environmental conditions, while still enforcing strict safety requirements. Although previous work has demonstrated use of Model Predictive Control (MPC) algorithms for obstacle avoidance on wheeled vehicles, more work is required to test the fidelity of these algorithms and determine where improvements are needed. One area in which MPC algorithms have yet to be tested is their ability to control a wheeled vehicle on granular terrain. Up to this point, the terrain has been assumed to be rigid and flat. Performance of MPC algorithms on deformable terrain raise additional questions. Do the current most commonly used vehicle models perform successfully within an MPC algorithm on deformable terrain? The present work evaluates, through numerical simulation, the robustness and validity of an MPC algorithm with different vehicle models in an environment more similar to what an off-road military vehicle would experience in combat.

The goal of this paper is to present the results of this study to understand how model fidelity of the controller model affects overall performance of the obstacle avoidance controller. Simulating a vehicle on granular terrain in such scenarios introduces additional challenges related to the sheer number of necessary particles required to properly model the desired terrain patch. This challenge has been addressed by employing a simulation framework for granular terrain described later in this paper.

The present study has the following three objectives:

1. Study the performance of the MPC Controller on granular terrain as compared to that on rigid terrain.
2. Analyze the impact of fidelity of the internal controller model on speed and performance of the obstacle avoidance controller.
3. Showcase the potential of controller testing in a high fidelity virtual test environment with **Chrono** to assist with initial control algorithm development before physical implementation for vehicular applications.

The remainder of this paper is organized as follows. In Section [8](#) we provide overviews of the MPC-based local obstacle avoidance algorithm and the **Chrono** multiphysics package used as the test environment. In Section [9](#) we describe the **Chrono::Vehicle** HMMWV vehicle model used in this study as the plant model to be controlled and our proposed method for simulating a vehicle driving on deformable terrain over large distances. Then, the specific MPC LIDAR-based local obstacle avoidance controller used for this study is presented. We introduce the different simplified, lower-fidelity analytical vehicle models used in the MPC algorithm to predict the **Chrono** vehicle behavior. We also provide descriptions of the test

cases considered for this study and a summary of the metrics used to compare performance of the various tested combinations. In Section 10 we present the results of the tests outlined in the previous section and provide comparisons when the internal controller vehicle model is varied and when the terrain is changed from rigid to granular. Section 12 wraps up the paper and presents potential future work based on the results of this study.

7.1 ProblemSummary

ss:ProblemSummary

7.2 Literature Review

ss:LiteratureReview

Many obstacle avoidance algorithms have been developed in the past that allow for smooth, continuous, fast motion of an AGV through an obstacle field. Early algorithms were primarily developed and tested on small ground robots and mainly focused on finding a collision free path. However, they did not necessarily guarantee optimality or the vehicle's ability to achieve that prescribed collision-free path. Some early researched algorithms were developed using artificial potential methods. Applied to a manipulator moving through space, the philosophy of the potential field approach can be summarized as: "The manipulator moves in a field of forces. The position or target to be reached is an attractive pole while all obstacles are repulsive surfaces for the manipulator parts" [1]. Khatib described the formulation and implementation of the artificial potential field concept towards obstacle avoidance. Using the kinematic relationships of the system to be controlled, a Lagrangian formulation is used to develop the equations of motion of the system. An artificial potential field is created that is nonnegative continuous and differentiable whose value goes to infinity as the vehicle approaches an obstacle. The overall artificial potential function is then a sum of the contributing potential fields caused by each of the obstacles within the obstacle field. Each of the obstacles are modeled as a composition of primitives with analytical equations representing their collision envelopes. Khatib successfully implemented this obstacle avoidance algorithm real-time in *Control in Operational Space of a Manipulator-with-Obstacles-System* (COSMOS) with links and moving obstacles [1].

Rimon and Koditschek then presented a technique for constructing artificial potential fields that could bring a bounded-torque actuated robot to a desired configuration without hitting obstacles [2]. Their formulation works for any n-DOF robot whose configuration space happens to be a generalized sphere world. However as with [1], this formulation requires *a priori* knowledge of the topology of the obstacle field. Therefore the implementation of the algorithm at the point of publication of [2] is infeasible for unknown obstacle fields.

Vector Field Histogram methods were then researched due to their ability to to navigate a robot through an unknown environment where topology is not known *a priori*. The Vector Field Histogram (VHF) method allows the detection of unknown obstacles and avoids collisions as it steers a mobile robot to some end destination [3]. The VHF method models the world as a two-dimensional Cartesian histogram grid and continuously updates the grid based on the input on-board sensor data. The algorithm reduces the histogram to a one-dimensional polar histogram around the current robot location such that each sector of the polar histogram has a polar obstacle density associated with it. The algorithm then selects the sector with a low obstacle density and aligns the robot's steering with that sector [3]. The VHF method outlined in [3] successfully navigated a mobile robot through an obstacle

field at an average speed of 0.6-0.7 m/s. This method is a local path planner, so there is no way of it purposefully following a globally optimal path. This algorithm is also susceptible to dead end situations and local minima [3]. [Borenstein&Koren1991](#)

In a much more recent study, Gong and Duan proposed a multi-strategy navigation method to combat some common problems with autonomous vehicle navigation [4]. The two common problems addressed were the vehicle reaching a local minimum and the vehicle navigating into a dead end. Both scenarios result in the vehicle remaining stuck forever with many algorithms. The proposed multistrategy algorithm identifies the current state of the vehicle and determines which navigation strategy is best for the current moment. Vector Polar Histogram is used until the vehicle gets stuck at a dead end or local minimum. The controller then switches modes to a wall-following algorithm, forcing the vehicle to find a wall and follow it until out of the dead end. If the vehicle is stuck in a local minimum, the the controller may switch to a move-towards-goal algorithm forcing the vehicle to progress straight to the goal if not blocked by obstacles for a short time until away from the local minimum. This would also apply to the situation where a vehicle's path to the goal is clear, but the goal is right next to a obstacle and therefore the Vector Polar Histogram prevents that path from being considered as a possibility. This method successfully navigated a vehicle in both simulation and real life at a top speed of 1 m/s [4]. [Gong&Duan2009](#)

Dynamic Window Approaches at their roots are most similar to the MPC algorithm used for the research of this thesis. Fox, Burgard, and Thrun focus specifically on the reactive avoidance of collisions with obstacles by a robot [5]. A dynamic window approach is proposed that deals with constraints imposed by velocity and acceleration limits. Steering commands are computed periodically thus avoiding the complexity of a general motion planning problem. First, they consider velocities that result in circular trajectories. Then, only velocities that can be reached within the next time window are considered, largeley decreasing the search space forming the dynamic window. Admissible velocities are weighed by an objective function. In experiments, this method successfully navigated a B21 mobile robot at a top speed of 0.95 m/s [5]. [Fox1997](#)

Brock and Khatib then proposed a global dynamic window approach to obstacle avoidance that combines path-planning and real-time obstacle avoidance algorithms to generate robot motions to complete a task and still remain safe in an unknown environment [6]. As the robot moves through the environment, it builds an occupancy grid based off sensor input data to represent the connectivity of free space. This allows the robot to learn about its environment without any global *a priori* knowledge of the obstacles. The global dynamic window approach combines the reactive collision avoidance of the dynamic window approach of [5] with a global, local minima-free navigation function *NF1*. An objective function is developed to choose the best path forward. This global dynamic window approach was successfully tested with a synchro-driven mobile robot Noman XR4000 at 1.0 m/s [6]. [Brock&Khatib1999](#)

Kunchev, Jain, Ivancevic, and Finn presented a review and comparison of path planning and obstacle avoidance techniques including those previously mentioned. Each obstacle avoidance method has its own benefits and disadvantages. It seems at this point the issue of local minima is significant for all developed algorithms. All the mentioned techniques need more development to be applied to a car-like vehicle at higher speeds [7]. More recent research aims to take these early collision avoidance algorithms and adapt them towards the navigation of high speed AGV's. The motion of these AGV's is not as simple as the small [Kunchev1999](#)

robots the algorithms were initially developed for in that the AGV cannot instantaneously move in any direction. Ignoring the effects of slip, a car-like AGV will move in its heading direction. Newer research has aimed to build upon or even combine previously mentioned algorithms for the application towards car-like AGV's.

Shimoda, Kuroda, and Iagnemma proposed a potential field based method of navigating an unmanned ground vehicle at high speeds across sloped and rough terrains [8]. With this proposed method, a potential field is generated in a two dimensional trajectory space of the path curvature and longitudinal velocity. The potential field is generated based on dynamic constraints, terrain slopes, obstacle proximities, and the target location. A maneuver is selected within a set of performance bounds based on the local potential field gradient. Each maneuver is mapped to the low level commands necessary to execute that maneuver. This method has successfully been tested at a speed of 7 m/s [8].

Koren and Borenstein performed a study presenting a systematic overview of the downfalls of potential field methods [9]. Four main drawbacks are noted and described. First, a well-documented issue is trap situations due to local minima. Second, even though the controlled robot may be slightly smaller than a certain gap between two obstacles, the combined repulsive force from the two obstacles prevents the robot from passing through gaps it should be able to. Third, the robot motion exhibits oscillatory behavior near obstacles. This same behavior is seen in the fourth noted downfall where if a robot is moving through a narrow-walled passage, the robot will oscillate close to each wall throughout the whole passage again due to the nature of the developed potential field and repulsive forces. For certain applications, potential field methods may be appropriate due to simple, elegant, and quick navigation of the robot through the obstacle field. However, Routh stability criterion presented in [9] prove this method would not be stable as vehicle mass and speed increase. This therefore makes the artificial potential field method unsuitable for the research of this study.. These drawbacks have been experimentally proven and motivated a move by Koren and Borenstein away from potential field methods [9].

To address the issue of optimality, more rigorous approaches have been developed leveraging the Model Predictive Control approach. With this method, control action is obtained by solving a finite horizon open-loop optimal control problem over a receding horizon.

Findheisen and Allgöwer provide an introduction to nonlinear Model Predictive Control [10]. Nonlinear MPC methods are being explored due to the method's ability to incorporate design driven constraints into the robot control. As time goes on, our systems are becoming subject to more safety and environmental constraints making the application of previously mentioned methods difficult. With nonlinear MPC, it is not too difficult to add more system constraints to the optimal control problem solved each time step. Multiple applications and derived nonlinear MPC methods are presented as well in [10]. MPC is therefore a promising approach for obstacle avoidance due to its capability to handle input saturation, system nonlinearities, and design state constraints in a dynamic environment.

Ogren and Leonard present the Dynamic Window Approach for fast and safe obstacle avoidance in unknown environments and recast the approach in a continuous nonlinear framework, drawing many similarities to MPC [11]. A method for generating a navigation function with a single unique minimum is presented. In this study, the control input space is discretized for a computationally tractable version of MPC, and exhaustive search is used to identify the best control input choice. Of all the control input possibilities, there always

exists one such that the robot will be able to stop before hitting any obstacle to insure safety [11]. [Ogren&Leonard2005](#)

Primbs, Nevisti, and Doyle explore Control Lyapunov Functions (CLF) and Receding Horizon Control or MPC approaches to nonlinear control [12]. [Primbs&Nevistic1999](#) Comparisons are made between the two methods and it is noted that they seek to solve the same problem optimally and some properties of each method are complementary to one another. The CLF methods are best interpreted in the context of Hamilton-Jacobi-Bellman equations while MPC relates more closely to a Euler-Lagrange framework. The CLF provides a global optimal solution, but the partial differential equation one must solve is very difficult and computationally infeasible. The MPC approach instead allows for on-line computation of an optimal solution locally, and resolves this problem every time step specified. The problem is only solved over a fixed time horizon. However, it is difficult to apply Lyapunov Stability Theory to this MPC method where numerical techniques are used to solve for optimal solutions. Also, the relation between time horizon length and stability are not necessarily linear as noted by results from experiments [12]. [Primbs&Nevistic1999](#)

Tahirovic and Magnani propose a passivity based MPC control for Robot Navigation through rough terrain [13]. [Tahirovic&Magnani2010](#) This method boasts simplicity of application for all vehicles as long as one can accurately model them. A virtual model of the vehicle is then made using shaped energy. The algorithm quantifies roughness of the terrain and uses this parameter when analyzing the cost of a specified path. The roughness is expressed as the relative height of the terrain locally. Simulations have proven successful navigation of a vehicle through hilly terrain, yet there is no mention of how the terrain data would actually be obtained [13]. [Tahirovic&Magnani2010](#) This passivity based MPC method can be applied on-line, but this would only be feasible if there was some sort of sensor and data analysis algorithm which can sense the local terrain roughness.

Bevan, Gollee, and O'Reilly proposed a new method for trajectory generation for road vehicle obstacle avoidance using convex optimization [14]. [Bevan&Gollee2010](#) The dynamics of a vehicle is a very nonlinear case which on the surface poses a non-convex optimization problem when searching for an optimal path trajectory. This study used their method on a vehicle performing an aggressive double lane change maneuver with the intent of designing a quick emergency obstacle avoidance algorithm for when cars either need to stop immediately or turn to move around an incoming obstacle. Most avoidance systems solely slow the vehicle down to prevent collision with a hazard in the vehicle heading direction. The optimization is performed in three stages in which each stage includes different assumptions to frame the problem as a convex optimization problem. The first stage solves for a trajectory assuming no slip, the second assumes slip but constant speed, and the third uses the values from the previous stages to insert to non-convex expressions and holds those terms constant. This study successfully simulated a general wheeled vehicle performing an optimal double lane change maneuver [14]. [Bevan&Gollee2010](#)

Nanao and Ohtsuka presented a nonlinear MPC algorithm which finds a solutions using continuation/generalized minimal residual (C/GMRES) algorithm to find a solution [15]. [Nanao&Ohtsuka2010](#) The algorithm was tested successfully in simulation and though runtimes are not yet quick enough for real-time implementation, they expect it in the near future of this paper. Tire ground interactions are modeled using the Pacejka Magic Formula. A friction circle is used to quickly determine the maximum friction force able to be generated and this is used to create

an "unavoidable region" in which the vehicle is unable to avoid hitting the obstacle [15]. However, this method relies heavily on accurate modeling of the ground tire interactions.

Gao, Lin, Borrelli, Tseng, and Hrovat presented and tested two different MPC algorithms for obstacle avoidance [16]. The first algorithm is a one level MPC algorithm. The algorithm combines the requirements of obstacle avoidance and the optimal trajectory planning into one step, solving a highly nonlinear optimal control problem within the horizon each time step. The objective function includes a proximity term which increases as the vehicle approaches an obstacles. The second algorithm is a two-level approach. A high-level path planner calculates an optimal trajectory for the vehicle disregarding any obstacle information. A low level algorithm then calculates the optimal control inputs that both route the vehicle around obstacles and maintain the vehicle along the desired trajectory. While both algorithms were implemented in realtime successfully on an icy road performing a double lane change maneuver, the two-level approach was successful at speeds of 55 kph while the one level could only be implemented up to 40 kph. Computational runtimes were compared and show the two-level approach runs much quicker than the one-level approach and shows promise for hierarchical MPC methods in the future [16].

The two-level MPC method was developed further and addressed the issue of the vehicle model predicting infeasible trajectories for the vehicle in the high-level planner [17]. Instead, motion primitives are used to develop a high-level trajectory assembled from trims which are connected by maneuvers. Possible trims are straight, left turn, right turn, drift left, and drift right. Maneuvers connect two trims. The possible maneuvers include straight to left turn, left turn to straight, straight to drift right, etc. This method proved successful in simulation. In experiments, a vehicle was able to successfully avoid obstacles, but no mention is made of the vehicle speed [17].

Abbas, Eklund, and Milman analyzed the possible realtime implementation of a nonlinear MPC algorithm for obstacle avoidance using Simulink and simulating the vehicle as a full Carsim nonlinear multibody vehicle [18]. Offline a trajectory is generated guiding the vehicle from a start point to the desired end location. Then on-line each controller time step a cost function is used to weigh possible trajectories and deviations from the reference trajectory are penalized. A pointwise potential function is used to increase cost as the vehicle gets closer to any obstacle. Simulation results showed the vehicle is capable of navigating around a single obstacle. The cost function is minimized using the gradient descent optimization method. The amount of time to find an optimal path in the simulation varies depending on the scenario the vehicle is in at each time step. A vehicle in close proximity to an obstacle will take longer to pick the best path than a vehicle in free space pointed towards the goal. Warm starting is used during the optimization routine to help quicken the optimization process, but there are still many time step scenarios where the amount of time needed to calculate a best path exceeds the available time the controller has to find that path [18].

A nonlinear MPC algorithm was then proposed for realtime obstacle avoidance that uses the software ACADO to autogenerate C-code to perform the high-level trajectory planning [19]. This increases algorithms speeds to the point where even with a complicated vehicle model, nonlinear tire model, and wheel dynamics considered, this proposed method was able to perform computations quickly enough to control a vehicle realtime at 10 m/s on rigid flat ground [19].

Researchers have proposed a fast motion planner that uses a half-car dynamical model

for a wheeled vehicle [20]. A fast local steering algorithm is developed to increase runtimes and splits into a geometric path planning step and then an optimal time parameterization step. The three control inputs are steering angle and the longitudinal slips of the front and rear tires much like a rally car driver would control. Simulation results support the realtime implementations of this obstacle avoidance controller [20].

Several studies have been performed examining MPC-based techniques for car-like vehicle collision avoidance. A MPC algorithm was developed that controls the braking and steering to better control the vehicle during a double lane change maneuver [21]. This algorithm was tested on a snowy road. In experiments, a controller that controlled both braking and steering performed better than an algorithm controlling steering alone [21].

Gray, Ali, Gao, Borrelli, and Hedrick used a MPC algorithm to maintain driver safety by keeping a vehicle within its lane [22]. The control has an internal four wheel vehicle model as well as a driver model, allowing the controller to predict driver commands and intentions. The controller aims to minimize its own input so that only when the driver is in danger such as departing from the lane will the controller take action. Otherwise, it seeks to maintain safety constraints so that the controller only intervenes when it senses safety constraints may be violated. Simulation results proved successful, but when implemented in real life at a Volvo facility, the only issue encountered was minor breaking of a safety constraint. This however was due to sensor delay and in the future this delay will be accounted for in the algorithm [22].

Beal and Gerdes proposed a solution to the issue of vehicle stabilization control near the limits of handling [23]. Normally, an ESC works by using a linear model of a vehicle to predict driver intent and when the vehicle deviates past a threshold of the allowed deviation between driver intent and vehicle behavior, then the controller activates and stabilizes the vehicle. However, this method does not work when the vehicle is being operated near the stable vehicle limits such as by an experienced driver because the linearized vehicle model is linearized about a conservative point for production level cars. The vehicle behavior near its dynamic limits is nonlinear and does not match the linearized model. Therefore, an MPC based controller was proposed and tested that operates at two levels and quickly enough for realtime implementation. The controller has two objectives. The first is to keep the vehicle inside of the safe-handling envelope and respond appropriately in case the vehicle leaves this safe envelope. The second objective is to allow the controller to track the driver's intended trajectory. Overall the controller was able to successfully achieve these objectives through experiments with Stanford's PI vehicle testbed [23].

7.3 Objectives and Methodology

ss:ObjectivesMethodology

8 THEORETICAL BACKGROUND

s:background

8.1 Chrono Multibody Physics Package

ss:Chrono

The physics modeling and simulation capabilities are provided by the multiphysics open-source package Chrono [24]. The core functionality of Chrono provides support for the modeling, simulation, and visualization of rigid multibody systems, with additional capabilities offered through optional modules. These modules provide support for additional classes of

problems (e.g., finite element analysis and fluid-solid interaction), for modeling and simulation of specialized systems (such as ground vehicles and granular dynamics problems), or provide specialized parallel computing algorithms (multi-core, GPU, and distributed) for large-scale simulations.

8.1.1 Vehicle Modeling

Built as a **Chrono** extension module, **Chrono::Vehicle** ^{ChronoVehicle_paper} is a C++ middleware library focused on the modeling, simulation, and visualization of ground vehicles. **Chrono::Vehicle** provides a collection of templates for various topologies of both wheeled and tracked vehicle subsystems, as well as support for modeling of rigid, flexible, and granular terrain, support for closed-loop and interactive driver models, and run-time and off-line visualization of simulation results.

Modeling of vehicle systems is done in a modular fashion, with a vehicle defined as an assembly of instances of various subsystems (suspension, steering, driveline, etc.). Flexibility in modeling is provided by adopting a template-based design. In **Chrono::Vehicle**, templates are parameterized models that define a particular implementation of a vehicle subsystem. As such, a template defines the basic modeling elements (bodies, joints, force elements), imposes the subsystem topology, prescribes the design parameters, and implements the common functionality for a given type of subsystem (e.g., suspension) particularized to a specific template (e.g., double wishbone). Finally, an instantiation of such a template is obtained by specifying the template parameters (hardpoints, joint directions, inertial properties, contact material properties, etc.) for a concrete vehicle (e.g., the HMMWV front suspension).

For wheeled vehicle systems, templates are provided for the following subsystems: *suspension* (double wishbone, reduced double wishbone using distance constraints, multi-link, solid-axle, McPhearson strut, semi-trailing arm); *steering* (Pitman arm, rack-and-pinion); *driveline* (2WD and 4WD shaft-based using specialized **Chrono** modeling elements, simplified kinematic driveline); *wheel* (simply a carrier for additional mass and inertia appended to the suspension's spindle body); *brake* (simple model using a constant torque modulated by the driver braking input).

In addition, **Chrono::Vehicle** offers a variety of tire models and associated templates, ranging from rigid tires, to empirical and semi-empirical models (such as Pacejka and Fiala), to fully deformable tires modeled with finite elements (using either an Absolute Nodal Coordinate Formulation or a co-rotational formulation). Driver inputs (steering, throttle, and braking) are provided from a *driver* subsystem with available options in **ChronoVehicle** including both open-loop (interactive or data-driven) and closed-loop (e.g., path-following based on PID controllers).

8.2 MPC Based Local Obstacle Avoidance : History and Current Algorithm ^{MPC}

The concept of MPC is to use an internal model of the system one desires to control to predict and optimize future system behavior from the current system state and inputs ^{Allgower&Findeisen} [10]. The system behavior is predicted over some defined finite time horizon and the optimal control sequence over the prediction horizon is output. The control sequence is executed for an execution time smaller than the prediction horizon, and the whole process is repeated.

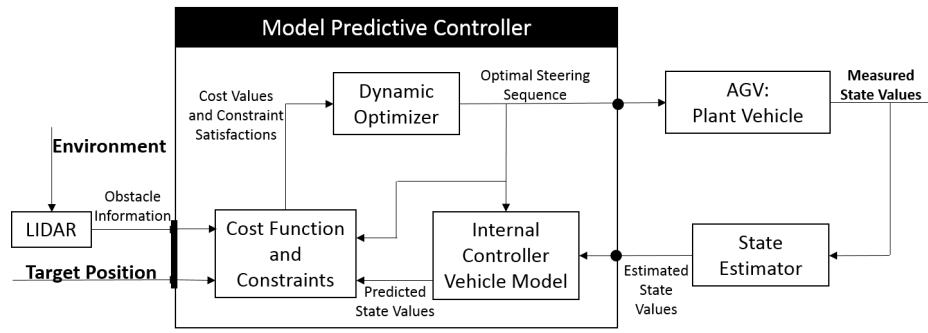


Figure 1: Schematic of MPC LIDAR-Based Constant Speed Local Obstacle Avoidance Controller

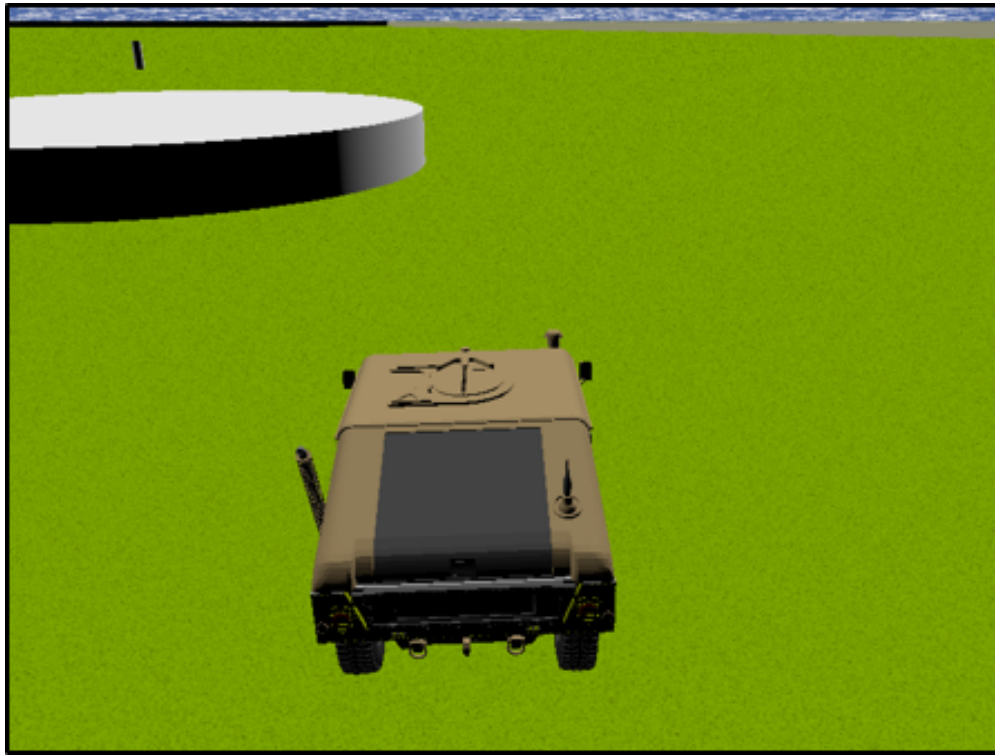
The repetition of this process over time creates a feedback loop which continually controls the system, pushing it towards an optimal path.

For this study, the system to be controlled is an AGV. Consider an AGV located in a level environment without roads or any other structures to guide its motion and assume the AGV has a known global target position. Between the target position and the current vehicle position there may or may not be obstacles of unknown size. Using the MPC formulation outlined in [25], the vehicle can navigate from the current position to the provided target position while avoiding obstacles as they are encountered. Obstacle information is assumed to be unknown a priori and only obtained through a planar LIDAR sensor. The MPC schematic is presented in Fig. 1.

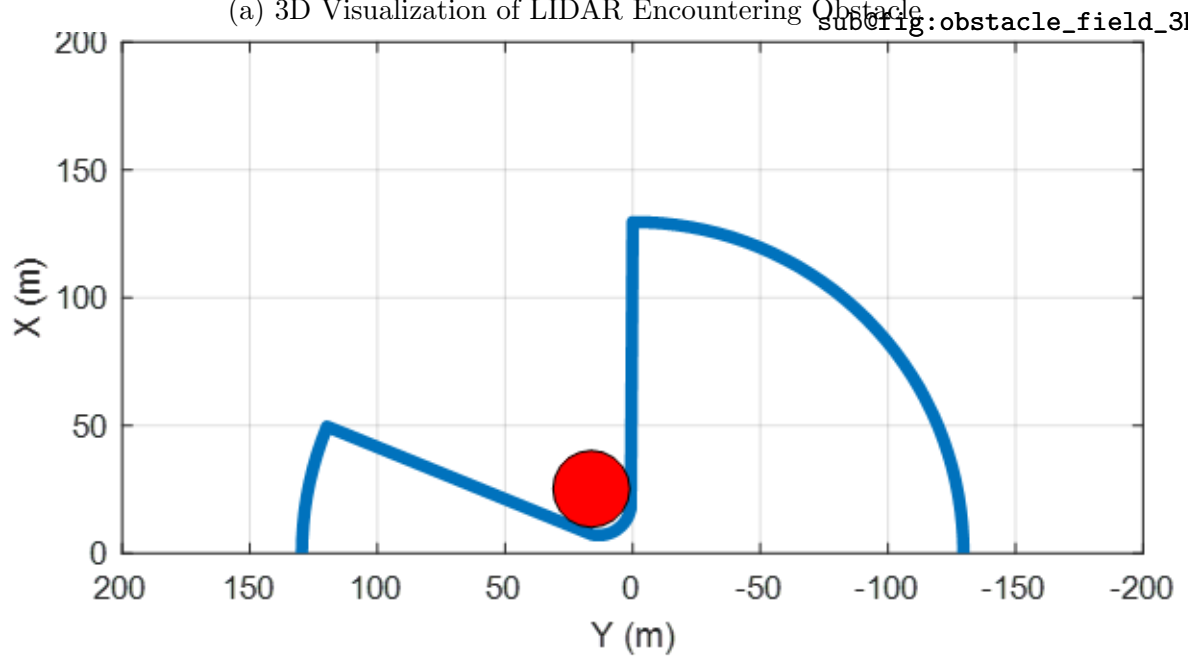
The planar LIDAR sensor, mounted at the front center location of the vehicle, returns the closest obstacle boundary in all radial directions of the sensor at an angular resolution ϵ . The sensor has a maximum range past which it cannot sense any obstacles. Therefore, if the closest obstacle boundary is greater than the LIDAR radius R_{LIDAR} , then the sensor returns R_{LIDAR} . The LIDAR sensor range is $[0^\circ, 180^\circ]$, with 90° being the vehicle heading direction. Since the AGV considered here is driving along level ground, whether granular or rigid, a planar LIDAR sensor is sufficient. The sensor is assumed to have no delay and zero noise and can therefore instantaneously generate a safe area polygon assembled from the returned points from the LIDAR. An overhead view of the AGV encountering an obstacle and the generated LIDAR safe area polygon are presented in Fig. 2.

For the purpose of these simulations, the only outputs of the MPC algorithm are the steering signals. As shown in Fig. 1, the MPC algorithm is made up of the internal controller vehicle model, the cost function and constraints, and the dynamic optimizer. The internal controller vehicle model predicts the future states of the AGV for a given steering sequence. Herein, the internal controller vehicle model is varied from test to test between a 2-DOF vehicle model and a 14-DOF vehicle model, as detailed in further sections. Cost functions and constraints are used to formulate the optimal control problem using the equations from the vehicle model, and the resulting optimal control problem is solved using dynamic optimization.

Since the ability of finding and executing an optimal solution, rather than solution speed, is the primary focus here, an exhaustive search is used to find the optimal solution to the problem at hand. With this method, the steering sequence is discretized and a finite set of path possibilities are tested and weighed by a cost function.



(a) 3D Visualization of LIDAR Encountering Obstacle



(b) LIDAR Sensed Safe Area

Figure 2: Sample obstacle field and LIDAR output

The MPC controller as formulated in [25] is used for this study. The cost function and constraints need to be specified to avoid collisions with obstacles and guarantee vehicle dynamical safety. The optimal control problem solved at each MPC time step consists of the following set of equations:

$$\begin{aligned}
 J &= s_T + wd && \text{e:MPCCost} && (1) \\
 \dot{\xi} &= v [\xi(t), \zeta(t)] && \text{e:State_ODE} && (2) \\
 \xi(0) &= \xi_0 && \text{e:InitialStates} && (3) \\
 \tilde{S}[x(t), y(t)] &\leq 0 && \text{e:SafeArea} && (4) \\
 |\delta_f(t)| &\leq \tilde{\delta}_{f,max}(U_0) && \text{e:SteerLimit} && (5) \\
 |\varsigma_f(t)| &\leq \varsigma_{f,max} && \text{e:SteerRateLimit} && (6) \\
 t &\in [0, T_P] && \text{e:TimeDomain} && (7)
 \end{aligned}$$

Equation (1) defines the cost function for this optimal control problem. This equation is a soft requirement which defines how the separate path possibilities are weighed against each other and how to determine which path is actually optimal. The cost function is comprised of two terms defined as:

$$\begin{aligned}
 s_T &= \sqrt{[x_G - x(T_P)]^2 + [y_G - y(T_P)]^2} && \text{e:DistanceCost} && (8) \\
 d &= \int_0^{T_G} |\varsigma(t)| dt && \text{e:TurningCost} && (9)
 \end{aligned}$$

Here, s_T seeks to minimize the distance between the prediction end-location and the target position. A prediction path that guides the vehicle towards the closest location to the target will have the smaller s_T term. The second term, d , aims to minimize the change in steering angle so that smoother, straighter paths are preferred over windy paths. A weighting factor w is used to scale the influence of d in the total cost.

Equations (2-7) are the constraints for this optimal control problem and represent hard requirements for vehicle safety and collision avoidance. Any paths that violate these constraints are not considered safe paths and are eliminated as potential options in this prediction window. Equation (2) defines a set of differential equations that describe the internal controller vehicle model, with the initial conditions of Eq. (3).

Equation (4) defines a safe area polygon, constructed from LIDAR data and including an additional safety buffer to account for vehicle size and to prevent collisions of the vehicle corners. All points along paths found from Eqs. (2)-(3) must fall inside this safe area polygon.

Equation (5) imposes a maximum limit on the steering angle, based on vehicle speed. This value is obtained from a lookup table which can be generated either experimentally or from simulation. Equation (6) imposes a maximum limit on steering rate. Equation (7) defines the time limits over which this optimal control problem is solved.

As noted above, Eq. (2) refers to an analytical vehicle model expressed as a set of differential equations which allow the controller to predict vehicle performance within the prediction

horizon. The accuracy of the internal vehicle controller model should directly influence the driven vehicle's controlled performance. If the internal vehicle controller model poorly predicts a vehicle's response to inputs from the driver or the environment, then these deficiencies will be witnessed when attempting to control the driven vehicle and result in incorrect trajectories that may lead to collisions. On the other hand, a highly complex internal vehicle controller model may provide accurate trajectory and vehicle response predictions, but with an unacceptable time required to solve the optimal control problem. An ideal controller should not only accurately predict vehicle responses, but also do it quickly enough to insure vehicle safety. In this study we consider two separate internal controller vehicle models to approximate the dynamics of the driven Chrono HMMWV vehicle (described in Section 8.2). The first controller uses a simple 2-DOF vehicle model to predict the trajectory for different series of steering sequences and is presented in detail in Section 8.3.1. The second controller, described in Section 8.3.2, uses a more complex 14-DOF vehicle model. These two models are compared in their ability to successfully navigate the driven HMMWV vehicle through two obstacle fields, on rigid flat terrain and then granular terrain.

As mentioned previously, exhaustive search space [25] is used to solve the optimal control problem at each time step. The steering space is discretized into five possible steering angles, while the prediction horizon is split into four intervals. This results in 625 different steering sequence possibilities and therefore 625 different path possibilities that are weighed by the controller to determine the optimal steering sequence and resulting forward path. A sample visualization of the different predicted trajectories from the 2-DOF model is presented in Fig. 3 where, for visualization purposes, only three steering angles over three intervals were considered. A point in polygon algorithm is used to determine which trajectories remain inside of the safe area polygon visualized in Fig. 2 and therefore should be compared to the other possibilities using Eq. (11).

8.3 Multi-DOF Vehicle Dynamics Models

The vehicle models embedded in the MPC are simplifications of the full, multi-body based, Chrono wheeled vehicle model which is being controlled. We consider two such models, providing different levels of fidelity, as described below.

8.3.1 2-DOF Vehicle Model

The standard vehicle model used in recently developed MPC obstacle avoidance algorithms such as [25] is the 2-DOF yaw plane vehicle model. These models normally either assume constant cornering stiffness or the nonlinear Pacejka Magic Formula Tire Model [?] to predict the ground tire interaction forces. For this study, the Pacejka Magic Formula is used to predict tire forces in the vehicle models.

A visual representation of the 2-DOF yaw plane vehicle model can be found in Fig. 4. The 2-DOF model is described by the following first-order ordinary differential equations:

$$\dot{V} = (F_{y,f} + F_{y,r}) / M - U_0 r \quad (10)$$

$$\dot{r} = (F_{y,f} - F_{y,r}) / I_{zz} \quad (11)$$

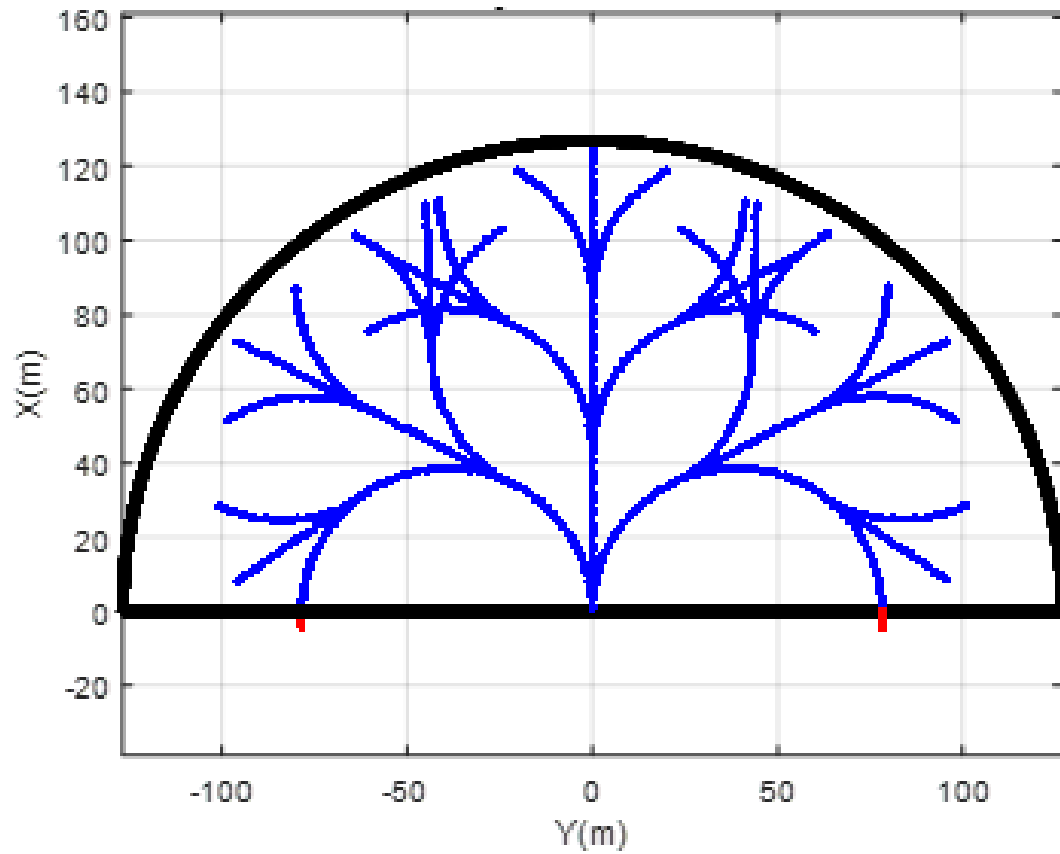


Figure 3: Potential Paths predicted by 2-DOF Internal Vehicle Controller Model

fig:PossiblePaths

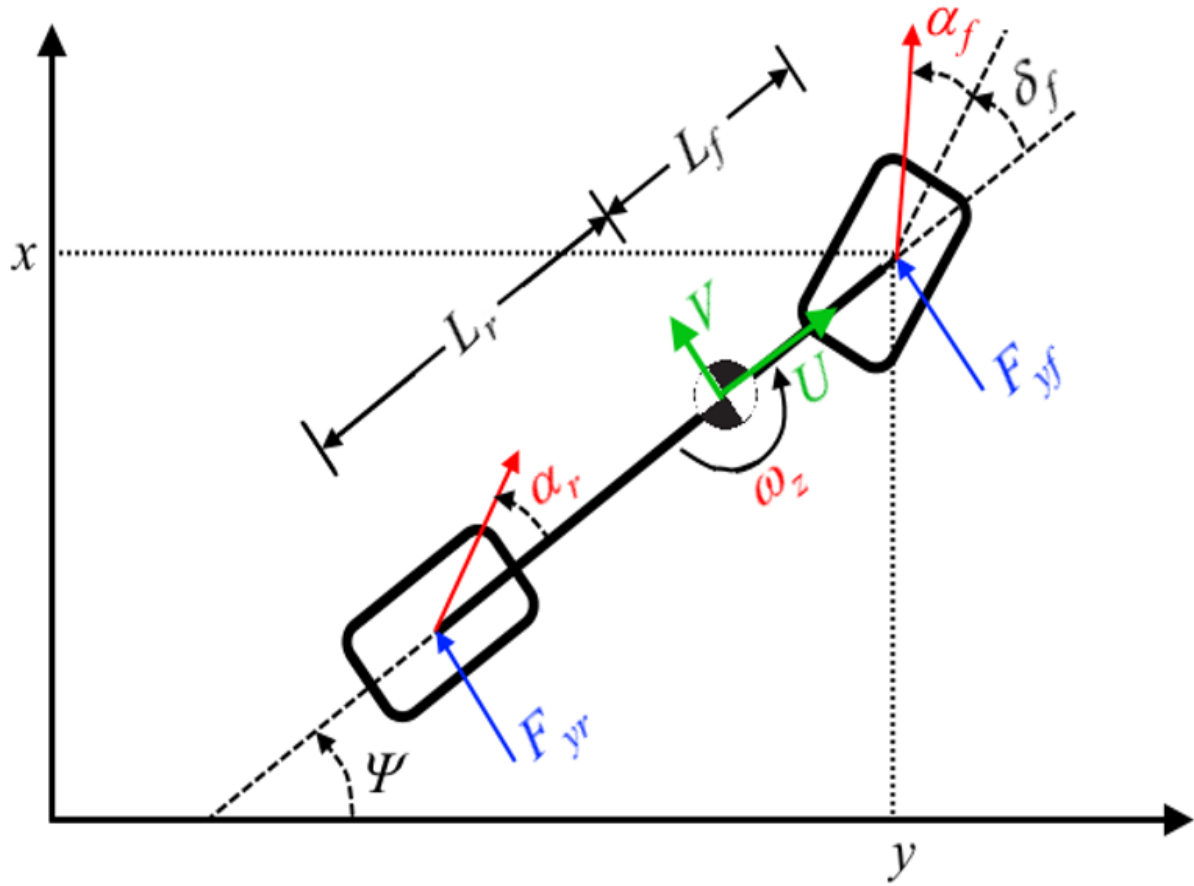


Figure 4: 2-DOF Vehicle Model

fig:2DOF

$$\dot{\psi} = r$$

 e:2DOF_psidot
(12)

$$\dot{x} = U_0 \cos \psi - (V + L_f r) \sin \psi$$

 e:2DOF_xdot
(13)

$$\dot{y} = U_0 \sin \psi + (V + L_f r) \cos \psi,$$

 e:2DOF_ydot
(14)

where $F_{y,f}$ and $F_{y,r}$ are the lateral tire forces at the front and rear axles, respectively. U_0 and V are the longitudinal speed and lateral speed of the vehicle in the vehicle's coordinate frame. ψ is the yaw angle and r is the yaw rate. (x, y) represent the front center location of the vehicle expressed in global coordinates. M is vehicle mass, I_{zz} is the moment of inertia of the vehicle, L_f is the distance from the front axle to the vehicle CoG, and L_r is the distance from the rear axle to the vehicle CoG. For this study, the model is constrained to a constant longitudinal speed. Then the planar 3-DOF body with one constraint results in the 2-DOF model used for this study.

8.3.2 14-DOF Vehicle Model

sss:14DOFModel

A 14-DOF model is often used in studies such as these to test the obstacle avoidance controller with a higher fidelity vehicle model [25, 26]. A benefit of using the 14-DOF in the controller

is the model's ability to predict tire liftoff and account for dynamic effects from suspension systems. For this paper, it is appropriate to also compare performance of the local obstacle avoidance controller running an internal 14-DOF on rigid terrain versus granular terrain.

The 14-DOF vehicle model consists of one sprung mass connected above four unsprung masses [27]. The sprung mass is allowed to roll, pitch, and yaw while also displacing laterally, vertically, and longitudinally. This sprung mass contributes six DOF to the model. Each of the four wheels are allowed to bounce vertically and rotate about the wheel horizontal axis. The front two wheels are also free to steer. Each wheel then contributes two DOF to the fourteen DOF model. The model is constrained at a constant longitudinal speed of 8.1 m/s. The equations used for this model, as well as their derivation can be found in [27].

8.4 Pacejka Magic Formula Tire Model

Pacejka

8.5 Optimization Problem

Optimization

9 Experimental Setup

ExpSetup

The physics modeling and simulation capabilities are provided by the multiphysics open-source package Chrono [24]. The core functionality of Chrono provides support for the modeling, simulation, and visualization of rigid multibody systems, with additional capabilities offered through optional modules. These modules provide support for additional classes of problems (e.g., finite element analysis and fluid-solid interaction), for modeling and simulation of specialized systems (such as ground vehicles and granular dynamics problems), or provide specialized parallel computing algorithms (multi-core, GPU, and distributed) for large-scale simulations.

9.1 Evaluation Metrics

ss:Metrics

Five evaluation metrics will be used to compare one test performance to the other. First, all test runs will be compared based on the time to reach the target, T_{target} to determine which controller leads the vehicle to the target point quickest. Second, the closest distance the vehicle reaches to any obstacle, d_{min} , will be measured. Third, the control effort will be calculated and compared between test cases. The better test case will have a lower controller effort value. Finally, the maximum and average lateral accelerations will be calculated and compared. The accelerations are calculated at the driver's position in the chassis. Using the DEM-P method of simulation results in noisy acceleration data. To address this, acceleration data is filtered to remove noise. These five evaluation metrics provide a consistent methodology for comparing one test case with another, regardless of the terrain type and underlying analytical controller model.

The model to be controlled (the *plant*) is a full-vehicle Chrono::Vehicle model of an HMMWV, which includes multi-body subsystems for the suspensions, steering, driveline, and powertrain, and is available in the Chrono package.

This vehicle model (see Fig. 5) has a curb weight of 2,550 kg. It includes independent front and rear double wishbone suspensions and a Pitman arm steering mechanism. The

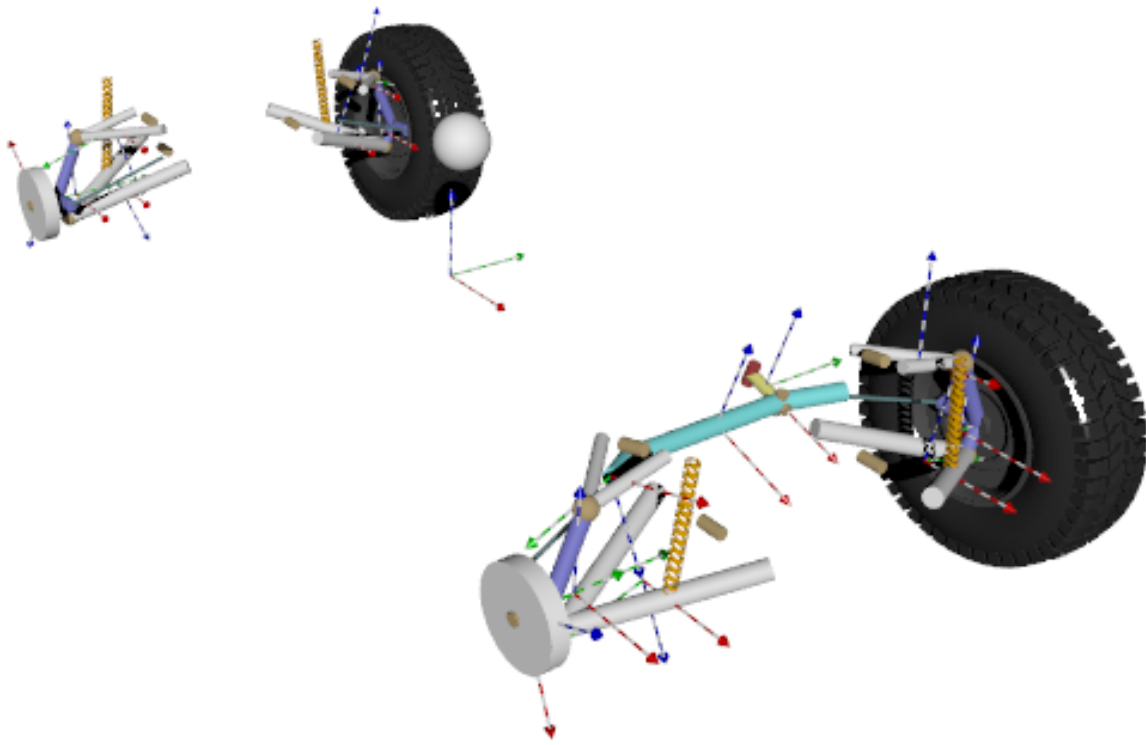


Figure 5: Full-vehicle HMMWV multibody model.

fig:hmmwv

shock absorber and coil spring, mounted between the lower control arm and the chassis, are modeled with **Chrono** nonlinear force elements and include the effects of bump stops.

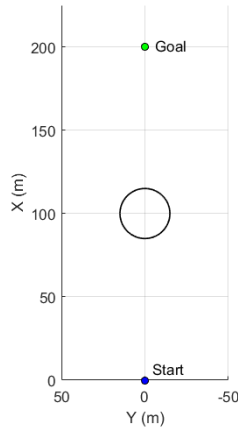
The AWD driveline is modeled using **Chrono** shaft elements and includes three power splitting elements (a central differential and front/rear differentials), as well as conical gears connected through 1-dimensional shaft elements which carry rotational inertia. The powertrain is also modeled using 1-dimensional shaft elements and includes models for a thermal engine specified through speed-torque maps for power and engine losses, a torque converter specified via maps for the capacity factor and the torque ratio, and an automatic transmission gearbox with three forward gears and a single reverse gear. The connection between powertrain and driveline is a force-displacement interface at the driveshaft (with torque applied from the powertrain and angular velocity provided by the driveline).

For mobility studies on deformable terrain, provided that the tire inflation pressure is comparable or larger than the average ground pressure, according to the postulate by Wong [28] the tire can be considered in a so-called *rigid regime*. As such, tire deformation can be ignored and the vehicle's tires modeled using rigid contact shapes. The HMMWV tire model used here is thus represented by a cylinder with a radius of 0.47 m (18.5 in) and a width of 0.254 m (10 in).

Test Number	Terrain Type	Controller Vehicle Model
1	Rigid	2-DOF
2	Rigid	14-DOF
3	Granular	2-DOF
4	Granular	14-DOF

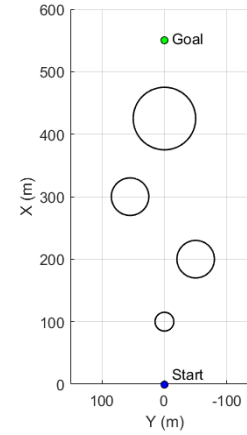
Table 1: Individual Simulation Test Information

t:TestMatrix



(a) Field 1

sub@fig:Obst1



(b) Field 2

sub@fig:Obst2

Figure 6: Obstacle fields.

fig:Obst

9.2 Validating Chrono Simulation with Previously Published Results

Validate

9.3 Simulation Parameters

SimParameters

Simulations on both rigid and granular terrain were compared to understand how the controller performs on non-ideal surfaces. The two internal controller vehicle models studied in these tests were the previously described 2-DOF and 14-DOF vehicle models. All other controller parameters were kept unchanged over all tests. Referring to Table 1, the effect of model fidelity on controller performance was gauged by comparing test 1 with 2 and test 3 with 4, for rigid flat and granular terrain, respectively. Similarly, comparing test 1 with 3 and test 2 and 4 allowed evaluating the performance of each internal vehicle model on rigid and granular terrain.

Each of the above four tests consisted of runs on two fields with different obstacle distributions. The first one, depicted in Fig. 6a, contains a single large circular obstacle located along the initial heading of the vehicle, with the target location at a large distance behind the obstacle. The second obstacle field (see Fig. 6b) consists of four circular obstacles of varying sizes placed in the vehicle's initial heading direction, and provides a more realistic obstacle-avoidance scenario. Table 2 summarizes the obstacle locations and dimensions, as well as the target location for the above two scenarios.

	x (m)	y (m)	Radius (m)
Field 1			
Target Location	200.0	0.0	-
Obstacle 1	100.0	0.0	15.0
Field 2			
Target Location	550.0	0.0	-
Obstacle 1	100.0	0.0	15.0
Obstacle 2	200.0	-50.0	30.0
Obstacle 3	300.0	55.0	30.0
Obstacle 4	425.0	0.0	50.0

Table 2: Obstacle Field Parameters

t:ObstSummary

The following vehicle parameters are maintained throughout all of the executed tests. A PID speed controller is used to maintain a near constant speed of 8.1 m/s longitudinally for the simulated plant **Chrono** wheeled vehicle. This constant speed is also enforced in the analytical models internal to the MPC controller. The LIDAR sensor has a maximum range $R_{LIDAR} = 129.6$ m and is sampled instantaneously at increments of 2.5° . The vehicle is limited to a maximum steering angle of 10° , with a maximum steering rate of $70^\circ/s$.

9.4 Rigid Terrain Simulation

RigidSim

9.5 Granular Terrain Simulation

GranSim

Due to the large-scale nature of the simulations in this study, generating granular particles distributed across the entire obstacle field is both computationally exhausting and unreasonable. Instead, we adopt a moving-patch approach. Consider an AGV on a large terrain patch of 100 by 100 meters. Since we are primarily interested in the vehicle behavior and performance and not explicitly in the terrain behavior, we assume only a small patch of granular terrain underneath and around the vehicle is significant. This idea promoted the development of a relocating granular patch which enables simulations of a vehicle traversing granular terrain over a large area, without the need to generate particles everywhere throughout that area.

This mechanism allows a user to generate granular particles within a specified distance of the vehicle's CoG. Figure 7 presents the patch of granular terrain underneath the **Chrono** HMMWV that moves with the vehicle. The particles are contained within four rigid walls to prevent them from escaping the desired terrain area. As the vehicle moves across the terrain, this patch maintains granular material underneath the vehicle by consistently relocating particles that are too far away from it. When the vehicle location comes within a certain predefined distance of any of the walls, a band of particles from the opposite end of the granular patch are relocated past the closest wall and all of the walls are shifted in that direction. Each of these relocation steps keeps the vehicle on top of granular terrain at all times. Depending on the vehicle, this granular patch can be resized for both computational

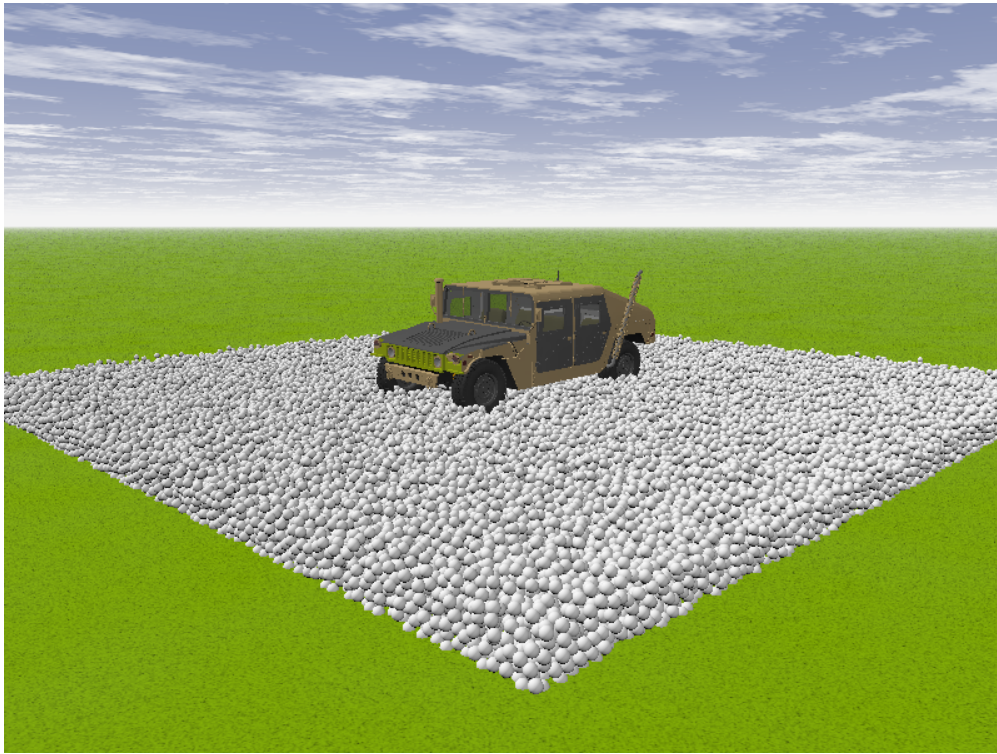


Figure 7: Relocating Granular Patch that follows the vehicle `fig:granPatch`

efficiency and to guarantee the vehicle maintains a reasonable distance from the walls to prevent any boundary effects. This newly developed method of simulating granular terrain for AGV's is omnidirectional in the ground plane, allowing the vehicle to move in any direction while the terrain patch relocates and responds to that movement, thus enabling simulations over arbitrarily large areas.

For our simulations, the granular patch was maintained at a size of roughly 6.6 by 6.6 meters. This number was assigned as two times the largest dimension of the vehicle. The dimensions of the patch are not constant, however, since the patch expands in the direction of relocation by two times the largest particle radius every time the advancing wall is shifted. This is done in order to avoid particle overlap, since the relocated particles are moved to a position that is shifted one largest particle radius ahead of the particles adjacent to the advancing wall. After relocation, the walls of the granular patch are given a recovery velocity such that the granular patch regains its original dimensions in 0.1 seconds. This recovery time should be small relative to the duration of the simulations, and in general it will also depend on the velocity of the vehicle, since the granular patch should completely recover its dimensions between subsequent relocations.

Within the granular patch, the granular terrain was modeled by 55,931 uniform spherical particles, with a micro-scale inter-particle sliding friction coefficient of $\mu = 0.8$, and particle diameter of 0.1 m. Previous studies [29] have shown that a randomly packed assembly of as few as 3,000 - 30,000 uniform spheres will exhibit macro-scale bulk granular material yield behavior (due to inter-particle sliding) that closely matches the Lade-Duncan yield surface, which is a well-established yield criterion in the field of geomechanics, where the

Test Number	1	2	3	4
Controller Model	2-DOF	14-DOF	2-DOF	14-DOF
Terrain	Rigid	Rigid	Granular	Granular
Time to Target (s)	26.67	26.15	28.32	28.03
Minimum Obstacle Distance (m)	0.897	5.462	3.491	4.721
Controller Effort	0.0340	0.0340	0.0340	0.0306
Max. Lateral Acceleration (m/s ²)	2.78	1.57	2.47	2.33
Avg. Lateral Acceleration (m/s ²)	0.54	0.51	0.55	0.46

Table 3: Test Evaluation Metrics Summary on Obstacle Field 1

macro-scale friction angle ϕ for the bulk granular material can be determined as a function of the inter-particle friction coefficient μ .

Referring to Fig. 10 of [29], a randomly packed assembly of uniform spheres with an inter-particle friction coefficient of $\mu = 0.8$ will exhibit macro-scale yield behavior corresponding to a bulk granular material with a macro-scale friction angle between roughly $35^\circ \leq \phi \leq 40^\circ$ if particle rotation is allowed (6 DOF particles), or with a macro-scale friction angle between roughly $65^\circ \leq \phi \leq 70^\circ$ if particle rotation is prohibited (3 DOF particles). Since the granular patch used in our simulations contains more than 50,000 uniform spheres, we can reliably conclude that it is accurately modeling the yield behavior of a true granular material on the macro-scale: either with a macro-scale friction angle of $35^\circ - 40^\circ$ if particle rotation is allowed, which is typical of a wide range of dry natural and crushed sands [30]; or with a macro-scale friction angle of $65^\circ - 70^\circ$ if particle rotation is prohibited, which is typical of crushed or fragmented rock, such as railway track ballast [31].

We note that in the case of the HMMWV, the vehicle tire width is equal to only approximately 2.5 particle diameters in the granular patch. This is less than ideal. However, a more suited value of 10 particle diameters per tire width would increase the number of particles in the granular patch to more than 3 million, which would result in prohibitively slow simulations, particularly for events corresponding to physical time durations on the order of 10 seconds or more.

10 EXPERIMENTAL RESULTS AND DISCUSSION

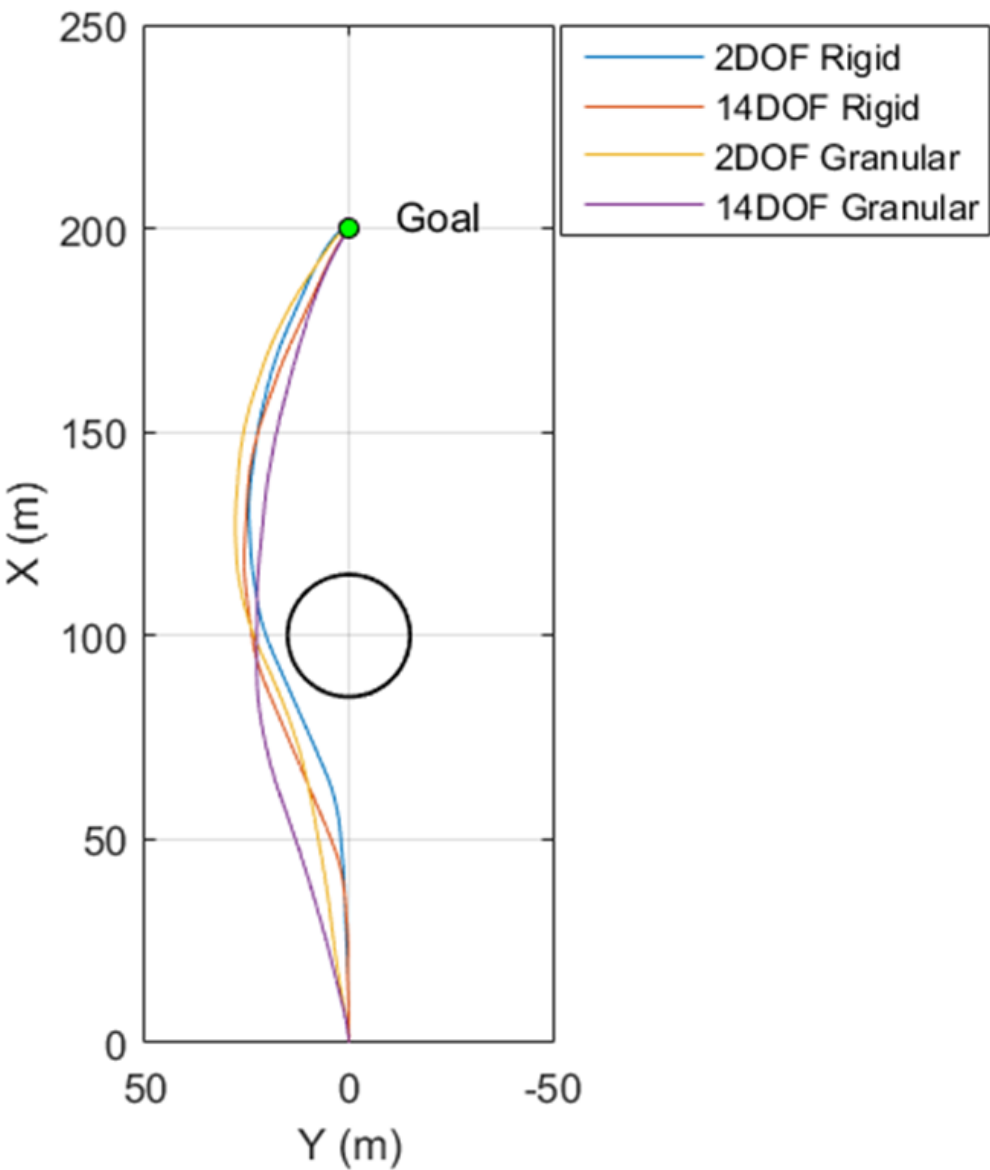
This section summarizes the results of the four tests listed in Table II.

The trajectories for the four performed tests are presented in Figs. 8a and 9a, for obstacle fields 1 and 2, respectively. The associated steering commands generated by the controller are presented in Figs. 8b and 9b.

The evaluation metrics for each test, as defined in Section 9.1, are tabulated for each obstacle field in Tables 3 and 4.

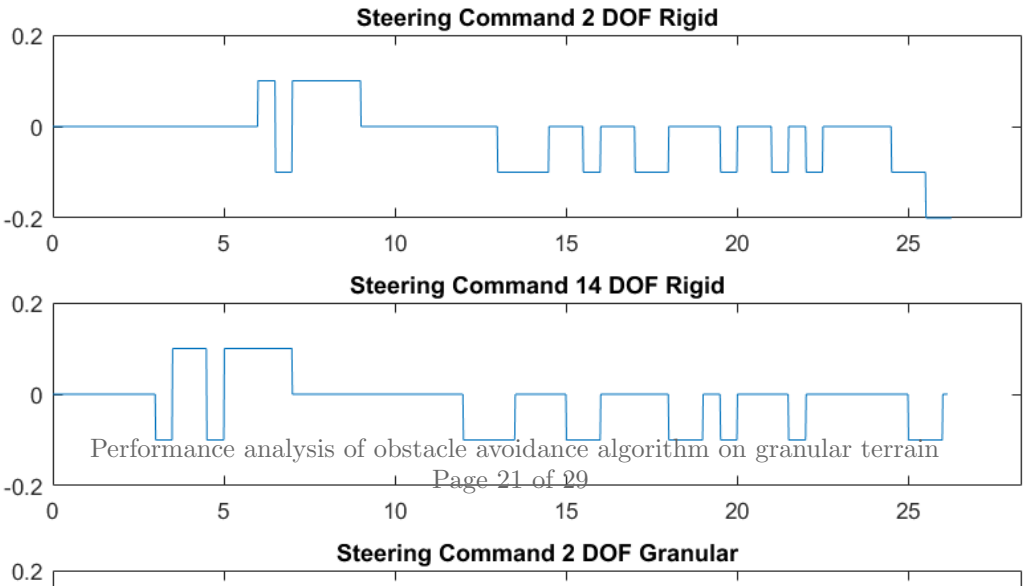
10.1 Influence of model fidelity

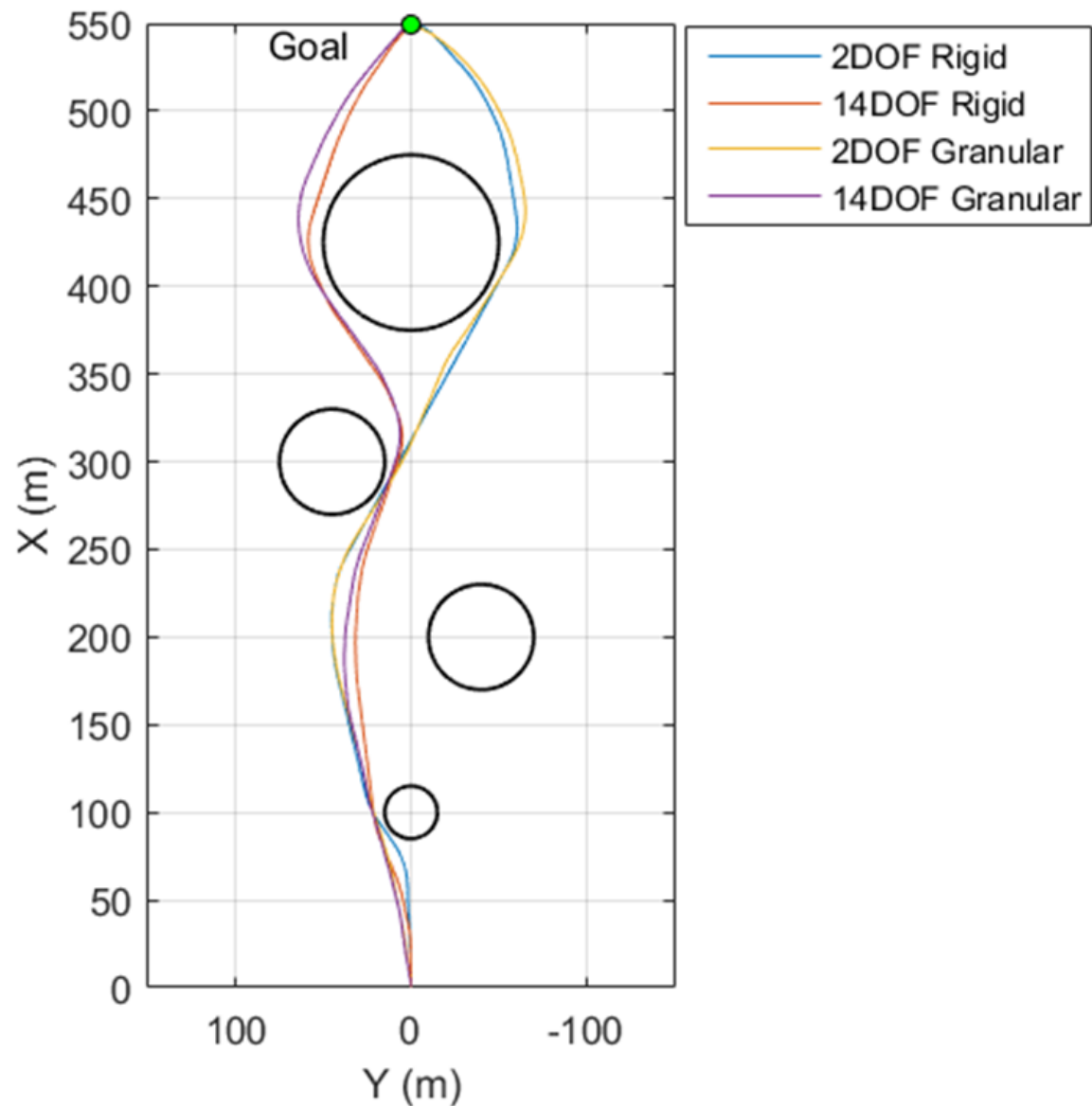
Assessing the effect of internal vehicle model fidelity on controller performance when the vehicle navigates on rigid terrain (comparison of tests 1 and 2), we make the following observations:



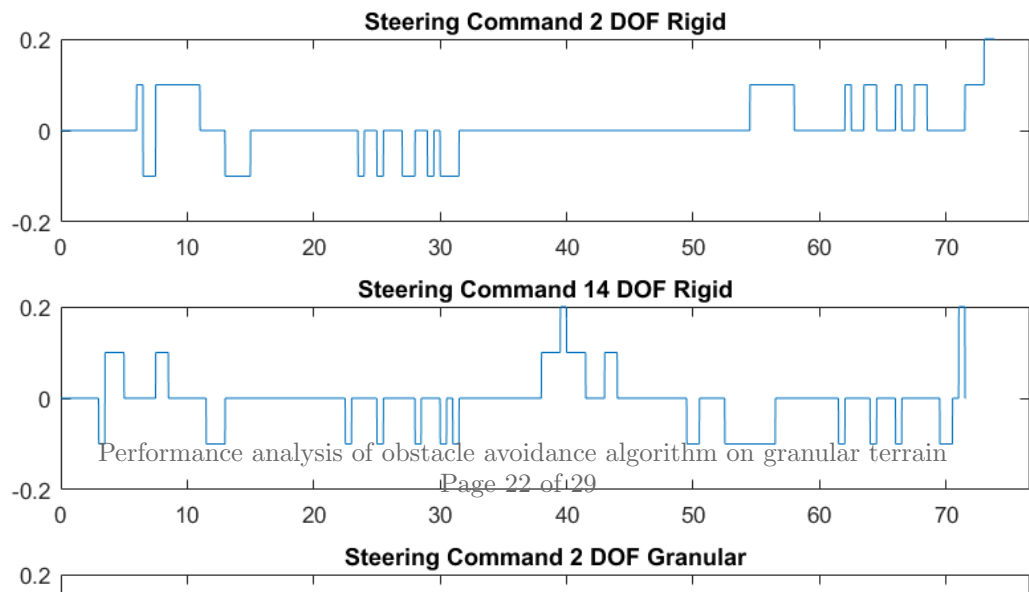
(a) Vehicle trajectories

Subfig:ObstacleField1Trajectories





(a) Vehicle trajectories



Test Number	1	2	3	4
Controller Model	2-DOF	14-DOF	2-DOF	14-DOF
Terrain	Rigid	Rigid	Granular	Granular
Time to Target (s)	73.85	71.55	76.64	74.70
Minimum Obstacle Distance (m)	0.331	2.599	1.083	1.152
Controller Effort	0.0510	0.0680	0.0714	0.0612
Max. Lateral Acceleration (m/s ²)	2.92	2.51	2.55	2.45
Avg. Lateral Acceleration (m/s ²)	0.41	0.43	0.53	0.58

Table 4: Test Evaluation Metrics Summary on Obstacle Field 2

- the 14-DOF model leads to marginally faster travel to the target location;
- the 2-DOF model leads to trajectories with lower obstacle clearance;
- the two models result in the same controller effort on the first obstacle course; however, the higher-fidelity 14-DOF model requires 33% more controller effort on the second obstacle course, a consequence of its decision of performing a course change to negotiate the last obstacle;
- the two models result in approximately the same number of command changes on both obstacle courses;
- the 14-DOF model results in lower maximum lateral accelerations.

Overall, while the higher-fidelity internal model leads to marginally better performance, the 2-DOF model is perfectly suitable for use in MPC-based local obstacle avoidance on rigid terrain at non-extreme speeds, as it can safely navigate the vehicle to its end goal. These results support the findings and conclusions in [25] even though a different AGV is considered here.

A similar comparison can be made for the case of obstacle avoidance on granular terrain (tests 3 and 4). As on rigid terrain, the higher fidelity model leads to faster travel to the end goal, larger clearances to the obstacles, and lower maximum lateral accelerations. However, on granular terrain, the controller effort required by using the 14-DOF model is always lower than that required by the 2-DOF model. We again conclude that using a higher-fidelity internal controller model leads to overall marginally better performance.

The 2-DOF and 14-DOF internal vehicle models were derived using rigid ground assumptions, yet they are still capable of successfully and safely navigating the simulated vehicle through an obstacle field on granular terrain. The monitored metrics indicate a slight drop in controller performance when using the 2-DOF model. However, these gains are outweighed by the benefits, in terms of required computational effort, offered by implementing the lower-fidelity 2-DOF model.

10.2 Influence of terrain type

Turning next to evaluating the 14-DOF internal controller model when navigating on rigid versus granular terrain (comparison of tests 2 and 4), we draw the following conclusions:

- on both obstacle courses, the target can be reached faster when navigating on rigid terrain;
- the resulting trajectories are slightly closer to the obstacles when navigating on granular terrain;
- the required controller effort is lower when navigating on granular terrain;
- the maximum lateral accelerations are similar on both types of terrain.

A similar analysis can be performed on the 2-DOF model when used to control a vehicle on either rigid or granular terrain (comparison of tests 1 and 3). Unlike the 14-DOF model, the simpler 2-DOF model does a better job of approaching the obstacles more closely when controlling a vehicle on rigid rather than on granular terrain. Related to this, the controller effort on rigid terrain (test 1) is lower than that required on granular terrain (test 3).

The forces the vehicle experiences from driving on granular terrain can be much different than forces on simple rigid ground. Examining Figs. 8a and 9a, the vehicle navigating on granular terrain does not turn as sharply for a given steering command as the vehicle does on rigid ground terrain. To understand the different behavior between a vehicle driving on rigid and granular terrain, a parametric study should be performed analyzing the vehicle driving on a variety of granular terrains with different granular parameters.

Tests 3 and 4 were performed on granular terrain modeled as randomly packed uniform spheres with an inter-particle friction coefficient of $\mu = 0.8$, diameter of 0.1m, and a macro-scale friction angle roughly in the range $65^\circ \leq \phi \leq 70^\circ$, with particle rotation prohibited. This granular material resembles railway track ballast. These same tests were attempted on the the same randomly packed uniform spheres with particle rotation allowed resulting in a macro-scale friction angle between $35^\circ \leq \phi \leq 40^\circ$. This granular material resembles a dry sand. The results of this second set of tests on dry sand are not plotted or tabulated because the vehicle failed to move from the initial position. Instead, the vehicle spins its wheels in place at its initial location, making no progress forward as the wheels slowly dig themselves down into the granular terrain. However, this may be due to the too large diameter spheres used for this study for computational reasons. Generally the MPC LIDAR-based constant speed local obstacle avoidance controller used in this study is not appropriate for use on all granular terrains. There are situations where a combined speed and steering controller similar to that described in [?], or even some new speed and steering controller which accounts for terrain parameters and mobility information to better predict vehicle movement, would be required. Neither of the models used in the present study, the rigid ground 2-DOF yaw plane model nor the rigid ground 14-DOF vehicle models, were appropriate for predicting vehicle behavior and performance on dry sand.

11 FINAL EXPERIMENT WITH NEW VEHICLE MODEL ON GRANULAR MATERIAL

s:finalResults

12 CONCLUSIONS

s:conclusion

In this study, using the multibody physics API Chrono a simulation has been developed of a HMMWV driving through a user specified obstacle field towards a defined target location.

Within this simulation, a MPC LIDAR based local obstacle avoidance controller has been implemented to navigate the vehicle around obstacles as it encounters them. The controller itself uses a simplified analytical vehicle model to predict the actual controlled **Chrono** vehicle states within a finite prediction horizon in order to determine the optimal path and steering sequence forward from the current vehicle state. A method of simulating granular terrain for this large obstacle field was developed and implemented for this study. Two separate MPC Algorithms were developed, one that uses a 2-DOF vehicle model to predict **Chrono** vehicle states and a 14-DOF model to do the same. These two controllers were used to navigate a **Chrono** HMMWV through two obstacle courses on both rigid terrain and then on granular terrain. The results were compared to understand what if any improvements need to be made to the MPC LIDAR based local obstacle avoidance controller to successfully control a vehicle on granular terrain.

As in [25], the controller with the 14-DOF internal vehicle model performs marginally better than the controller with the 2-DOF internal vehicle model in all situations. However, tests with the 2-DOF controller prove that this controller can still successfully navigate a vehicle through an obstacle field when the vehicle is moving at non-extreme speeds. Using the 2-DOF controller also allows for faster calculation of optimal steering sequences which is required for eventual realtime implementation. Both of the controllers navigated the vehicle worse on granular terrain than on rigid ground terrain, but this is expected since the internal vehicle controller models were derived using rigid ground assumptions. This study also highlights the complexities introduced to vehicle modeling when the terrain is no longer rigid. Looking at the vehicle trajectories and lateral accelerations on granular terrain, there are clear differences in tire ground interactions when the terrain is now granular. The terrain granular parameters affect the turning characteristics of the vehicle, acceleration abilities, and overall vehicle dynamic performance which is not control predictive with the currently used 2-DOF and 14-DOF analytical vehicle models.

The results of this study are promising. Since the 2-DOF controller is able to navigate the **Chrono** vehicle on this study's granular terrain, then there is some small set of granular terrain on which a 2-DOF model is sufficient for predicting vehicle performance well enough to guide the vehicle successfully and safely to a target point. However, these results also emphasize the need for future research and studies relating to vehicle simulation on granular terrain. This same sort of study should be performed parametrically on a comprehensive set of different granular terrains to understand the difference between vehicle performance on rigid ground and granular terrain. From there, a new intermediate analytical model should be research and developed that is not as complex as the 14-DOF model but does take into account granular parameters such as the friction angle.

12.1 Future Work

FutureWork

13 BIBLIOGRAPHY

s:bibliography

REFERENCES

- [1] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.
- [2] E. Rimon and D. E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, 1992.
- [3] J. Borenstein and Y. Koren. The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Journal of Robotics and Automation*, 7(3):278–288, 1991.
- [4] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *IEEE Intelligent Vehicles Symposium*, pages 417–424, 2009.
- [5] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, pages 23–33, 1997.
- [6] O. Brock and O. Khatib. High-speed navigation using the global dynamic window approach. In *Proceedings of the 1999 IEEE International Conference on Robotics & Automation*, pages 341–346, 1999.
- [7] V. Kunchev, L. Jain, V. Ivancevic, and A. Finn. Path planning and obstacle avoidance for autonomous mobile robots: A review. In *The Knowledge-Based Intelligent Information and Engineering Systems Conference*, pages 537–544, 2006.
- [8] S. Shimoda, Y. Kuroda, and K. Iagnemma. High-speed navigation of unmanned ground vehicles on uneven terrain using potential fields. *Robotica*, 25:409–424, 2007.
- [9] Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 1398–1404, 1991.
- [10] F. Allgower and R. Findeisen. An introduction to nonlinear model predictive control. In *21st Benelux Meeting on Systems and Control*, 2002.
- [11] P. Ogren and N. E. Leonard. A convergent dynamic window approach to obstacle avoidance. *IEEE Transactions on Robotics*, 21:188–195, 2005.
- [12] J. Primbs, V. Nevistic, and J. Doyle. Nonlinear optimal control: A control lyapunov function and receding horizon perspective. *Asian Journal of Control*, 1(1):14–24, 1999.
- [13] A. Tahirovic and G. Magnani. Passivity-based model predictive control for mobile robot navigation planning in rough terrains. 2010.
- [14] G. P. Bevan, H. Gollee, and J. O'Reilly. Trajectory generation for road vehicle obstacle avoidance using convex optimization. *Journal of Automobile Engineering*, 224:455–473, 2010.

- [15] M. Nanao and T. Ohtsuka. Nonlinear model predictive control for vehicle collision avoidance using c/gmres algorithm. 2010.
- [16] Y. Gao, F. Borrelli, T. Lin, E. Tseng, and D. Hrovat. Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads. 2010.
- [17] A. Gray, Y. Gao, J. K. Hedrick, F. Borrelli, T. Lin, and E. Tseng. Predictive control for agile semi-autonomous ground vehicles using motion primitives. pages 4239–4244, 2012.
- [18] M. A. Abbas, J. M. Eklund, and R. Milman. Real-time analysis for nonlinear model predictive control of autonomous vehicles. 2012.
- [19] J. V. Frasch, A. Gray, M. Zanon, H. J. Ferreau, S. Sager, F. Borrelli, and M. Diehl. An auto-generated nonlinear mpc algorithm for real-time obstacle avoidance of ground vehicles. pages 4136–4141, 2013.
- [20] J. H. Jeon, R. V. Cowlagi, S. C. Peters, S. Karaman, E. Frazzoli, P. Tsiotras, and K. Iagnemma. Optimal motion planning with the half-car dynamical model for autonomous high-speed driving. pages 188–193, 2013.
- [21] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat. A model predictive control approach for combined braking and steering in autonomous vehicles. 2007.
- [22] A. Gray, M. Ali, Y. Gao, J. Hedrick, and F. Borrelli. A unified approach to threat assessment and control for automotive active safety. volume 14, pages 1490–1499, 2010.
- [23] C. E. Beal and J. C. Gerdes. Model predictive control for vehicle stabilization at limits of handling. volume 21, pages 1258–1269, 2013.
- [24] A. Tasora, R. Serban, H. Mazhar, A. Pazouki, D. Melanz, J. Fleischmann, M. Taylor, H. Sugiyama, and D. Negrut. Chrono: An open source multi-physics dynamics engine. In T. Kozubek, editor, *High Performance Computing in Science and Engineering – Lecture Notes in Computer Science*, pages 19–49. Springer, 2016.
- [25] J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal. A study on model fidelity for model predictive control-based obstacle avoidance in high-speed autonomous ground vehicles. *Vehicle System Dynamics*, 2016.
- [26] J. Liu, P. Jayakumar, J. L. Stein, T. Ersal, and J. L. Overholt. The role of model fidelity in model predictive control based hazard avoidance in unmanned ground vehicles using lidar sensors. *Dynamic Systems and Control Conference*, 2013.
- [27] T. Shim and C. Ghike. Understanding the limitations of different vehicle models for roll dynamics studies. *Vehicle System Dynamics*, 2007.
- [28] J. Y. Wong. *Theory of ground vehicles*. J. Wiley, New York, N.Y., 2nd edition, 1993.

- [29] J. A. Fleischmann, M. E. Plesha, and W. J. Drugan. Determination of yield surfaces for isotropic non-cohesive particulate materials by the discrete element method. *Geotechnical and Geological Engineering*, 32(4):1081–1100, 2014.
- [30] G.-C. Cho, J. Dodds, and J. C. Santamarina. Particle shape effects on packing density, stiffness, and strength: Natural and crushed sands. *Journal of Geotechnical and Geoenvironmental Engineering*, 132(5):591–602, 2006.
- [31] B. Indraratna, D. Ionescu, and H. D. Christie. Shear behavior of railway ballast based on large-scale triaxial tests. *Journal of Geotechnical and Geoenvironmental Engineering*, 124(5):439–449, 1998.

14 APPENDICES

s:appendices