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

by

Nicholas Haraus, B.S.M.E.

A Thesis submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Master of Science

Milwaukee, Wisconsin

December 2017

ABSTRACT

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

Nicholas Haraus, B.S.M.E.

Marquette University, 2017

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.

ACKNOWLEDGEMENTS

Nicholas Haraus, B.S.M.E.

insert acknowledgements

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

c: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. This controller could be applied to any car-like wheeled vehicle.

The remainder of this paper is organized as follows. The remainder Chapter c:intro ?? consists of the problem statement, a thorough literature review into the problem, and the objectives of this research based on the findings of the literature review. In Chapter c:background 2 consists of theoretical background and overviews pertinent to this research. This chapter consists of sections presenting multibody dynamics, the penalty method, Chrono to handle the multibody dynamics, the MPC Controller development, the LIDAR model used, the Pacejka MagicFormula, and an overview of the exhaustive search algorithm used to solve this control problem. In this chapter are also the different simplified, lower-fidelity analytical vehicle models used in the MPC algorithm to predict the Chrono vehicle behavior for Experiment 1. Chapter c:ExpSetup 3 describes 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. In this chapter the first numerical simulation used to analyze controllerperformance is outlined, detailing the different simulations that will be run, the evaluation metrics that will be used to compare them, and the hardware components of the computer used to perform these simulations. Descriptions of the test cases considered for this first experiment and a summary of the metrics used to compare performance of the various tested combinations are provided. Chapter c:results 4 presents the results of the tests performed in Experiment 1 and provides comparisons when the internal controller vehicle model is varied and when the terrain is changed from rigid to granular. Chapter c:improvedModel 5 presents a new intermediate vehicle model that uses terramechanics relations to better predict vehicle trajectories on granular terrain. This chapter also contains open loop comparisons to confirm this model does in fact predict more accurate granular terrain trajectories. Chapter c:Exp2 6 contains the description and results of a second experiment performed to test the new vehicle model within a controller. Chapter c:conclusion 7 wraps up the paper and presents potential future work based on the results of this study.

1.1 Problem Summary

s:ProblemSummary

Regarding the MPC control algorithm, many studies have been performed previously to validate its potential to develop local obstacle avoidance controllers. Specifically, studies by Liu, Jayakumar, Overholt, Stein, and Ersal explored the implementation of an MPC local obstacle avoidance algorithm in AGVs. The research group tested a steering controller for an AGV on rigid flat terrain ModelFidelity2013[3], then a combined steering and speed controller SpeedSteer2015[4], and finally studied the role of model fidelity in the implementation of the MPC algorithm ModelFidelity2016[5] to understand how detailed the controller model needs to be to effectively and quickly control the vehicle. However, in these studies and those performed before them, the AGV has been tested on rigid flat terrain, an assumption which does not hold when driving a vehicle in reality. Therefore, the research for this thesis will perform studies on the promising models from ModelFidelity2016[5] and test them on granular terrain to understand how changing the ground from rigid and flat to granular affects the controlled AGV performance.

This research also addresses another larger problem for the military. The North Atlantic Treaty Organization (NATO) is seeking to replace their current NATO Reference Mobility Model (NRMM) NATONRMM[6]. The current NRMM is an empirically based computer model developed during the 1970s, which computes tactical mobility metrics by correlating engineering level performance of various ground vehicle systems with different terrain conditions, allowing for successful comparison of these ground systems and capabilities over varied terrain surfaces, obstacles, vegetation, weather scenarios, grades, and other features which can adversely impact ground vehicle performance. NATO seeks to replace this older model with a new physics-based computer simulation solution which can still answer the same questions as the older model, but can then be extended past situations with only empirical results. One of the key areas of work in this group effort is the ability to develop and deliver to the vehicle go-nogo maps of upcoming terrain, either from recent satellite imagery or other sorts of sensors. With this research, the development of this simulation and test will seek to control an AGV only with the sensor data or terrain inputs provided by the mission planner. Any research achieved relating to vehicle mobility across a variety of terrains can help with the NRMM effort.

Significant studies have been carried out and work accomplished testing the MPC Algorithm and validating Chrono as a viable program for Multibody simulations. However, the two topics have not combined to be tested together. The MPC algorithm has been tested on rigid flat terrain, yet with Chrono Parallel one has the ability to simulate granular terrain if desired with reasonable runtimes. Therefore, one can develop a simulation of a vehicle on granular terrain with scattered obstacles in Chrono Parallel, and then use the MPC algorithm from [5] to control that simulated vehicle. Doing so would benefit both the simulations field and the controls field. By bridging the gap between these two areas, those designing the control algorithms have a quicker and less expensive method of testing a control algorithm on a full vehicle on a variety of landscapes to gain quick understanding of how well their controller behaves. On the other hand, developing this sort of simulation improves the amount of applications Chrono Parallel can be used in. This research has performd higher fidelity testing of the control algorithm than has previously been accomplished, a step which is necessary if this algorithm is to eventually be implemented in an actual vehicle.

ModelFidelity2016

The uniqueness of this research comes from two areas: controls and simulation. Most Chrono simulations created so far do not test any sort of complex control algorithm for an autonomous vehicle. More interest is placed in understanding vehicle mobility over different terrains. However, this research does test an autonomous vehicle and its ability to navigate around local sensed obstacles. Therefore, a frmework has been developed to allow the simulated vehicle to sense obstacles in the area and generate its own commands while it navigates across the terrain instead of relying on externally supplied commands. Can this MPC control algorithm generate the appropriate commands to navigate a vehicle through an obstacle field? On the controls side, this is a unique effort as well. As mentioned, the MPC algorithm has yet to be tested on granular terrain. However, considering an AGV will be driving off road on soil or sand, this is an appropriate next step for testing. This research answers the question of if simulation can be used to effectively test an autonomous vehicle on granular terrain. Computationally this requires a lot of resources, but if a team can test their control algorithm on the computer before going out to the test field with a vehicle, this could save both time and money for that team.

1.2 Literature Review

s: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" [7]Khatib1986. 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 [7] Khatib1986.

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 [8] Rimon&Koditschek1992. Their formulation works for any n-DOF robot whose configuration space happens to be a generalized sphere world. However as with [7] Khatib1986, 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 [8] Rimon&Koditschek1992 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 [9] Borenstein&Koren1991. 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 [9] Borenstein&Koren1991. The VHF method outlined in [9] Borenstein&Koren1991 successfully navigated a mobile robot though 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 Borenstein&Koren1991 [9].

In a much more recent study, Gong and Duan proposed a multi-strategy navigation method to combat some common problems with autonomous vehicle navigation Gong&Duan2009 [10]. 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 Gong&Duan2009 [10].

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 Fox1997 [11]. 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 Fox1997 [11].

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 Brock&Khatib1999 [12]. 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 Fox1997 [11] with a global, local minima-free navigation function *NF1*. An objective function is developed to choose the best path forward. This global dynami window approach was successfully tested with a synchro-driven mobile robot Noman XR4000 at 1.0 m/s Brock&Khatib1999 [12].

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 Kunchev1999 [13]. 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 robots the algorithms were intially 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 Shimoda&Kuroda&Iagnemma2007 [14]. 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 in 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 Shimoda&Kuroda&Iagnemma2007 [14].

Koren and Borensteain the performed a study presenting a systematic overview of the downfalls of potential field methods Koren&Borenstein1991 [15]. 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 passafe 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, Rauth stability criterion presented in Koren&Borenstein1991 [15] 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 Koren&Borenstein1 [15].

To address the issue of optimality, more rigorous approaches have been developed leveraging the Model Predictive Cotnrol 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 Allgower&Findeisen2002 [16]. 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 Allgower&Findeisen2002 [16]. 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 Ogren&Leonard2005 [17]. 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 Ogren&Leonard2005 [17].

Primbs, Nevisti, and Doyle explore Control Lyapunov Functions (CLF) and Receding Horizon Control or MPC approaches to nonlinear control Primbs&Nevistic1999 [18]. 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 interpretted 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 Primbs&Nevistic1999 [18].

Tahirovic and Magnani propose a passivity based MPC control for Robot Navigation through rough terrain Tahirovic&Magnani2010 [19]. 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 Tahirovic&Magnani2010 [19]. 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 Bevan&Gollee2010 [20]. 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 Bevan&Gollee2010 [20].

Nanao and Ohtsuka presented a nonlinear MPC algorithm which finds a solutions using continuation/generalized minimal residual (C/GMRES) algorithm to find a solution Nanao&Ohtsuka2010 [21]. 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 Nanao&Ohtsuka2010 [21]. 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 Gao&Borrelli2010 [22]. 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 Gao&Borrelli2010[22].

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 Gray&Gao2012[23]. 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, stright 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 Gray&Gao2012[23].

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 Abbas&Eklund2012[24]. 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 [24] Abbas&Eklund2012.

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 [25] Frasch&Gray2013. 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 [25] Frasch&Gray2013.

Researchers have proposed a fast motion planner that uses a half-car dynamical model for a wheeled vehicle [26] Jeon&Cowlagi2013. 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 fron and rear tires much like a rally car driver would control. Simulation results support the realtime implementations of this obstacle avoidance controller [26] Jeon&Cowlagi2013.

Several studies have been performed examing 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 [27] Falcone2007. This algorithm was tests on a snowy road. In experiments, a controller that controlled both braking and steering performed better than an algorithm controlling steering alone [27] Falcone2007.

Gray, Ali, Gao, Borrelli, and Hedrick used a MPC algorithm to maintain driver safety by keeping a vehicle within its lane [28] Ali&Gray2013. 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 Ali&Gray2013 [28].

Beal and Gerdes proposed a solution to the issue of vehicle stabilization control near the limits of handling Beal&Gerdes2013 [29]. 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 Beal&Gerdes2013 [29].

Ersal, Jayakumar, Lui, and Stein developed a constant speed LIDAR-based local obstacle avoidance controller using an MPC algorithm and performed a study on the role of model fidelity with this controller ModelFidelity2016 [5]. The internal vehicle models used for this study were a 2-DOF and 14-DOF constant speed vehicle models while lateral tire forces were modeled either using the pure slip Pacejka Magic Formula or a linearized version of the PAcejka Magic Formula. All of these tests were performed on rigid ground and a 14-DOF vehicle model was used as the plant vehicle. The vehicle when controlled by the controller with the 14-DOF model with nonlinear tire forces performed marginally better in this study than all other vehicle and tire force model combinations, but runtimes were sacrificed to achieve this boost of performance. However, the 2-DOF models with linear tire forcesdid perform successfully in most tests which led to the conclusion that using this controller would be appropriate doe to the benefits of quicker runtimes. The 14-DOF model can then be used to determine appropriate safety constraints to include in the optimal control problem formulation ModelFidelity2016 [5].

Ersal, Jayakumar, Lui, and Stein alsodeveloped a combined speed and steering local obstacle avoidance controller using an MPC algorithm which is an improvement on the controller from ModelFidelity2016 [5]. A multi-phase optimal control problem was formulated to optimize speed and steering within the prediction window SpeedSteer2015 [4]. A powertrain model is used to determine acceleration capabilities as a function of speed of the vehicle and safety constraints to prevent tire lift-off are imposed in the optimal control problem formulation. The vehicle does not need *a priori* knowledge of the obstacle field and instead develops a safe area polygonfrom LIDAR points. Simulation results from this study show this newer controller is capable of navigating through more obstacle distributions and sharper turns than the constant speed controller. A 3-DOF vehicle model is used as the internal controller model used to predict vehicle states SpeedSteer2015 [4].

A common aspect to all of the previously studied papers and studies is that they were performed on rigid ground. Some sources used height maps to test the vehicle over hilly terrain or used a roughness factor to quantify any deviations from flat terrain. However, most of the previous studies had been primarily focused on the case where a vehicle needs to move to a target location on rigid terrain. Depending on the applications, this sort of testing may be appropriate such as for a car driving on a paved road. For applications such as military, agricultural, off-road sport, and mining this rigid ground assumption does not hold. Therefore a fundamental understanding of the relationship between vehicle performance and terrain characteristics is necessary. The testing performed previously in ModelFidelity2016 [5] should be extended to granular terrain. This presents another two questions. What is the best way to simulate granular terrain and what effect do the granular terrain parameters have on vehicle performance?

The first question depends on the application and the computational resources available for the study. Some methods are quicker than others yet do not completely capture exactly what is happening at the tire ground contact patch while other methods more accurately model that interaction but are computationally exhausting. Generally in terramechanics there are two high fidelity techniques to model soft soils for a vehicle on granular terrain applications: finite element models and particle or mesh-free models.

Finite Element (FE) soil models model the soil by discretizing it into a FE mesh and use some elasto-visco-plastic continuum mechanics contitutive material model to approximate the behavior of the soil such as yielding, flow, cohesion, and internal friction. A review of basic soil constitutive models was compiled in SoilModels2009 [30]. These models aim to approximate the Mohr-Coulomb yield behavior of the soil. Most FE soil studies model the soil deformation using the motion of the FE nodes using a Lagrangian formulation. However, the Lagrangian method requires remeshing when deformations approach moderate to large values which are both computationally expensive and still may not prevent element distortion. This also fails to capture effects of soil deformation such as bulldozing or soil seperation. This is an active area of research where there is even an effort to develop a soil continuum mechanics model which includes cohesion and can properly account for soil flow, plasticity, interparticle friction, cohesion, and the effect of current stress values on the material properties.

The high fidelity alternative to FE soil models is mesh-free particle-based models where instead of treating the terrain as a continuum, the terrain consists of a set of particles with inter-particle forces then representing the soil mechanical behavior. A brief summary of three different particle-based methods is presented in SoilCharacteristics2017 [31]. These methods are successfully able to capture the effects of soil flow, bulldozing, and ruts. Though particle-based methods are computationally expensive as well since typically they can have millions of particles, the benefits of handling bulldozing, seperation, flow, and ruts make it a desirable method. They may even be less expensive than an FE method which is constantly being remeshed. One comman particle-based method is the Disrete Element Method (DEM) where the soil is modeled as a large number of individual bodies which collide and roll along each other to simulate bulk soil material behavior. Normal contact forces generally in this method prevent particles from penetrating one another. Often DEM soil models include tangential contact forces, cohesive forces such as if the soil had some water content, or even distant dependent forces such as gravitational or electrostatic interactive forces.

Jayakumar, Mechergui, and Wasfy performed a simulation study to understand the effects of soil characteristics on vehicle mobility over granular terrain SoilCharacteristics201 [31]. Different cohesive and non-cohesive soils were modeled using DEM. Cone penetrometer simulations were performed to study the effect of the inter-particle friction coefficient and cohesion on soil strength measured using the cone index (CI). A multibody dynamics model of a vehicle was developed consisting of 33 rigid bodies connected with spherical, revolute, prismatic, and constant velocity joints. To simulate the vehicle over granular terrain, a moving soil patch was used which takes particles behind the vehicle, deletes them, and then re-emits them in front of the vehicle and compresses them before the vehicle moves over them again. This patch is implemented in the x-direction. This research group was able to conclude from their results that for non-cohesive or weakly cohesive particles when particle size increases, wheel slip and sinkage decreases, vehicle speed increases,and tractive force slightly decreases. As soil cohesion increases, soil cone index increases resulting in vehicle speed increasing, wheel slip and sinkage decreasing, and tractive force decreasing. For weakly cohesive soil, inter-particle friction and soil density have a big influence on mobility parameters. Mobility parameters seem to level off once inter-particle friction reaches a certain point SoilCharacteristics2017[31].

The performed literature review has assisted with refining the goals and means for achieving the goals of this research. Researchers for decades have been exploring obstacle avoidance techniques to navigate vehicles operating at their dynamic limits. However, the next and also logical step forward from this point is to implement these control algorithms on granular terrain. This task for this research will be accomplished virtually by creating a test environment to perform both rigid ground and granular simulations using Chrono.

1.3 Objectives and Methodology

s:ObjectivesMethodology

The goal of this research is to first understand how model fidelity of the controller model affects overall performance of the obstacle avoidance controller. This leads to a better understanding of vehicle dynamics and performance on granular terrain as well as promotes the development of a controller designed for general terrain travel in the future. From this understanding, the controller and internal models were improved to allow for better controlled vehicle performance on granular terrain. 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 five objectives:

1. Develop a robust simulation environment using Chrono to simulate granular terrain efficiently over large large terrains without actually modeling every single particle over the large terrain.

2. Study the performance of the MPC Controller on granular terrain as compared to that on rigid terrain.

3. Analyze the impact of fidelity of the internal controller model on speed and performance of the obstacle avoidance controller.

4. Improve the controller for better performance on granular terrain.

5. 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.

From previous literature research, this MPC algorithm has been tested successfully on flat rigid terrain. However, using Chrono Parallel one has the capabilities to develop a simulation of a vehicle on hilly, deformable, and even granular terrain. In this research, Chrono Parallel is used to develop a simulation of a AGV driving on customizable granular terrain through a user defined obstacle field to a defined target location. Chrono already possesses the ability to create new geometries easily with a modular framework, so one will then be able to easily create their own obstacle field for the vehicle to maneuver through. Using theory and developments from ModelFidelity2016[5], the MPC algorithm has been implemented in the simulation to determine optimal steering commands for the AGV. Tests similar to those executed in ModelFidelity2016[5] were conducted within this simulation on rigid flat terrain for the two models presented previously to confirm the controller operates appropriately. The algorithm was then tested on granular terrain to understand how controlled vehicle performance changes on granular terrain versus previously tested rigid ground simulations. The MPC algorithm solves a very large optimization problem to determine the quickest safe path to the destination. An exhaustive search based method was used to solve this optimal control problem. Improvements were explored to boost vehicle performance on granular terrain. A second virtual experiment was conducted using the improvements to the controller and those results are discussed at the end of this paper.

2 THEORETICAL BACKGROUND

c:background

2.1 Multibody Dynamics Background

s:MBD

A system of index 3 Differential Algebraic Equations (DAE's) are used to characterize the dynamics of a system composed of rigid and flexible bodies as presented in Chrono2016 [1].

e:LinTrans

$$\dot{\boldsymbol{q}} = \boldsymbol{L}\left(\boldsymbol{q}\right)\boldsymbol{v} \tag{2.1}$$

e:ForceBalance

$$\boldsymbol{M}\left(\boldsymbol{q}\right)\dot{\boldsymbol{v}} = \boldsymbol{f}\left(t, \boldsymbol{q}, \boldsymbol{v}\right) - \boldsymbol{g}_{\boldsymbol{q}}^{\boldsymbol{T}}\left(\boldsymbol{q}, t\right)\hat{\lambda} + \sum_{i \in \boldsymbol{\mathcal{A}}(\boldsymbol{q},\delta)} \left(\hat{\gamma}_{i,n}\boldsymbol{D}_{i,n} + \hat{\gamma}_{i,u}\boldsymbol{D}_{i,u} + \hat{\gamma}_{i,w}\boldsymbol{D}_{i,w}\right) \tag{2.2}$$

e:Constraints

$$\boldsymbol{0} = \boldsymbol{g}\left(\boldsymbol{q}, t\right) \tag{2.3}$$

e:Inequalities

$$i \in \mathcal{A}\left(\boldsymbol{q}\left(t\right)\right) : \begin{cases} 0 \leq \boldsymbol{\Phi}_i\left(\boldsymbol{q}\right) \perp \hat{\gamma}_{i,n} \geq 0 \\ \left(\hat{\gamma}_{i,u}, \hat{\gamma}_{i,w}\right) = \underset{\sqrt{\bar{\gamma}_{i,u}^2 + \bar{\gamma}_{i,w}^2} \leq \mu_i \hat{\gamma}_{i,n}}{\operatorname{argmin}} \boldsymbol{v}^T \cdot \left(\bar{\gamma}_{i,u}\boldsymbol{D}_{i,u} + \bar{\gamma}_{i,w}\boldsymbol{D}_{i,w}\right) \end{cases} \tag{2.4}$$

Equation e:LinTrans 2.1 relates the time derivative of the generalized coordinates $\boldsymbol{q}$ to the velocities $\boldsymbol{v}$ using a linear transformation defined by $L\left(\boldsymbol{q}\right)$. Equation e:ForceBalance 2.2 relates the inertial forces of the system to the applied forces $\boldsymbol{f}\left(t, \boldsymbol{q}, \boldsymbol{v}\right)$ and constraint forces $-\boldsymbol{g}_{\boldsymbol{q}}^{\boldsymbol{T}}\left(\boldsymbol{q}, t\right)\hat{\lambda}$. Equation e:Constraints 2.3 is a set of bilateral constraints imposed on the system that enforce the kinematic behavior of all the bodies in the system and restrict their relative motion. The bilateral constraints of Equation e:Constraints 2.3 can then be augmented by unilateral constraints due to contact/impact phenomena. This point allows the concept of equations of motion to be extended to include differential inclusions filippov88 [32]. A simple example that outlines this concept is of a single body that interacts with the ground through both friction and contact and the equations of motion become an inclusion $\boldsymbol{M}\ddot{\boldsymbol{q}} - \boldsymbol{f} \in \boldsymbol{F}\left(\boldsymbol{q}, t\right)$. In this case $\boldsymbol{M}$ is the inertia matrix of the body, $\ddot{\boldsymbol{q}}$

is the body acceleration, $\boldsymbol{f}$ is the external force, and $\boldsymbol{F}(\boldsymbol{q}, t)$ is some set-valued function. This inclusion then states that the frictional contact force remains somewhere inside of a friction cone with a not yet determined value controlled by the stick or slip state of the interaction between the body and the ground. The red terms in Equations e:LinTrans 2.1-e:Inequalities 2.4 are introduced to the commonly known equations represented in black when in multibody dynamics the differential inclusion is posed as a differential variational inequality problem. The unilateral constraints then presented in Equation e:Inequalities 2.4 define a set of contact complementarity conditions $0 \leq \boldsymbol{\Phi}_i(\boldsymbol{q}) \perp \hat{\gamma}_{i,n} \geq 0$. These conditions make the following simple point regarding contact forces. For a potential contact i in the active set, $i \in \mathcal{A}(\boldsymbol{q}(t))$, either the gap $\boldsymbol{\Phi}_i$ between two geometries is zero and therefore the normal contact force $\hat{\gamma}_{i,n}$ is greater than 0, or $\boldsymbol{\Phi}_i$ is greater than 0 meaning two geometries are not in contact and therefore $\hat{\gamma}_{i,n}$ is 0. Equation e:Inequalities 2.4 then poses an optimization problem whose first order Karush-Kuhn-Tucker optimality conditions are equivalent to the Coulomb dry friction contact model st00 [33]. Therefore in Equation e:ForceBalance 2.2 the force associated with contact i leads to the red generalized forces obtained by using projectors $\boldsymbol{D}_{i,n}$, $\boldsymbol{D}_{i,u}$, and $\boldsymbol{D}_{i,w}$.

2.2 Dealing with Contacts: The Penalty Method

s:Penalty

The method of modeling described above is implemented in Chrono to analyze the dynamics of large multibody systems and granular media in a Discrete Element Method (DEM) framework. This method uses the complementarity conditions in Equation e:Inequalities 2.4 to enforce non-penetration of bodies that come into contact. This method is then referred to as DEM-C. This research however does not use the DEM-C method, but instead employs a Penalty (P) method that accounts for partial body deformations in mutual contact to calculate contact forces and thus this method is referred to as DEM-P. Therefore the Penalty Method is used to calculate contact forces one body exerts on another within the DEM simulation framework in Chrono chosen for this research Chrono2016 [1]. With this DEM-P approach or soft-body approach, bodies are considered ”soft” in that their collision geometries are able to overlap each timestep or experience local deformation. A collision detection algorithm is used to find when and where these overlaps δ_n occur. Contact force vectors $\boldsymbol{F}_n$ and $\boldsymbol{F}_t$ normal and tangential to the contact plane respectively at the point of contact are calculated using a chosen constitutive law based on the local body deformations at the point of mutual contact. In the contact-normal direction $\boldsymbol{n}$, this local body deformation is defined as the overlap of the two quasi-rigid bodies $\boldsymbol{u}_n = \delta_n \boldsymbol{n}$. Then in the contact-tangential direction, the deformation is defined by the vector $\boldsymbol{u}_t$ that keeps track of the total tangential displacement of the intial contact points on the two quasi-rigid bodies while in contact, projected onto the current contact plane.

An example of a DEM-P contact constitutive law similar to one implemented in Chrono is a viscoelastic model based on either Hookean or Hertzian contact theory outlined in Equations e:normalContact e:tangentContact 2.5-2.6. Here $\boldsymbol{u} = \boldsymbol{u}_n + \boldsymbol{u}_t$ is the overlap or local contact displacement of two interacting bodies. The quantities $\bar{m} = m_i m_j / (m_i + m_j)$ and $\bar{R} = R_i R_j / (R_i + R_j)$ are the effective mass and effective radius of curvature respectively for bodies i and j in contact. The vectors $\boldsymbol{v}_n$ and $\boldsymbol{v}_t$ are the normal and tangential components respectively of the relative velocity at the contact point. For a Hookean contact model, $f\left(\bar{R}, \delta_n\right) = 1$ while for a Hertzian contact model, $f\left(\bar{R}, \delta_n\right) = sqrt\bar{R}\delta_n$ silbert2001granular,zhang2005jamming,MachadoMoreiraFloresLankarani2012 [34, 35, 36]. The normal and tangential stiffnesses and damping coefficients k_n, k_t, γ_n, and γ_t are obtained through various constitutive laws derived from contact mechanics and physically measurable properties of materials of the contacting bodies such as Young's Modulus, Poisson's ratio, coefficient of restitiution, etc.

e:normalContact

$$\boldsymbol{F}_n = f\left(\bar{R}, \delta_n\right)\left(k_n \boldsymbol{u}_n - \gamma_n \bar{m} \boldsymbol{v}_n\right) \tag{2.5}$$

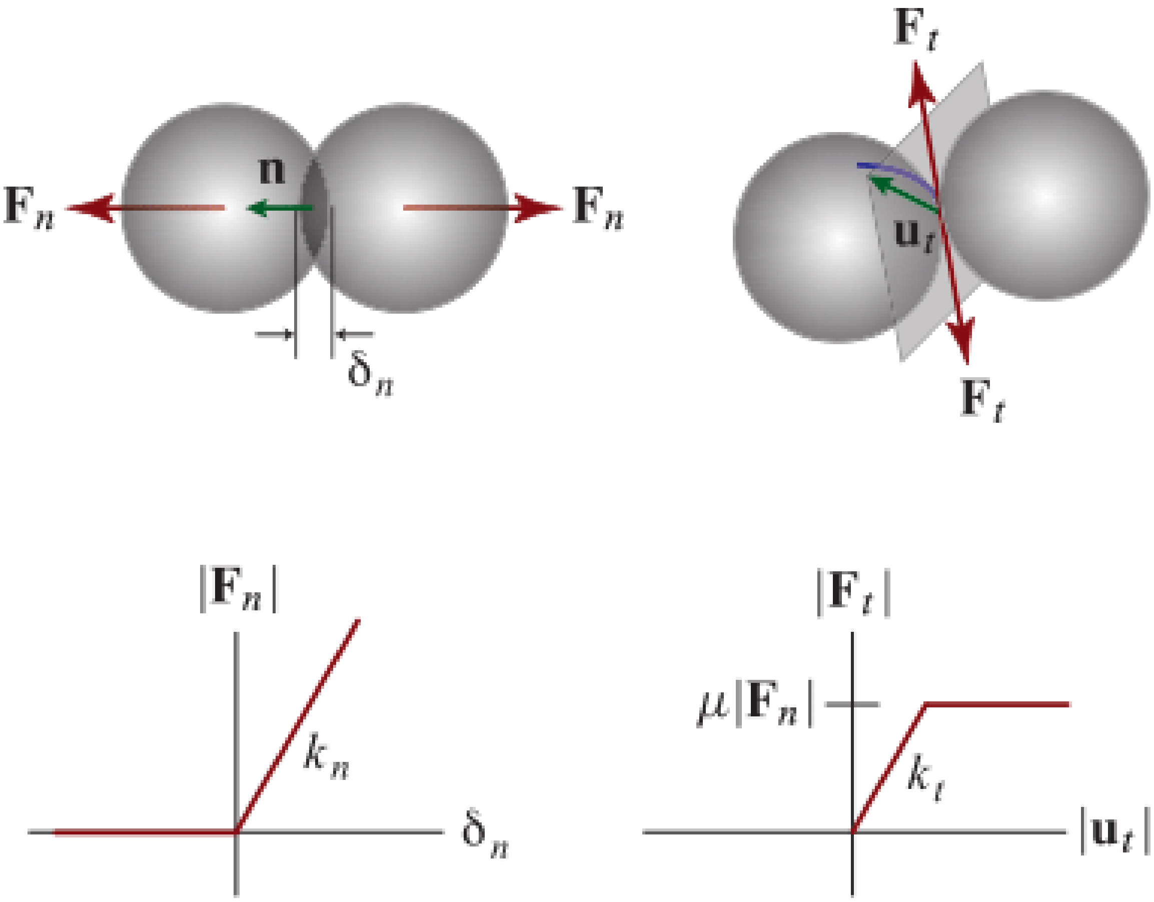


Figure 2.1: Reproduced from Chrono2016 [1]. DEM-P contact model with normal overlap distance δ_n, contact-normal unit vector $\boldsymbol{n}$, and tangential displacement vector $\boldsymbol{u}_t$ (top), and a Hookean-linear contact force-displacement model with constant Coulomb sliding friction (bottom). fig:Penalty

e:tangentContact

$$\boldsymbol{F}_t = f\left(\bar{R}, \delta_n\right)\left(-k_t \boldsymbol{u}_t - \gamma_t \bar{m} \boldsymbol{v}_t\right) \tag{2.6}$$

The component of the contact displacement vector $\boldsymbol{u}$ in the contact normal direction $\boldsymbol{u}_n = \delta_n \boldsymbol{n}$ is obtained directly from the contact detection algorithm which identifies and quantifies the magnitude of body overlap between interacting bodies. The tangential contact displacement vector $\boldsymbol{u}_t$ is a bit more complicated and formulated in Equation e:TangentialContactHistory 2.7. In Equation e:TangentialContactHistory 2.7, t is the current time and t_0 is the time when the contact was initiated. Then for a true tangential contact displacement history model,the vector $\boldsymbol{u}_t$ must be stored and updated at each time for each contact point from the time when contact is initiated up until the contact is terminated.

e:TangentialContactHistory

$$\boldsymbol{u}_t = \int_{t_0}^{t} \boldsymbol{v}_t dt - \left(\boldsymbol{n} \cdot \int_{t_0}^{t} \boldsymbol{v}_t dt\right) \boldsymbol{n} \tag{2.7}$$

To enforce the Coulomb friction law, if $|\boldsymbol{F}_t| > \mu |\boldsymbol{F}_n|$ at any given time step, before the contact forces are added to the resultant force and torque on each body, the stored value of $|\boldsymbol{u}_t|$ is scaled such that $|\boldsymbol{F}_t| = \mu |\boldsymbol{F}_t|$, where μ is the Coulomb friction coefficient. An example of this is presented in Equation e:tanLimit 2.8 if a Hookean contact model is used and $f\left(\bar{R}, \delta_n\right) = 1$ in Equations e:normalContact 2.5-e:tangentContact 2.6.

e:tanLimit

$$k_t |\boldsymbol{u}_t| > \mu |\boldsymbol{F}_n| \Rightarrow \boldsymbol{u}_t \leftarrow \boldsymbol{u}_t \frac{\mu |\boldsymbol{F}_n|}{k_t |\boldsymbol{u}_t|} \tag{2.8}$$

Through the processes described above, once the contact forces $\boldsymbol{F}_n$ $\boldsymbol{F}_t$ are computed for each contact and their contributions are summed to obtain resultant forces and torques for each interacting body, the Newton-Euler equations of motion are integrated subject to the Courant-Friedrichs-Lewy (CFL) stability condition. The CFL stability condition limits the integration time step-size to $h < h_{crit}\ \sqrt{m_{min}/k_{max}}$ OSullivan&Bray2004 [37]. For multibody dynamics simulations with frictional contact modeled using the penalty approach as with this research, Chrono implements index 3 DAE solutions Chrono2016 [1].

2.3 Chrono Multibody Physics Package

s:Chrono

The physics modeling and simulation capabilities are provided by the multiphysics open-source package Chrono Chrono2016[1]. 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.

2.3.1 Vehicle Modeling

ss:Chrono_Vehicle

Built as a Chrono extension module, Chrono::Vehicle ChronoVehicle_TR[38] 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).

2.4 MPC LIDAR-Based Local Obstacle Avoidance : History and Current Algorithm

s: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&Findeisen2002 [16]. 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. The repetition of this process over time

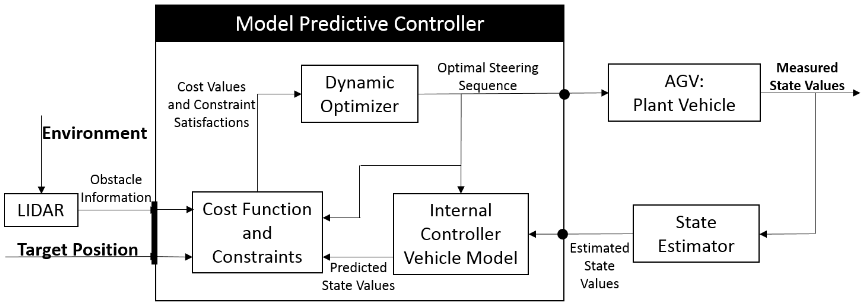
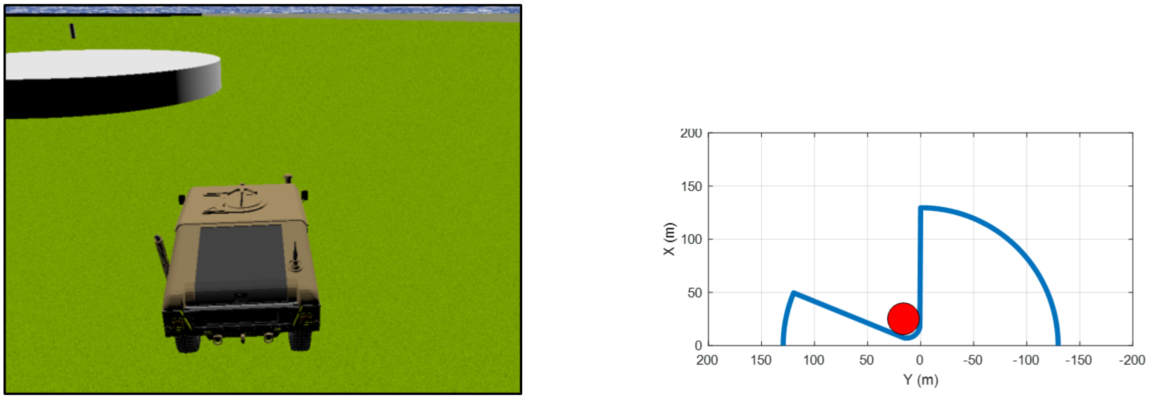


Figure 2.2: Schematic of MPC LIDAR-Based Constant Speed Local Obstacle Avoidance Controller fig:MPC_schematic

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 ModelFidelity2016 [5], 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. fig:MPC_schematic 2.2.

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°, 180°], 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



(a) 3D Visualization of LIDAR Encountering Obstacle sub@fig:obstacle_field_3D

(b) LIDAR Sensed Safe Area sub@fig:obstacle_field_LIDAR

Figure 2.3: Sample obstacle field and LIDAR output fig:LIDARExample

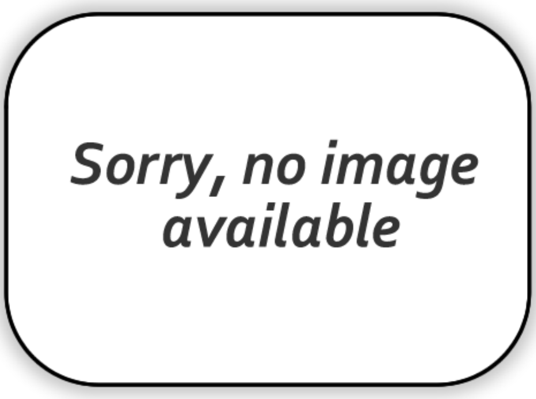


Figure 2.4: Ray-Circle Intersection Visualization fig:RayCircleIntersection

points from the LIDAR. An overhead view of the AGV encountering an obstacle and the generated LIDAR safe area polygon are presented in Fig. fig:LIDARExample 2.3.

In this research, all of the obstacles are upright cylinders so a ray-circle intersection algorithm is used to determine when each LIDAR ray intersects with a nearby obstacle. The ray-circle intersection algorithm is developed as follows. Consider a two dimensional ray in the x-y plane with the start point $\boldsymbol{E}$ and $\boldsymbol{L}$ some point along the length of the ray we are interested in. In our case, this point is the LIDAR radius. Also consider a circle with a center point defined as $\boldsymbol{C}$ and radius r. This is visualized in Fig. fig:RayCircleIntersection 2.4. We then form the following vectors:

e:RayVector
$$\boldsymbol{d} = \boldsymbol{L} - \boldsymbol{E} \tag{2.9}$$

e:CircleCenterVector
$$\boldsymbol{f} = \boldsymbol{E} - \boldsymbol{C} \tag{2.10}$$

e:Intersection
$$\boldsymbol{P} = \boldsymbol{E} + t * \boldsymbol{d} \tag{2.11}$$

Here, $\boldsymbol{d}$ is a vector describing the ray from its start point to end point. $\boldsymbol{f}$ is a vector from the circle center to the start of the vector $\boldsymbol{d}$. $\boldsymbol{P}$ is the location of the point where the ray intersects with the circle. t is a scalar which defines how far along the length of $\boldsymbol{d}$ the intersection between the ray and the circle actually occurs. The circle is then defined by:

e:Circle
$$(x - C_x)^2 + (y - C_y)^2 = r^2 \tag{2.12}$$

e:CircleExpanded
$$x^2 - 2C_x x + C_x^2 + y^2 - 2C_y y + C_y^2 - r^2 = 0 \tag{2.13}$$

We seek the point or points where the ray intersects with this circle which also happens to be described by Eq. e:Intersection 2.11 as well. Noting this yields:

e:XIntersection
$$x = E_x + t * d_x \tag{2.14}$$

e:YIntersection
$$y = E_y + t * d_y \tag{2.15}$$

Substituting Eqs. e:XIntersection 2.14 and e:YIntersection 2.15 into e:CircleExpanded 2.13 results in the following.

e:CircleSubst
$$(E_x + t * d_x)^2 - 2C_x (E_x + t * d_x) + C_x^2 + (E_y + t * d_y)^2 - 2C_y (E_y + t * d_y) + C_y^2 - r^2 = 0 \tag{2.16}$$

Expanding and grouping like terms yields the following.

$$t^2\left(d_x^2+d_y^2\right)+2t\left(E_x d_x+E_y d_y-C_x d_x-C_y d_y\right)+E_x^2+E_y^2-2E_xC_x-2E_yC_y+C_x^2+C_y^2-r^2=0 \tag{2.17}$$

e:CircleGrouped

Equation e:CircleGrouped 2.17 can be simplified further by identifying previously defined vectors and dot products can be substituted in.

$$t^2\left(\boldsymbol{d}\cdot\boldsymbol{d}\right)+2t\left(\boldsymbol{f}\cdot\boldsymbol{d}\right)+\boldsymbol{f}\cdot\boldsymbol{f}-r^2=0 \tag{2.18}$$

e:RayCircleIntersection

Equation e:RayCircleIntersection 2.18 is a quadratic equation that can be solved for two values of t. The ray-circle intersection algorithm then implemented for this research checks the discriminant of Eq. e:e:RayCircleIntersection ??. If the discriminant is less than zero, then the roots of this quadratic are imaginary and there is no intersection with the circle. If the discriminant is greater than zero, then there is an intersection and the quadratic formula is used to solve for the two values of t that satisfy Eq. e:RayCircleIntersection 2.18. Of the two solved t values, the lower value is checked to see if it is less than one but greater than zero. If so, this t value is the ray-circle intersection point used as the LIDAR detected point for the algorithm for this specific ray. Otherwise, the algorithm returns the maximum LIDAR radius. This algorithm does a quick and efficient job of simulating a perfect LIDAR sensor when the algorithm is performed on a set of rays all sweeping iteratively from the vehicle's location.

For the purpose of these simulations, the only outputs of the MPC algorithm are the steering signals. As shown in Fig. fig:MPC_schematic 2.2, 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.

The MPC controller as formulated in ModelFidelity2016 [5] 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:

e:MPCCost
$$J = s_T + wd \tag{2.19}$$

e:State_ODE
$$\dot{\xi} = v\left[\xi\left(t\right), \zeta\left(t\right)\right] \tag{2.20}$$

e:InitialStates
$$\xi\left(0\right) = \xi_0 \tag{2.21}$$

e:SafeArea
$$\tilde{S}\left[x\left(t\right), y\left(t\right)\right] \leq 0 \tag{2.22}$$

e:SteerLimit
$$\left|\delta_f\left(t\right)\right| \leq \tilde{\delta}_{f,max}\left(U_0\right) \tag{2.23}$$

e:SteerRateLimit
$$\left|\varsigma_f\left(t\right)\right| \leq \varsigma_{f,max} \tag{2.24}$$

e:TimeDomain
$$t \in \left[0, T_P\right] \; . \tag{2.25}$$

Equation e:MPCCost (2.19) 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:

e:DistanceCost

$$s_T = \sqrt{\left[x_G - x\left(T_P\right)\right]^2 + \left[y_G - y\left(T_P\right)\right]^2} \tag{2.26}$$

e:TurningCost

$$d = \int\limits_0^{T_G} \left|\varsigma\left(t\right)\right| dt\,. \tag{2.27}$$

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 (e:State_ODE 2.20-e:TimeDomain 2.25) 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 (e:State_ODE 2.20) defines a set of differential equations that describe the internal controller vehicle model, with the initial conditions of Eq. (e:InitialStates 2.21).

Equation (e:SafeArea 2.22) 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. (e:State_ODE 2.20)-(e:InitialStates 2.21) must fall inside this safe area polygon.

Equation (e:SteerLimit 2.23) 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 (e:SteerRateLimit 2.24) imposes a maximum limit on steering rate. Equation (e:TimeDomain 2.25) defines the time limits over which this optimal control problem is solved.

As noted above, Eq. (e:State_ODE 2.20) 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 s:FullVehicleModel ??). 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 ss:2DOFModel 2.5.1. The second controller, described in Section ss:14DOFModel 2.5.2, 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 ModelFidelity2016 [5] 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. fig:PossiblePaths 2.5 where, for visualization purposes, only three steering angles over three intervals were

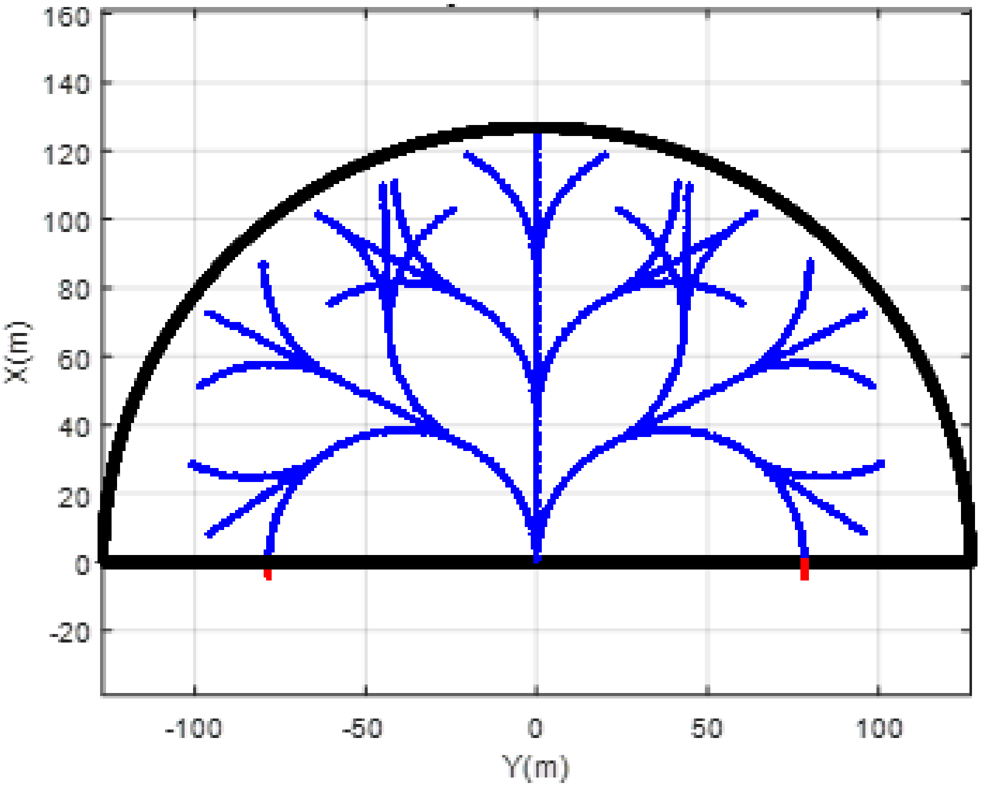


Figure 2.5: Potential Paths predicted by 2-DOF Internal Vehicle Controller Model. fig:PossiblePaths

considered. A point in polygon algorithm is used to determine which trajectories remain inside of the safe area polygon visualized in Fig. fig:LIDARExample 2.3 and therefore should be compared to the other possibilities using Eq. (e:MPCCost 2.19).

To determine if any point along the path is within the LIDAR sensed safe area polygon, a point in polygon algorithm is used. The algorithm developed for this research is similar to the winding method where the winding method tracks how many times the edges of the polygon wind around the current point of interest. However, the algorithm developed and used in this research is much simpler and quicker and is very similar to that provided in pointinpolygon2014 [39]. This algorithm checks the quadrant each vertex of the polygon is located in in a frame with origin at the point of interest, and counts the number of times that quadrant changes as the algorithm cycles through all of the vertices to determine if the point of interest is inside the polygon. The motivation for this actually comes from real llife. Imagine sitting in a closed room. A way one could test if they were actually in the room would be to iteratively look at all the corners of the room in sequence. In the person's relative reference frame fixed in its orientation, cycling through the corners of the room one would note the net quadrant change is 4, regardless of how many corners there are. Now if the person were standing outside of the room and performed this same test, they would not that the net quadrant changes was a number other than 4 meaning the room did not enclose them and therefore they are outside of the room. Again the method is similar to the winding method but the algorithm performs and checks different aspects to reach the same conclusion quicker. A note is since for this research a planar LIDAR sensor is used, this algorithm is only developed in two dimensions in the x-y plane.

To start, the polygon is simply an ordered list of its vertices. Edges are line segments connecting one vertex to the next. The first vertex of the list and the last vertex of the list are also connected to close the polygon. Next, we consider a reference frame at the point we are testing and express all of the verteces in this relative reference frame. The first vertex is chosen and the quadrant that it resides in is identified. Then the next vertex is chosen and its quadrant is identified as well. If the two points fall in the same quadrant, then nothing happens and the next vertex in the list is checked. However, if quadrant is different from the previous vertex, this changes is noted and a net quadrant change counter records this change. Clockwise is positive so changes in this direction result in +1 being added to the net quadrant change counter while counterclockwise changes result in -1 being added to the counter. In the case where the quadrant changes by two aka. quadrant 1 changes to 3 or 2 to 4 or vice versa, a vector is expressed connecting the old vertex and new vertex. Depending on the direction of the vector and if the y-intercept is above or below zero determines if +2 or -2 is added to the net quadrant change counter. Once the algorithm cycles through all of the vertices, it checks the original

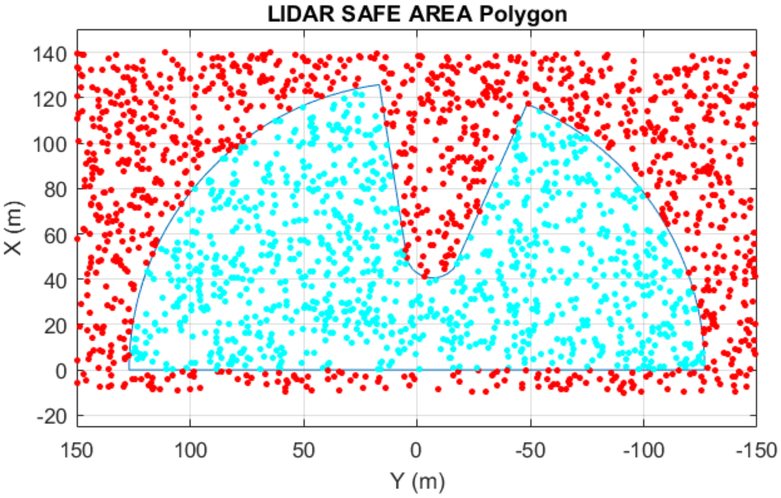


Figure 2.6: Visualization of Point in Polygon Algorithm Working Successfully. fig:PointInPolygon

vertex in order to close the polygon. If the net quadrant change is 4or -4, then the point is in the polygon. Any other situation means the point is outside of the polygon. This algorithm is efficient, simple, and has been implemented successfully for use in this research. However,this algorithm would need to be rethought for application in 3D space. Figure fig:PointInPolygon 2.6 presents a set of sample points used to show the algorithm effectively checks to see if the points fall within a sample LIDAR SAFE AREA polygon.

2.5 Multi-DOF Vehicle Dynamics Models

s:IntModel

The vehicle models embedded in the MPC are simplifications of the full, multi-body based, Chrono wheeled vehicle model which is being controlled. Two models were considered initially, providing different levels of fidelity, as described below.

2.5.1 2-DOF Vehicle Model

ss:2DOFModel

The standard vehicle model used in recently developed MPC obstacle avoidance algorithms such as ModelFidelity2016 [5] is the 2-DOF yaw plane vehicle model. These models normally either assume constant cornering stiffness or the nonlinear Pacejka Magic Formula Tire Model [?]Pacejka1997 to predict the ground tire interaction forces. For this study, the Pacejka Magic Formula is used to predict tire forces in the vehicle models. This force model is presented in the future Section 2.6s:Pacejka.

A visual representation of the 2-DOF yaw plane vehicle model can be found in Fig. 2.7fig:2DOF. The 2-DOF model is described by the following first-order ordinary differential equations:

e:2DOF_Vdot
$$\dot{V} = \left(F_{y,f} + F_{y,r}\right)/M - U_0 r \tag{2.28}$$

e:2DOF_rdot
$$\dot{r} = \left(F_{y,f} L_f - F_{y,r} L_r\right)/I_{zz} \tag{2.29}$$

e:2DOF_psidot
$$\dot{\psi} = r \tag{2.30}$$

e:2DOF_xdot
$$\dot{x} = U_0 \cos\psi - \left(V + L_f r\right)\sin\psi \tag{2.31}$$

e:2DOF_ydot
$$\dot{y} = U_0 \sin\psi + \left(V + L_f r\right)\cos\psi\,, \tag{2.32}$$

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 research, 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.

2.5.2 14-DOF Vehicle Model

ss: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 [5, 3]ModelFidelity2016, ModelFidelity2013. A benefit of using

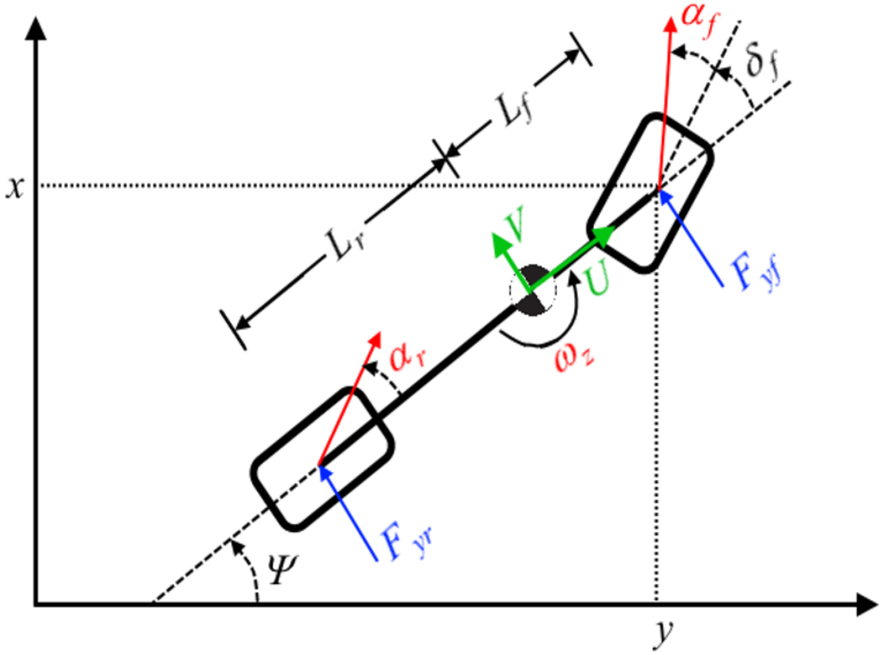


Figure 2.7: 2-DOF Vehicle Model fig:2DOF

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 RollStudies2007 [2]. 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 RollStudies2007 [2] and visualized in Fig. fig:14DOF 2.8.

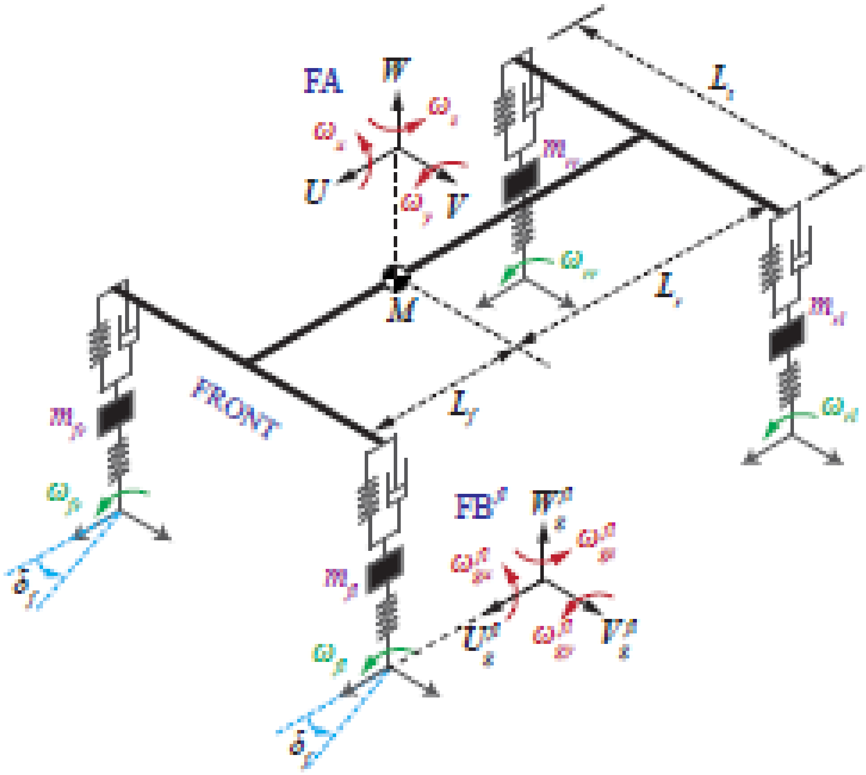


Figure 2.8: Reproduced from [2]. 14-DOF Vehicle Model

RollStudies2007

fig:14DOF

2.6 Tire Ground Force Model: Pacejka Magic Formula

s:Pacejka

The Pacejka Magic Formula is a semi-empirical tire model developed in 1987 by Hans B. Pacejka from Delft University of Technology, Egbert Bakker from Volvo Cars, and Lars Nyborg from Volvo Cars TireSensitivityStudy [40]. This model was developed further by Pacejka and commercially known as the "MF-Tire." The Magic Formula is capable of characterizing a tire with a large set of coefficients obtained from experiments. These coefficients are then used to calculate lateral, longitudinal forces as well as the self aligning torque experienced by a tire interacting with the ground.

e:MagicA

$$A = B\left(F_z\right) \cdot \alpha \tag{2.33}$$

e:MagicLateral

$$F_y = D\left(F_z\right) \cdot sin\left\{C\left(F_z\right) \cdot atan\left[A - E\left(F_z\right) \cdot A + E\left(F_z\right) \cdot atan\left(A\right)\right]\right\} \tag{2.34}$$

The lateral forces at a wheel used in the previously described 2-DOF vehicle model are calculated using Equations e:MagicA 2.33 and e:MagicLateral 2.34 for the case of pure lateral slip. F_z is the vertical load at the axle. α is the lateral tire slip angle. $B\left(\cdot\right)$, $C\left(\cdot\right)$, $D\left(\cdot\right)$, and $E\left(\cdot\right)$ are functions of the tire vertical load F_z and these relationships are described in Pacejka2012 [41]. The slips at a front and rear tire are calculated using Equations e:SlipFront 2.35 and e:SlipRear 2.36 respectively where δ_f is the steering angle of the front wheel.

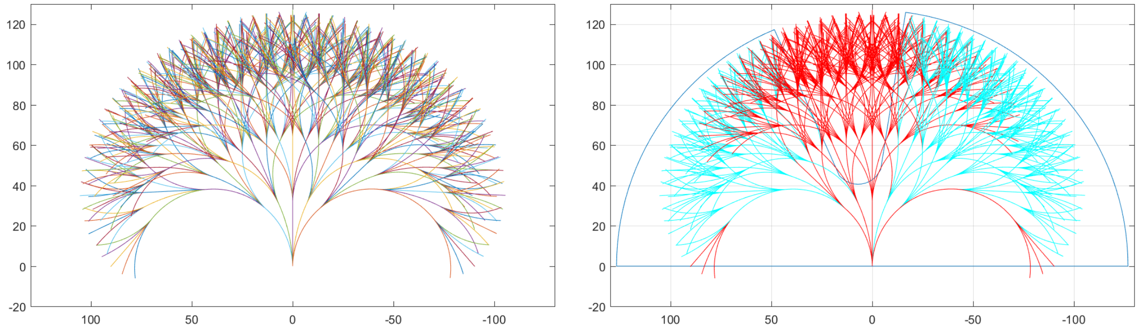
e:SlipFront

$$\alpha_f = tan^{-1}\left(\frac{V + L_f r}{U_0} - \delta_f\right) \tag{2.35}$$

e:SlipRear

$$\alpha_r = tan^{-1}\left(\frac{V - L_r r}{U_0}\right) \tag{2.36}$$

The tire vertical load is then required as an input to Equations e:MagicA 2.33 and e:MagicLateral 2.34. Longitudinal load tranfer is accounted for when calculating front and rear tire vertical loads. Justification for this inclusion can be found in ModelFidelity2016 [5]. Equations e:FrontLoad 2.37 and e:RearLoad 2.38 are then used to calculate the front and rear tire loads $F_{z,f}$ and $F_{z,r}$ and



(a) All Possible Path Possibilities sub@fig:Paths

(b) All Path Possibilities Determined if Safe or Not sub@fig:PathsSafe

Figure 2.9: Exhaustive Search Space Paths Checked with Point in Polygon Algorithm fig:SearchPaths

input into Equations e:MagicA 2.33 and e:MagicLateral 2.34.

$$F_{z,f} = \frac{MgL_r + MVrh_CG}{L_f + L_r} \tag{2.37}$$

e:FrontLoad

$$F_{z,r} = \frac{MgL_f - MVrh_CG}{L_f + L_r} \tag{2.38}$$

e:RearLoad

The Magic Formula characterizes the tire's interaction with rigid ground with a set of coefficients and uses those coefficients to match the Magic Formula with experimental data. Then a tire interacting with one type of pavement will have different Magic Formula parameters than the same tire interacting with a different type of pavement. However, the Magic Formula models tire ground interactions on rigid ground, not granular terrain i.e. there is not a granular terrain Magic Formula. There is then a mismatch between the internal controller models and the plant vehicle when controling the vehicle on granular terrain, but this research first seeks to identify what role that mismatch plays in controller performance.

maybe clean up the end here

2.7 Optimization Problem: Exhaustive Search Space

s:Optimization

insert description of the exhaustive search space method used to find the optimal solution in this study

3 EXPERIMENTAL SETUP

c:ExpSetup

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. fig:hmmwv 3.1) has a curb weight of 2, 550 kg. It includes independent front and rear double wishbone suspensions and a Pitman arm steering mechanism. The 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 wong93 [42] 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).

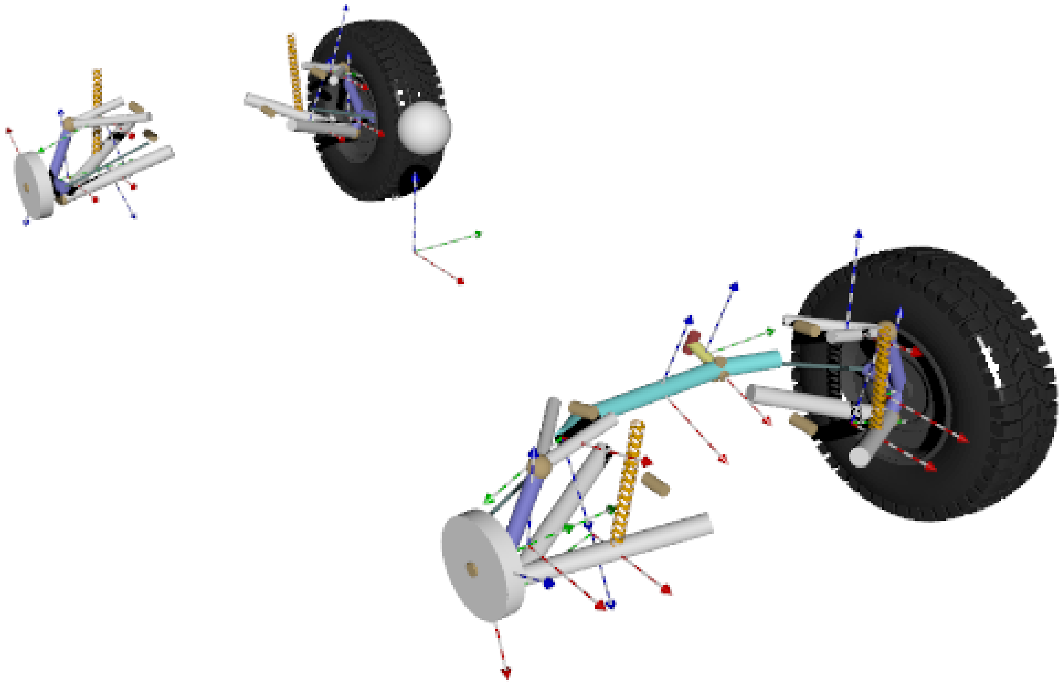


Figure 3.1: Full-vehicle HMMWV multibody model. fig:hmmwv

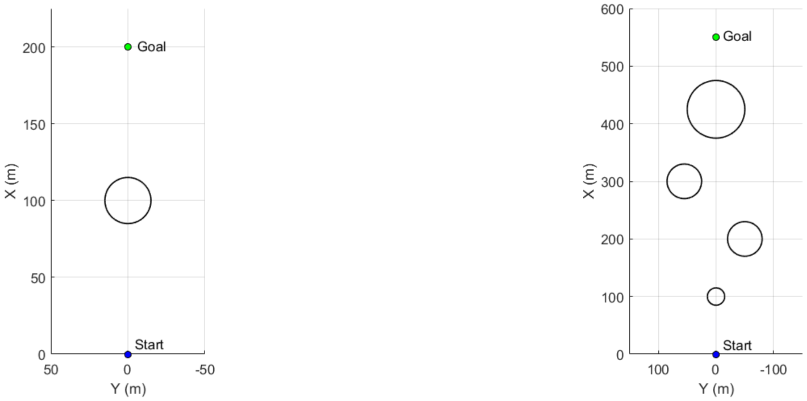
3.1 Evaluation Metrics

s: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

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

Table 3.1: Individual Simulation Test Information t:TestMatrix



(a) Field 1 sub@fig:Obst1

(b) Field 2 sub@fig:Obst2

Figure 3.2: Obstacle fields. fig:Obst

terrain type and underlying analytical controller model.

3.2 Simulation Parameters

s: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 t:TestMatrix 3.1, 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.

	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 3.2: Obstacle Field Parameters t:ObstSummary

Each of the above four tests consisted of runs on two fields with different obstacle distributions. The first one, depicted in Fig. fig:Obst1 3.2a, 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. fig:Obst2 3.2b) 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 t:ObstSummary 3.2 summarizes the obstacle locations and dimensions, as well as the target location for the above two scenarios.

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°/s.

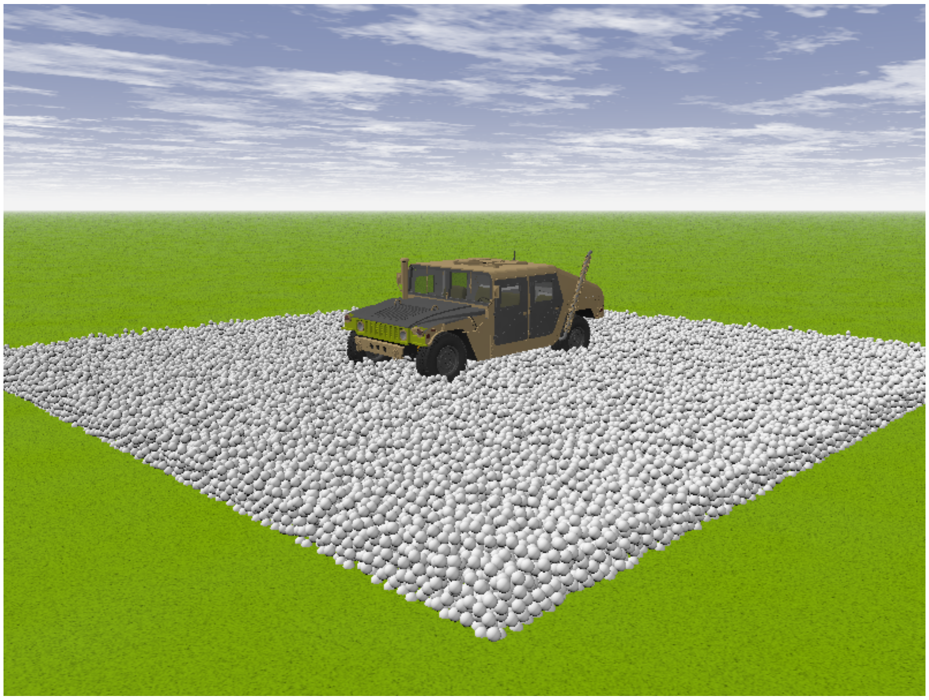


Figure 3.3: Relocating Granular Patch that follows the vehicle fig:granPatch

3.3 Granular Terrain Simulation

s: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.

create in depth description of the relocating particles algorithm

This mechanism allows a user to generate granular particles within a specified distance of the vehicle's CoG. Figure fig:granPatch 3.3 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 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 fleischmannetalGEGE2014 [43] 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 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 fleischmannetalGEGE2014 [43], 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 $35° \leq \phi \leq 40°$ if particle rotation is allowed (6 DOF particles), or with a macro-scale friction angle between roughly $65° \leq \phi \leq 70°$ 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 $35° - 40°$ if particle rotation is allowed, which is typical of a wide range of dry natural and crushed sands Cho&Dodds&Santamari [44]; or with a macro-scale friction angle of $65° - 70°$ if particle rotation is prohibited, which is typical of crushed or fragmented rock, such as railway track ballast Indraratnaetal1998 [45].

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.

3.4 Hardware and Materials

s:Hardware

The simulations throughout this research were performed on a Quantum TXR430-512R computer with Scientific Linux 7.2 installed as the primary operating system and an Ubuntu 16.04 virtual machine. The specifications of the computer are outlined in Table t:galerkin 3.3. MATLAB 2015b was used to generate all plots in a postprocessing scripts and determine evaluation metrics from each test case. POVRAY was used to generate visualizations of the performed simulations. Due to the long runtimes of each granular simulation, realtime visualization is not possible or necessary. Therefore, visualization with POVRAY is used. Simulations output large POVRAY data files that describe the body locations, sizes, and orientations at each time step. A set of scripts in Python 2.7 were developed to generate images from the POVRAY output data files and then stitch the images together to create a 60 fps video with a prescribed camera angle. The user could define using the scripts the location and orientation of the camera, what point the camera should look at, and if the camera should follow the vehicle throughout the video. This tool allows for simple and effective development of videos and images as a postprocessing method. Overall, all work will be accomplished using this computer and resources accessible on this computer. This computer outlined below is designed to run simulations with large amounts of bodies in a quick efficient manner.

Identifier	Basic Information	Additional Information	Quantity
Architecture	Quantum TXR430-512R	· Tower / 4U Convertable Chassis · Supports 2x E5-2600 v3 Processors · DDR4 Memory · Supports Up To 3x 5.25IN Bays · 8x 3.5IN Hot-Swap HDD Bays · Up To 4x Double-Width GPU · Up To 1TB DDR4 Memory · Dual Gigabit Ethernet LAN · 2000W Redundant Power Supplies	1
Processor	Intel Xeon E5-2687W v3	10C, 3.1 GHz 25M, Support DDR4-2133	2
Memory	32GB DDR4	2133MHz LR ECC LRDIMM	8
Primary Storage	512 GB 2.5 SATA III	Internal Solid State Drive (SSD) ** OS DRIVE **	1
Secondary Storage	1.8 TB SAS	12Gb/s 2.5InternalHard-Drive**RAID10**	8
RAID Controller	LSI 9300 MegaRAID SAS 9361-8i (LSI00417) PCI	· Express 3.0x8 Low Profile SATA/SAS · 1 GB Cache · High Performance Eight-Port 12 Gb/s RAIDController	1
RAID Cache	LSI LSICVM02	(LSI00418) CacheVault Accessory Kit for 9361 Series)	
GPU	Nvidia Tesla K80	24 GB GDDR5 384-bit PCI Express 3.0 - Passive Cooling	3
GPU Cooling	Passive Cooling	Passive GPU Kit for 4U 4GP USystem	1
Power	PC Smart UPS X 3000VA	Rack/Tower LCD100-127V	1
Internal Cabling		0.6m Internal Cable SFF8643 to x4 SATA HDD (mini SASHD to SATA data port)	2
		2.5” to 3.5” SATA/SAS6Gb SSD Hard Drive Converter / Adapter / Bracket	9
Additional Components		4U / Tower Conversion Kit / Rail Kit	1
		Windows 8.1 Profesional 64Bit	1
		Samsung S24D300H 24IN LCD Monitor	1

Table 3.3: Hardware Components of Computer Used for Parallel Simulations

t:galerkin

4 EXPERIMENTAL RESULTS AND DISCUSSION

c:results

t:TestMatrix

This section summarizes the results of the four tests listed in Table 3.1.

fig:ObstacleField1Traj fig:ObstacleField2Traj

The trajectories for the four performed tests are presented in Figs. 4.1a and 4.2a, for obstacle fields 1 and 2, respectively. The associated steering commands generated by the controller are presented in Figs. 4.1b and 4.2b.

fig:SteeringCommandsField1 fig:SteeringCommandsField2

s:Metrics

The evaluation metrics for each test, as defined in Section 3.1, are tabulated for each obstacle field in Tables 4.1 and 4.2.

t:EvalMetricsObst1 t:EvalMetricsObst2

4.1 Influence of model fidelity

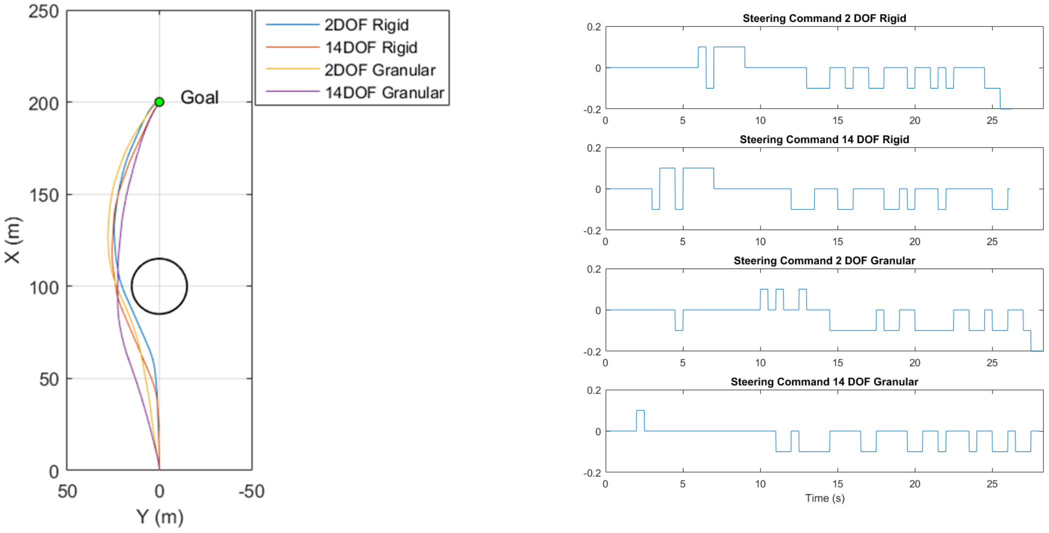
Assessing the effect of internal vehicle model fidelity on controller performance when the vehicle navigates on rigid terrain (comparison of tests 1 and 2), we make the following observations:

- 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;

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 4.1: Test Evaluation Metrics Summary on Obstacle Field 1

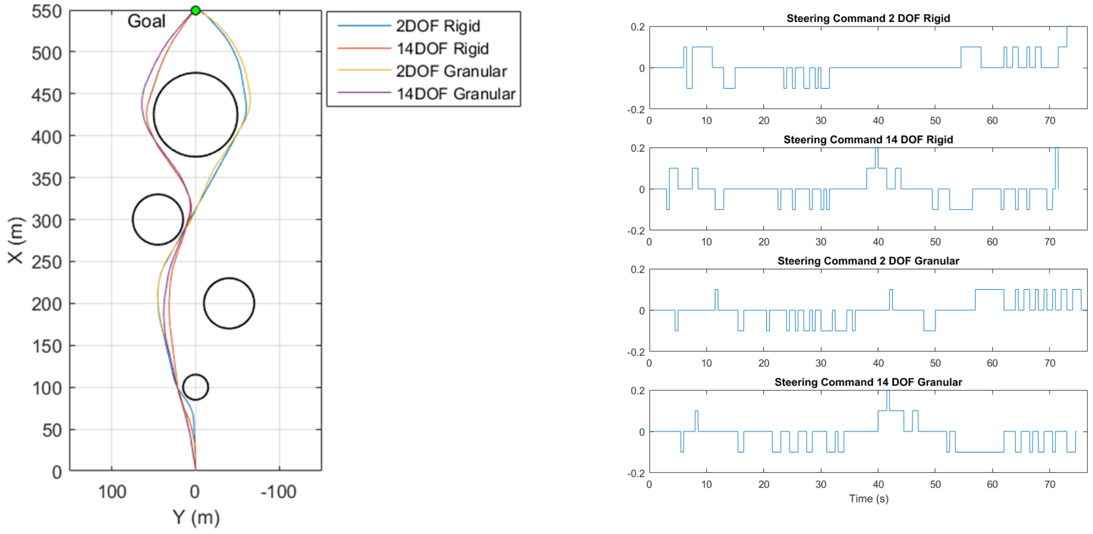
t:EvalMetricsObst1



(a) Vehicle trajectories sub@fig:ObstacleField1Trajectories

(b) Steering commands sub@fig:SteeringCommandsField1

Figure 4.1: Test results on obstacle field 1. fig:Obst1TestData



(a) Vehicle trajectories sub@fig:ObstacleField2Trajectories

(b) Steering commands sub@fig:SteeringCommandsField2

Figure 4.2: Test results on obstacle field 2. fig:Obst2TestData

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 4.2: Test Evaluation Metrics Summary on Obstacle Field 2.

t:EvalMetricsObst2

- 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 [5] even though a different AGV is considered here.

ModelFidelity20

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.

4.2 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. 4.1a and 4.2a, 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 $65° \leq \phi \leq 70°$, 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 $35° \leq \phi \leq 40°$. 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 SpeedSteer2015 [4], 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.

5 IMPROVED VEHICLE MODEL ON GRANULAR MATERIAL

c:improvedModel

5.1 Bekker/Wong Load Sinkage Relationship

s:BekkerWong

Often the interaction between a rigid tire and soft soil is modelled using terramechanics relations originating from the work of Bekker and Wong wong93 [42]. Penetration tests were performed to simulate the contact area of a tire or track interacting with soft ground and the pressure p and sinkage z were measured for the array of experiments. Based on this experimental data, Bekker proposed the well known load-sinkage relationship in Equation e:BekkerLoadSinkage 5.1 wong93 [42]. n is an exponent of deformation, b is the width of the tire, k_c is the cohesive modulus of deformation, and k_ϕ is the frictional modulus of deformation.

$$p = \left(\frac{k_c}{b} + k_\phi \right) z^n \qquad (5.1)$$

e:BekkerLoadSinkage

If the normal stress is modelled anywhere along the wheel-terrain interface as Bekker proposed, then Equation e:BekkerLoadSinkage 5.1 becomes Equation e:modifiedNormal 5.2 where θ is defined in Figure fig:wheelTerrain 5.1.

$$\sigma\left(\theta\right) = \left(\frac{k_c}{b} + k_\phi \right) \zeta\left(\theta\right)^n \qquad (5.2)$$

e:modifiedNormal

Using terramechanics relations as in wong93 [42], the shear stress τ can then be determined along the wheel-terrain interface with Equation e:shearStress 5.3 where c is the terrain cohesion, ϕ is the internal friction angle of the granular terrain, r is the wheel radius, K is the shear deformation modulus, and i is the wheel slip ratio defined in Equation e:wheelSlipRatio 5.4. ω is the angular speed of the wheel and v_t is the speed of its center Ghotbi2016 [46].

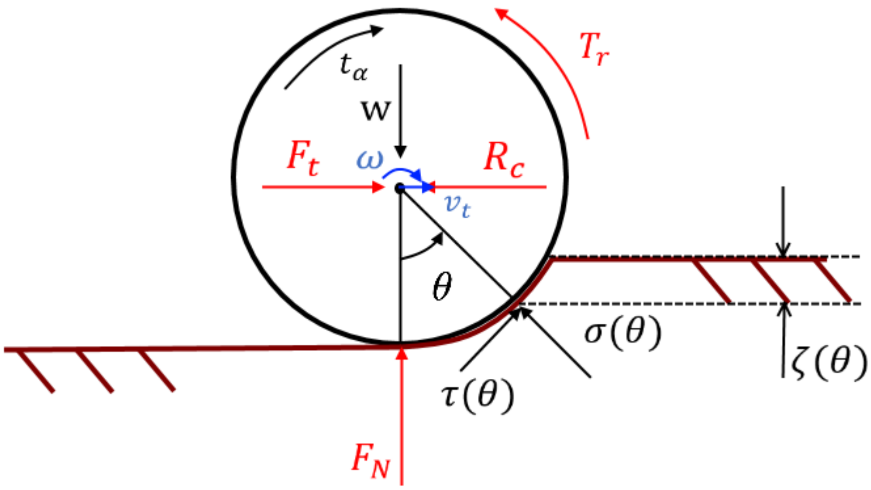


Figure 5.1: Terrain interactions on a single wheel on soft soil. fig:wheelTerrain

e:shearStress

$$\tau\left(\theta\right) = [c + \sigma\left(\theta\right) tan\phi][1 - e^{-r[\theta_1 - \theta - (1-i)(sin\theta_1 - sin\theta)]/K}] \tag{5.3}$$

e:wheelSlipRatio

$$i = \frac{(r\omega - v_t)}{r\omega} \tag{5.4}$$

5.2 New Simplified Force Model

s:NewForce

.

The results in Chapter c:results 4 promote the need for a simple yet improved method of estimating the tire forces at the ground tire contact patch when a vehicle is operating on soft or granular terrain. A benefit to the Pacejka Magic Formula presented in Section s:Pacejka 2.6 is tire force calculation is backed by experimental data and the functions are shaped by a large number of experimentally determined parameters. However, the Magic Formula was derived for rigid ground applications only and does not succeed in estimating tire forces on granular terrain. The Magic Formula overestimates the lateral and longitudinal tire forces the vehicle experiences for any given slip angle as it does not take into account the shear strength of the soil. Therefore a simplified method of estimating the lateral tire forces at a tire ground contact patch was developed. This model is then used on the 2-DOF analytical vehicle modelto estimate lateral tire forces and better predict vehicle trajectories on granular terrain.

Rather than using Equation e:modifiedNormal 5.2 to calculate the normal stress at the tire ground contact patch, instead the normal load is calulated using Equations e:FrontLoad 2.37 and e:RearLoad 2.38. Then the following assumptions are made. First, the tire ground contact patch is assumed to be square of side length b which represents the width of the tire. Second, we assume a constant normal stress σ throughout the contact patch such that the total force contribution from this normal stress is equal and opposite the normal load calculated in Equations e:FrontLoad 2.37 and e:RearLoad 2.38. This leads to the simplified estimate of tire ground patch normal stress in Equation e:simplifiedNormal 5.5. Third, we assume the tangential stress along the tire contact patch to be constant as well. These assumptions lead to the simplification of e:shearStress 5.3 and when we only consider the pure lateral slip case simplifies to Equation e:simplifiedShear 5.6.

e:simplifiedNormal

$$\sigma = \frac{F_z}{b^2} \tag{5.5}$$

e:simplifiedShear

$$\tau = [c + \sigma tan\phi][1 - e^{-r[\frac{b}{r} - sin(\frac{b}{r})]/K}] \tag{5.6}$$

Using the simplified constant stress relations and trigonometry, the lateral force is then calculated using Equation e:newLatForce 5.7. Though this development is the result of a handful of simplifying assumptions, the results of the formulation make sense. A comparison between the force prediction as a fucntion of slip as predicted by the Magic Formula is compared to this new force model prediction in Figure fig:newForceComparison 5.3. Intuitively one would expect these sorts of results because on granular terrain the tire ground contact patch is subject to a much more limited shear strength than a

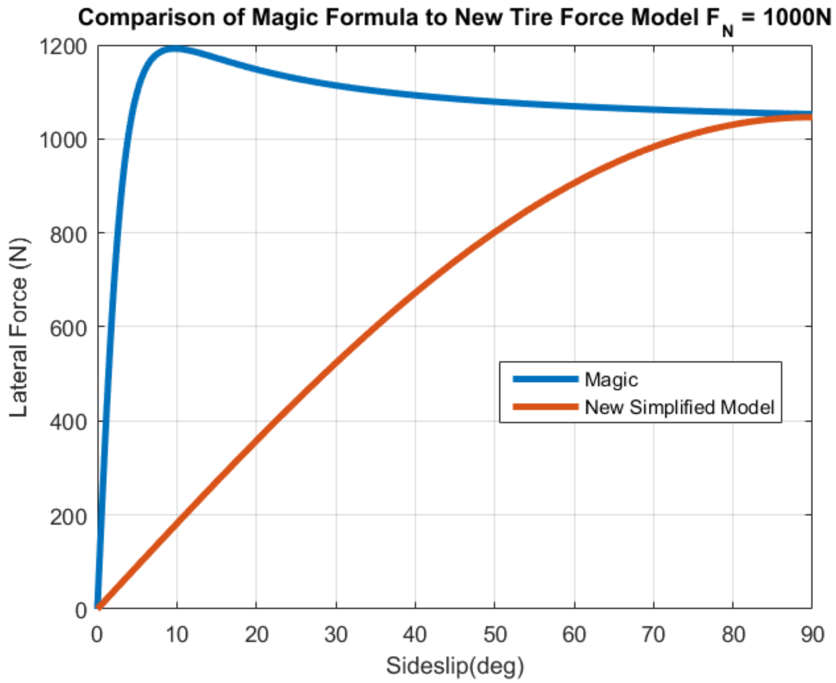


Figure 5.2: Comparison of Magic Formula Lateral Force Prediction to New Force Model fig:newForceComparison

tire operating on rigid ground. This new force model then takes into account the cohesion and internal friction angle of the granular terrain in a meaningful way. The new model predicts higher lateral forces when a more cohesive soil or terrain with higher friction angle is used and vice versa.

$$F_y = -\tau b^2 sin\left(\alpha\right) \qquad (5.7)$$

e:newLatForce

5.3 Open Loop Comparisons

s:OpenLoop

To justify that the new force model better models the lateral ground interaction force between a tire and granular terrain, a set of open loop simulations were performed. In these simulations, the HMMWV was initialized at rest, quickly accelerated to 8.3 m/s, and continued at this speed until the simulation has run for 10 seconds. Then the vehicle undergoes the steering sequence presented in Figure fig:openSteer 5.4 which consists of the vehicle steering left with a steering angle of 10 degrees for

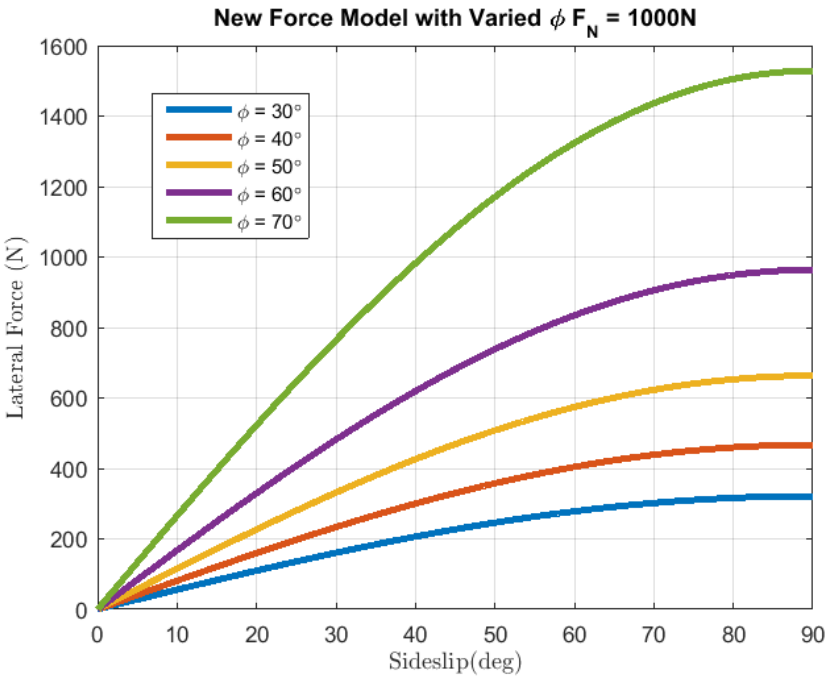


Figure 5.3: Effect of varying internal friction angle ϕ on lateral force prediction fig:newForceComparison

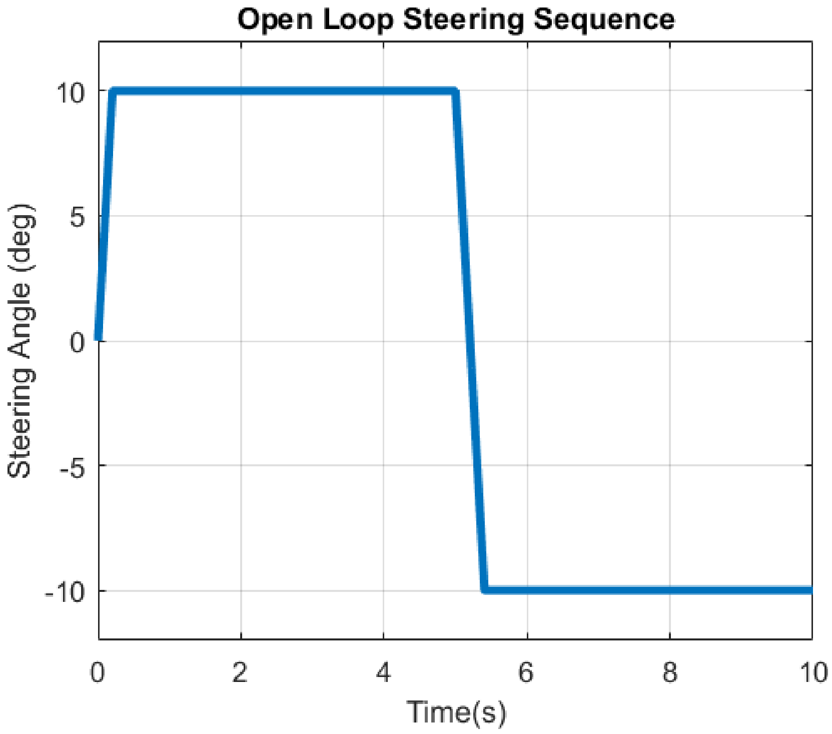


Figure 5.4: Steering Sequence used for Open Loop Simulations fig:openSteer

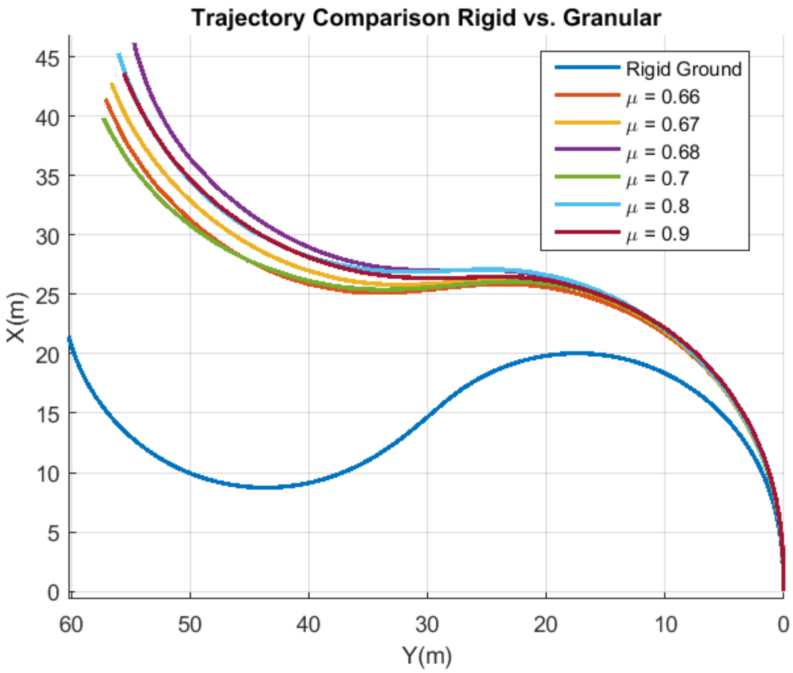


Figure 5.5: Open Loop Trajectory of HMMWV over non-rotating granular terrains with different interparticle friction coefficients μ fig:openLoopmu

5 seconds followed by steering right at a steering angle of -10 degrees for another 5 seconds. In this open loop test we are only interested in the vehicle's trajectory for the last 10 seconds of the simulation so only this time frame is considered and shown in plots.

Figure fig:openLoopmu 5.5 presents the resulting trajectories of 7 different open loop simulations run with this steering sequence with the parameters just discussed. One simulation was run on rigid flat ground while the other 6 were run on granular terrain consisting of non-rotating particles of inter-particle friction coefficients varying from μ=0.66 to μ=0.9. Lower values of μ were considered and tested, but these simulations resulted in the vehicle simply initializing and remaining stuck at the location where it started, constantly spinning its tires without making any forward progress. From these results it is shown that a vehicle operating on granular terrain will steer less sharply than the same vehicle operating on rigid ground undergoing the same steering sequence at the same speed. From these simulations we see there

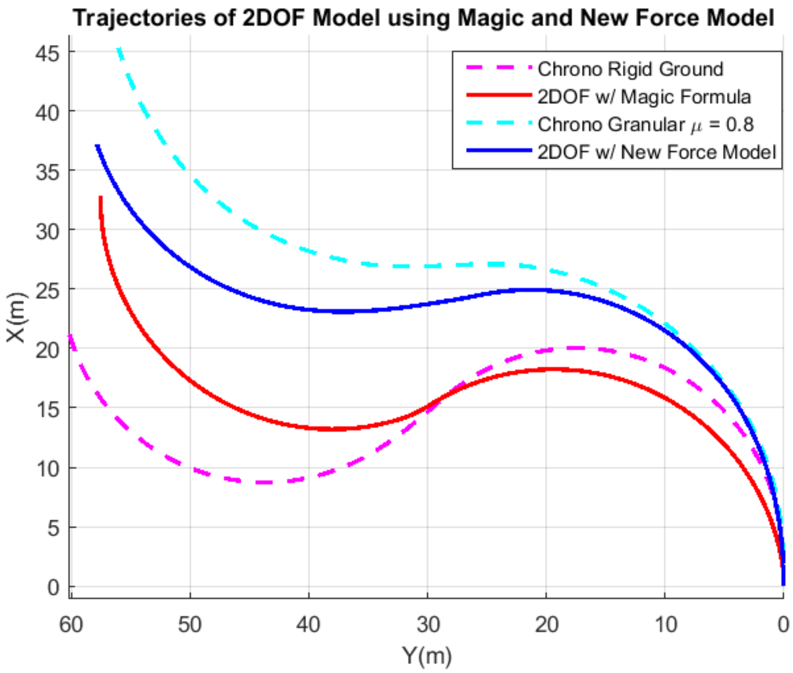


Figure 5.6: Comparison of Trajectory Predicted by 2-DOF model using the Magic Formula to estimate Tire Forces and the 2-DOF model using the new simplified force model fig:MagicvsNew

are clear differences in how a vehicle performs on rigid ground versus on granular terrain and this difference needs to be accounted for when attempting to model vehicle dynamic performance using simplified analytical equations.

The new force model developed in the previous section seeks to address this difference in vehicle performance based on terrain type. To confirm the new force model does a better job of modeling the lateral force experienced by a tire at the ground tire contact patch, two more open loop simulations were performed. Here the 2-DOF analytical yaw plane vehicle model was simulated for 10 seconds while undergoing the same steering sequence shown in Figure fig:openSteer 5.4. In one simulation, the 2-DOF model uses the Pacejka Magic Formula to estimate lateral tire forces experienced by the vehicle. The second simulation uses the new force model to estimate the lateral tire forces experienced by the vehicle. These simulation trajectories are presented in Figure fig:MagicvsNew 5.6 along with the Chrono open loop simulations trajectories on rigid ground and nonrotating granular terrain with μ=0.8. The

2-DOF vehicle model using the new force model to predict tire forces matches the Chrono simulation on granular terrain much more closely than the 2-DOF model with using the Magic formula. Instead the 2-DOF model with the Magic Formula matches the Chrono simulation on rigid ground much better. Even though the results in Chapter c:results 4 show the controlled HMMWV was still able to successfully navigate to a target location while avoiding obstacles on granular terrain, there was a clear model mismatch between the internal vehicle controller model and the actual plant vehicle. The 2-DOF model using the PAcejka Magic Formula was overestimating the lateral tire forces because it did not take into account the granular terrain. This new force model does account for the granular terrain and based on Figure fig:MagicvsNew 5.6 does a better job of predicting vehicle trajectories on granular terrain.

6 SECOND EXPERIMENT TO TEST NEW VEHICLE MODEL

c:Exp2

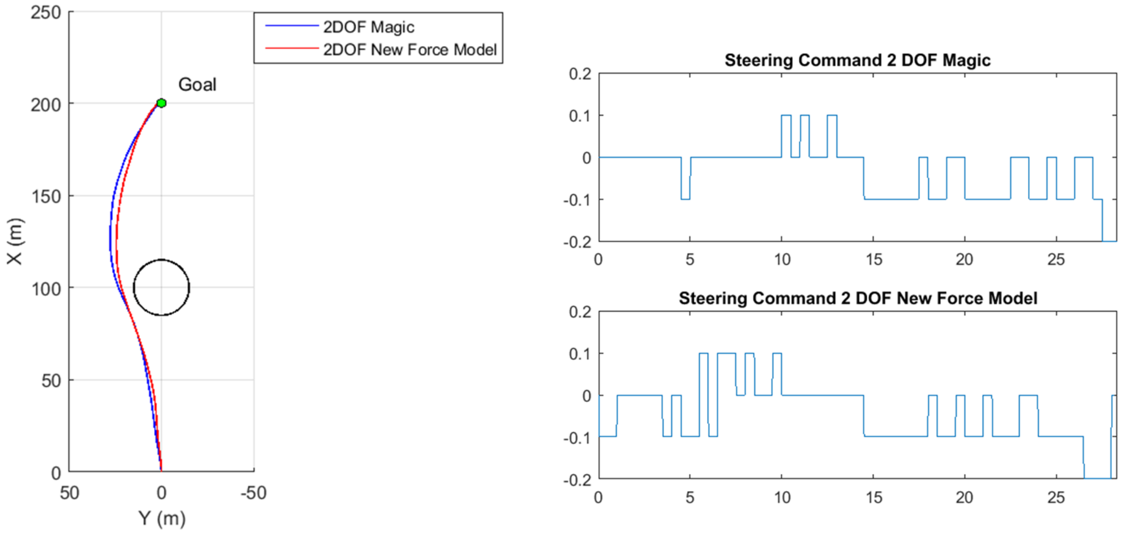
6.1 Experimental Setup

s:ExpSetup2

Another set of simulations were performed outlined in Table t:TestMatrix2 6.1. This experiment consisted of four simulations on the same two obstacle fields as the previous experiment. Two analytical controller models are considered for this experiment, namely the 2-DOF model using the Pacejka Magic Formula to estimate lateral tire forces on the tires and another 2-DOF yaw plane vehicle model using the newly developed force model to estimate lateral tire forces. As was shown in Section s:OpenLoop 5.3, open loop simulations enforced that this 2-DOF vehicle model with the new force model does a better job of predicting vehicle trajectories on granular terrain than the previously used Magic Formula 2-DOF models. The same evaluation metrics summarized in Section s:Metrics are used in this experiment to compare simulation runs and determine which controller actually worked best. However, we are primarily concerned with the Time to Target evaluation metric in this experiment and therefore this will be the most influential factor in determining which controller worked best.

Test Number	Obstacle Field	Terrain Type	Controller Vehicle and Force Model
1	1	Granular	2-DOF Magic
2	2	Granular	2-DOF Magic
3	1	Granular	2-DOF New Model
4	2	Granular	2-DOF New Model

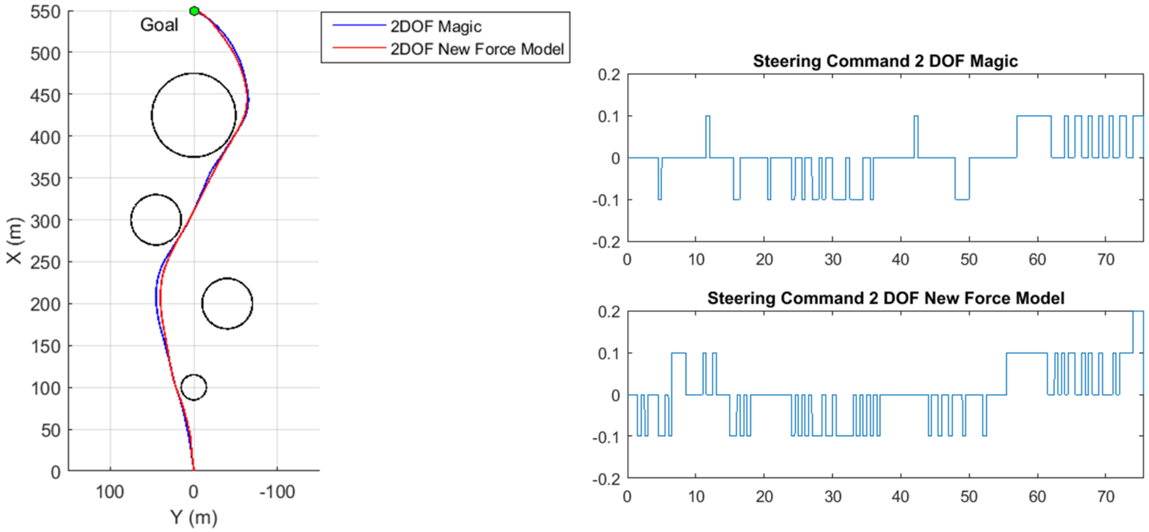
Table 6.1: Compared Simulations for Second Experiment t:TestMatrix2



(a) Vehicle trajectories. sub@fig:ObstacleField1TrajectoriesExp2

(b) Steering commands. sub@fig:SteeringCommandsField1Exp2

Figure 6.1: Test results on obstacle field 1 fig:Obst1TestDataExp2



(a) Vehicle trajectories. sub@fig:ObstacleField2TrajectoriesExp2

(b) Steering commands. sub@fig:SteeringCommandsField2Exp2

Figure 6.2: Test results on obstacle field 2 fig:Obst2TestDataExp2

Test Number	1	3
Controller Model	2-DOF	2-DOF
Terrain	Granular	Granular
Force Model	Magic Formula	New Force Model
Time to Target (s)	28.32	28.15
Minimum Obstacle Distance (m)	3.491	2.837
Controller Effort	0.0340	0.0381
Max. Lateral Acceleration (m/s^2)	2.47	2.69
Avg. Lateral Acceleration (m/s^2)	0.55	0.53

Table 6.2: Test Evaluation Metrics Summary on Obstacle Field 1 t:EvalMetricsObst1Exp2

Test Number	2	4
Controller Model	2-DOF	2-DOF
Terrain	Granular	Granular
Force Model	Magic Formula	New Force Model
Time to Target (s)	76.64	75.58
Minimum Obstacle Distance (m)	1.083	1.440
Controller Effort	0.0714	0.0952
Max. Lateral Acceleration (m/s^2)	2.55	2.75
Avg. Lateral Acceleration (m/s^2)	0.53	0.51

Table 6.3: Test Evaluation Metrics Summary on Obstacle Field 2 t:EvalMetricsObst2Exp2

6.2 Results

s:Results2

The trajectories for the four performed tests are presented in Figs. 6.1a and 6.2a, for obstacle fields 1 and 2, respectively. The associated steering commands generated by the controller are presented in Figs. 6.1b and 6.2b.

The evaluation metrics for each test, as defined in Section 3.1, are tabulated for each obstacle field in Tables 6.2 and 6.3.

From the results summarized in the plots and tables, the 2-DOF model using the new force model to predict lateral tire forces results in the HMMWV navigating to the target quicker than the 2-DOF controller using the Magic Formula to estimate tire forces. This improved performance is witnessed on both obstacle fields. On Obstacle Field 1, the new force 2-DOF model navigates closer to the single obstacle than the 2-DOF Magic model but this is just the opposite on obstacle field 2. On both obstacle fields, the controller effort of the new force 2-DOF model is slightly higher than the controller effort of the 2-DOF Magic controller. Finally, the average and lateral accelerations experienced by the vehicle when controlled by each different 2-DOF model are similar and the magnitude differences are negligible.

Qualitatively examining the trajectories of all four tests, the trajectory of the 2-DOF new forcemodel run appears to be much smoother on both obstacle fields than its 2-DOF Magic controller counterpart. This is evident when looking at the runs on the first obstacle field. The 2-DOF Magic model commands result in the HMMWV taking a much wider turn followed my a more aggresive turning maneuver to reach the target position. However, when the vehicle is controlled by the 2-DOF model with the new force model, the HMMWV takes a more direct route to the target position and this improved performance is expressed in the decreased Time to Target. This difference is due to the Magic Model's overestimation of lateral tire forces that the HMMWV experiences. Since the 2-DOF vehicle model using the

Magic formula estimates the vehicle can turn sharper at a given steering angle than the terrain actually allows, the predicted trajectories for any given steering angle are wrong and the HMMWV ends up changing its heading less than the controller predicts. This can also be seen comparing the trajectories on obstacle field 2. Both controllers command the HMMWV to follow similar trajectories, navigating the HMMWV in the same direction around all of the obstacles. Again though the 2-DOF new force model controller trajectory seem smoother of the two. The internal analytical vehicle model matches the plant vehicle better which leads to predicted trajectories that match the actual behavior of the HMMWV. These two facts combined lead to a better controlled HMMWV that navigates the the target quicker than if its trajectories were predicted using the 2-DOF Magic formula model.

Overall based on the comparison of evaluation metrics and qualitative comparison of the trajectories, in this experiment the controller with the 2-DOF vehicle model using the new force model performed better than the controller using a 2-DOF model with the Magic formula. The trajectories of the controlled HMMWV are smoother and result in quicker Time to Target metrics. It is therefore possible to develop a tire force model which takes into account the soil parameters to estimate the lateral forces at the tire ground contact patch and use this new tire model to better predict vehicle trajectories in a MPC controller. In the future, this 2-DOF vehicle model with the new force model should be used or developed further to better control a wheeled vehicle on flat granular terrain of varying cohesive and shear strengths as well as interparticle friction coefficients.

7 CONCLUSIONS

c: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 initially, one that uses a 2-DOF vehicle model using the Pacejka Magic Formula as a tire 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 ModelFidelity2016[5], 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 introoduced 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.

From the results of the first experiment, a new lateral tire force model was developed using common terramechanics relationships developed by Wong and Bekker. This new force model uses the cohesive strength and internal friction angle of the granular terrain to calculate the lateral forces experienced by a tire provided the current slip angle and wheel speed. From the first experiment,it was determined that though the 14-DOF internal vehicle model resulted in better controller performance, the complexity of this model made it undesirable for realtime implementation. Though the 2-DOF controller navigated to the target marginally slower, it still did succeed in reaching the target. Therefore the 2-DOF model was deemed desirable for future studies due to its success in reaching the target as well as the quicker runtime speeds necessary to determine a optimal path. For this reason then a second experiment was conducted using twodifferent controllers. The first is the same controller from the first experiment consisting of an internal 2-DOF vehicle model using the Pacejka Magic Formula to predict vehicle trajectories. The second controller uses the 2-DOF vehicle model but uses the new lateral force model to better predict trajectories on granular terrain. These controllers were tested on the same two obstacle fields from the first experiment but solely on a granular terrain resembling a stiff railroad ballast, the same granular terrain as from the first experiment. The new controller performed better than the old controller using the Magic formula to predict tire forces and this improvement is shown by a lower Time to Target as well as smoother trajectories with the second controller. This new intermediate analytical model is not as complex as the 14-DOF model but does take into account granuler parameters such as the friction angle to result in more accurate vehicle trajectory predictions on granular terrain and ultimately a better controlled HMMWV.

Each of the five objectives have been accomplished through the development and execution of the experiments described in this study. For clarification, each of the five objectives are summarized and how they were accomplished are presented as follows.

1. Develop a robust simulation environment using Chrono to simulate granular terrain efficiently over large large terrains without actually modeling every single particle over the large terrain. This was accomplished by leveraging the multibody and parallel computational abilities provided by Chrono. A novel relocating soil patch was developed to make simulation of a vehicle traversing large granular terrains possible without requiring months of runtime per experiment.

2. Study the performance of the MPC Controller on granular terrain as compared to that on rigid terrain. The results of this specific performance analysis are summarized in Chapter c:results 4. These results enforced the need for more research into a better method of predicting tire forces at the tire ground patch in a simple manner.

3. Analyze the impact of fidelity of the internal controller model on speed and performance of the obstacle avoidance controller. These results are summarized in Chapter c:results 4 for Experiment 1. Overall similar conclusions were made to those drawn in ModelFidelity2016 [5]. Namely, the controller using the more complex 14-DOF model resulted in marginally quicker Time to Target metrics than the

2-DOF controller model, but the increased complexity of the 14-DOF model increased controller runtimes making the 14-DOF model an undesirable choice for realtime implementation. The 2-DOF controller did still perform successfully and was chosen for more development to improve its performance in Experiment 2. The 14-DOF model should be used to determine safety consrtaints to impose on a 2-DOF model used as the controller model.

4. Improve the controller for better performance on granular terrain. Chapter 5 c:improvedModel presents the development of this improved 2-DOF vehicle model which uses a newly developed lateral force model stemming from Bekker and Wong's terramechanics relations to more accurately predict vehicle trajectories on granular terrain. 6 c:Exp2 presents the results of a second experiment run to confirm the newly developed vehicle model results in better controlled HMMWV performance.

5. 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. All of the simulations and experiments from this study support and prove this objective. Using Chrono to create a virtual testing environment made it possible to quickly change aspects of the controller design to understand specific aspects of the control algorithm such as model fidelity or terrain's effect on controller performance. Testing the controller through simulation allowed for quicker and safer experiments than if one were to attempt to run each of these experiments in real life. Though the goal of these virtual experiments is to eventually prepare the controller for real life implementation and tests, these virtual tests should speed up the design process and make real life experiments that much more beneficial and meaningful to the design team.

Richard Hamming once said the purpose of computing is insight, not numbers NumMethods [47]. All mathematical models are an abstraction of reality, so what the user is interested in exploring or modeling should drive the complexity of the model. Detailed analysis of vehicle performance requires sophisticated high fidelity models often leveraging complex contact algorithms and sophisticated tire behavior representations. Control engineers make use of mathematical models to describe the dynamics of the system they are attempting to control. Linearizations, model simplifications, and other assumptions allow the controller to increase its speed while still describing the desired aspect of the system appropriately. This point is important when considering all of the research and work done to achieve the results of this study. Computer simulations can be extremely beneficial to the design process as long as the user has a strong understanding of the background, mathematical models, and methods that have driven the simulations. However, simulation simply cannot and more importantly should not replace all real life experimentation. No matter how complex a developer makes their simulation, it will still always be an abstraction of reality to some degree. However, it can be an extremely powerful tool in the design process, allowing a user to test core functionalities of their mechanical or controller design to gain insight towards how to improve thier design. When used in this manner, virtual and real life experimentation can cooperatively contribute to quicker and more effective design processes. If more tests are run virtually then this saves money and timefor the team that then runs physical tests on the most promising designs from virtual experimentation.

This applies directly to all of the work and research performed in this study. This research has provided insight into the MPC LIDAR-based local obstacle avoidance controller. Due to this experiments outlined in this thesis, there is now a better understanding of how model fidelity and terrain affect the performance of this researched controller. From the insight gained from the first experiment, the controller and internal model were improved and tested virtually. The results of the second experiment emphasized that it is possible to develop a simplified vehicle model that predicts its lateral tire forces using simple terramechanics relations and allow the controlled vehicle to better navigate on granular terrain of varying cohesive strength and frictional properties. In these experiments, a high fidelity sophisticated multibody physics model of a HMMWV was used as the plant vehicle due to the goal of this simulation which is to analyze vehicle performance when controlled by different MPC controllers. Inside of the controller then, simplified analytical vehicle models are used which are another level of abstraction in order to quickly predict the trajectory of the simulated HMMWV based on terrain parameters. As simulation methods improve and computational speeds continue increasing, simulation's role in the design process will increase which can lower design costs and improve design speeds as well.

7.1 Future Work

s:FutureWork

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 using the Magic formula is sufficient for predicting vehicle performance well enough to guide the vehicle successfully and safely to a target point. However, a new force model was developed using terramechanics relations to better approximate the lateral tire forces experienced by the 2-DOF vehicle model which results in better approximations of vehicle trajectories on granular terrain. This research opens up many opportunities and avenues for future research topics. This research lends itself to a variety of different parametric studues to understand how changing aspects of the terrain or controller affects the overall vehicle performance. For example, parametric studies examining controller performance on a wider array of granular terrains would allow for for insight into how granular terrain parameters effect vehicle performance. In this research exhaustive search space was used to find an optimal solution. In the future though, a much quicker optimization method would be necessary to solve the optimal control problem each time step and ultimately implement the controller in realtime applications.

A newer control algorithm could be developed all together with the goal of navigating a vehicle on all sorts of terrains. If instead of using a constant speed internal vehicle model and searching for an optimal steering sequence only, a controller could search for an optimal steering and acceleration input. If this were the case then the terrain parameters would impose constraints on how quickly a vehicle could actually accelerate on the terrain thus constraining the search space and resulting in a much smarter granular terrain navigation controller.

Another avenue of research would be to explore improvements to the granular simulations developed in this study. To simulate granular terrain, in this study a novel relocating granular patch mechanism was developed that simulates a significant thickness of granular terrain underneath the vehicle at all times and relocates particles in front of the vehicle as it moves in any direction. This research would not have been feasible without the development of this mechanism, but as with any research topic there are further improvements that could help simulation speeds for these sorts of experiments. One possible improvement would be the implementation of periodic boundary conditions at the walls of the relocating patch to minimize any boundary effects at the tire ground contact patch.

Finally, the research performed here would benefit greatly from physical tests to compliment the results of this study. Insight would be gained relating to hardware and computational speeds necessary to implement this controller on a physical vehicle.

Overall the terramechanics field is a highly active and current area of research. The research performed in this study promotes the opportunity for many equally exciting research opportunities building off the results of this study.

Bibliography

[1] 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.

[2] T. Shim and C. Ghike. Understanding the limitations of different vehicle models for roll dynamics studies. *Vehicle System Dynamics*, 45:191–216, 2007.

[3] 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.

[4] J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal. An mpc algorithm with combined speed and steering control for obstacle avoidance in autonomous ground vehicles. *Dynamic Systems and Control Conference*, 2015.

[5] 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.

[6] Next-generation nato reference mobility model (nrmm) development. `https://www.cso.nato.int/ACTIVITY_META.asp?ACT=6243`. Accessed: September 21, 2016.

[7] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.

[8] E. Rimon and D. E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, 1992.

[9] 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.

[10] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *IEEE Intelligent Vehicles Symposium*, pages 417–424, 2009.

[11] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, pages 23–33, 1997.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] F. Allgower and R. Findeisen. An introduction to nonlinear model predictive control. In *21st Benelux Meeting on Systems and Control*, 2002.

[17] P. Ogren and N. E. Leonard. A convergent dynamic window approach to obstacle avoidance. *IEEE Transactions on Robotics*, 21:188–195, 2005.

[18] 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.

[19] A. Tahirovic and G. Magnani. Passivity-based model predictive control for mobile robot navigation planning in rough terrains. 2010.

[20] 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.

[21] M. Nanao and T. Ohtsuka. Nonlinear model predictive control for vehicle collision avoidance using c/gmres algorithm. 2010.

[22] Y. Gao, F. Borrelli, T. Lin, E. Tseng, and D. Hravot. Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads. 2010.

[23] 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.

[24] M. A. Abbas, J. M. Eklund, and R. Milman. Real-time analysis for nonlinear model predictive control of autonomous vehicles. 2012.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] C. E. Beal and J. C. Gerdes. Model predictive control for vehicle stabilization at limits of handling. volume 21, pages 1258–1269, 2013.

[30] K.S. Ti, B. B. K. Huat, J. Noorzaei, M. D. S. Jaafar, and G. S. Sew. A review of basic soil constitutive models for geotechnical applications. *Journal of Geotechnical Engineering*, 14:1–18, 2009.

[31] P. Jayakumar, D. Mechergui, and T. M. Wasfy. Understanding the effects of soil characteristics on mobility. 2017.

[32] A. F. Filippov. *Differential Equations with Discontinuous Righthand Sides.* Springer, 1988.

[33] David E. Stewart. Rigid-body dynamics with friction and impact. *SIAM Review*, 42(1):3–39, 2000.

[34] L. E. Silbert, D. Ertaş, G. S. Grest, T. C. Halsey, D. Levine, and S. J. Plimpton. Granular flow down an inclined plane: Bagnold scaling and rheology. *Physical Review E*, 64(5):051302, 2001.

[35] H. P. Zhang and H. A. Makse. Jamming transition in emulsions and granular materials. *Physical Review E*, 72(1):011301, 2005.

[36] M. Machado, P. Moreira, P. Flores, and H. M. Lankarani. Compliant contact force models in multibody dynamics: evolution of the Hertz contact theory. *Mechanism and Machine Theory*, 53:99–121, 2012.

[37] C. O'Sullivan and J. D. Bray. Selecting a suitable time step for discrete element simulations that use the central difference time integration scheme. *Engineering Computations*, 21(2-4):278–303, 2004.

[38] R. Serban and D. Negrut. Chrono::Vehicle – Template-Based Ground Vehicle Modeling and Simulation. Technical Report TR-2016-10: `http://sbel.wisc.edu/documents/TR-2016-10.pdf`, Simulation-Based Engineering Laboratory, University of Wisconsin-Madison, 2016.

[39] S.G. Jairo. point in polygon 'by quadrants' very fast algorithm. `https://www.youtube.com/watch?v=f-DIsdvwJ7E`. Accessed: 2016-11-18.

[40] F. Mohammadi. Tire characteristics sensitivity study. Master's thesis, Chalmers University of Technology, Gothenburg, Sweden, 2012.

[41] H. Pacejka. *Tire and Vehicle Dynamics.* Butterworth-Heinemann, 3rd edition, 2012.

[42] J. Y. Wong. *Theory of ground vehicles.* J. Wiley, New York, N.Y., 2nd edition, 1993.

[43] 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.

[44] 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.

[45] 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.

[46] Bahareh Ghotbi, Francisco González, József Kövecses, and Jorge Angeles. *Mobility Assessment of Wheeled Robots Operating on Soft Terrain*, pages 331–344. Springer International Publishing, Cham, 2016.

[47] R. W. Hamming. *Numerical Methods for Scientists and Engineers.* McGraw-Hill, 2nd edition, 1973.