

Development of a novel neural network-based control scheme for autonomous farming vehicles

Zeinab Sobhanigavgani

Master of Engineering

Electrical Engineering Department

McGill University

Montreal, Quebec

2019-12-01

A thesis submitted to McGill University in partial fulfilment of
the requirements for the degree of Master of Engineering

© Zeinab Sobhanigavgani, 2019

DEDICATION

This thesis is dedicated to my dearest husband for constantly being by my side, for being my best friend and critique, and to my parents who have supported me at every moment of my life.

ACKNOWLEDGEMENTS

I gratefully acknowledge the great support received from Professor Benoit Boulet, my supervisor, who tirelessly navigated and helped me through the path. I would like to thank Peiman Karimi Eskandary, Mario Landry, Marc-Antoine Legault, and Elmec Inc. for their support and collaboration, and InnovÉE for supporting this project. I gratefully acknowledge that this work was supported by Mitacs through the Mitacs Accelerate program.

ABSTRACT

Agriculture industries call upon autonomous tractors to speed up the process of farming tasks. Although there has been a lot of research on autonomous vehicles, this research mostly concerns road vehicles, which operate in a controlled road environment.

This thesis is a part of the ongoing Arion project, from Institut du Véhicule Innovant (IVI). The ultimate goal of the Arion project is to develop an autonomous farming vehicle, which can operate independently and safely. This research is dedicated to designing a robust controller to maintain the accuracy of the autonomous farming vehicle path on different types of soil.

The autonomous farming vehicle designed at IVI is an electrical four-wheel-drive vehicle. In this study, the dynamics equations of the vehicle are derived. The model used here is a four-wheel model, with two steerable wheels in the front. Moreover, realistic terramechanics equations, regarding the interaction between the soil and the wheels, are established. The equations of terramechanics are then solved offline, to be used as a lookup table online. This method will increase the efficiency, as the equations need to be solved numerically, and are computationally expensive. The lookup tables are created for four different soil types: Clay, Loam, Sand, and Sandyloam.

We introduce a novel scheme for the control of the vehicle. The system contains two controllers: a PID controller for longitudinal speed, and a Model Predictive Controller (MPC) for steering angle. In this scheme, a neural network is used to

classify the soil type, and to act as a switch to adapt the MPC controller to the new soil type. The neural network is trained on physical features that can be measured with inexpensive sensors on the vehicle, such as velocity, slip angle, slip ratio, and wheel's sinkage in the soil.

The performance of the controller for each soil type is evaluated independently, as well as the performance of the vehicle operating on a field with various soil types. The results show a considerable improvement in accuracy when the proposed controller is used.

RÉSUMÉ

L'industrie agricole réclame des tracteurs autonomes pour accélérer le processus des tâches agricoles. Bien que de nombreuses recherches aient été menées sur les véhicules autonomes, ces recherches concernent principalement les véhicules routiers, qui fonctionnent dans un environnement routier contrôlé.

Cette thèse fait partie du projet Arion en cours de l'Institut du véhicule innovant (IVI). Le but ultime du projet Arion est de développer un véhicule agricole autonome, capable de fonctionner de manière autonome et en toute sécurité. Cette recherche est dédiée à la conception d'un contrôleur robuste pour maintenir la précision de la trajectoire du véhicule agricole autonome sur différents types de sol.

Le véhicule agricole autonome conçu par l'IVI est un véhicule électrique à quatre roues motrices. Dans cette étude, les équations dynamiques du véhicule sont établis. Le modèle dynamique utilisé est le modèle à quatre roues, avec deux roues directrices orientables à l'avant. De plus, les équations de la mécanique des sols, concernant l'interaction entre le sol et les roues, sont développées. Les équations de la mécanique des sols sont alors résolues hors ligne, pour servir de table de recherche en ligne. Cette méthode augmentera l'efficacité, car les équations doivent être résolues numériquement et sont coûteuses en calculs. Les tables de recherche sont créées pour quatre types de sol, l'argile, le terreau, le sable et le terreau sablonneux.

L'auteur propose une nouvelle architecture pour le contrôle du véhicule. Le système contient deux contrôleurs: Un PID pour le contrôle de la vitesse longitudinale et un contrôleur à modèle prédictif pour le contrôle de la direction. Dans cette

architecture, un réseau de neurones est utilisé pour classifier le type de sol et pour servir de commutateur pour adapter le contrôleur MPC au nouveau type de sol. Le réseau de neurones est entraîné en utilisant des variables physiques, qui peuvent être mesurées avec des capteurs peu coûteux sur le véhicule, tels que la vitesse, l'angle de glissement et le taux de glissement, ainsi que l'enfoncement des roues dans le sol.

Par la suite, la performance du contrôleur est évaluée indépendamment pour chaque type de sol, aussi que la performance du véhicule dans une ferme avec un sol mixte. Les résultats montrent une amélioration considérable de la précision lorsque le contrôleur proposé est utilisé.

TABLE OF CONTENTS

DEDICATION	ii
ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
RÉSUMÉ	vi
LIST OF TABLES	x
LIST OF FIGURES	xi
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	2
1.3 Literature Review	4
1.3.1 Types of Controllers for Autonomous Vehicles	6
1.3.2 Terrain Classification	9
2 Dynamics and Terramechanics	12
2.1 Dynamics	12
2.1.1 Vehicle Dynamics	12
2.1.2 Wheel Dynamics	16
2.1.3 Defining Trajectory and Error calculation	17
2.1.4 Vehicle Specifications	18
2.2 Terramechanics	19
2.2.1 Longitudinal Force Generation	22
2.2.2 Lateral Force Generation	27
2.2.3 Combined lateral and longitudinal	30
2.2.4 Parameters	32

3	Controller Design	34
3.1	Longitudinal Control	34
3.2	Lateral Control	35
3.3	Neural Network	42
4	Results	52
4.1	Clay	52
4.2	Loam	57
4.3	Sand	61
4.4	Sandyloam	64
4.5	Mixed Soil	67
5	Conclusion and Future Work	73
5.1	Conclusion	73
5.2	Future Work	74
	References	74

LIST OF TABLES

<u>Table</u>	<u>page</u>
2-1 Vehicle Specifications	19
2-2 Soils Parameters [50]	33
3-1 Linearization Times	40
3-2 MPC Parameters	40
3-3 Features of the input vector	44
3-4 Average Accuracy for Train and Validation sets	47
3-5 Average Accuracy for Train and Validation sets	50

LIST OF FIGURES

<u>Figure</u>	<u>page</u>
2-1 Vehicle Schematic Diagram	13
2-2 Left: Side view ; Right: Front view of the car and normal forces . . .	16
2-3 Wheel Dynamics	17
2-4 Wheel stresses and variables	22
2-5 Velocity at angle θ	23
2-6 Normal Stress as a function of θ	25
2-7 Left: Front view ; Right: Top view of the wheel	27
3-1 The scheme of the control system	36
3-2 Neuron and Perceptron	42
3-3 Neural Network	43
3-4 Common activation functions	43
3-5 The architecture of the neural network	46
3-6 Cross-Entropy Loss per epoch during the training for Train, Validation and Test sets	47
3-7 Confusion Matrix for train and validation sets	48
3-8 Confusion Matrix test and overall sets	48
3-9 The architecture of the neural network	49
3-10 Cross-Entropy Loss per epoch during the training for Train, Validation and Test sets	50
3-11 Confusion Matrix for train and validation sets	51

3-12	Confusion Matrix test and overall sets	51
4-1	Vehicle path on Clay	53
4-2	MPC controller output: Steering angle of the front wheel	53
4-3	Heading reference for the S-shape path	54
4-4	Errors as feedback to the MPC controller	55
4-5	PID controller output: wheel torque	55
4-6	Longitudinal speed	56
4-7	Class prediction	56
4-8	Vehicle Path on Loam	57
4-9	MPC controller output: Steering angle of the front wheel	58
4-10	Errors as feedback to the MPC controller	58
4-11	PID controller output: wheel torque	59
4-12	Longitudinal speed	59
4-13	Class prediction	60
4-14	Vehicle Path on Sand	61
4-15	MPC controller output: Steering angle of the front wheels	61
4-16	Errors as feedback to the MPC controller	62
4-17	PID controller output: wheel torque	63
4-18	Longitudinal speed	63
4-19	Class Prediction	63
4-20	Vehicle Path on Sandyloam	64
4-21	MPC controller output: Steering angle of the front wheels	64
4-22	Errors as feedback to the MPC controller	65

4-23 PID controller output: torque of each wheel	66
4-24 Longitudinal speed	66
4-25 Class Prediction	66
4-26 Vehicle Path on Mixed Soil	67
4-27 MPC controller output: Steering angle of the front wheels	68
4-28 Errors as feedback to the MPC controller	68
4-29 PID controller output: wheel torque	69
4-30 Longitudinal speed	69
4-31 Class prediction	70
4-32 Left: Path of the vehicle with and without NN Controller; Right: Zoomed view of the part of the path in the Rectangle	70
4-33 Lateral deviation of the vehicle along the path	71
4-34 MPC feedback errors at the beginning of the simulation	72

Chapter 1

Introduction

1.1 Motivation

Autonomous agriculture has many benefits, especially for the developing world, where many countries have to deal with hunger over food shortage. Autonomous tractors and automatic planting systems have high accuracy, which may result in a substantially improved return on investment for farmers, making food planting more economical. Moreover, the tractors can collect information on soil conditions, which can lead to improved maintenance of the crops, prevent blights, and achieve higher efficiency and higher food quality. Autonomous tractors can increase farming hours per day, as they can work even after dark, which will speed up the process of farming tasks. Autonomous farming vehicles can disrupt agriculture, and provide a solution to world hunger.

Although there has been a lot of research and work on autonomous vehicles, this research mostly concerns road vehicles. Autonomous cars, designed to operate in the controlled environment of the road, are now starting to penetrate the market. However, autonomous farming vehicles need to operate in a largely unpredictable and very different environment. Looking at the developments of farming autonomous vehicles, we are still a long way to a completely independent autonomous vehicle as many challenges need more research to overcome.

Companies such as John Deere, Autonomous Tractor Corporation, Fendt, and Case IH, are currently developing autonomous farming vehicles. John Deere has developed an autonomous tractor that follows a path, defined by the farmer, although an operator should always be present inside the cabin. Autonomous Tractor Corporation (ATC) developed an electric autonomous vehicle, which initially was designed to follow another tractor, driven by an operator. Fendt applied the same idea for its autonomous vehicle, to follow a leader tractor. Case IH also developed its tractors based on the same idea; however, in 2016, they presented a concept of a completely independent autonomous vehicle, which is still under research and development.

The ultimate goal of the Arion project is to develop an autonomous farming vehicle, which can operate independently and safely. For this purpose, A robust controller is required to maintain the accuracy of the vehicle path on different types of soil and condition. The first step to achieve this goal is for the vehicle to understand the soil and soil reaction to the wheel movement. Computational efficiency is key factor in solving the interaction equations, as the operations are occurring mostly in real time.

1.2 Objectives

One of the challenges for autonomous farming vehicles is the variety of terrains the vehicle should be able to drive on. This fact leads us to one of the most important challenges in the control of these systems, which is the estimation of the soil type, and the interaction forces between the soil and the tires. The type of soil, the humidity of the soil, and the condition of terrain (e.g., snow-covered) will greatly

affect the controller performance. Moreover, the vehicle should also be able to drive on asphalt. The mechanics of the interaction between the tires and terrain are completely different on soil and asphalt. Therefore, finding the terrain type enables us to adapt the vehicle and/or the controller to different situations of the soil. This can be done through the estimation of tire forces, in the controller system, or using machine learning methods to find the soil type and humidity. The objectives of this project are to use machine learning methods to predict the soil type and to develop a speed and steering controller for the vehicle to follow a predefined trajectory with high accuracy, and adapt the controller to a different type of soil.

This thesis is organized as follows. In Chapter 2, Dynamics, the dynamics equations of the vehicle are derived, then the terramechanics equations of the soil-wheel interaction are formulated. The terramechanics equations are solved numerically for four different types of soil, and the solutions are stored in four lookup tables to be used in the simulation. The simulation of the system is built in SimulinkTM.

Chapter 3 is dedicated to the design process of the control strategy. In this chapter, the vehicle's longitudinal and steering controllers are developed. Moreover, the process of data acquisition for the neural network and the architecture of the network are covered. The performance of the neural network is evaluated for noisy and clean datasets.

In Chapter 4, the results of the simulations are explored. In this chapter, the advantages of the designed control scheme are evaluated over the typical Model Predictive Control (MPC).

1.3 Literature Review

There are many challenges associated with autonomous farming vehicles. An autonomous vehicle should be able to understand and interpret its environment, make decisions based on that, and execute the decisions. Compared to road vehicles, farming vehicles have different challenges at each step. The environment in fields, where farming vehicles are mostly working in, is highly uncertain. There are different types of objects that the vehicle should avoid, including the plant rows. The soil characteristics in one area might be different from another one in the same field. Moreover, wheel slippage could not be ignored. The vehicle should be able to work in different conditions of soil and weather. The surface of the terrain could be uneven and bumpy.

Two main different fields of research can be defined on an autonomous vehicle project.

Environment Perception: Use deep learning and other AI tools to interpret and understand the environment based on the data received from the sensors, including cameras, radar, etc. The vehicle should understand the obstacles and their positions, and regarding the dynamic obstacles, their speed and the direction of their movement. Moreover, in the agriculture fields, the vehicle should understand the type of plant and/or the condition of the soil. There is a big scope for research in this field, as the understanding of the vehicles of their surroundings is still far from perfect, and there is a huge variety of scenarios that a vehicle can encounter.

Control: Use the knowledge of the environment acquired from the previous step to decide (control) future actions of the vehicle. These actions include: move

straight, change direction, accelerate, decelerate, stop, use tools, etc. Furthermore, in case of any change in the environment including changes in the type of crop, the type and condition of the soil, the weather, the vehicle should be able to adapt itself to the new environment.

The control problem consists of two tasks. The first task is path planning, in which the path that the vehicle should follow considering the obstacles and their predicted movement should be determined. Many other constraints can contribute to the planned path such as the fact that the vehicle should not cross the plant rows or time minimization.

The second task is path tracking. Path tracking is the act of lateral and longitudinal control of the vehicle to follow the planned path in real time. In the case of agriculture vehicles, this is not an easy job, since these vehicles work in a complex environment. Agricultural vehicles often work on different types of terrain, even and uneven, or changing and unpredictable terrain ranging from asphalt to spongy topsoil in the field. Even the condition of the soil, such as humidity, can be very different which affects the dynamics of the vehicle. Add to all these the fact that the weight of the vehicle can be largely changing based on the tools it has been given for a specific task. Path tracking control in autonomous vehicles covers speed (longitudinal) control and steering (lateral) control. The steering controller is aimed to provide sufficient steering input to control the direction of the vehicle, and the speed controller provides throttle and braking input (in case of an electrical vehicle, torque) to control the speed of the vehicle and to guide the vehicle along a predefined path in the desired time.

1.3.1 Types of Controllers for Autonomous Vehicles

Researchers have developed different steering and speed control methods in the past decades for autonomous vehicles. The controllers can be divided into six categories based on the exercised tools.

Geometric and Kinematic Controller

This type of controller is developed based on the geometric and kinematic vehicle models. It is one of the most popular types of controllers used in autonomous steering control due to its simplicity and stability. However, this type of controller is not robust to the changes the tractor may encounter in the field. The most basic type of this controller, "Follow the Carrot", has been reviewed and applied by Arturo L. Rankin et al. [1] and Barton [2], and was improved by Wit [3] for a smoother controller, "Pure Pursuit". Stanley, Stanford University's autonomous vehicle that won the second DARPA Grand Challenge in 2005 [4], exerted a steering controller, named Stanley method, detailed by Hoffmann [5]. Lenain et al. [6] designed an extended kinematic model accounting for sliding effects for high accuracy path tracking of the vehicle in the presence of sliding.

Dynamic Controllers

Dynamic controllers include the dynamic model of the vehicles in their control law. Rossetter [7] in his work proposed a lane keeping strategy using the virtual force concept, using Kalman Filters to process position and yaw rate data for the controller to determine sufficient virtual forces. Fierro [8] modified the standard kinematic

controller to include a dynamic backstepping extension to the control structure. A dynamic controller requires dynamic feedbacks such as force and torque, which require expensive sensors, or alternatively, should be done with a great computational cost.

Optimal Controller

LQR is one of the most popular optimal control theories where the controller gain was determined using the linear quadratic optimization approach. Osinenko [9], developed an optimal traction control, in order to improve the efficiency of the farming vehicles, with feedback of the drive torque.

Adaptive Controller

The adaptive controller is used when the controller needs to be highly robust to changes in dynamics in certain ranges. Lucet [10] developed an extended kinematic model of the robot considering the effects of wheel-ground skidding. Based on the extended kinematic model, an adaptive and predictive controller for path tracking is developed to drive the front and rear steering angles. In the study of Fang et al. [11] the kinematic model of the vehicle was modified to account for the time-varying sliding effects in the lateral deviation, and a robust adaptive control law was designed. Other notable studies can be found in [12], and [13].

Model-Based Controller

The Model Predictive Controller (MPC) typically uses a linear or nonlinear plant model to predict the required control input for the plant. This method usually involves an optimization procedure in order to get the optimal value for the plant input. MPC is the most common controller used for autonomous vehicles. Kong [14] presents a study in which, a simplified bicycle kinematic and dynamic model was used in an MPC controller, to reduce the computational cost. Gao [15] used MPC to control the autonomous vehicle with obstacle avoidance explicitly considered in the control design, represented by a cost in the optimization problem. In a similar work, Yoon [16] developed an optimal tracking problem while avoiding collision with obstacles is formulated in terms of cost minimization under constraints. Information on obstacles is incorporated online in the nonlinear model-predictive framework as they are sensed within a limited sensing range.

Neural Network Controller

Intelligent controllers are used to make the vehicle even more robust to a wide range of disturbances and uncertainty in the system. However, the training of such a controller is still a challenge. NN predictive controller uses a neural network model of a nonlinear plant to predict future plant performance. The first step in this model predictive control is to determine the neural network plant model (system identification). Next, the plant model is used by the controller to predict future performance [17]. NARMA-L2 Control is another type of NN controller that transforms nonlinear system dynamics into linear dynamics by canceling the nonlinearities [18].

Model Reference Control uses two neural networks: a controller network and a plant model network. First, the plant model is identified by the first neural network, and then the second neural network is trained as a controller such that the plant output follows the reference model output [19]. Zhang [20] used the MPC controller to train a NN controller, to reduce the computational cost of MPC in realtime, while achieving the same accuracy as MPC.

Other than the controller, machine learning methods can be used for other goals. Zhu [21] used a neural network to model the vehicle behavior on sloping terrains, and then used a fuzzy logic controller based on the constructed NN vehicle model to guide the tractor along the path. Matusko [22] used the neural network to estimate tire/road friction force, which can be used to adapt the controller to different conditions of the soil. In this study, we aim to use a neural network to predict the soil type that the vehicle is traversing, and use this information to adapt the controller to different types of soils.

1.3.2 Terrain Classification

There have been many research studies towards terrain classification for autonomous ground vehicle or exploratory rovers. Most previous researchers used vision data to predict the soil type of the terrain that the vehicle is traversing. Bellutta from CalTech classified the terrain into classes of "green vegetation, dry vegetation, soil/rock, and outliers" based on color [23]. Later on, they performed the classification using other visual features such as texture and shape beside color [24].

Howard and Seraji [25] presented a technique for realtime terrain characterization for a field mobile robot using a vision system and artificial neural networks.

The terrain characteristics including roughness, slope, discontinuity, and hardness are extracted from image data and are represented in a fuzzy logic framework.

Kim [26] used vision data to classify the terrains into four groups and estimated the friction coefficient.

Methods involving vision and image will have a problem identifying soil type in bad situations; for example, when the weather is foggy, or when a layer of snow has covered the ground, the images are not sufficient for an accurate prediction.

Another widely used method to estimate the terrain type is based on the vibration of the vehicle caused by soil-wheel interaction. In Brooks and Iagnemma's method [27], vibrations are measured using an accelerometer mounted on the vehicle. The classifier identifies the terrain class to be sand, gravel, or clay [28].

In related work, Valada [29] used a microphone as a sensor to measure the acoustic sounds of a mobile robot on different terrains, and used a deep convolutional neural network to classify the terrain.

However, methods based on only vibrations have their own difficulties. In the case of bumpy grounds, the vibrations caused by bumps and holes could be mistaken by the vibration caused by the nature of the soil. Moreover, these methods could be applied to a limited number of classes.

Combining different methods will help us overcome the limitations of each method. In another work by Iagnemma [30], a combination of vision and auditory sensors was used. First, the upcoming terrain was predicted based on image

data. Then, the terrain parameters were estimated online based on wheel-soil interaction analysis. In the end, based on the auditory data from the wheel-soil contact signature, the current terrain could be classified.

Weiss [31] presented a terrain classification method which fuses predictions based on vision data with predictions made by a vibration-based method. Based on colors in the image data, the robot can predict the terrain in front of it, and then it will verify the prediction using the vibration data of the soil-wheel interaction when the robot is traversing the classified area.

Wang [32] used Laser Measurement System (LMS) to observe anomalies in surface reflection properties of terrains. The data were classified using the Support Vector Machine (SVM) classifier.

In a more recent work done by Gonzalez and Iagnemma [33], first, the slip was estimated using physical features including wheel torque and velocities. Then, a deep network was trained on images from NASA's Planetary Data System, MSL Curiosity rover to classify the terrain into asphalt, sand, grass, gravel, mars ground and pavement.

Inspired by the research done on this matter, we propose a new method for terrain classification. In this method, the physical data such as car velocity, wheel angular velocities, and slip ratios, measured with inexpensive sensors, are used to classify the terrain. Then, the information is used to adapt the MPC controller for steering control.

Chapter 2

Dynamics and Terramechanics

2.1 Dynamics

To design a control system and to simulate the behavior of the system, dynamic models of the vehicle and the wheels are essential.

2.1.1 Vehicle Dynamics

In this thesis, we use a double-track model to account for differences in the wheels' forces. In Figure 2–1 the forces acting on each wheel and the velocities are specified. F_l represents the longitudinal force and F_s represents the lateral force acting on the wheel. L_1 and L_2 are the longitudinal distances from the center of gravity to front and rear wheels, respectively. W is the width of the vehicle. v_1, v_2, v_3, v_4 are the total velocities of the centers of the wheels, v_x, v_y are the velocities of the vehicle in body-fixed coordinates $x - y - z$, ω_z is the angular velocity of the vehicle around the z -axis, and ψ is the vehicle heading. Note that $\omega_z = \dot{\psi}$. We assume that only the front wheels are steerable. δ_1 and δ_2 represent the steering angles of the front wheels.

Total forces and the moment acting on the vehicle are calculated in Equations 2.1.

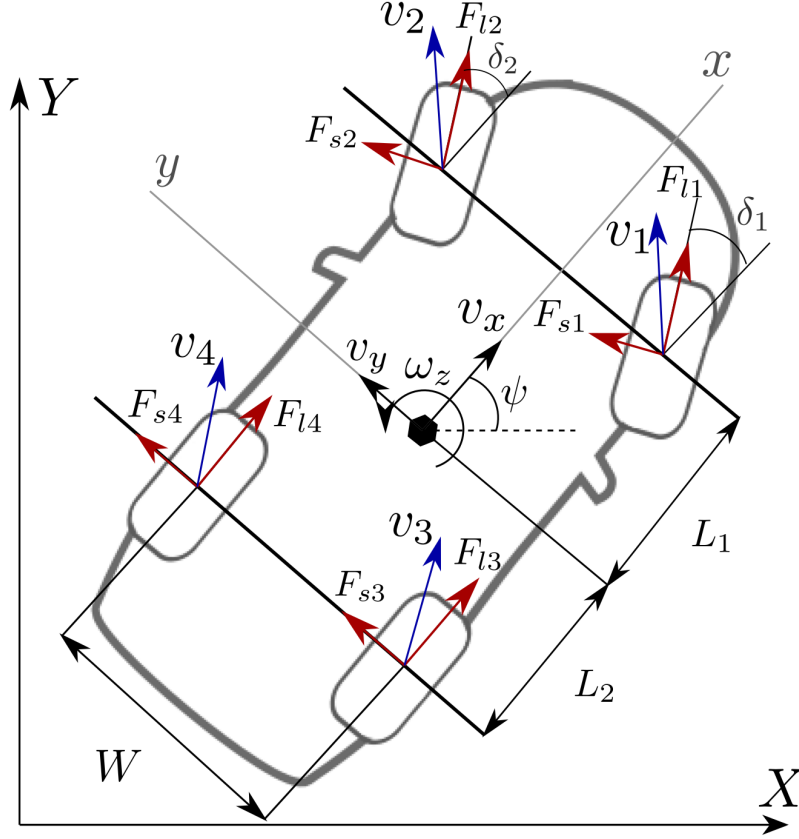


Figure 2-1: Vehicle Schematic Diagram

$$F_{xtotal} = F_{l1} \cos \delta_1 + F_{l2} \cos \delta_2 + F_{l3} + F_{l4} - F_{s1} \sin \delta_1 - F_{s2} \sin \delta_2 \quad (2.1a)$$

$$F_{ytotal} = F_{s1} \cos \delta_1 + F_{s2} \cos \delta_2 + F_{s3} + F_{s4} + F_{l1} \sin \delta_1 + F_{l2} \sin \delta_2 \quad (2.1b)$$

$$M_z = (-F_{s1} \sin \delta_1 + F_{l1} \cos \delta_1 - F_{l2} \cos \delta_2 + F_{s2} \sin \delta_2 + f_{l3} - f_{l4}) \frac{W}{2} + (F_{l1} \sin \delta_1 + F_{s1} \cos \delta_1 + F_{l2} \sin \delta_2 + F_{s2} \cos \delta_2) L_1 - (F_{s3} + F_{s4}) L_2 \quad (2.1c)$$

By applying the Euler-Newton Equations, the dynamic model of the system is obtained. The effect of the road bank and slope is considered.

$$m(a_x - v_y\omega_z) = F_{xtotal} - mgsin\Theta \quad (2.2a)$$

$$m(a_y + v_x\omega_z) = F_{ytotal} + mgsin\Phi cos\Theta \quad (2.2b)$$

$$I_{zz}\alpha_z = M_z \quad (2.2c)$$

where m is the vehicle mass, I_{zz} is the inertia, Θ is the road slope, Φ is the road bank, a_x is the longitudinal acceleration, a_y is the lateral acceleration, α_z is the vehicle's angular acceleration around the z -axis, and g is the acceleration of gravity. Since the farming vehicle does not have any suspension system, we ignore the pitch and roll dynamics of the vehicle. Having v_x , v_y and ω_z , the velocities of each wheel, $v_i = \begin{bmatrix} v_{xi} \\ v_{yi} \end{bmatrix}$ for $i = 1, 2, 3, 4$ are computed, which will be used in terramechanics, to calculate the slip ratio and slip angle.

$$v_{x1} = v_x + \omega_z \frac{W}{2} = v_{x3} \quad (2.3a)$$

$$v_{x2} = v_x - \omega_z \frac{W}{2} = v_{x4} \quad (2.3b)$$

$$v_{y1} = v_y + \omega_z L_1 = v_{y2} \quad (2.3c)$$

$$v_{y3} = v_y - \omega_z L_2 = v_{y4} \quad (2.3d)$$

In the next step, the interaction forces between the wheels and the soil should be determined to simulate the dynamic model. The interaction forces are dependent on the load on the wheels, F_{zi} for $i = 1, 2, 3, 4$. The load on the wheels is calculated by solving the equilibrium equations of the moments and forces on the center of gravity.

The tractor does not have high speed and acceleration, therefore the effect of load transfer can be ignored. We assume $F_{z1} = F_{z2}$ and $F_{z3} = F_{z4}$. From equilibrium,

$$(F_{z1} + F_{z2})L1 = (F_{z3} + F_{z4})L2 \quad (2.4a)$$

$$mg = F_{z1} + F_{z2} + F_{z3} + F_{z4} \quad (2.4b)$$

Solve the Equations 2.4a and 2.4b,

$$F_{z1} = Mg \frac{L2}{L1 + L2} = F_{z2} \quad F_{z3} = Mg \frac{L1}{L1 + L2} = F_{z4} \quad (2.5)$$

This model for computing the load on the wheels is simplified, since it does not consider the effect of cornering, road bank and slope. A more complete model can be formulated solving Equations 2.6a-2.6f. See Figure 2-2. In this model, considering that the vehicle has little or no acceleration mostly, we ignore the effects of acceleration and load transfer.

$$F_{x1} + F_{x2} + F_{x3} + F_{x4} = mgsin\Theta \quad (2.6a)$$

$$F_{y1} + F_{y2} + F_{y3} + F_{y4} = mgcos\Theta sin\Phi \quad (2.6b)$$

$$F_{z1} + F_{z2} + F_{z3} + F_{z4} = mgcos\Theta cos\Phi \quad (2.6c)$$

$$(F_{z2} + F_{z3} - F_{z4} - F_{z1})\frac{W}{2} + (F_{y1} + F_{y2} + F_{y3} + F_{y4})h = 0 \quad (2.6d)$$

$$(F_{z1} + F_{z2})L1 - (F_{z3} - F_{z4})L2 + (F_{x1} + F_{x2} + F_{x3} + F_{x4})h = 0 \quad (2.6e)$$

$$F_{z2} - F_{z1} = F_{z3} - F_{z4} \quad (2.6f)$$

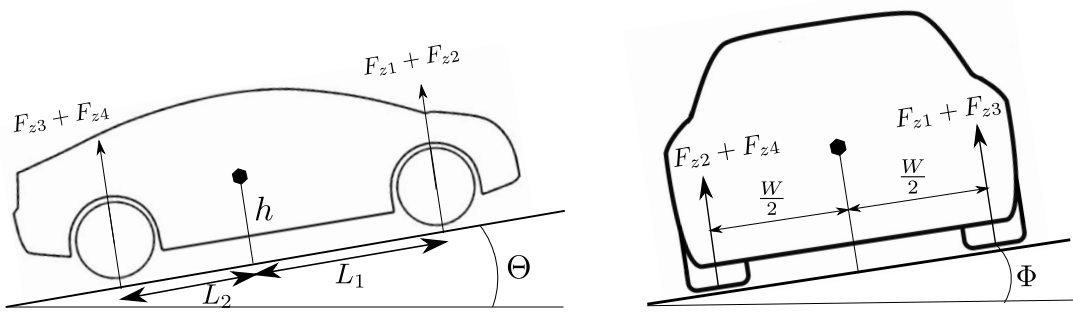


Figure 2-2: Left: Side view ; Right: Front view of the car and normal forces

where, h is the height of C.G. We obtain,

$$F_{z1} = \frac{1}{2} \left[(mg \cos \Theta \cos \Phi) \frac{L_2}{L_1 + L_2} - mg \sin \Theta \frac{h}{L_1 + L_2} + mg \sin \Phi \cos \Theta \frac{h}{W} \right] \quad (2.7a)$$

$$F_{z2} = \frac{1}{2} \left[(mg \cos \Theta \cos \Phi) \frac{L_2}{L_1 + L_2} - mg \sin \Theta \frac{h}{L_1 + L_2} - mg \sin \Phi \cos \Theta \frac{h}{W} \right] \quad (2.7b)$$

$$F_{z3} = \frac{mg \cos \Theta \cos \Phi}{2} - F_{z1} \quad (2.7c)$$

$$F_{z4} = \frac{mg \cos \Theta \cos \Phi}{2} - F_{z2} \quad (2.7d)$$

2.1.2 Wheel Dynamics

For the simulation of the vehicle, the wheel dynamics are also important. Having the force and torque acting on the wheel, one can obtain the angular velocity of the wheel, which is used in the computation of slip ratio. Based on Figure 2-3 we have,

$$I_w \alpha_w = T - F_l R \quad (2.8)$$

where, T is the wheel torque, α_w is the wheel angular acceleration, and I_w is the wheel inertia, which is $\frac{m_w R^2}{2}$. In this formula, m_w is the mass of the wheel, and R is the radius.

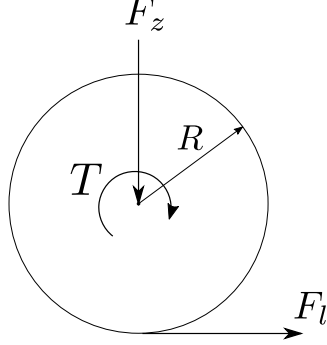


Figure 2-3: Wheel Dynamics

2.1.3 Defining Trajectory and Error calculation

The desired trajectory is defined by the desired longitudinal vehicle velocity, v_{des} , and the desired heading rate, $\dot{\psi}_{des}$. The trajectory in $X - Y - Z$ coordinate is formulated as,

$$\dot{X}_{des} = \cos\psi_{des}v_{des}$$

$$\dot{Y}_{des} = \sin\psi_{des}v_{des}$$

The trajectory that the vehicle is traversing, based on computed v_x , v_y and ω_Z is as follows:

$$\dot{X} = \cos\psi v_x - \sin\psi v_y$$

$$\dot{Y} = \sin\psi v_x + \cos\psi v_y$$

The longitudinal, lateral and heading errors, e_ψ , $e_{\dot{x}}$, and $e_{\dot{y}}$ in $x-y-z$ coordinate is computed.

$$\begin{aligned} e_\psi &= \psi_{des} - \psi \\ e_{\dot{x}} &= v_{des}\cos(\psi_{des} - \psi) - v_x \\ e_{\dot{y}} &= v_{des}\sin(\psi_{des} - \psi) - v_y \end{aligned} \tag{2.9}$$

The lateral deviation, e_y , the lateral velocity deviation $e_{\dot{y}}$, and heading deviation, e_ψ are used as the feedback for the MPC controller, and $e_{\dot{x}}$ is used as the feedback for PID controller.

2.1.4 Vehicle Specifications

In the Arion project, the farming vehicle is designed, and a prototype has been built. The vehicle specifications used in the dynamics model, are listed in Table 2–1. Note that, all wheels have the same radius. In this study, we assume the road slope and bank angles are zero.

Parameter	Symbol	Value	Unit
Distance from front wheels to C.G.	L_1	1.15	m
Distance from rear wheels to C.G.	L_2	1.28	m
Vehicle Width	W	1	m
Height of C.G.	h	0.8613	m
Wheels Radius	R	0.43	m
Mass of the vehicle	m	1000	Kg
Mass of the Wheels	m_w	29	Kg
Inertia of the vehicle around Z axis	I_{zz}	1100	$Kg.m^2$
Standard gravity	g	9.8	m/s^2

Table 2-1: Vehicle Specifications

2.2 Terramechanics

The main objective of terramechanics is to calculate forces and moments interacting between the wheel and the terrain. Modeling this interaction is a key factor for evaluating vehicle performance, thus simulation and controller design. Pacejka [34] developed a formulation named Magic Formula which can model interaction forces between wheel and hard terrain. However, we cannot use this model directly for the vehicle on soft soil. Deformation of the soil, different characteristics of different soil types and conditions, and uncertainties that exist in its parameterization make the modeling of tire-soil interaction a very complex problem.

Researchers have developed a variety of models to formulate the interactions. The models can be categorized into three groups: 1) empirical models, 2) physics-based models, and 3) semi-empirical models. These models will result in different accuracy and computational cost; therefore, based on the application, one of the methods should be chosen.

The empirical models are completely based on laboratory or field experiments. A function is fitted to the data gathered for a soil type, and a look-up table is created. In the WSE VCI model developed by Army Engineer Waterways Experiment Station [35], Cone Index was used as a measure for terrain interaction forces. In a similar approach, Wismer and Luth [36] proposed a model to correlate wheel torque, motion resistance, tractive and net pull efficiency to wheel load, soil strength, tire geometry, and tire deflection. This model introduced the effect of wheel slip on tire force. STIREMODE [37] was developed by Systems Technology Inc. for on-road vehicles and then extended to off-road condition by applying Metz [38] shaping functions. In this model, longitudinal and lateral force function vs slip was introduced.

The empirical models are simple tools to evaluate wheel behavior on the soil in a similar condition to the experiment. However, outside the scope of the tests, these models are not accurate, and thus not appropriate to use in real world applications.

Physics-based models use physical principles and analytical methods to model tire-soil interaction and structure. VTIM (Vehicle Terrain Interaction Model) proposed by Madsen [39], considers the deformation of both the tire and the soil. The model combines the 3D tire model that consists of a lumped mass, spring, and damper system with analytical soil mechanics. The model was developed for use in realtime.

Another method for physics-based modeling is DEM (Discrete Element Method). The main concept of DEM is to consider the soil as a system of discrete particles and model the interaction of each particle individually. [40, 41]

One alternative numerical model to capture more details of vehicle soil interaction is the Finite Element Method (FEM). Past advancements in computational resources made FEM a practical tool to analyze complex systems. Many researchers utilized FEM to model soil-tire interactions [42–47]. Physics-based models are very accurate, but also have high computational cost.

Semi-empirical models are the most commonly used models to use in a full-vehicle simulation and control design. These models combine experimental measurements, empirical formulation, and analytical methods, to have accurate and computationally efficient modeling of the interaction.

In this thesis, we use Chan and Sandu’s [48] revised model for wheel-soil longitudinal and lateral interaction. The following assumptions have been made in this study considering the application.

1. The soil is deformed.
2. The tire is rigid.
3. The tire inflation pressure and the temperature remain constant during the entire simulation.
4. The tire remains in contact with the ground at all times.

Figure 2–4 shows a schematic of the wheel and the variables used in this modeling. The interaction between a wheel and soil will result in the normal and tangential stress field under the wheel. The forces and moments acting on the wheel are the integral of these stresses over the contact patch. The variables in the figure, are as follows: R is the radius of the wheel, ω_r is the angular velocity, v_{xr} is the longitudinal velocity, θ_r is the exit angle, θ_e is the entry angle, θ_m is the angle where the

maximum normal stress will happen, z is the sinkage of the wheel into the soil, σ_n is the normal stress and τ is the shear stress.

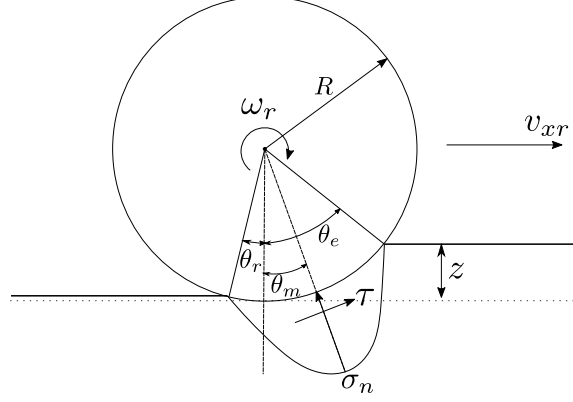


Figure 2-4: Wheel stresses and variables

2.2.1 Longitudinal Force Generation

Janosi and Hanamoto [49] provided a formulation to calculate the shear stress using the Mohr-Coulomb failure criterion and the Janosi-Hanamoto relation:

$$\tau_x(\theta) = (c + \sigma_n(\theta)\tan\phi)(1 - e^{-\frac{j_x(\theta)}{K_x}}) \quad (2.10)$$

where σ_n is the normal stress, c is the soil cohesion, ϕ is the internal friction angle, j is the shear displacement, K_x is the shear deformation modulus. To use this relation, we need to calculate the shear displacement of the soil underneath the wheel and the normal stress. A key variable for analysis of the interaction is the wheel slip ratio, which has a direct effect on the traction. The slip ratio is defined as:

$$s = \frac{R\omega_r - v_{xr}}{\max(R\omega_r, v_{xr})} \quad (2.11)$$

In this definition, during acceleration, the slip ratio is positive, and during braking, it is negative.

Wong [50, 51] demonstrated that the shear displacement for any arbitrary angle θ on the wheel can be calculated using the tangential interface velocity of the wheel. The absolute velocity of each point on the contact patch can be expressed as:

$$V_x = R\omega_r - v_{xr}\cos\theta = R\omega_r(1 - (1 - s)\cos\theta) \quad (2.12)$$

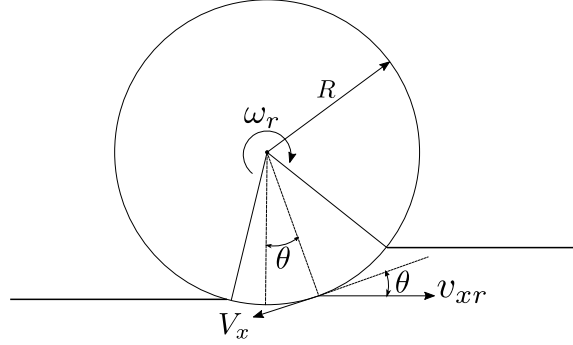


Figure 2-5: Velocity at angle θ

Therefore, the shear displacement can be calculated by integrating the surface velocity over time.

$$\begin{aligned} j_{dx} &= \int_0^t V_x dt = \int_0^t R(1 - (1 - s)\cos\theta)\omega_r dt = R \int_{\theta}^{\theta_e} (1 - (1 - s)\cos\theta) d\theta \\ &= R[(\theta_e - \theta) - (1 - s)(\sin\theta_e - \sin\theta)] \end{aligned} \quad (2.13)$$

Bekker [52] developed a formulation to express the pressure as a function of soil empirical parameters and tire sinkage.

$$p = \left(\frac{k_c}{b} + k_\phi \right) z^n \quad (2.14)$$

where k_c is the soil cohesion related parameter, k_ϕ is the friction angle related parameter, n is the sinkage index, and these three parameter will be obtained with sinkage plate tests. b is the contact patch width, and z is the sinkage.

Later, Wong and Reece [51] presented a revised and more accurate version of the normal stress-sinkage relation.

$$p = (ck'_c + \frac{\gamma bk'_\phi}{2}) \left(\frac{z}{b} \right)^n \quad (2.15)$$

where k'_c is the soil cohesion related parameter, k'_ϕ is the friction angle related parameter, γ is the unit weight of the soil, and c is the soil cohesion. In this equation, k'_c and k'_ϕ are non-dimensional parameters.

Computation of radial tire stresses by Wong and Reece also shows that the maximum normal stress does not happen directly under the center of the wheel. It is shifted forward and will move further with the increase in slip ratio. The distribution of normal stress under the wheel is shown in Figure 2-6. θ_N is the point where the maximum normal stress happens. As seen in the figure, the distribution of normal stress has different curves in the ranges θ_r to θ_N and θ_N to θ_e .

Wong developed a piece-wise formulation to calculate normal stress in the two regions. By substituting the sinkage z in Equation 2.14 and Equation 2.15 with the following term, the normal stress is obtained as a function of θ .

$$z = R(\cos\theta - \cos\theta_e) \quad (2.16)$$

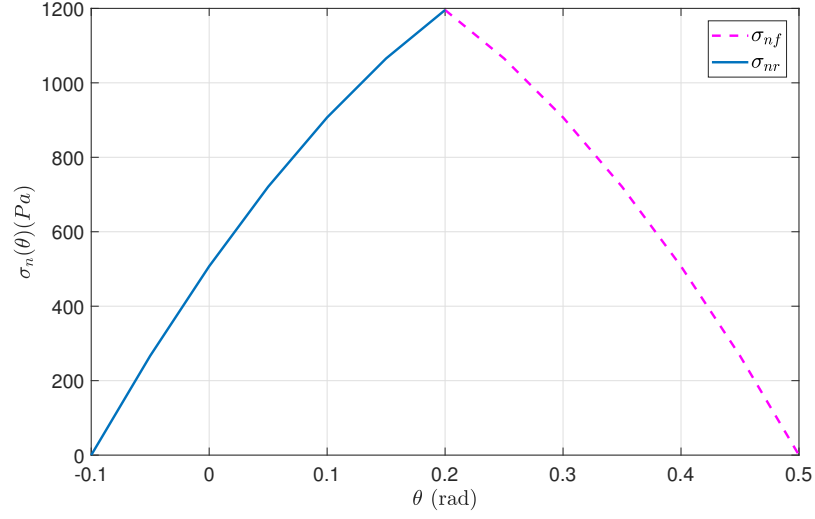


Figure 2-6: Normal Stress as a function of θ

$$\sigma_n = \begin{cases} \sigma_{nf} = C_\sigma (\cos\theta - \cos\theta_e)^n & \theta_N \leq \theta \leq \theta_e \\ \sigma_{nr} = C_\sigma \left(\cos\left(\theta_e - \left(\frac{\theta - \theta_r}{\theta_N - \theta_r}\right)(\theta_e - \theta_N)\right) - \cos\theta_e \right)^n & \theta_r \leq \theta \leq \theta_N \end{cases} \quad (2.17)$$

where,

$$C_\sigma = \begin{cases} \left(\frac{k_c}{b} + k_\phi\right) R^n & \text{Bekker} \\ \left(ck'_c + \frac{\gamma bk'_\phi}{2}\right) \left(\frac{R}{b}\right)^n & \text{Wong} \end{cases} \quad (2.18)$$

In terramechanics, Equations 2.19-2.21 provide a framework to calculate Draw-bar Pull (DP), Vertical Force (VF) and the angular Torque (T), by integrating the normal and tangential stresses over the contact patch.

$$DP = Rb \left[\int_{-\theta_r}^{\theta_e} \tau_x(\theta) \cos \theta d\theta - \int_{-\theta_r}^{\theta_e} \sigma_n(\theta) \sin \theta d\theta \right] \quad (2.19)$$

$$VF = Rb \left[\int_{-\theta_r}^{\theta_e} \tau_x(\theta) \sin \theta d\theta + \int_{-\theta_r}^{\theta_e} \sigma_n(\theta) \cos \theta d\theta \right] \quad (2.20)$$

$$T = R^2 b \left[\int_{-\theta_r}^{\theta_e} \tau_x(\theta) d\theta \right] \quad (2.21)$$

These equations are complex, and need to be solved numerically. In order to solve the equations efficiently, the following assumptions are made:

1. The tire is rigid, that means R is constant.
2. θ_r is usually small and does not vary. Therefore, we assume a constant value for θ_r . $\theta_r = -0.09$ rad (-5°), is a reasonable assumption.
3. θ_N is directly in between of θ_r and θ_e .

$$\theta_N = \frac{\theta_e + \theta_r}{2} \quad (2.22)$$

4. The normal stress can be approximated with a parabolic curve. This will allow us to use the Simpson's rule to simplify the integrals. Based on Simpson's rule, we can approximate the integral of a parabola using Equation 2.23.

$$\int_a^b F(x) dx = \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \quad (2.23)$$

After numerical computations, the accuracy of this estimation was deemed acceptable.

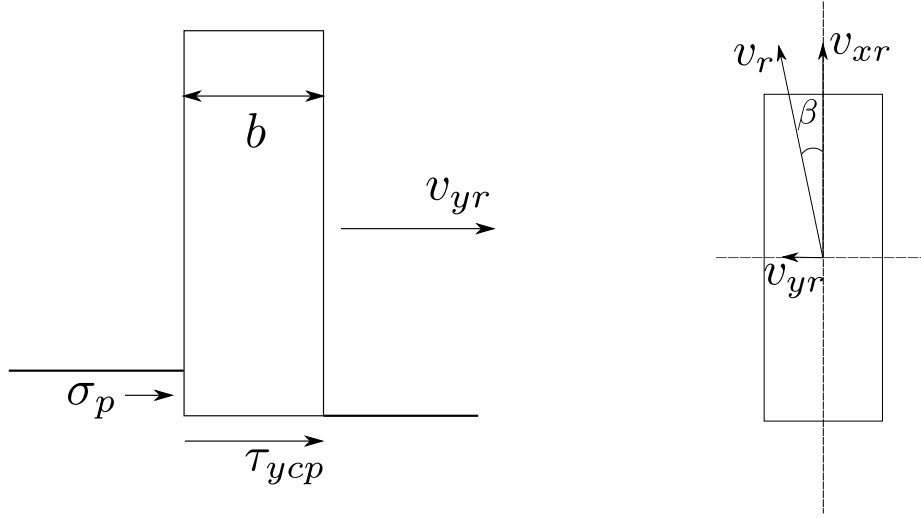


Figure 2-7: Left: Front view ; Right: Top view of the wheel

For all sets of θ_e and θ_r , the vertical force should be balanced with the wheel load. That means:

$$F_z = VF \quad (2.24)$$

Given the assumptions, we can solve the Equation 2.24 for θ_e . Then, solve the Equation 2.19 to find the drawbar pull, which is equal to F_x .

2.2.2 Lateral Force Generation

The lateral force is the summation of the forces caused by the lateral shear stress, F_{ycp} , and the bulldozing effect of the soil, F_{ybd} .

$$F_y = F_{ycp} + F_{ybd} \quad (2.25)$$

Figure 2-7 shows the variables used for lateral shear modelling.

The variables in the figure are as follows: v_{yr} is the lateral velocity, v_r is the total velocity, b is the width of the wheel, β is the slip angle, σ_p is lateral bulldozing pressure acting on the side of the wheel and τ_{ycp} is the shear stress in the lateral direction. Similarly, for the lateral shear deformation, we need to define the slip angle.

$$\beta = \tan^{-1}\left(\frac{v_{yr}}{v_{xr}}\right) \quad (2.26)$$

Hence, the lateral velocity v_y is defined as:

$$v_{yr} = v_{xr} \tan \beta \quad (2.27)$$

Similar to Equation 2.13 the lateral shear displacement can be calculated.

$$j_{dy} = \int_0^t v_{yr} dt = \int_0^t v_{xr} \tan \beta dt = \tan \beta \int_{\theta}^{\theta_e} \frac{v_{xr}}{\omega_r} d\theta = R(\theta_e - \theta)(1 - s) \tan \beta \quad (2.28)$$

Using Janosi and Hanamoto relation with lateral shear displacement,

$$\tau_{ycp}(\theta) = (c + \sigma_n(\theta) \tan \phi) \left(1 - e^{-\frac{j_y(\theta)}{K_y}}\right) \quad (2.29)$$

where k_y is the shear deformation module in the lateral direction, the force acting on the wheel that is caused by the lateral shear stress can be computed.

$$F_{ycp} = Rb \int_{-\theta_r}^{\theta_e} \tau_{ycp} d\theta \quad (2.30)$$

For analysis of the soil bulldozing effect during cornering, Schwanghart [53] proposed using the Fundamental EarthMoving Equation to calculate the resultant force. Based on Reece's [54] Fundamental EarthMoving Equation, the passive ground

resistance for a wall moving into a body of soil is composed of four terms, which represent the effects of soil density, cohesion, the surcharge on the soil surface, and adhesion between the blade and soil. The force of the bulldozing effect is computed by integrating over the soil and wall contact patch. In this approach, it is assumed that the sidewall of the wheel acts similar to the cutting blade of a bulldozer. McKyes [55] proposed the following equation to calculate the bulldozing effect using N -factors.

$$F = \gamma_s z^2 N_\gamma + cz N_c + qz N_q + c_a z N_a \quad (2.31)$$

where the N -factors are: N_γ , the soil specific weight coefficient, N_c , the soil cohesion coefficient, N_q , the soil surcharge load coefficient, and N_a , the coefficient of adhesion between soil and blade. c is the soil cohesion, q is the surcharge load from accumulated bulldozed soil, c_a is the soil-blade adhesion, z is the sinkage, and γ is the unit weight of soil. In this study, we ignore the effect of surcharge load and soil-blade adhesion. From the static equilibrium of soil and blade, the N -factors can be determined.

$$N_\gamma = \frac{(\cot \rho_w + \cot X_c) \sin(\phi + X_c)}{2 \sin(\rho_w + X_c + \phi)} \quad (2.32a)$$

$$N_c = \frac{\cos \phi}{\sin X_c \sin(\rho_w + X_c + \phi)} \quad (2.32b)$$

where ρ_w is the blade-soil angle, which is 90° most of the time, X_c is the soil failure angle, and ϕ is the soil friction angle. X_c can be approximated as, [55]

$$X_c = \frac{\pi}{4} - \frac{\phi}{2} \quad (2.33)$$

Using the assumed values, we can rewrite Equations 2.31 and 2.32 as follows with a different notation.

$$F(\theta) = D_1(cz + D_2\gamma\frac{z^2}{2}) \quad (2.34)$$

where

$$\begin{aligned} D_1 &= \cot X_c + \tan(X_c + \phi) \\ D_2 &= \cot X_c + \frac{\cot^2 X_c}{\cot \phi} \\ z &= R(\cos \theta - \cos \theta_e) \end{aligned}$$

Finally, F_{ybd} is calculated by integrating the bulldozing effect over the submerged portion of the wheel.

$$F_{ybd} = R \int_{-\theta_r}^{\theta_e} F(\theta) \cos \theta \sin \beta d\theta \quad (2.35)$$

The slip angle is introduced into the equation, as in the real world, there is no bulldozing effect when there is no lateral movement. It should be mentioned that in pure cornering situation, θ_e and θ_r are equal to θ_s which is the sinkage of the wheel in the static condition.

2.2.3 Combined lateral and longitudinal

During combined manoeuvres caused by soil shear stresses in lateral and longitudinal directions, the limits of total traction generated by soil shear stress should be taken into account. The shear strength of the soil is defined by the Mohr-Coulomb

failure criterion,

$$\tau_{max} = c + \sigma_n \tan \phi \quad (2.36)$$

Therefore, the total shear stress is bounded by the inequality in Equation 2.37.

$$\left(\frac{\tau_x}{\tau_{max}}\right)^2 + \left(\frac{\tau_{ycp}}{\tau_{max}}\right)^2 \leq 1 \quad (2.37)$$

After some simplification, the boundary curve of the inequality 2.37 is,

$$\left(1 - e^{-\frac{j_x}{K_x}}\right)^2 + \left(1 - e^{-\frac{j_y}{K_y}}\right)^2 = 1 \quad (2.38)$$

or

$$\left(1 - e^{-\frac{R[(\theta_e - \theta_a) - (1-s)(\sin \theta_e - \sin \theta_a)]}{K_x}}\right)^2 + \left(1 - e^{-\frac{R(\theta_e - \theta_a)(1-s)\tan \beta}{K_y}}\right)^2 = 1 \quad (2.39)$$

where θ_a is the point where the combination of lateral and longitudinal shear stresses are equal to the maximum shear stress allowed, and after this point, the wheels will slide. By solving Equation 2.39 numerically, θ_a is found. The new expressions for lateral and longitudinal shear stresses can be defined as,

$$\tau_x = \begin{cases} \tau_{max} \left(1 - e^{-\frac{j_x}{K_x}}\right) & \theta_a \leq \theta < \theta_e \\ \tau_{max} & -\theta_r \leq \theta < \theta_a \end{cases} \quad (2.40)$$

$$\tau_{ycp} = \begin{cases} \tau_{max} (1 - e^{-\frac{j_y}{K_y}}) & \theta_a \leq \theta < \theta_e \\ \tau_{max} & -\theta_r \leq \theta < \theta_a \end{cases} \quad (2.41)$$

Equations 2.40 and 2.41 ensure that the tangential stresses will satisfy the soil failure criterion.

2.2.4 Parameters

The aforementioned equations are complex, and need to be solved numerically. This will make this approach inefficient, as solving the equation at each timestep in realtime will add delay to the system, which is not efficient. One solution to this problem is to solve the equations offline for different soil types and ranges of possible values of load on the wheel, slip ratio and slip angle, and create lookup tables. Therefore, in real time the system will only use the look up table to find the interaction forces, base on the type of the soil, current load, slip ratio and slip angle, which is more efficient.

Four types of soil are considered in this study; Clay, Loam, Sandy Loam, and Sand. The empirical parameters of these soils are listed in Table 2–2.

Parameters \ Terrain Type	Clay	Loam	Sandy Loam	Sand	Unit
Cohesion C	7.58	0.8	2.2	1.3	kPa
Internal Friction Angle ϕ	14	37.2	39.4	31.1	degree
Shear Deformation Parameter K	2.5	3.6	6.1	1.2	mm
Sinkage index n	0.85	1	1.1	0.79	-
Soil cohesion related parameter k_c	43.08	1.37	74.6	102	kPa/ m^{n-1}
Friction angle related parameter k_ϕ	499.7	814	2082	5301	kPa/ m^n

Table 2-2: Soils Parameters [50]

Chapter 3

Controller Design

3.1 Longitudinal Control

For the control of the longitudinal movement, we use a Proportional-Integral-Derivative (PID) controller, with the feedback of measured longitudinal velocity of the vehicle. Having an integral term (I-term) is important in this controller since the speed error will accumulate during the simulation, and the vehicle will fall behind the desired trajectory, along the $x - axis$. The I-term will help eliminate this accumulated error, and achieve better accuracy on the trajectory tracking. It is worth mentioning that the longitudinal and lateral dynamics are related to each other, and the performance of one will impact the performance of the other. For example, one of the feedback errors for the lateral controller is the lateral deviation from the path. If the error along the $x - axis$ of the vehicle is large, the computed lateral error will not represent the shortest distance between the vehicle and the path. Instead, it represents the distance between the vehicle and where the vehicle is supposed to be on the path, which means $\sqrt{dx^2 + dy^2}$. This will affect the performance of the lateral controller.

3.2 Lateral Control

The most common method for steering control of a vehicle is MPC, which is a powerful tool that can deal with difficult situations. The behavior of the system, considering the equations of terramechanics, is highly nonlinear. Some simplification is possible. However, if we use a more accurate model of terramechanics instead of simplified linear relations, the system will remain nonlinear. In this study, we approximate the behavior of the system at a stable point and in different soil types with linear systems, to use as the models in the MPC controller. Therefore, we have different estimations for different types of soil. Based on each linearized system, the MPC controller is designed and tuned to have the best performance on that specific soil type. Therefore, in real time, we need to identify the soil type in order to choose the appropriate MPC controller. In the proposed structure, the neural network block's purpose is to identify the soil and act as a switch. Figure 3–1 shows a schematic block diagram of the proposed system.

MPC Configuration

Model Predictive Control (MPC) uses a model of the plant to predict the future inputs and outputs of the system. Considering this prediction for future inputs, at each time step t , the current plant state is sampled and a cost minimizing control strategy is computed under operating constraints for a specified time horizon in the future: $t + T$, in order to best follow a given trajectory.

The computed optimal move is the control action applied to the plant at time t . At the time $t + 1$, a new optimization is solved over a shifted prediction horizon.

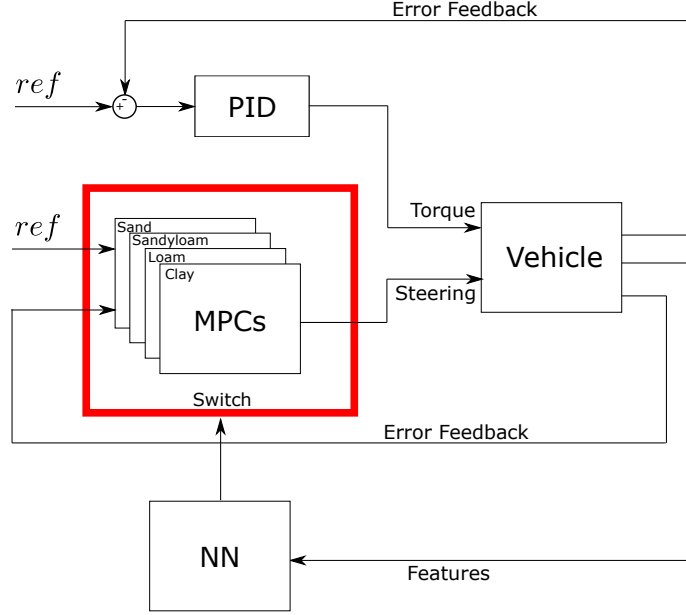


Figure 3–1: The scheme of the control system

In other words, MPC is based on iterative, finite-horizon optimization of a plant model. MPC allows the current timeslot to be optimized while taking future timeslots into account. MPC needs the dynamic model of the system to predict the future. The model should first be validated, meaning that the model should be close to the real system considering disturbance and uncertainty in the system.

The dynamic model of the system is developed in Chapter 2. The states of the system and the input are,

$$\zeta = [v_x \ v_y \ \dot{\psi} \ x \ y \ \psi \ \omega_1 \ \omega_2 \ \omega_3 \ \omega_4]^T, \quad u = \delta_1. \quad (3.1)$$

In this application, we assume δ_3 and δ_4 are zero, and δ_2 is related to δ_1 with Ackermann's relation. This means:

$$\delta_2 = \cot^{-1}(\cot\delta_1 - \frac{W}{L_1 + L_2}). \quad (3.2)$$

Consider the system,

$$\begin{aligned} \dot{\zeta} &= f_{soil}(\zeta, u) \\ \eta &= h(\zeta) \end{aligned} \quad (3.3)$$

where f_{soil} is a nonlinear function which is different for each soil type, and the output $\eta = [e_\psi, e_y, e_{\dot{y}}]$ is the vector of heading deviation, lateral deviation, and lateral velocity deviation from the desired path, obtained from Equation 2.9. In order to obtain a finite dimensional optimization problem, the discrete version of the system is defined for timestep k .

$$\begin{aligned} \zeta_d(k+1) &= f_{soil,d}(\zeta_d(k), u_d(k)) \\ \eta_d(k) &= h_d(\zeta_d(k)) \end{aligned} \quad (3.4)$$

The optimization problem at timestep k is defined as,

$$\min_{\Delta u_k \in \mathbb{R}^{H_c}} J(\zeta_k, \Delta u_k) = \sum_{i=1}^{H_p} \|\hat{\eta}_{k+i,k}\|_Q^2 + \sum_{j=0}^{H_c-1} \|\Delta u_{k+j,k}\|_R^2$$

Subject to: $\delta_{1,min} \leq u_j \leq \delta_{1,max}$ (3.5)

$$\Delta \delta_{1,min} \leq \Delta u_j \leq \Delta \delta_{1,max} \quad \text{for } j = k, \dots, k + H_c - 1$$

$$u_j = u_{j-1} + \Delta u_j$$

In this formulation, $\hat{\eta}_{k+i,k}$ is the output of the system at timestep $k+i$, calculated at timestep k , $\Delta u_{k+j,k}$ is the difference between the input of the system at timestep $k+j$ calculated at time k and the reference input at the same timestep, H_p is the prediction horizon, H_c is the control horizon, and $\Delta \delta_1$ is the rate of steering. Note that:

$$\|\cdot\|_Q^2 := (\cdot)^T Q (\cdot)$$

$$\|\cdot\|_R^2 := (\cdot)^T R (\cdot)$$

Linearization is a linear approximation of the system around an operating point. Different methods of linearization have been developed by researchers. In our application, because of the complexity of the model, we use numerical perturbation to linearize the MPC internal plant. In the perturbation method, a small perturbation is introduced into the nonlinear model and the response to this perturbation is measured. Looking back at the continuous-time system in Equation 3.3, we define a new set of the perturbed states, inputs, and outputs centered about the operating point, $[\zeta_0, u_0, \eta_0]$:

$$\begin{aligned}
\Delta\zeta(t) &:= \zeta(t) - \zeta_0 \\
\Delta u(t) &:= u(t) - u_0 \\
\Delta\eta(t) &:= \eta(t) - \eta_0
\end{aligned} \tag{3.6}$$

Therefore, the linearized state-space equations in terms of these new variables are obtained.

$$\begin{aligned}
\Delta\dot{\zeta}(t) &= A\Delta\zeta(t) + B\Delta u(t) \\
\Delta\eta(t) &= C\Delta\zeta(t) + D\Delta u(t)
\end{aligned} \tag{3.7}$$

The state-space matrices are computed as follows:

$$\begin{aligned}
A(i, j) &= \left. \frac{\partial f_i}{\partial \zeta_j} \right|_{\zeta_0, u_0} & B(i, m) &= \left. \frac{\partial f_i}{\partial u_m} \right|_{\zeta_0, u_0} \\
C(p, j) &= \left. \frac{\partial h_p}{\partial \zeta_j} \right|_{\zeta_0, u_0} & D(p, m) &= \left. \frac{\partial h_p}{\partial u_m} \right|_{\zeta_0, u_0}
\end{aligned} \tag{3.8}$$

for $i, j = 1, 2, \dots, N$ with N the number of states, $m = 1, 2, \dots, M$ with M the number of control inputs, and $p = 1, 2, \dots, P$ with P the number of outputs. In this formulation, $\dot{x}_i = f_i(\zeta, u)$, and $\eta_i = h_i(\zeta, u)$. In this study, matrix D is zero.

Using this method, we need to find a point where the system is nearly at steady state and linearize the plant around that point. Each soil has different characteristics, that means the operating point and the linearized plant are different for each soil type.

The operation points for linearization are selected at specific times in the simulations, which are listed in Table 3–1. The best value of linearization time was obtained by trial and error, to find the best model that minimizes the error.

Soil	Linearization time(s)
Clay	1.5
Loam	1
Sand	1.6
Sandyloam	1.1

Table 3–1: Linearization Times

In the next step, we tune the MPC controllers’ parameters for linearized plants of each soil type. The tuned values of Prediction Horizon, H_p (steps), Control Horizon, H_c (steps), and sampling period, t_s (seconds) are listed in Table 3–2.

Soil	H_p	H_c	t_s
Clay	15	10	0.1
Loam	15	6	0.1
Sand	12	7	0.1
Sandyloam	12	7	0.1

Table 3–2: MPC Parameters

Furthermore, the weight matrices Q , are tuned as in Equation 3.9.

$$\begin{aligned}
Q_{Clay} &= \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 3 \end{bmatrix} & Q_{Loam} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 40 & 0 \\ 0 & 0 & 4 \end{bmatrix} \\
Q_{Sand} &= \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 30 & 0 \\ 0 & 0 & 2 \end{bmatrix} & Q_{Sandyloam} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 30 & 0 \\ 0 & 0 & 3 \end{bmatrix}
\end{aligned} \tag{3.9}$$

For all soil types the input weight matrix, $R = [1]$

The constraints of the optimization problem are defined based on the physical properties and limits of the system. In this application, the wheels cannot steer more than 45° , and the steer rate is not more than 0.2 rad/s that means

$$\begin{aligned}
\delta_{1,min} &= -45^\circ = -0.8 \text{ rad} & \delta_{1,max} &= +45^\circ = 0.8 \text{ rad} \\
\Delta\delta_{1,min} &= -0.02 \text{ rad/timestep} & \Delta\delta_{1,max} &= 0.02 \text{ rad/timestep}
\end{aligned} \tag{3.10}$$

3.3 Neural Network

A neural network consists of units called perceptron or neuron. The concept of a perceptron is inspired by human brain neurons, which receive the information from one terminal and produce output from the other terminal. Figure 3–2 compares the functionality of a neuron and a perceptron [56].

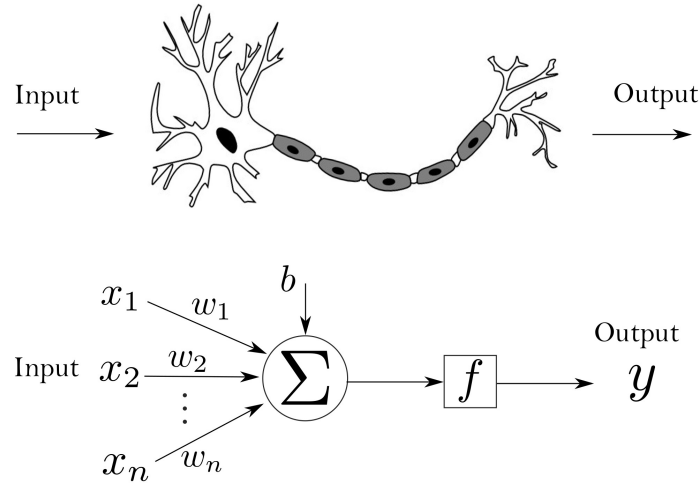


Figure 3–2: Neuron and Perceptron

The input vector $x = [x_1, x_2, \dots, x_n]$ is multiplied by the weight vector $w = [w_1, w_2, \dots, w_n]$. The bias term b will be added to the result. This weighted sum will then pass through an activation function f , to produce the output y . Note that w and b are the learnable parameters, which are determined with a learning algorithm, such that the output reaches a goal. A neural network is built up of many neurons interacting with each other. A neural network consists of an input layer, which does not modify the input and only transfers the input to the network, the output layer, which is the last layer and the output of this layer is the output of the neural network, and hidden layers, which are layers between input and output layers. The output of

each layer is the input to the next layer. Figure 3–3 shows an example of a network with two hidden layers. In this structure, the first hidden layer has five and the second hidden layer has three neurons.

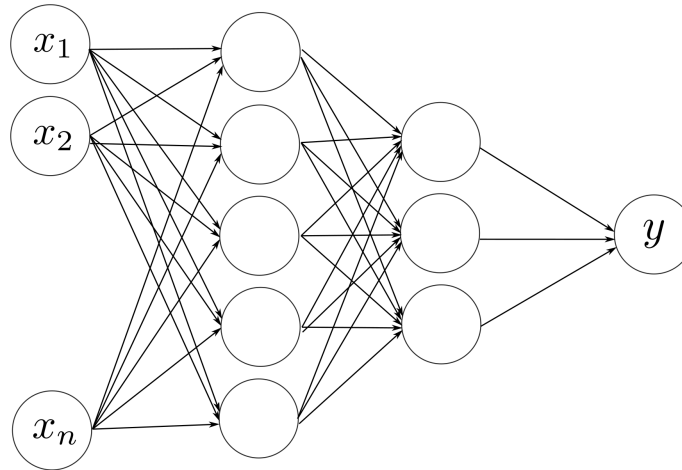


Figure 3–3: Neural Network

The activation function is chosen based on the application. Different perceptrons can have different activation functions. Three common choices for f are the sigmoid, tanh, and relu functions, shown in Figure 3–4, and defined in Eq 3.11.

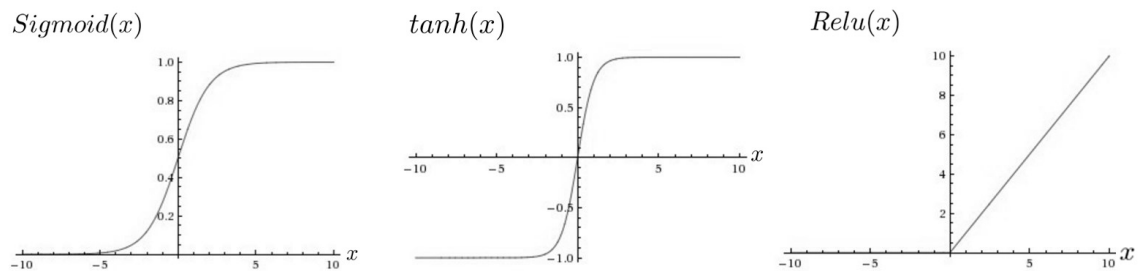


Figure 3–4: Common activation functions

$$\text{sigmoid}(x) := \frac{1}{1 + e^{-x}} \quad \text{tanh}(x) := \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad \text{Relu}(x) := \max(0, x) \quad (3.11)$$

The goal of the neural network in this study is to predict the soil type, based on the input data. The first step is to define the input vector features of the system. As mentioned in Chapter 2 on terramechanics, there is a nonlinear relation between the soil type and physical variables in the system. Inspired by terramechanics relations, we consider the physical parameters listed in Table 3–3 as features of the input vector to the system.

parameter	symbol	parameter	symbol
Angular velocity wheel 1	ω_1	Slip ratio wheel 1	s_1
Angular velocity wheel 2	ω_2	Slip ratio wheel 2	s_2
Angular velocity wheel 3	ω_3	Slip ratio wheel 3	s_3
Angular velocity wheel 4	ω_4	Slip ratio wheel 4	s_4
Vehicle longitudinal velocity	v_x	Slip angle wheel 1	β_1
Vehicle lateral velocity	v_y	Slip angle wheel 2	β_2
Vehicle angular velocity around z-axis	ω_z	Slip angle wheel 3	β_3
Wheel torque	T	Slip angle wheel 4	β_4
average of front wheels sinkage	z		

Table 3–3: Features of the input vector

The parameters should be easy to measure in the real application. The angular velocity of the wheels will be measured using encoders on the wheels. Vehicle longitudinal, lateral and angular velocity will be measured using GPS, or an IMU. Since the vehicle is electrical, we have access to torque data using the voltage of the motors in the wheels. Slip Ratios and Slip Angles are estimated using measured

velocities. For measuring the sinkage of the wheels inside the soil, we use two wide range ultra-sonic sensors mounted on a structure above the front wheels. Since the surface of the ground is not even, we need to get the average of sensor data over its range to reduce noise. As the vehicle moves on the ground, the front wheels will compact the soil, and the sinkage of the rear wheels can be misleading. Therefore, we only include the front wheels sinkage in the data.

We also use the information from four previous time-steps for more accurate classification. Therefore each input sample is a vector consists of parameters listed in Table 3–3, stacked with the same features from previous timesteps:

$$x_{(1 \times 85)} = [x \ x^1 \ x^2 \ x^3 \ x^4]$$

where x is the current timestep feature vector,

$$x_{(1 \times 17)} = [\omega_{r1} \ \omega_{r2} \ \omega_{r3} \ \omega_{r4} \ s_1 \ s_2 \ s_3 \ s_4 \ \beta_1 \ \beta_2 \ \beta_3 \ \beta_4 \ v_x \ v_y \ \omega_z \ T \ z]$$

and x^i is the feature vector from i_{th} previous timestep. The target of the classification problem is the soil type; Clay (class 1), Loam (class 2), Sand (class 3) and Sandy-loam(class 4). For each type of soil, we collect data, in different situations, to make sure the network is generalized.

First Approach: The input layer of the network has 85 neurons (number of features) and the output layer has 4 neurons (number of classes). The tuned network architecture consists of three hidden layers with 10, 5, and 10 neurons in each layer, and the activation functions for hidden layers are *tanh*, *Relu*, and *softmax* respectively, as in Figure 3–5.

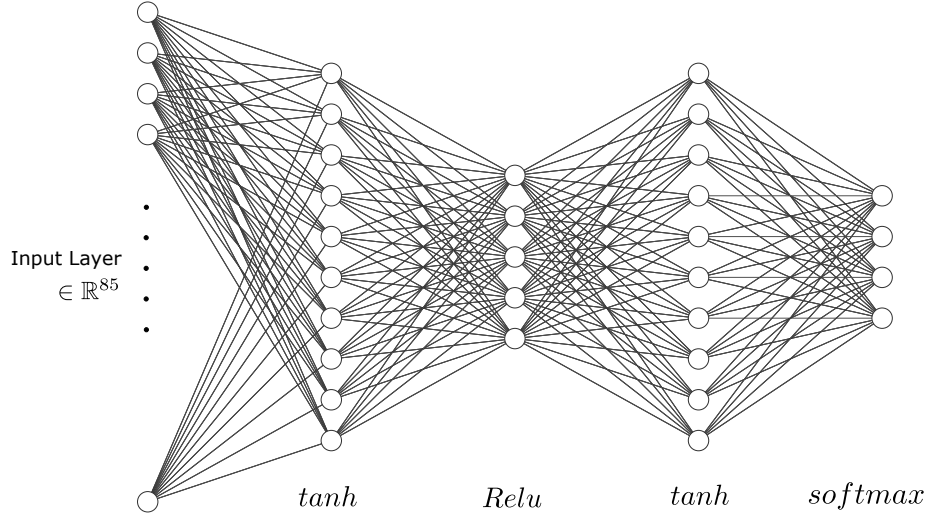


Figure 3-5: The architecture of the neural network

The network uses scaled conjugate gradient backpropagation as the training algorithm, with regularization parameter $\lambda = 0.1$, and Cross-Entropy as a measurement for loss. Cross entropy is calculated as

$$\mathcal{L} = - \sum_{i=1}^M \hat{y}_i \log(y_i)$$

where M is the number of classes, in this case, 4, $\hat{y}_i$ is the true probability of class i , and y_i is the predicted probability of class i .

Moreover, the data is standardized before training in order to have the same scale for all features. This will assure that all the features have the same contribution, regardless of their mean and variance. In this process, the values of a row are mapped between -1 and 1 range.

The dataset is split into three randomly-selected partitions to be used for training (75% of the dataset), validation (15% of the dataset) and test (10% of the dataset).

Using 5-fold Cross-Validation, the average accuracy obtained for train and validation sets are listed in Table 3-4.

	accuracy
Train set	99.94%
Validation set	99.98%

Table 3-4: Average Accuracy for Train and Validation sets

Figure 3-6 shows the performance (loss) of the network during the training process, and Figure 3-7 and 3-8 show the confusion matrices of the network. As seen in the confusion matrix, the accuracy of the network on the test set is 99.6%.

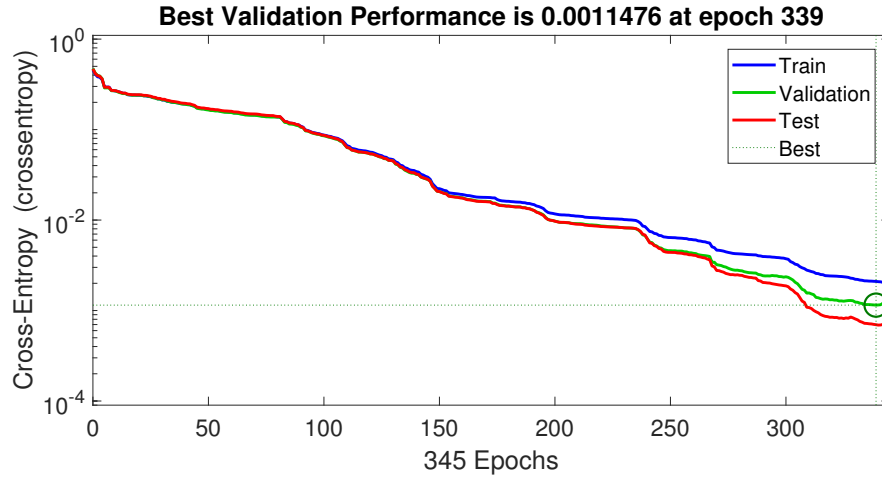


Figure 3-6: Cross-Entropy Loss per epoch during the training for Train, Validation and Test sets

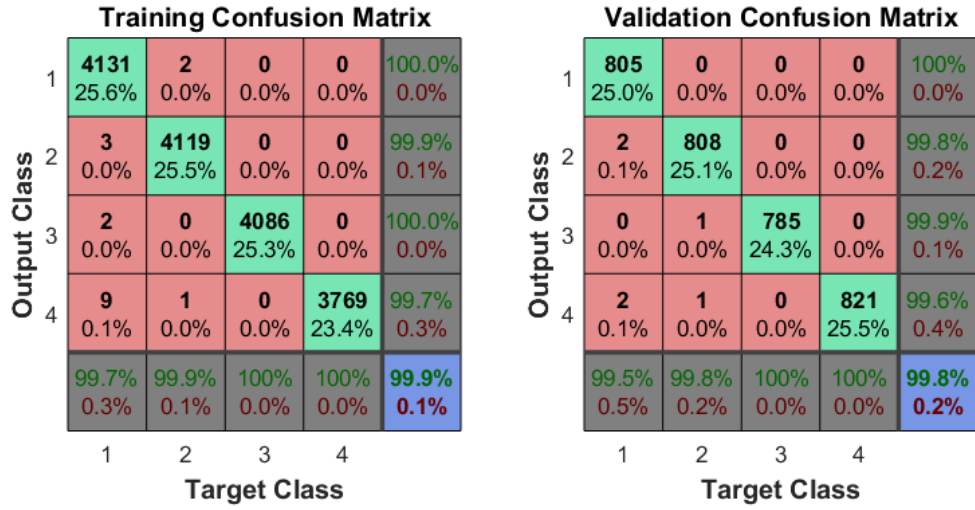


Figure 3-7: Confusion Matrix for train and validation sets

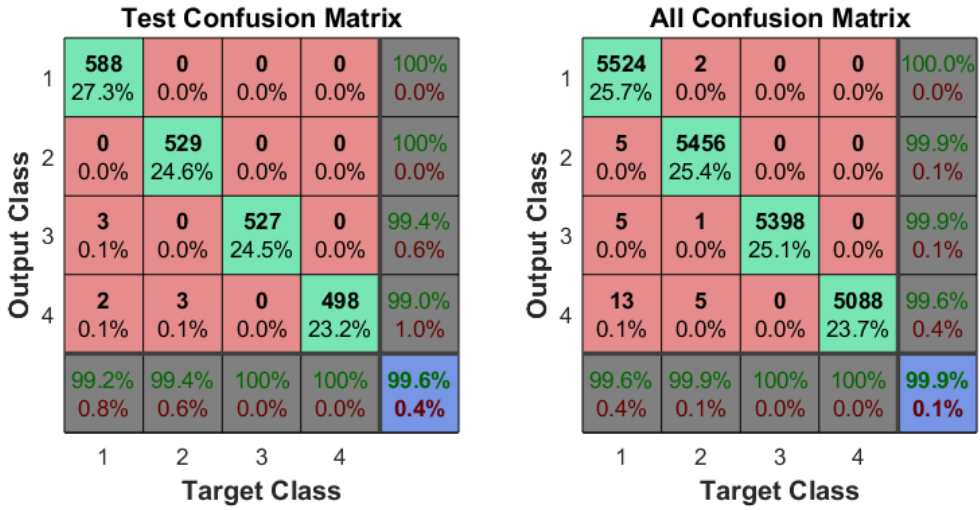


Figure 3-8: Confusion Matrix test and overall sets

Second Approach: The measured data for the previous approach comes from the simulation. However, the real data cannot be as perfect as the simulation data, due to the measurement noises. In the second approach, to have more realistic data, we induce noise with about $20dB$ Signal Noise Ratio (SNR), into the input data, then train the network with the imperfect data. In this case, the network needs to be more complex in order to handle the noise.

The new network architecture consists of four hidden layers with 12 , 15, 10 and 8 neurons in each layer, and the activation functions for hidden layers are $\tanh$, $Relu$, $Relu$, and $\tanh$ respectively, as in Figure 3–9.

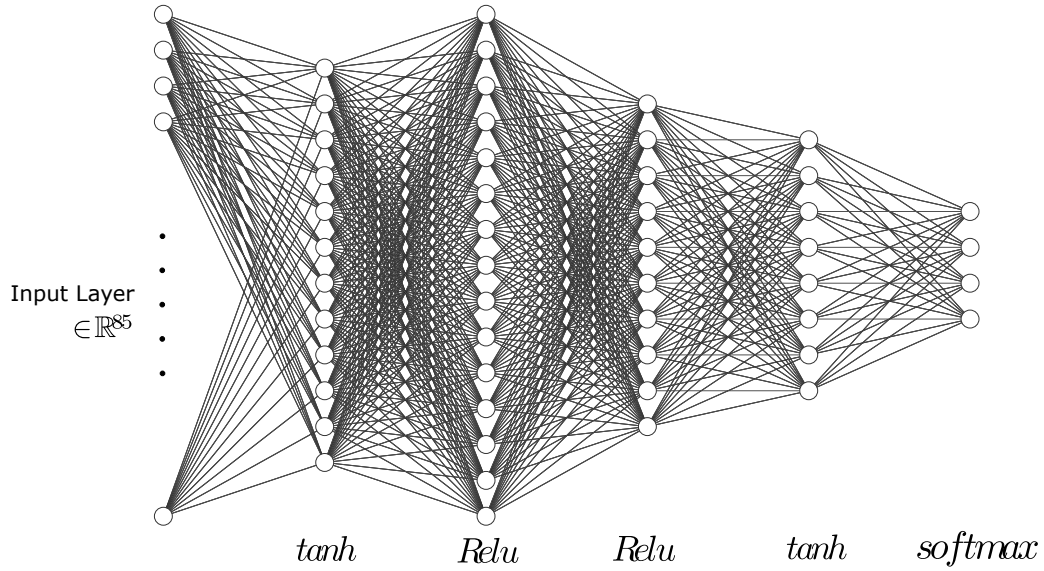


Figure 3–9: The architecture of the neural network

The network uses scaled conjugate gradient backpropagation as the training algorithm, without regularization, and Cross-Entropy as a measurement for loss. The data is standardized before training.

The dataset is split into three randomly-selected partitions to be used for training (75% of the dataset), validation (15% of the dataset) and test (10% of the dataset).

Using 5-fold Cross-Validation, the average accuracy obtained for train and validation sets are listed in Table 3–5.

	accuracy
Train set	99.88%
Validation set	99.85%

Table 3–5: Average Accuracy for Train and Validation sets

Figure 3–10 shows the performance (loss) of the network during the training process, and Figure 3–11 and 3–12 show the confusion matrices of the network. As seen in the confusion matrix, the accuracy of the network on the test set is 99.7%.

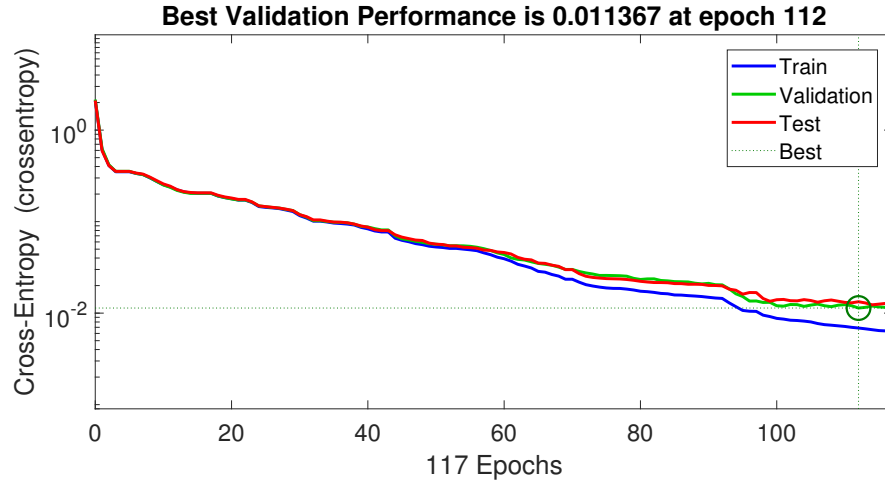


Figure 3–10: Cross-Entropy Loss per epoch during the training for Train, Validation and Test sets

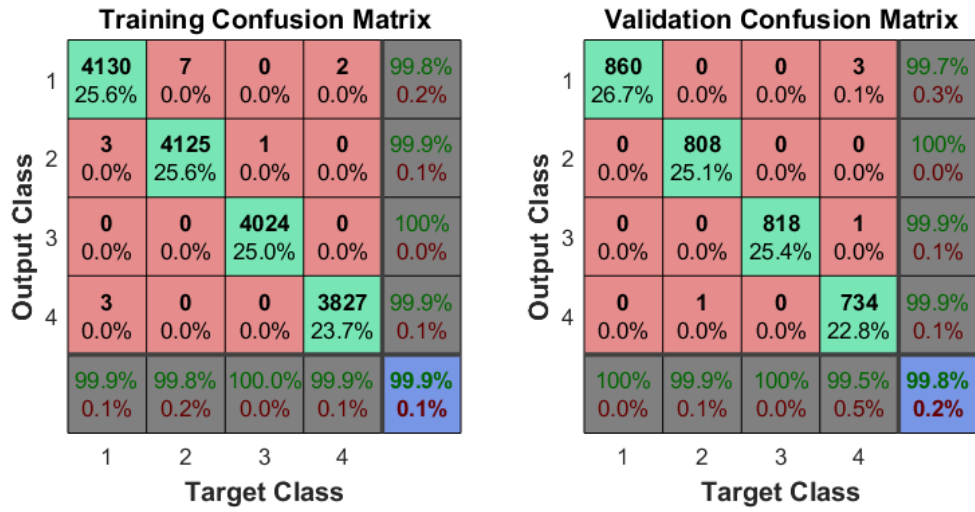


Figure 3–11: Confusion Matrix for train and validation sets

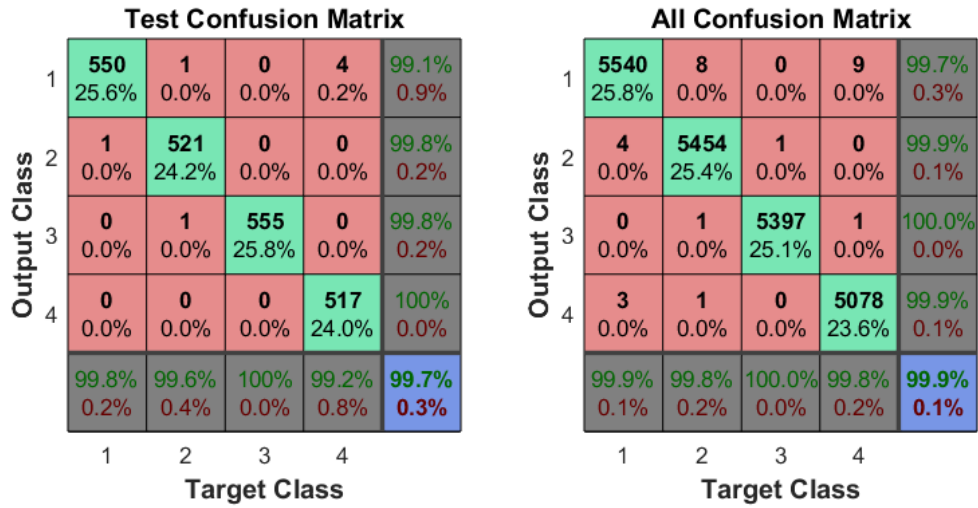


Figure 3–12: Confusion Matrix test and overall sets

Chapter 4

Results

In this chapter, we present the results obtained from the simulation. First, the behavior of the system and controllers are evaluated in a field of one soil, for Clay, Loam, Sand, and Sandyloam, independently. Then, the system is evaluated in a field of mixed soils. All simulation results are gathered at a constant speed of $4.5 \frac{m}{s}$, or $16 \frac{km}{h}$. It is assumed that the vehicle is moving on the XY plane, and has no movement along the Z axis.

4.1 Clay

Figures 4–1 to 4–7 show the simulation results for the vehicle moving on Clay soil. Figure 4–1 demonstrates the path of the vehicle, and the desired path, in global fixed coordinates $X - Y - Z$. This path is designed based on the fact that the farming vehicle needs to move between crops rows, and perform tasks such as seeding or harvesting. This path consists of two U-turns and 65-meter straight line in between.

Figure 4–2 represents the MPC controller’s input command to the system, which is the steering of the left front wheel(δ_1). The step between $t=17$ s and $t=32$ s represents the steering angle input needed for the first U-turn, and the step between $t=48$ s and $t=63$ s is the steering angle input needed for the second U-turn.

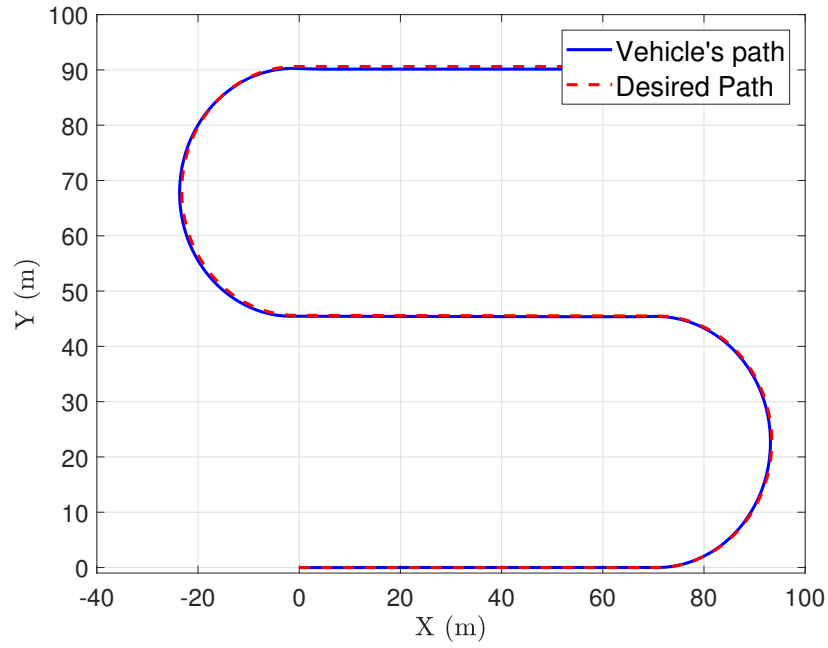


Figure 4-1: Vehicle path on Clay

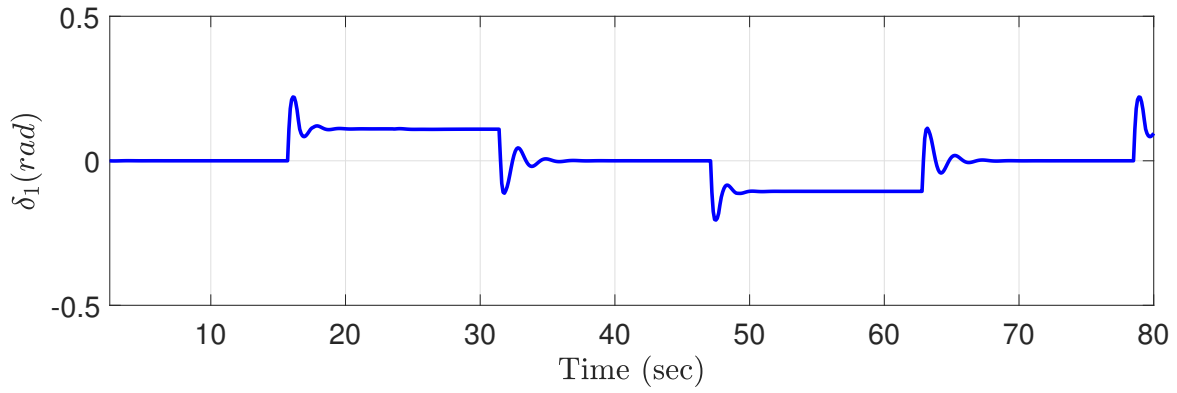


Figure 4-2: MPC controller output: Steering angle of the front wheel

The errors used as the feedback signal for the MPC controller are shown in Figure 4-4. The errors are defined in Chapter 2, Equation 2.9. The maximum value of the heading deviation is less than 0.1 rad, which happens at the beginning and at

the end of a turn, when the heading reference, Figure 4–3, changes from zero to 0.2 rad.

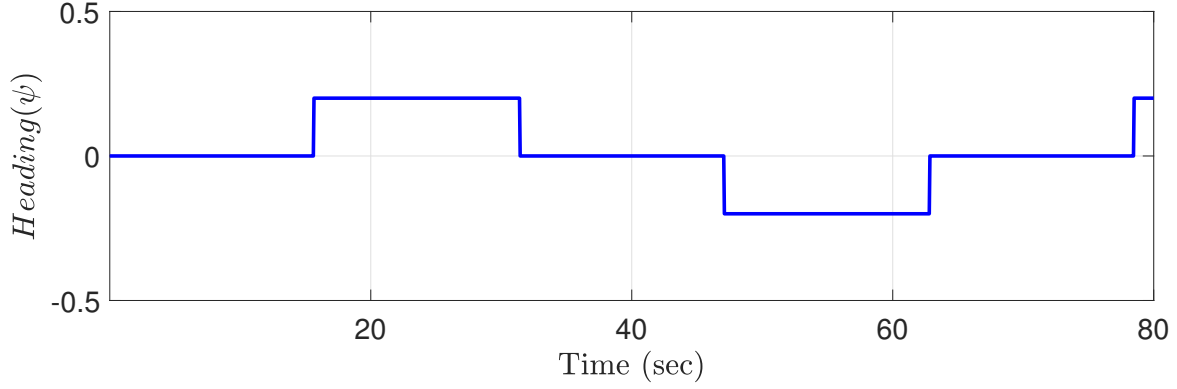


Figure 4–3: Heading reference for the S-shape path

As can be seen in Figure 4–4, the maximum value of the lateral deviation is 0.1 m, which comparing to the common width of the crop rows and the width of the vehicle itself, which is around 2 m, is 5% error.

Figure 4–5 displays the PID controller’s input command to the system, which is the torque of a wheel. At the beginning of the maneuver, the torque is generated to accelerate the vehicle and increase the velocity of the vehicle from zero to the desired value. In the turn, the vehicle will have a drop in speed, as the forces caused by the normal and shear stress of soil is divided into two motions, longitudinal and lateral. This will cause an increase in the torque at the beginning of each turn.

The longitudinal speed is shown in Figure 4–6, in which the difference between the desired and actual speed of the vehicle along its x axis can also be observed. The red dashed line represents the desired constant speed. The longitudinal speed error is 2% of the given reference speed.

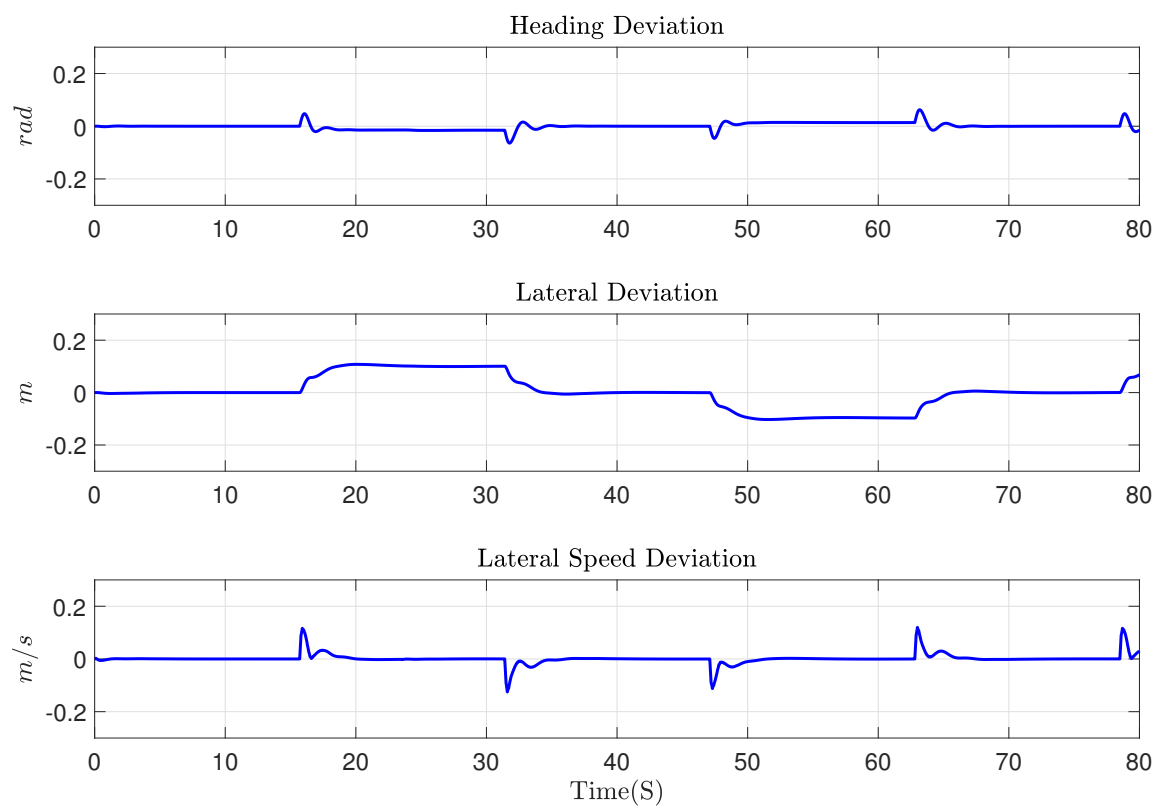


Figure 4-4: Errors as feedback to the MPC controller

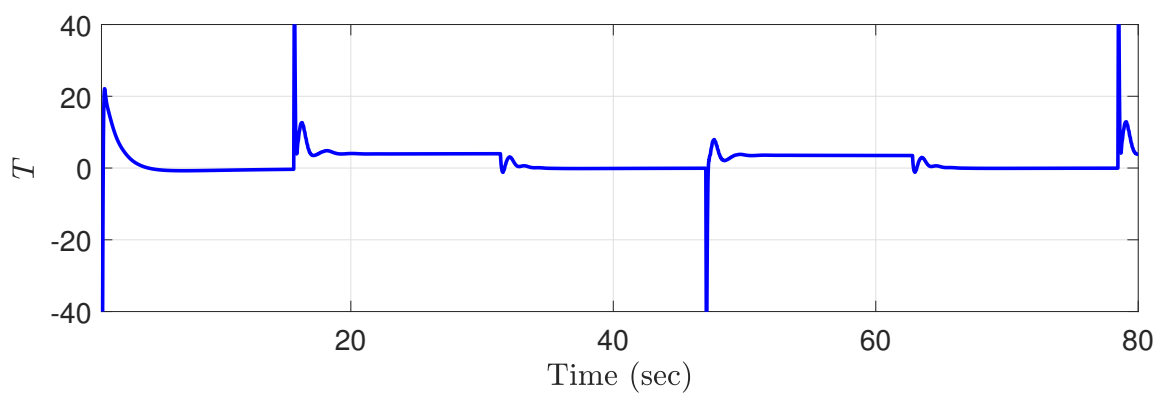


Figure 4-5: PID controller output: wheel torque

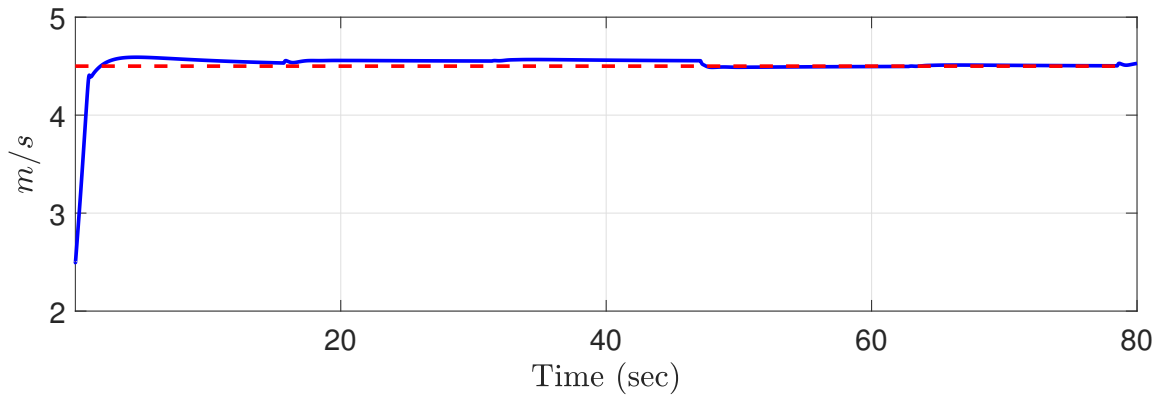


Figure 4-6: Longitudinal speed

Figure 4-7 shows the decision made by the Neural Network block on the soil type. In this figure, there is a misclassification in the first second of the simulation, which provides 99% accuracy.

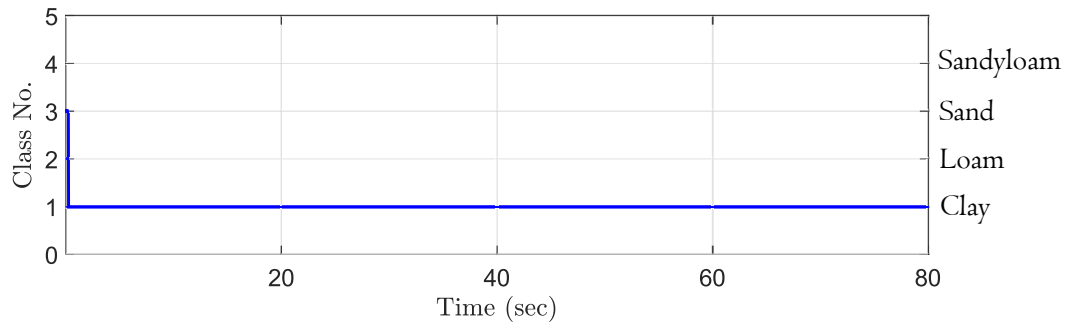


Figure 4-7: Class prediction

4.2 Loam

Similar to Clay, Figures 4–8 to 4–13 demonstrate the simulation results for the vehicle moving on Loam soil. Figure 4–8 shows the path of the vehicle, and the desired path, in the global fixed coordinates $X - Y - Z$.

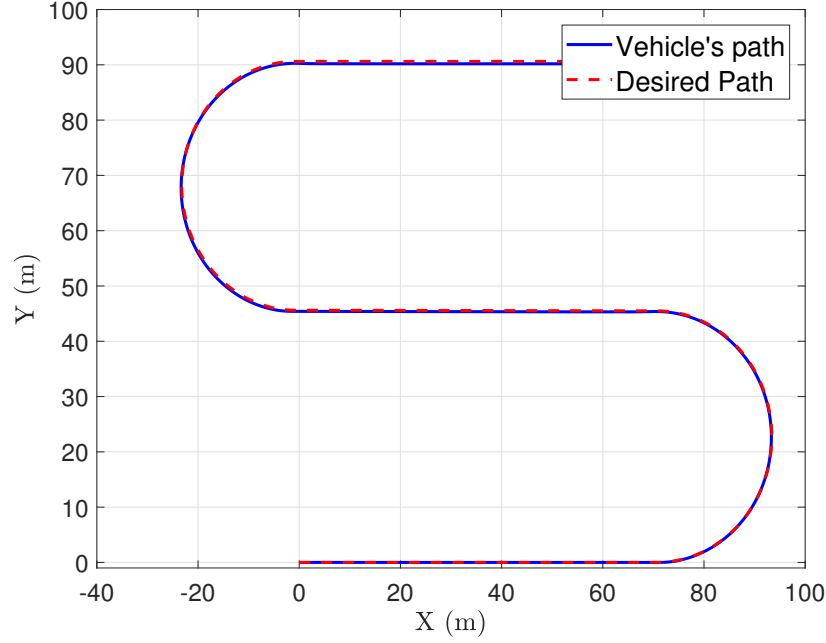


Figure 4–8: Vehicle Path on Loam

Figure 4–9 shows the MPC controller's input to the system.

The errors used as the feedback signal for the MPC controller are shown in Figure 4–10. As can be seen, the maximum values of the heading deviation and lateral deviation are 0.2 rad and 0.08 m (0.02% error), respectively.

The deviation of the longitudinal speed from its desired value is shown in Figure 4–12. Apart from the transient portion of the result, the longitudinal speed error

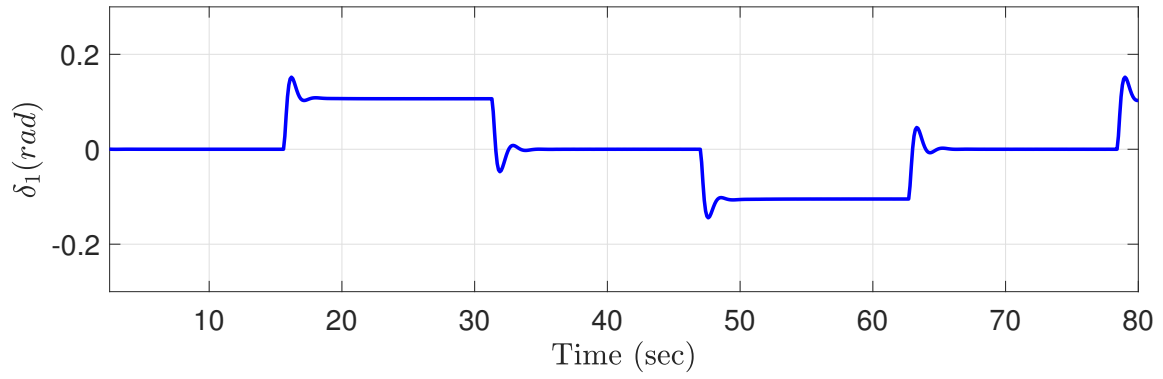


Figure 4-9: MPC controller output: Steering angle of the front wheel

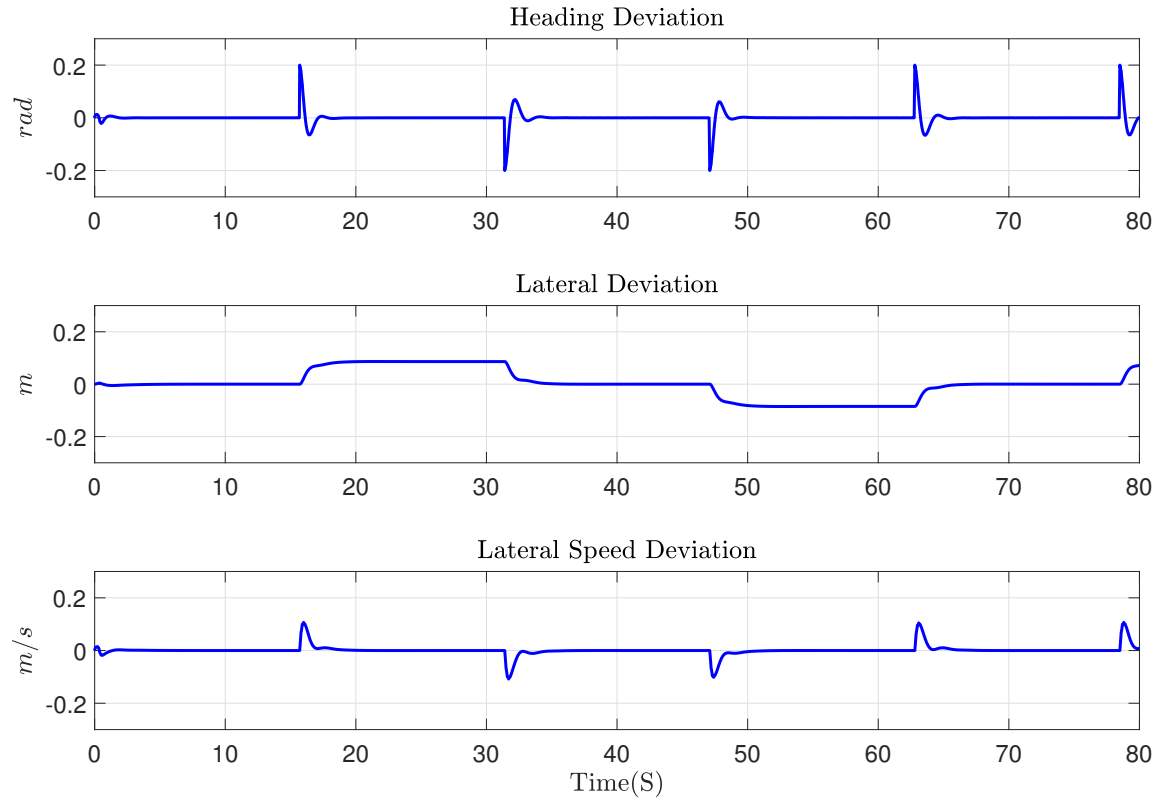


Figure 4-10: Errors as feedback to the MPC controller

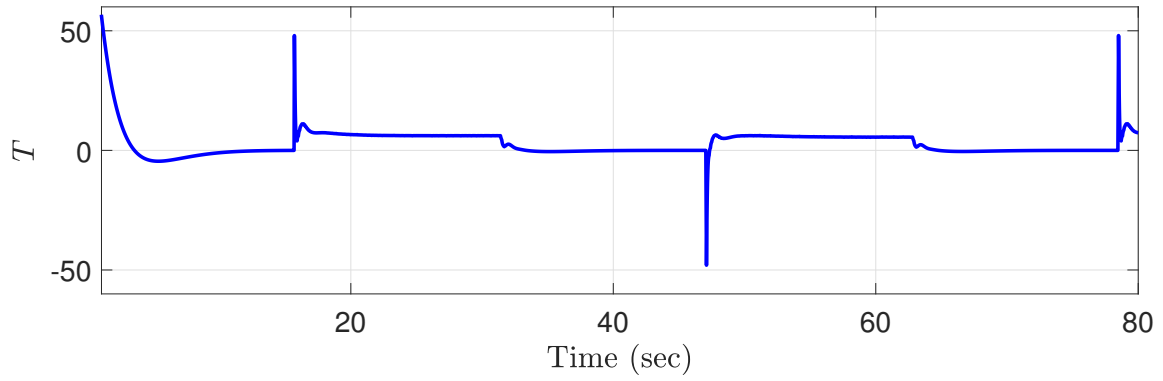


Figure 4–11: PID controller output: wheel torque

is 1% of the reference value. In this case, the speed overshoot acts in our favor. It compensates for the distance the vehicle falls behind, along the X axis, because of the difference in the desired speed and the actual speed. In other words, the ideal vehicle on the desired path starts its movement with the velocity of $4.5 \frac{m}{s}$, but the vehicle, in reality, will start from 0 velocity.

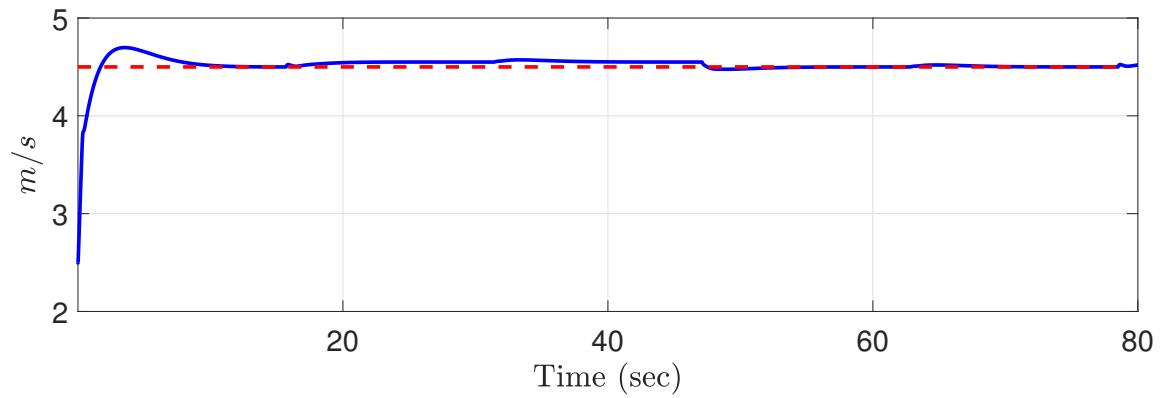


Figure 4–12: Longitudinal speed

Figure 4–13 shows the decision made by the Neural Network block on the soil type. In this figure, there is a misclassification for 0.5 second in the simulation, which provides 99.4% accuracy.

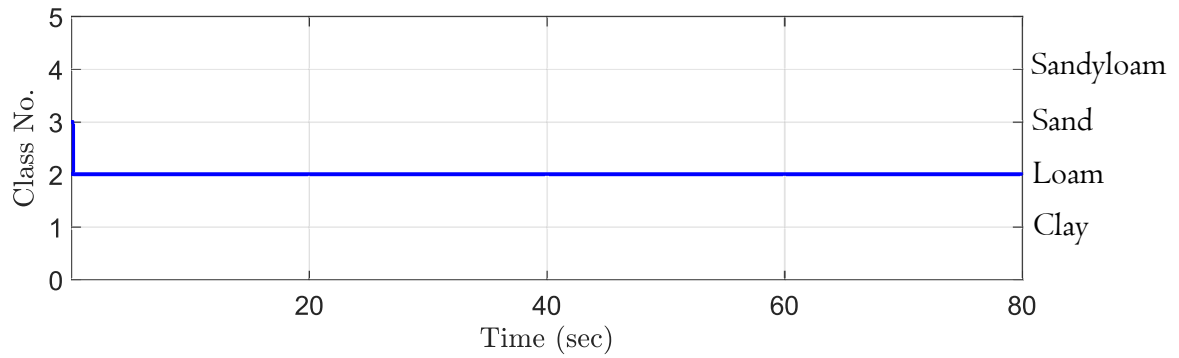


Figure 4–13: Class prediction

4.3 Sand

Similarly, Figures 4–14 to 4–19 show the simulation results for the vehicle moving on Sand soil.

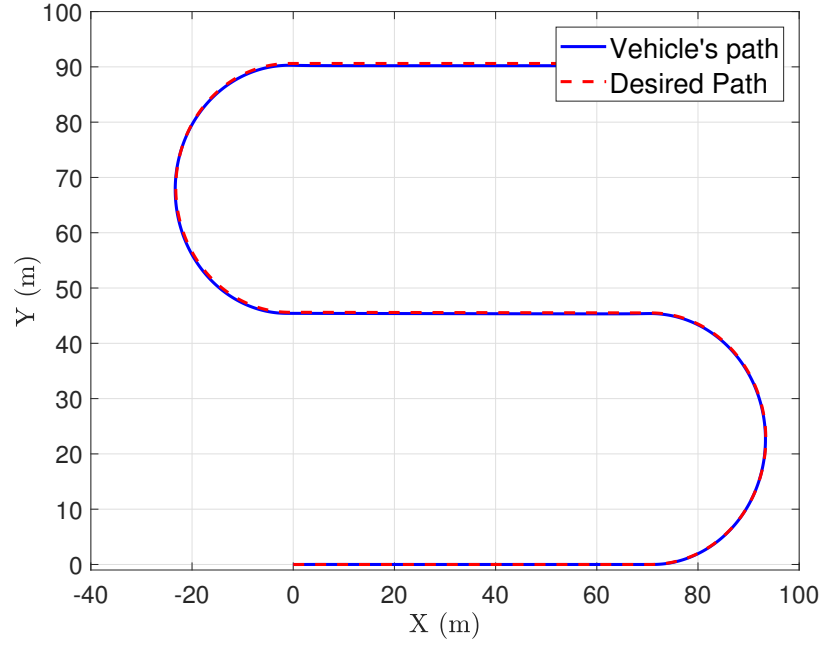


Figure 4–14: Vehicle Path on Sand

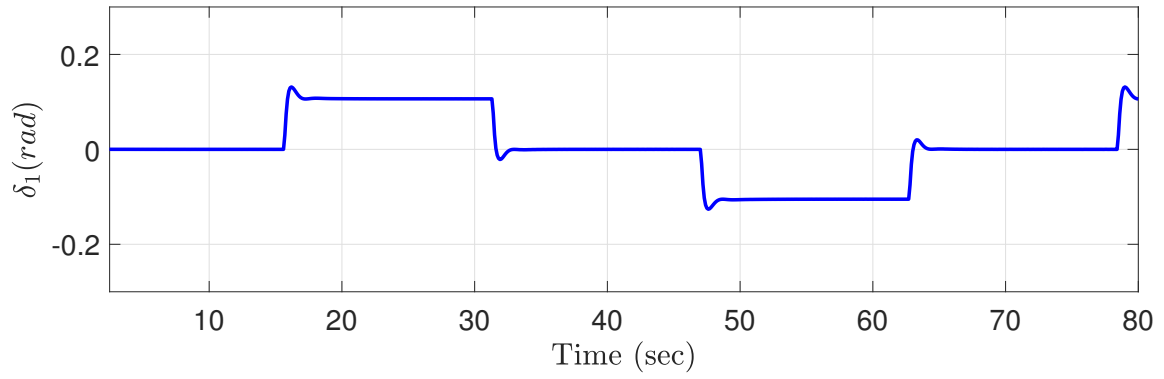


Figure 4–15: MPC controller output: Steering angle of the front wheels

The peak value of the heading deviation is 0.2 rad , which happens at the beginning and the end of a turn when the heading reference- the S-shape path- changes from zero to 0.2 rad . As can be seen in 4-16, lateral deviation peaks at $0.09m$, which leads to 0.02% error.

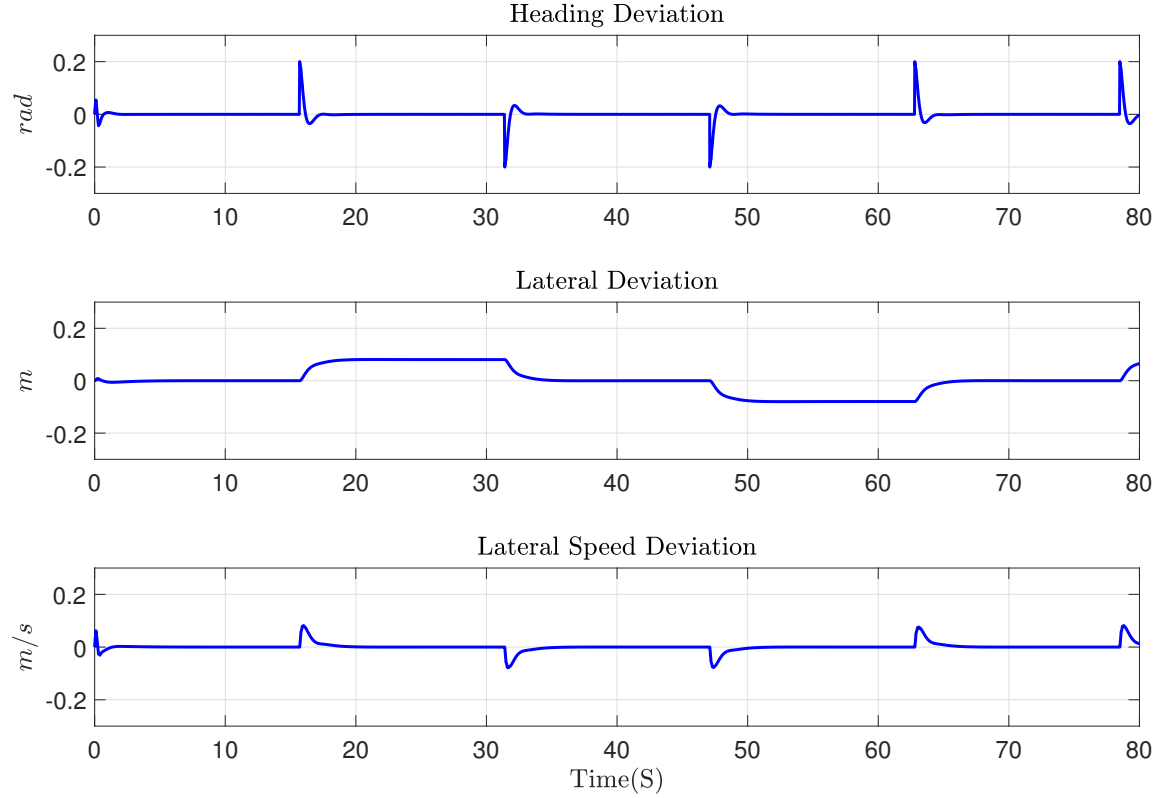


Figure 4-16: Errors as feedback to the MPC controller

The longitudinal speed error is 1% of the reference speed.

Figure 4-19 shows the decision made by the Neural Network block on the soil type. In this figure, there is a misclassification for 0.6 second of the simulation, which provides 99.3% accuracy.

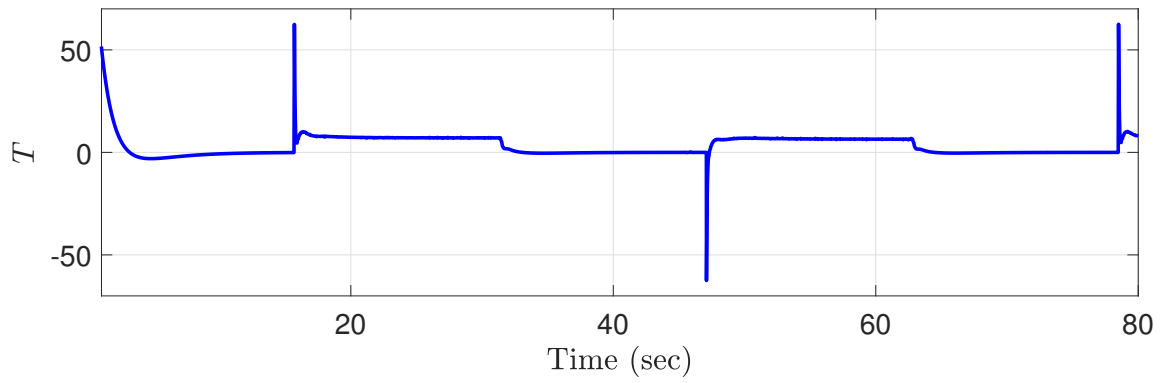


Figure 4-17: PID controller output: wheel torque

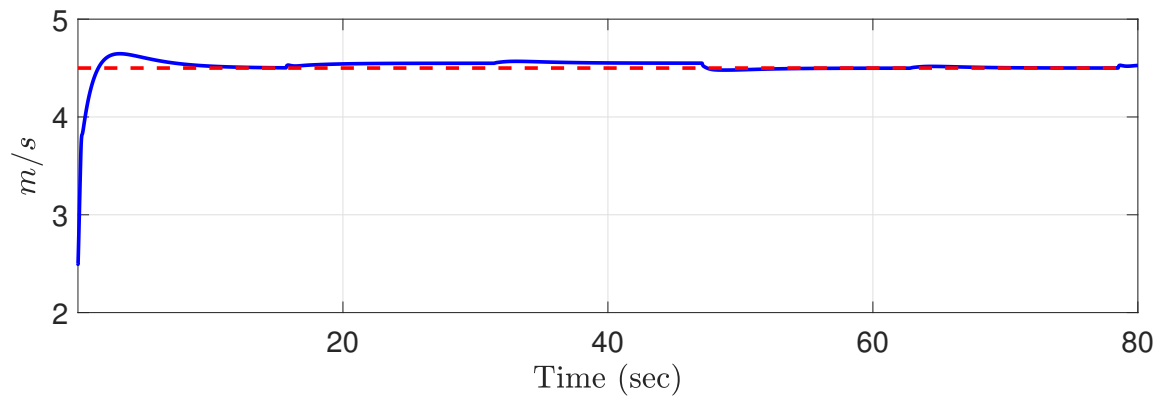


Figure 4-18: Longitudinal speed

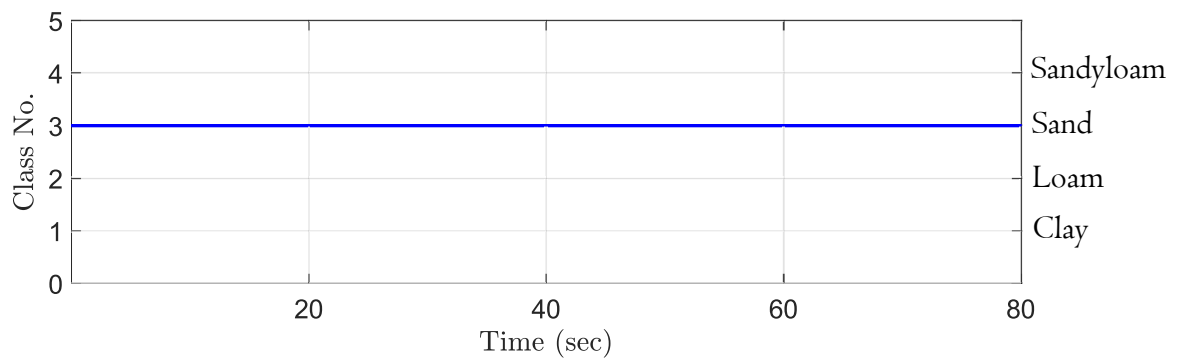


Figure 4-19: Class Prediction

4.4 Sandyloam

Figures 4–20 to 4–25 shows the simulation results for the vehicle moving on Sandyloam soil.

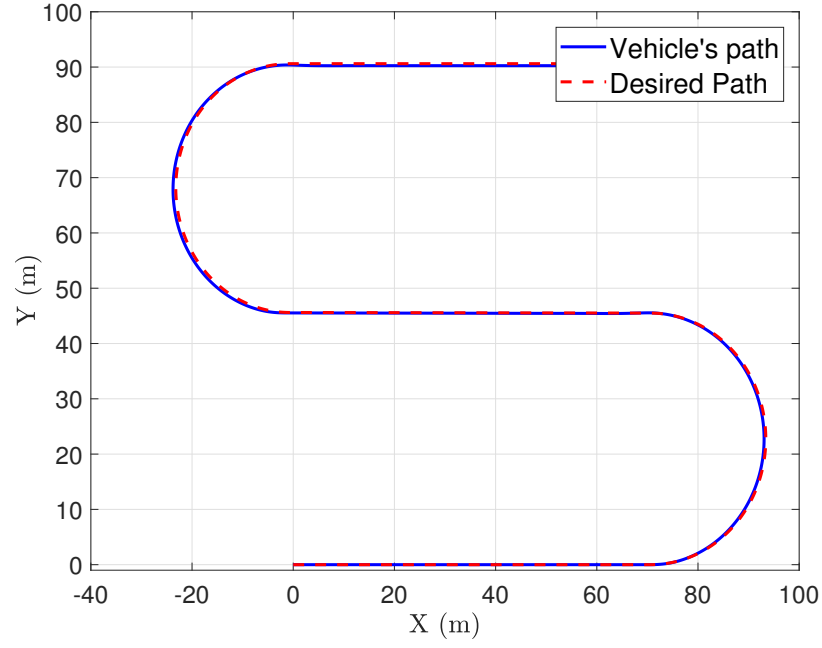


Figure 4–20: Vehicle Path on Sandyloam

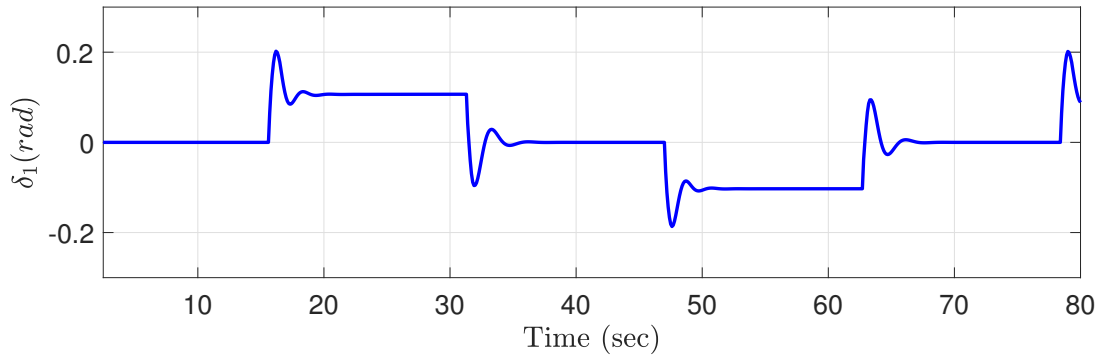


Figure 4–21: MPC controller output: Steering angle of the front wheels

The peak of the heading deviation is 0.2 rad . As can be seen in Figure 4–22, lateral deviation peaks at 0.15 m , which leads to 0.04% error.

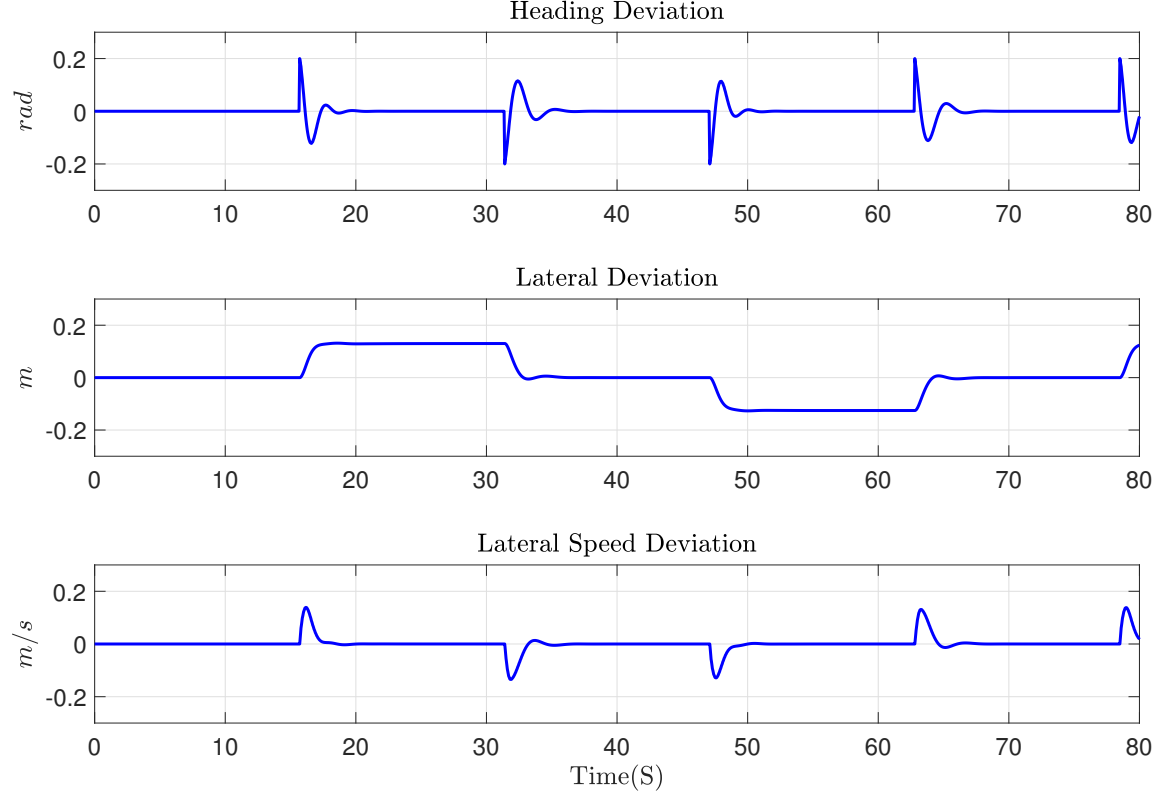


Figure 4–22: Errors as feedback to the MPC controller

The longitudinal speed error is 3% of the reference speed. Figure 4–25 shows the decision made by the neural network block on the soil type. In this figure, there is a misclassification for 1 second of the simulation, which provides 99% accuracy.

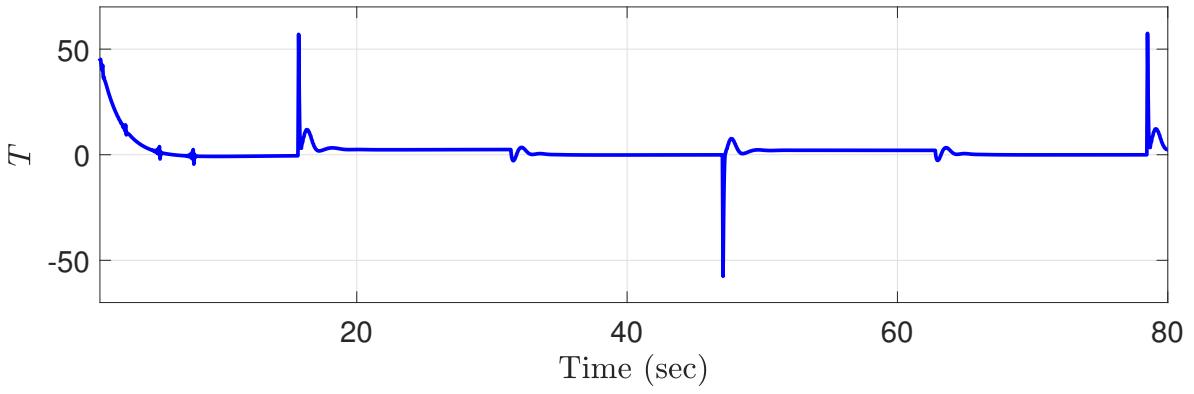


Figure 4-23: PID controller output: torque of each wheel

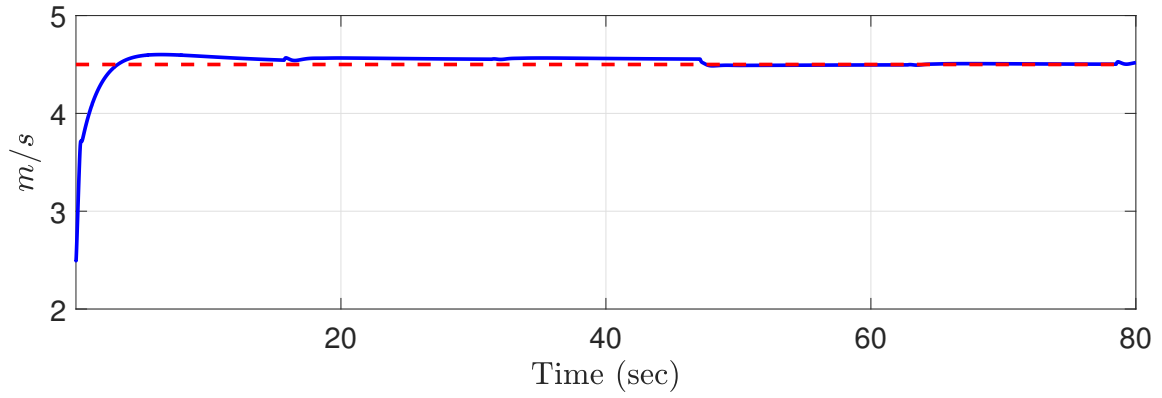


Figure 4-24: Longitudinal speed

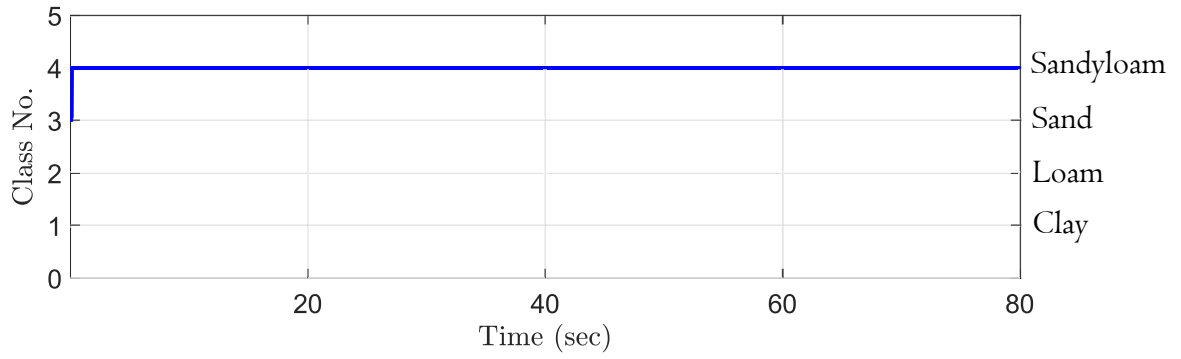


Figure 4-25: Class Prediction

4.5 Mixed Soil

In this section, the field consists of four different types of soils: Clay, Loam, Sand, and SandyLoam, as shown in Figure 4–26. Figures 4–26 to 4–31 demonstrate the results, in this case, using the neural network controller for smart adaptation to the soil type. As can be seen in Figure 4–26, the effect of changes in soil type on the vehicle path is not considerable, as the controller can handle the situation well. However, the small changes in the control inputs, steering angle and torque, as well as errors, can be seen in Figures 4–27 to 4–31, at $t = 20$, $t = 40$, $t = 60$ second, when the soil type changes.

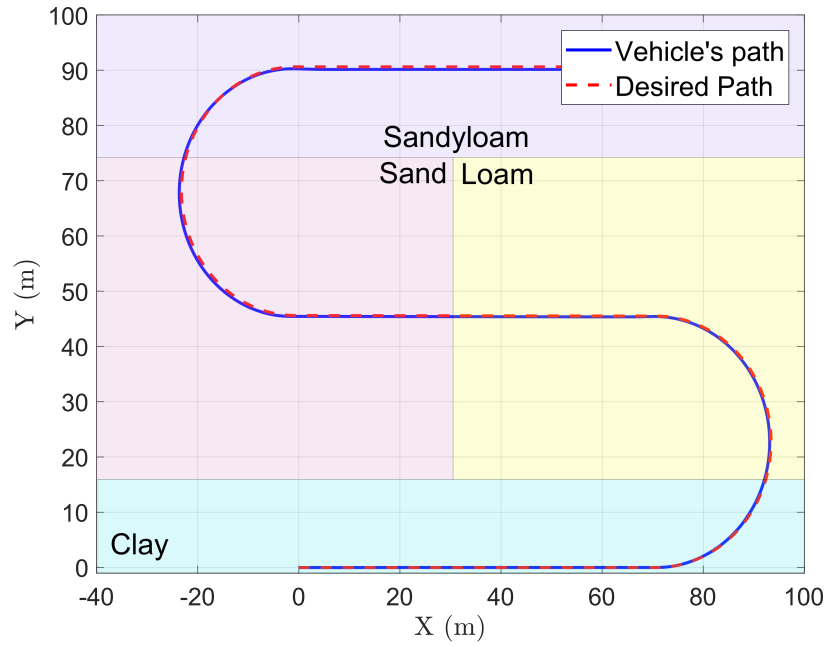


Figure 4–26: Vehicle Path on Mixed Soil

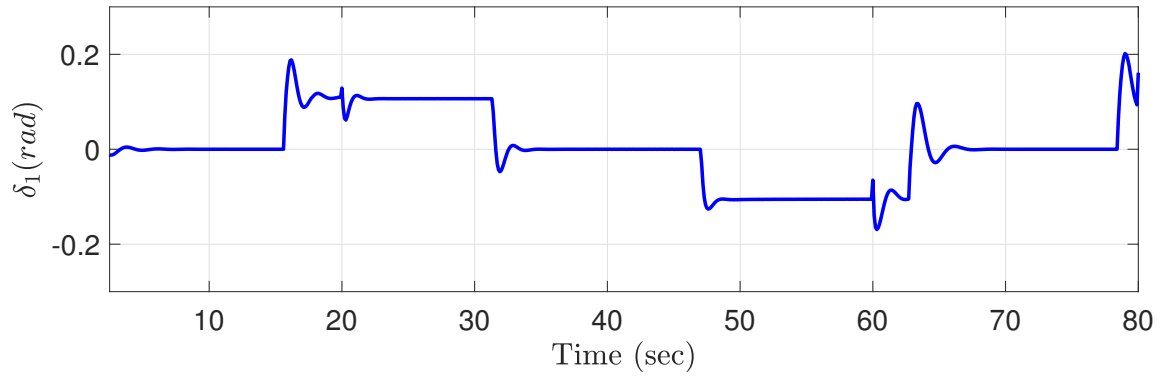


Figure 4-27: MPC controller output: Steering angle of the front wheels

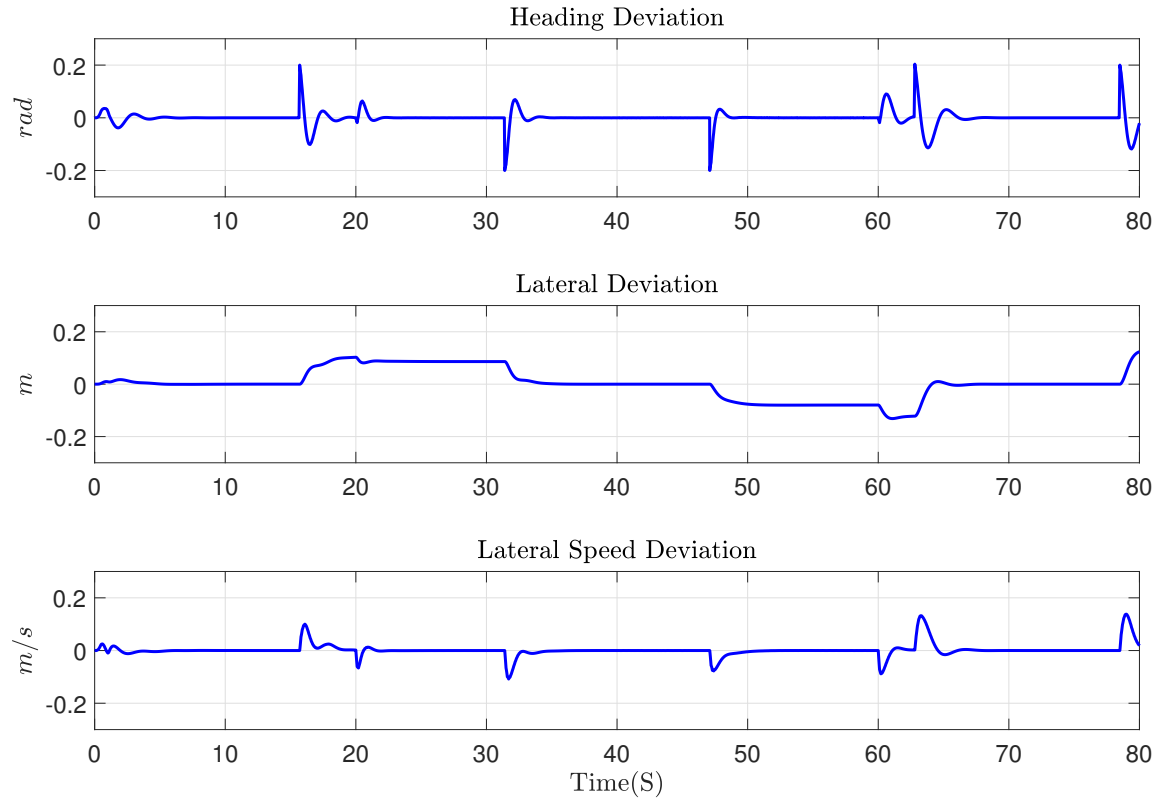


Figure 4-28: Errors as feedback to the MPC controller

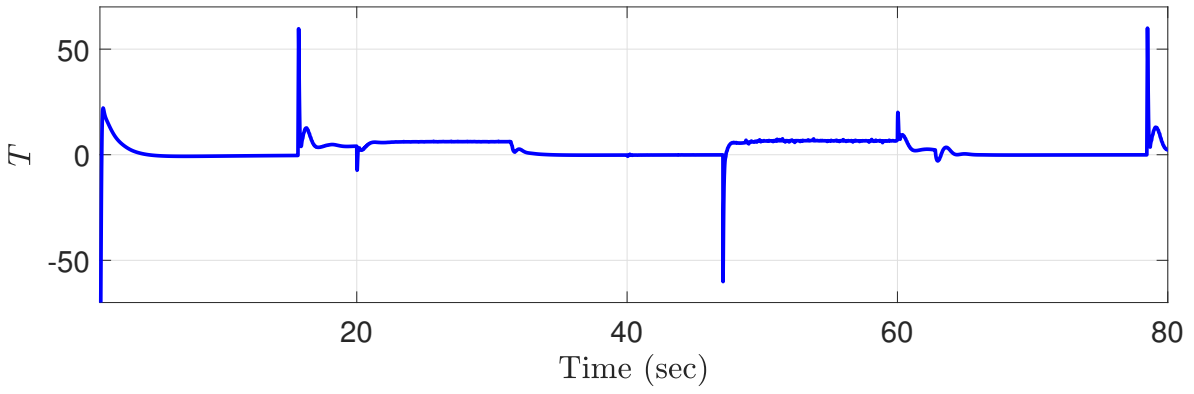


Figure 4-29: PID controller output: wheel torque

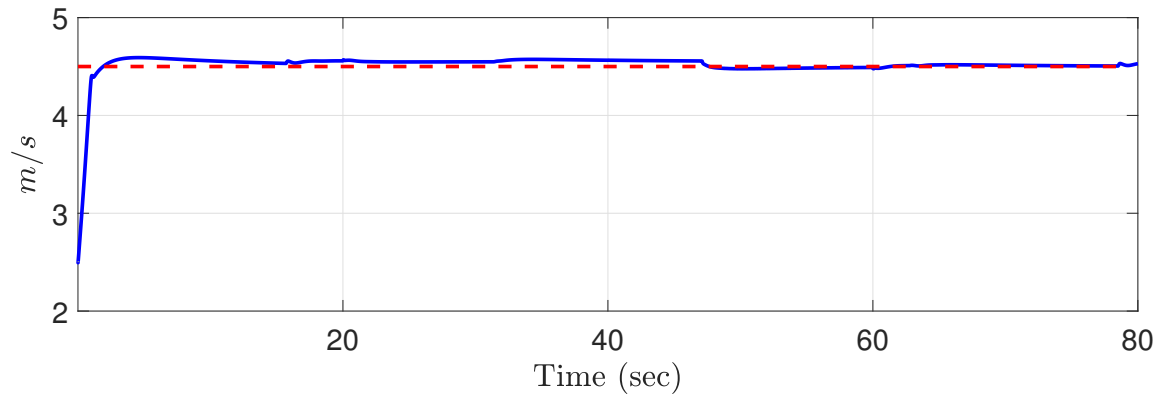


Figure 4-30: Longitudinal speed

Figure 4-31 shows the predicted soil type and the true soil type during the simulation. The prediction accuracy, in this case, is 99%.

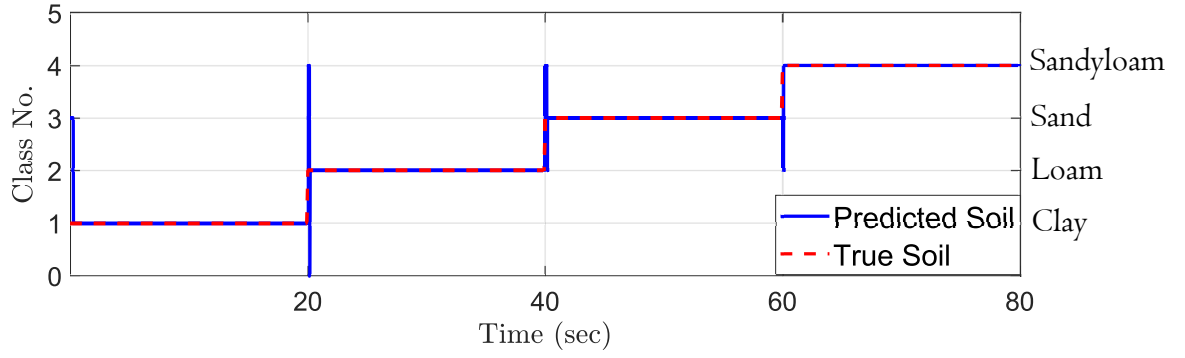


Figure 4-31: Class prediction

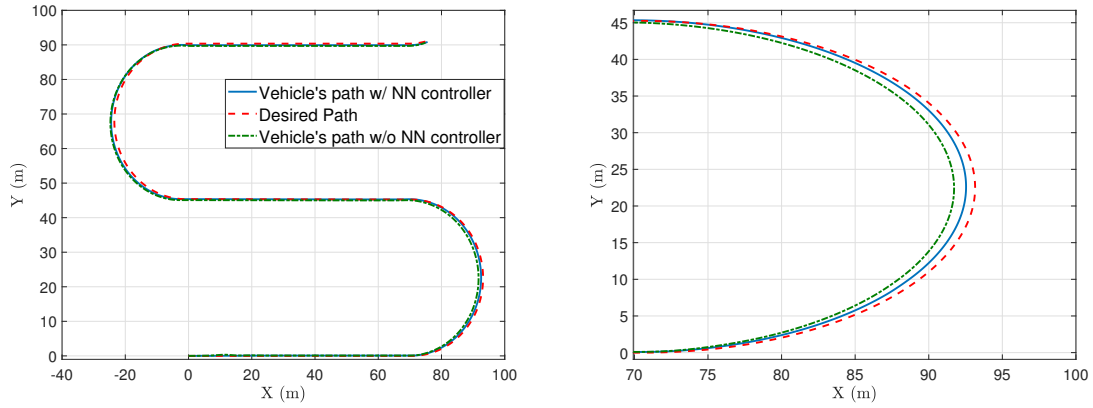


Figure 4-32: Left: Path of the vehicle with and without NN Controller; Right: Zoomed view of the part of the path in the Rectangle

Figures 4-32 compares the performance of the controller with the neural network block and a single MPC controller (loam) in the mixed soil field. Figure 4-33 shows the deviation of the vehicle with two controllers along the S-shape path of Figure 4-32. The error of the single MPC is at some points 2.5 times more than the error of the smart controller with neural network block.

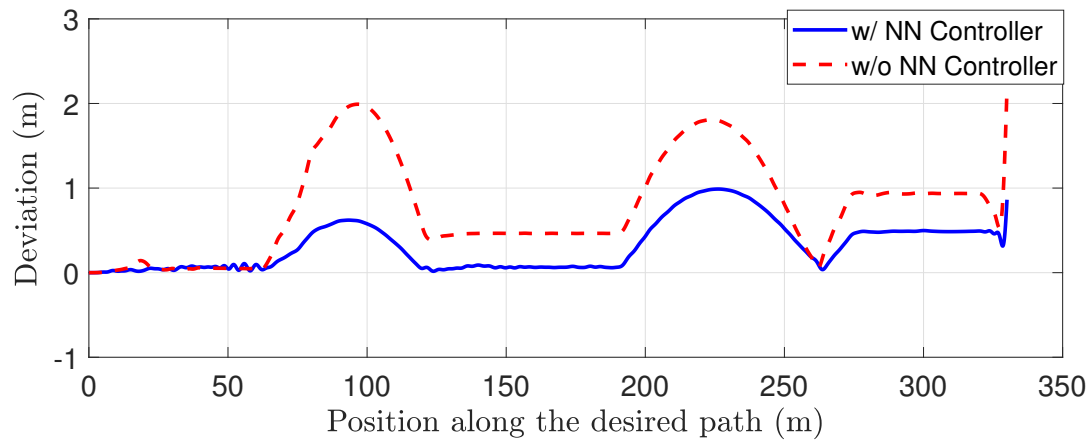


Figure 4–33: Lateral deviation of the vehicle along the path

Figure 4–34 is an example of the performance of the two controllers at the start of the motion, where the vehicle is in an unstable condition. In this situation, torque is applied on the wheels to reach the desired velocity, and the wheels tend to slip.

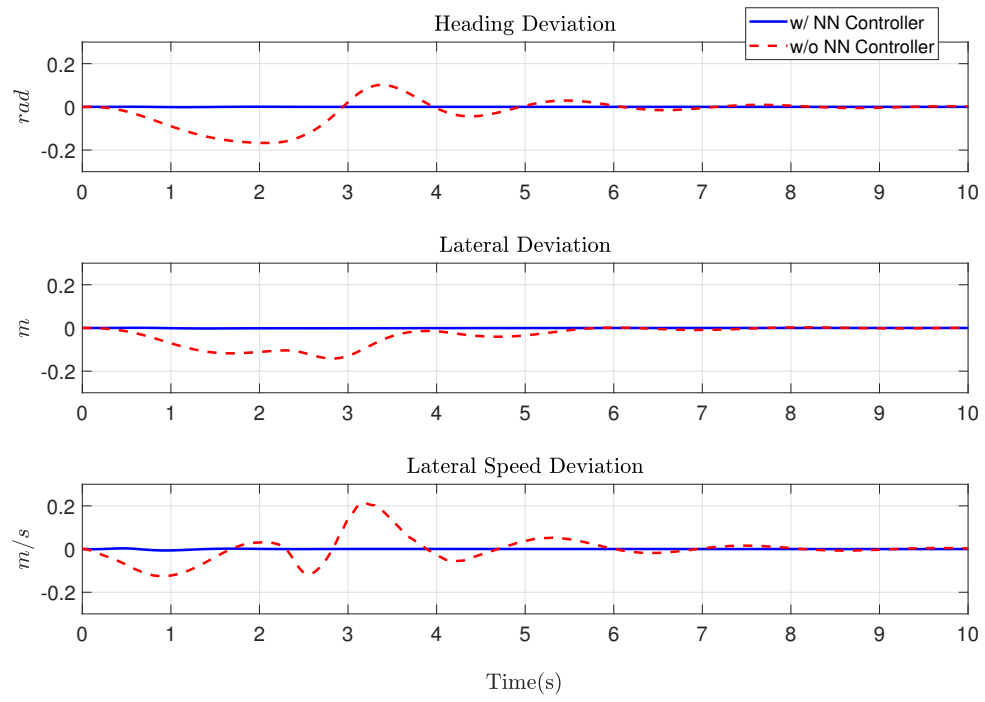


Figure 4–34: MPC feedback errors at the beginning of the simulation

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we provided a novel scheme for steering control of an autonomous farming vehicle, to adapt to different and difficult terrain situations. This study contains two sections:

First, the design of a Neural Network that can classify the current soil the vehicle is driving on, based on physical features such as vehicle velocity, wheel torques and angular velocities, wheel sinkage, slip ratios and slip angles. In this study, we designed a neural network that can classify the soil type with at least 99% accuracy. The Neural Network is trained offline and used in real time, to classify the soil. The soil types defined in Chapter 2 are Clay, Loam, Sandyloam, and Sand.

Second, the structure of the vehicle controller of the system is designed. This structure contains one PID to control the longitudinal speed, and four MPCs to control the steering of the vehicle, one for each type of soil. In this structure, the neural network makes the decision to switch between the MPCs, based on the soil type, to have the most suitable MPC.

The implementation of this system on an autonomous vehicle is evaluated. The variables needed for feedback to the system, and features of the neural network input can be measured using cheap and easily accessible sensors. The encoders are

needed for measuring angular velocities of the wheels. We can use IMU or GPS to measure the vehicle's longitudinal, lateral and angular speed. The torque applied to the wheels can be calculated knowing the input current to the wheel's electric motors. The average sinkage of the front wheels is measured using two wide range ultra-sonic sensors, mounted on a structure on front wheels. Slip ratio and slip angle are estimated using measured velocities.

The results demonstrate the performance of the designed controller and its advantages over a simple MPC controller. The smart controller is more accurate in terms of lateral deviation, and more stable in case of difficult situations where wheel slippage is high.

5.2 Future Work

For future studies, this scheme can be expanded to a variety of different situations. The Neural Network can be generalized to a wider variety of situations, such as more classes of soil types, a wider range of speed, and slippage. Moreover, as the neural network is designed for a specific vehicle, the effect of the weight is not considered in the Neural Network. This feature could be used in training the neural network, to make it compatible with other vehicles, and also to consider the effect of load transfer in high-acceleration situations, and harsh maneuvers.

We also propose that the neural network can be used to adapt the PID to different soil types. Since the dynamics of the vehicle and soil terramechanics are nonlinear with respect to slippage, adapting the PID controller to slippage and soil type will have a considerable effect on stability and accuracy of trajectory tracking, especially in high slip condition.

References

- [1] Rankin AL, Crane III CD, Armstrong II DG, Nease AD, Brown HE., "Autonomous path-planning navigation system for site characterization." *Navigation and Control Technologies for Unmanned Systems*, 1996, Vol. 2738, pp. 176-186, International Society for Optics and Photonics.
- [2] Barton MJ., "Controller development and implementation for path planning and following in an autonomous urban vehicle.", Undergraduate thesis, University of Sydney. 2001.
- [3] Wit J, Crane III CD, Armstrong D., "Autonomous ground vehicle path tracking.", *Journal of Robotic Systems*, 2004, 21(8), pp. 439-49.
- [4] Buehler M, Iagnemma K, Singh S, "The 2005 DARPA grand challenge: the great robot race.", Springer, 2007.
- [5] Hoffmann GM, Tomlin CJ, Montemerlo M, Thrun S., "Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing.", *American Control Conference*, 2007, pp. 2296-2301, IEEE.
- [6] Lenain R, Thuilot B, Cariou C, Martinet P., "High accuracy path tracking for vehicles in presence of sliding: Application to farm vehicle automatic guidance for agricultural tasks.", *Autonomous robots*, 2006, 21(1), pp. 79-97.
- [7] Rossetter EJ., "A potential field framework for active vehicle lanekeeping assistance", Doctoral dissertation, Stanford University, 2003.

- [8] Fierro R, Lewis FL., "Control of a nonholomic mobile robot: Backstepping kinematics into dynamics.", *Journal of robotic systems*. 1997, 14(3), pp. 149-63.
- [9] Osinenko PV, Geissler M, Herlitzius T., "A method of optimal traction control for farm tractors with feedback of drive torque.", *Biosystems engineering*, 2015, 129, pp. 20-33.
- [10] Lucet E, Lenain R, Grand C., "Dynamic path tracking control of a vehicle on slippery terrain.", *Control Engineering Practice*, 2015, 42, pp. 60-73.
- [11] Fang H, Lenain R, Thuilot B, Martinet P., "Robust adaptive control of automatic guidance of farm vehicles in the presence of sliding.", *InProceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 3102-3107, IEEE.
- [12] Hima S, Glaser S, Chaibet A, Vanholme B., "Controller design for trajectory tracking of autonomous passenger vehicles.", *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 1459-1464, IEEE.
- [13] Pratama PS, Jeong JH, Jeong SK, Kim HK, Kim HS, Yeu TK, Hong S, Kim SB., "Adaptive backstepping control design for trajectory tracking of automatic guided vehicles.", *AETA: Recent Advances in Electrical Engineering and Related Sciences*, 2016, pp. 589-602, Springer, Cham.
- [14] Kong J, Pfeiffer M, Schildbach G, Borrelli F., "Kinematic and dynamic vehicle models for autonomous driving control design.", *IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 1094-1099, IEEE.

- [15] Gao Y, Lin T, Borrelli F, Tseng E, Hrovat D., "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads.", ASME 2010 dynamic systems and control conference, 2010, pp. 265-272.
- [16] Yoon Y, Shin J, Kim HJ, Park Y, Sastry S., "Model-predictive active steering and obstacle avoidance for autonomous ground vehicles.", Control Engineering Practice, 2009, 17(7), pp. 741-50.
- [17] Soloway DI, Haley PJ., "Neural generalized predictive control: A newton-raphson implementation", NASA Langley Technical Report Server, 1997.
- [18] Slotine JJ, Li W., "Applied nonlinear control.", Englewood Cliffs, NJ: Prentice hall, 1991.
- [19] Narendra KS, Parthasarathy K., Identification and control of dynamical systems using neural networks.", IEEE Transactions on neural networks, 1990, 1(1), pp. 4-27.
- [20] Zhang T, Kahn G, Levine S, Abbeel P., "Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search.", IEEE international conference on robotics and automation (ICRA), 2016, pp. 528-535.
- [21] Zhu ZX, Torisu R, Takeda JI, Mao ER, Zhang Q., "Neural network for estimating vehicle behaviour on sloping terrain.", Biosystems engineering, 2005, 91(4), pp. 403-11.
- [22] Matusko J, Petrović I, Perić N., "Neural network based tire/road friction force estimation.", Engineering Applications of Artificial Intelligence, 2008, 21(3), pp. 442-56.

- [23] Bellutta P, Manduchi R, Matthies L, Owens K, Rankin A, "Terrain perception for DEMO III", InProceedings of the IEEE Intelligent Vehicles Symposium 2000, 2000, pp. 326-331.
- [24] L. Matthies et al., "Performance evaluation of UGV obstacle detection with CCD/FLIR stereo vision and LADAR," Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC) held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA) Intell, Gaithersburg, MD, USA, 1998, pp. 658-670.
- [25] Howard A, Seraji H., "Vision-based terrain characterization and traversability assessment", Journal of Robotic Systems, 2001, pp. 577-87.
- [26] Kim J, Kim D, Lee J, Lee J, Joo H, Kweon IS., "Non-contact terrain classification for autonomous mobile robot.", IEEE International Conference on Robotics and Biomimetics (ROBIO), 2009, pp. 824-829.
- [27] Brooks CA, Iagnemma K., "Vibration-based terrain classification for planetary exploration rovers.", IEEE Transactions on Robotics, 2005, pp. 1185-91.
- [28] Sathukhan D, Moore C, Collins E., "Terrain estimation using internal sensors.", InProceedings of the IASTED International Conference on Robotics and Applications, 2004.
- [29] Valada A, Spinello L, Burgard W., "Deep feature learning for acoustics-based terrain classification." In Robotics Research, Springer, Cham, 2018, pp. 21-37.
- [30] Iagnemma KD, Dubowsky S., "Terrain estimation for high-speed rough-terrain autonomous vehicle navigation.", Unmanned Ground Vehicle Technology IV, 2002, Vol. 4715, pp. 256-267.

- [31] Weiss C, Tamimi H, Zell A., "A combination of vision-and vibration-based terrain classification.", IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, pp. 2204-2209.
- [32] Wang S, Kodagoda S, Ranasinghe R. "Road terrain type classification based on laser measurement system data.", Australasian Conference on Robotics and Automation, ACRA, 2012.
- [33] Gonzalez R, Iagnemma K., "Deep Terramechanics: Terrain Classification and Slip Estimation for Ground Robots via Deep Learning.", arXiv preprint arXiv:1806.07379, 2018.
- [34] Pacejka H., "Tire and vehicle dynamics.", Elsevier, 2005.
- [35] Rula AA, Nuttall Jr CJ., "An analysis of ground mobility models (ANAMOB).", Army Engineer Waterways Experiment Station, 1971.
- [36] Wismer RD, Luth HJ., "Off-road traction prediction for wheeled vehicles.", Journal of Terramechanics. 1973, 10(2), pp. 49-61.
- [37] Allen RW, Rosenthal TJ, Chrstos JP., "A vehicle dynamics tire model for both pavement and off-road conditions.", SAE Technical Paper, 1997.
- [38] Metz LD., "Dynamics of four-wheel-steer off-highway vehicles.", SAE Technical Paper, 1993.
- [39] Madsen J, Seidl A, Negrut D, Ayers P, Bozdech G, Freeman J, O'Kins J, Reid A., "A physics based vehicle terrain interaction model for soft soil off-road vehicle simulations.", Wisconsin University-Madison Department of Mechanical Engineering, 2012.

- [40] Tanaka H, Momozu M, Oida A, Yamazaki M., "Simulation of soil deformation and resistance at bar penetration by the distinct element method.", *Journal of terramechanics*. 2000, 37(1), pp. 41-56.
- [41] Cundall PA, Strack OD., "A discrete numerical model for granular assemblies.", *Geotechnique*. 1979, 29(1), pp. 47-65.
- [42] Yong RN, Fattah EA, Boonsinsuk P., "Analysis and prediction of tyre-soil interaction and performance using finite elements.", *Journal of Terramechanics*, 1978, 15(1), pp. 43-63.
- [43] Yong RN, Boonsinsuk P, Fattah EA., "Tyre flexibility and mobility on soft soils.", *Journal of Terramechanics*, 1980, 17(1), pp. 43-58.
- [44] Shoop SA., "Finite element modeling of tire-terrain interaction.", *Engineer Research and Development Center*, Hanover, NH, 2001.
- [45] Darnell I, Hulbert GM, Mousseau CW., "An efficient three-dimensional tire model for vehicle dynamics simulations.", *Journal of Structural Mechanics*, 1997, 25(1), pp. 1-9.
- [46] Fervers CW., "Improved FEM simulation model for tire-soil interaction.", *Journal of Terramechanics*, 2004, 41(2-3), pp. 87-100.
- [47] Xia K., "Finite element modeling of tire/terrain interaction: Application to predicting soil compaction and tire mobility.", *Journal of Terramechanics*, 2011, 48(2), pp. 113-23.
- [48] Chan BJ, Sandu C., "Development of a 3-D quasi-steady-state tyre model for on-road and off-road vehicle dynamics simulations: Part II-off-road rigid wheel model.", *International Journal of Vehicle Systems Modelling and Testing*, 2014,

9(2), pp. 107-36.

- [49] Janosi Z, Hanamoto B., "Analytical determination of drawbar pull as a function of slip for tracked vehicles in deformable soils: Mechanics of Soil-Vehicle Systems.", Editone Minerva Tecnica, Torino, 1962.
- [50] Wong JY., "Terramechanics and off-road vehicle engineering: terrain behaviour, off-road vehicle performance and design.", Butterworth-heinemann, 2009.
- [51] Wong JY, Reece AR., "Prediction of rigid wheel performance based on the analysis of soil-wheel stresses part I. Performance of driven rigid wheels.", Journal of Terramechanics, 1967, 4(1), pp. 81-98.
- [52] Bekker, G., "Introduction to Terrain-Vehicle Systems", University of Michigan Press, 1969.
- [53] Schwanghart H., "Lateral forces on steered tyres in loose soil.", Journal of Terramechanics, 1968, 5(1), pp. 9-29.
- [54] Hettiaratchi DR, Reece AR., "The calculation of passive soil resistance.", Geotechnique, 1974, 24(3), pp. 289-310.
- [55] McKyes E., "Soil cutting and tillage.", Elsevier, 1985.
- [56] Hopfield JJ., "Artificial neural networks.", IEEE Circuits and Devices Magazine, 1988, 4(5), pp. 3-10.