DEGREE PROJECT IN MECHANICAL ENGINEERING,
SECOND CYCLE, 30 CREDITS
*STOCKHOLM, SWEDEN 2020*

# Traction Adaptive trajectory planning for autonomous racing


**FELICIA ABOUD VIEIDER**

**ANIRUDH NARASIMHA KULKARNI**



**KTH ROYAL INSTITUTE OF TECHNOLOGY**
**SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT**

# Traction Adaptive trajectory planning for autonomous racing

Felicia Aboud Vieider
Anirudh Narasimha Kulkarni

Master of Science Thesis TRITA-ITM-EX 2020:444
KTH Industrial Engineering and Management
Machine Design
SE-100 44 STOCKHOLM

**Examensarbete TRITA-ITM-EX 2020:444**

KTH Industriell teknik och management

# Greppadaptiv rörelseplanering för autonom racing

Felicia Aboud Vieider
Anirudh Narasimha Kulkarni

| Godkänt | Examinator<br>DeJiu Chen | Handledare<br>Lars Svensson |
|---|---|---|
| | Uppdragsgivare<br>AFRY AB | Kontaktperson<br>Anders Mattsson |

## *Sammanfattning*

De senaste åren har den autonoma fordons industrin genomgått en stor utveckling genom forskning, företag har gjort stora investeringar för att förbättra teknologin för att kunna nå den privata marknaden. Industrin och akademin jobbar fortfarande för att göra autonoma bilar säkra, pålitliga och robusta. Autonom racing tillhandahåller en plattform för att förbättra tekniken så att den kan utnyttja fordonets fulla fysiska förmåga i ett brett spektrum av driftsförhållanden. Flera funktioner krävs för att göra bilen autonom, detta arbete fokuserar på rörelseplaneringsmodulen för autonom racing. Vi har utvärderat hur rörelseplanerings algoritmen presterar vid användning av en dynamisk modell med dynamiska begräsningar.

Utvärderingen är baserad på ett ramverk för optimal rörelseplanering [1] vilken löser optimerings problemet genom användning av "Sampling Augmented Real Time Iteration (SAARTI) motion planning scheme". Fyra olika modeller jämfördes vilka inkluderade en dynamisk cykelmodell med både statiska och dynamiska begränsningar. De parametrar som påverkade prestandan identifierades, och avvägningen mellan modell komplexitet och planerings horisont undersöktes genom att studera skillnader i prestanda för olika parameter konfigurationer. Generaliserbarhet av resultaten undersöktes genom att studera prestandan för olika parameter konfigurationer under olika körförhållanden.

Batch simuleringar utfördes för att ta hänsyn till många olika scenarion, för att säkerställa att resultaten var så nära verkligheten som möjligt. Simuleringarna visade att användning av dynamiska begränsningar vid rörelse planering förbättrar prestandan jämfört med att använda statiska begränsningar vid extrema körförhållanden.

Observation av resultaten från simuleringarna visade att användning av den grepp adaptiva modellen resulterade i robust och konsistent prestanda. Att kombinera estimering av friktion och samtidigt ta hänsyn till en varierande normal kraft, ökar förmågan att planera för variationer i friktion, minskar chansen att bilen kör av vägen och förbättrar varvtiden.

KTH Industrial Engineering and Management

**Master of Science Thesis TRITA-ITM-EX 2020:444**

# Traction Adaptive trajectory planning for autonomous racing

Felicia Aboud Vieider

Anirudh Narasimha Kulkarni

| Approved | Examiner<br>DeJiu Chen | Supervisor<br>Lars Svensson |
|---|---|---|
| | Commissioner<br>AFRY AB | Contact person<br>Anders Mattson |

## *Abstract*

The autonomous driving industry has undergone leaps and bounds of research to reach the mainstream market, with major players investing heavily to improve the technology further. Industry and academia are currently working to make the technology safe, reliable and robust. Autonomous racing provides this opportunity, to improve the technology to the point, where it can utilize the full physical capability of the vehicle in a wide range of operational conditions. Multiple functionalities are required to make the car autonomous, this thesis focuses on the path planning module for autonomous racing. We evaluated the performance of dynamic models, using different adaptive dynamic constraints, implemented for path planning.

The evaluation is based on framework for optimization based motion planning[1] The optimization problem is solved by "Sampling Augmented Real Time Iteration (SAARTI) motion planning scheme". Four different models were studied during this thesis and include dynamic bicycle models, with static and dynamic constraints. Parameters affecting the planning performance were identified, and the trade-off between model complexity and planning horizon, was investigated by varying these parameters and the differences in performance was studied. The generalizability of results for different driving conditions was investigated for these parameter configurations.

Batch simulations were performed to account for various possible scenarios of different parameter configurations, to ensure results closest to reality. The simulation was conducted with, hardware in the loop setup running the planning node, to get a realistic estimation of the computation resources. Batch simulations were instrumental in showing interesting trends of how the input parameters affected the planning performance. Simulations provided extensive proof of the different dynamic constraints improving the planning performance, over the basic dynamic model under extreme driving conditions.

When reviewing the results from the simulations, the traction adaptive model showed robust and consistent performance. The results showed that combining friction estimation and load adaption, increased the ability to plan for local variations, reduced failure and improved laptime. When designing a planner for a race car that is exposed for local variations, the friction estimation is needed to reduce errors and the load adaption is needed for better handling of the vehicle to perform critical maneuvers.

# Acknowledgements

We would like to express our appreciation and gratitude to all the people who made this thesis possible by their contributions. Firstly, we thank our KTH supervisor Lars Svensson who contributed with both the planning framework and his time to provide feedback throughout our work. The discussions about theory and results were instrumental in completing the thesis. We would also like to thank our AFRY supervisor Stefan Ionescu and manager Anders Mattson for welcoming us at the office and supporting our work by weekly updates to help our project go forward and by giving us feedback to make improvements on the work done. Finally, we would like to thank KTH, our course supervisor Fredrik Asplund for helping us developing the research question, our examiner DeJiu Chen, and our opponents Sara Soltaniah and Johanna Ollas for their feedback to finalize and refine our work.

# Contributions

This thesis was completed by the authors as part of their Masters of Science in Mechatronics at Kungliga Tekniska Högskolan (KTH). When preparing for the simulations the distribution of work is 50% for both members. All sections were worked through together. Work was done in collaboration using paired programming and discussion. Both members have an equal understanding of the system and theory governing it. The largest aspects of individual work was reading papers for the theoretical section or writing small functions. In addition, the simulations was done by Anirudh and data visualizations was done by Felicia.

# Contents

# List of Figures

# List of Tables

# Acronyms

**DoF** Degrees of freedom. 8

**FSSIM** Formula Student Simulator. 6, 7, 26, 27

**MPC** Model predictive control. 2, 9, 10

**NMPC** Nonlinear Model predictive control. 10, 11

**ROS** Robot Operating System. 7, 25–27

**RTI** Real Time Iteration Sequential Quadratic programming. 11, 18, 19

**SAARTI** Sampling Augmented Real Rime Iteration. 11

**SLAM** Simultaneous localization and mapping. 6

# Chapter 1

# Introduction

## 1.1 Background

Autonomous driving has undergone a lot of research in recent years, to make it safe and reliable [2]. Some functionalities involved in autonomous driving, include perception, localization, path planning, and path following. The dream of large scale commercial deployment of autonomous cars is closer to reality today, with the industry taking a keen interest. AFRY as an engineering, design, and advisory company, works together with other companies to improve and develop the technology. This thesis was conducted with AFRY Sweden and focuses on path planning for autonomous racing to improve the autonomous driving competence of the company.

In autonomous racing, cars drive in an environment void of human intervention and therefore provides a platform to develop and test new technologies, pushing the limits of vehicle capabilities. This setup ensures a low risk of human injuries [3]. Racing is a challenging task for an autonomous system. Primarily due to the need for handling the vehicle close to its stability limits to reach high speeds, in cases like sharp corners or slippery surfaces. Also, the dynamic limits of the vehicle are time-dependent due to local variations in terms of varying road conditions, and high accelerations causing varying distribution of normal force. Implementation of a real-time system with the need for fast computations on an embedded platform with limited computational power makes the task even more challenging [4].

Trajectory planning is essential in autonomous vehicles, a trajectory describes how configurations, for example, position, and velocity of the vehicle will

evolve with time. By using the input from the surroundings it is possible to predict feasible trajectories for the vehicle. Feasible and optimal path planning has been studied extensively, and a large number of algorithms have been developed [5]. This thesis will be based on the thesis done by Ionescu and Jonsson [6], which used an algorithm based on Model predictive control (MPC) for trajectory planning. One of the factors affecting this planning method is, the selection of the vehicle model. Ionescu and Jonsson [6] showed improved performance for planning, in terms of laptime, when increasing the model complexity from a kinematic bicycle model to a dynamic bicycle model. This thesis [6] stresses the need for having a more complex way of describing the constraints on the tire forces to improve the constraints at high accelerations.

Another factor affecting performance is the planning horizon[6], which decides how far ahead in time planning will be done. The previous thesis investigated the trade-off between model accuracy and planning horizon, by employing complex models. We extend this work by using a dynamic model with dynamic constraints adapting to time-varying traction limitations, and investigate if it improved the planning performance in real-time.

## 1.2 Problem description

Optimal performance in autonomous racing is achieved when the vehicle operates close to its physical limits [7]. To fully use the physical capabilities of the vehicle an accurate description of the dynamics and dynamic constraints are required [1]. The dynamic model used by Ionescu and Jonsson [6] is based on the assumption of static force constraints on the tires. This model will only be an accurate representation of the system when the vehicle is operating in static traction conditions of the road.

Increased model complexity for trajectory planning will permit the cars to reach their physical limits in terms of speed and control. However, given limited computational resources, a more complex vehicle model would also increase the computational time which would negatively affect performance in real-time applications. Planning horizon length decides for how many steps the planning will be done, a high planning horizon allows the planner to see further and allow for better planning. At the same time, longer planning horizon leads to more computations and a longer planning time. Ionescu and Jonsson [6] show that the optimal planning horizon is varying with the shape of the track [6]. Planning horizon length and model accuracy, both affect the

computation time and performance of the trajectory planner, meaning there is a trade-off between model accuracy and planning horizon.

Most models used for trajectory planning is based on assuming a constant tire-road friction coefficient and normal forces acting on the tires. Although the dynamic limits are typically varying with time due to local variations in terms of road, tire condition, temperature and distribution of the normal force. Hence, Svensson et al. [1] proposed to incorporate, an adaptation of tire force constraints of the vehicle at run time for motion planning to avoid obstacles. The traction adaptive algorithm allowed the vehicle to avoid obstacles with increased capacity, in comparison to an equivalent non-adaptive control scheme [1]. Thus we hypothesize that including traction adaptation in trajectory planning for racing will improve performance, in terms of increased laptime while keeping the vehicle inside the track, compared to using static constraints.

## 1.3 Hypothesis

Four types of constraints on the maximum tire force will be examined:

- Static constraints
- Dynamic constraints depending on friction estimation
- Dynamic constraints depending on the normal load
- Dynamic constraints depending on both friction estimation and varying normal load

Using dynamic constraints require more computations, meaning a limited computation force could result in a longer computation times.

We hypothesize that all models will have decreased failure with higher planning horizons although there will be a trade-off when the planning time gets close the sample time, and failure will start to increase with higher planning horizons. The model using static constraints will have a lower increase in planning time and therefore be less negatively affected by increased planning horizon.

Using static constraints will allow for higher planning horizon at the point of failure compared to dynamic constraints, but will have high failure when there are high variations in surface friction. Models using friction estimation will

lead to lower failure due to the decreased environmental error. Models with dynamic constraints considering varying normal load without friction estimation permit higher accelerations resulting in shorter laptime. Models running close to physical limits become prone to model errors, hence resulting in higher failure when considering varying normal load. Using dynamic constraints with both friction estimation and adaption to normal load will give lower failure compared to only normal load adaption and a lower laptime compared to only using friction estimation.

## 1.4 Research Question

Previous research[6] has shown that increased accuracy and planning horizon improves planning performance while causing an increase in computation time, when given limited computational resources. Therefore, considering a real-time application. Increased computation time will at some point affect the vehicle's performance negatively, as the benefit of high plan quality will be outweighed by extensive planning delay. Ionescu and Jonsson [6] showed that the optimal planning horizon is varying with the features of the racing track, and that, the S-curve, a U-turn, and a double U-turn had the most impact on planning horizon.

Optimal trade-off between model accuracy and planning horizon length should optimize laptime and failure. At the same time, utilizing all computational resources available should at times mean that the vehicle is operating just at the point of failure, i.e. that the latency and planning are both just enough to keep the vehicle from going off the track. Given these circumstances:

- What are the implications of using a dynamic model with dynamic constraints (traction adaption) compared to a dynamic model with static constraints?
- Investigated for the above mentioned race track features, is it possible to generalize these implications for an arbitrary race track?

## 1.5 Methodology

Using case study is feasible when studying "a contemporary phenomenon within its real-life context, especially when the boundaries between the phenomenon and context are not clearly evident" [8]. To answer the research ques-

tion the performance of the path planner was to be evaluated during different scenarios and input parameters. Case study was a suitable choice of research method since this method allowed control of the real-life context to investigate how the shape of the track affects the result. Case studies permit in-depth study of specific cases [9], each case was evaluated by means of quantitative metrics like laptime and success rate of path planning.

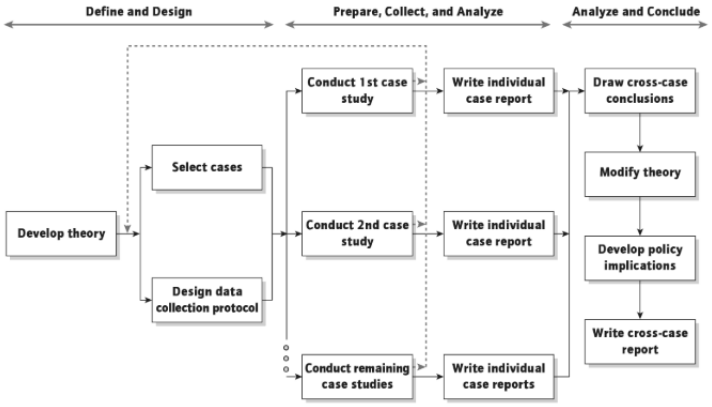



Figure 1.1: Design flow of general Case Study [8]

Figure 1.1 shows the important steps that were required in making a case study. During the initial design phase, a hypothesis was set up from the existing literature. The models and planning algorithm was set up during this phase. The next step involved the selection of different cases and setup of the simulation environment. The research investigated the implications of different vehicle models and if these implications were general for various tracks. Hence, different track shapes were chosen as cases. Different sets of input variables and models were evaluated for these track sections by running simulations. Inputs included planning horizon and friction estimate accuracy, and the output was quantified as laptime and fail rate. The different parameters were plotted to observe interesting behavior and relations, which provided quantitative data for basing qualitative analysis. The qualitative analysis answered the initial hypothesis and contributed to the research.

Case studies needed to be competent, current, and research worthy. An effective way to ensure this was to have studies on relations between interesting input parameters, and how varying input parameters affect the planning performance. Therefore numerical experiments were used to evaluate different algorithm configurations for each case. The following checklist [10] was used

as a baseline when planning the experimental setup used for data collection.

1. Define the objectives of your experiment.
2. Define all sources of variation, including:
   (a) Treatment factors and their levels.
   (b) Experimental units.
   (c) Blocking factors, noise factors and constraints.
3. Choose a rule for assigning the experiment units to the treatments.
4. Specify the measurements to be made, the experimental procedure and the anticipated difficulties.
5. Run a pilot experiment.
6. Specify the model.
7. Outline the analysis.
8. Calculate the number of observations that need to be taken.
9. Review the above decisions and revise, if necessary.

A detailed description of the important steps is given in chapter 4.

## 1.6 Scope and Limitations

Autonomous driving is a huge area of interest, however, the main focus will be on the planning module, with inputs from other modules used for localization of the vehicle like perception and Simultaneous localization and mapping (SLAM), being considered constant and ideal. This thesis is relying on the simulation environment Formula Student Simulator (FSSIM) to match the simulation to the actual vehicle performance. The tracks for experimentation are created in the simulation environment, and boundaries of the track are implemented using cones for each side of the track, to ease implementation. Different road conditions are considered for evaluating the planning performance by varying the surface friction. However, the road is considered to be flat, with no banking. The module used to estimate friction is not implemented in this work, and is assumed to be ideal. The value used for friction estimation is given by using the real friction value from the simulation environment. The complete hardware implementation requiring multiple modules, on 1:12 scale car for autonomous driving will not be done during this thesis.

## 1.7 Implementation

The initial hypothesis was validated by employing batch simulations using Robot Operating System (ROS) and FSSIM. Batch simulations were used for validating our initial hypothesis as it permitted different parameters affecting the planning performance to be modeled in form of different levels of treatment and then investigate how a single or a combination of parameters affected the overall planning performance. The number of simulations performed decided the generalisability of the results, and a higher number of simulations correlated to the results being closer to reality. This permitted the results to be generalized for different sections of the track and models in the study. Batch simulations was performed using a hardware in the loop setup, with the planning module running on a Jetson TX2 microcontroller and host machine running the other nodes essential for autonomous driving.

## 1.8 Verification

This project compares performance at the point of failure using different parameter configurations for path planning. Laptime and failure were used as success parameters, with planning time decides the point of failure. These parameters were used as measurements to verify our initial hypothesis and answer the research question.

## 1.9 Ethics

The trolley problem [11], [12] of decision making appears in motion planning, and show how ethical questions can arise when there is an interaction between human and autonomous vehicles. The trolley problem refers to the dilemma situations involving two groups of people, where it is unavoidable that one of the groups get harmed to spare the other. For example should the car manage to avoid pedestrians on the road and harm the driver or harm the pedestrians to spare the driver? The moral question is how should we program the car to make these decisions, and who is responsible in case of failure? Paper [13] proposes a way in which the stakeholders can give weights to ethical decisions for different autonomous robots. Paper [14] proposes hierarchical decision making to reduce fatal accidents. These methods can be utilized to build algorithms capable of ethical decision making and they can be tested by means simulation experiments.

# Chapter 2

# Related Work

During this thesis, we investigate the implications of using a more complex model when using a model predictive controller for trajectory planning. This chapter describes previous research where model predictive control is used for trajectory planning.

## 2.1 Parameters affecting trajectory planning

### 2.1.1 Model complexity

Including more complex dynamics to describe the vehicle increases the model accuracy. Although it is not certain that this would lead to better performance of the planning since higher complexity could also affect the robustness of the planner. Previously, research has been done to compare different levels of model fidelity, by comparing the dynamic model with the kinematic model. Ionescu and Jonsson [6] examine the trade-offs between model accuracy and planning horizon. Having a more complex model improved racing performance and showed that consideration of lateral tire forces is needed at high speeds conditions. Ionescu and Jonsson [6] concluded that factors like sample time and planning horizon had more impact on the computation time than model complexity [6]. Liu et al. [15] compared a dynamic bicycle model to a four-wheel model with 14 Degrees of freedom (DoF) and concluded that by using steering limits of the four-wheel model, the bicycle model could perform similar to the four-wheel model. The research established the importance of a vehicle being able to operate close to performance limits for better performance. While they concluded that it might not be as important, having these complex models inside the planner as the constraints to set limits to the control

inputs [16]. John K. Subosits and J. Christian Gerdes [17] present a replanning algorithm using a point-mass model for planning to minimize the computational force needed for the planning. The algorithm uses a simple model for planning while adding longitudinal weight transfer and road topography to model the vehicle limits. This scheme successfully manages to avoid obstacles in real-time scenarios [17], although it is still dependent on a nominal path to be computed offline for the whole track. Additional studies have been made to further investigate what levels of model complexity is needed, Liu et al. [18] performed a study to determine the model fidelity required in an MPC to plan online trajectories for obstacle avoidance. The paper presents a comparison of four variations of the dynamic bicycle model, the models have either a linear tire model or a pure-slip Magic Formula tire model [19] in combination with constant force or varying force on the axle loads. The results showed that the dynamic model with linear tire force and constant axle load had a good performance for low speed situations but fails when moving at a higher speed of 30 m/s. Therefore, a nonlinear tire model and longitudinal load transfer were shown to be important factors to include in the dynamic bicycle model to accurately predict vehicle trajectory.

### 2.1.2 Local variations

The previous section showed that the model plays an important role in optimization based planning and the dynamic capabilities of road vehicles are affected by local variations [1]. Papers [20] [1] talk about the trade-offs between model accuracy, planning horizon and computation time. Kabzan et al. [7] [3] further emphasize the challenging trade-off between computation time and model accuracy given by the racing environment. Ionescu and Jonsson [6] showed that a longer planning horizon has no impact on performance if the race track has a simple form as a straight line, while in the case of a more complex track like a double U-turn a large planning horizon led to an improvement on the planning performance [6]. An increased planning horizon in turn negatively affected the computation time.

Other factors affecting performance in trajectory planning, include varying track surface and axle load variations due to longitudinal accelerations, which cause the maximum tire force to vary with time, especially in racing applications [21]. Svensson et al. [1] propose a model predictive adaptation of tire force constraints of the vehicle at run time for motion planning to avoid obstacles. By adding online friction estimation the traction adaptive algorithm

allowed the vehicle to avoid obstacles with increased capacity, in comparison to equivalent non-adaptive control scheme, this was confirmed by extensive numerical simulations [1]. The work [1] assumes a state of the art friction estimation and focuses on the planning module. Study [22] presents the different available model based and experimental based techniques for tire-road friction estimation.

As seen, previous research has been done to increase the ability to plan feasible trajectories for autonomous vehicles. This thesis further investigated combining friction estimation and longitudinal load transfer for a dynamic bicycle model using a simplified nonlinear tire model, to study the effects of their incorporation into a trajectory planner used for racing.

## 2.2 Optimisation

Optimisation based planning involves minimising a cost function, that accounts for the behavior of the system. Used for racing, the optimisation function needs to be set up considering, maximisation of the distance traveled [23] [4] and needs to be solved by numerical methods such as dynamic programming [24]. Previous research has been done to improve planning performance without a significant increase in computation time. Kabzan et al. [7] use a simple model and tighten the dynamic constraints to achieve safe driving behavior, however, this limited the racing performance. A learning based control approach was implemented to overcome this limitation, based on information from previous laps to improve current performance [7]. Gao [25] suggests the usage of hierarchical MPC, which is simpler than nonlinear MPC [26], for real-time implementation. The planner used a complex 4 wheel model for local planning and a simpler model for global planning. The hierarchical planner showed good performance in avoiding obstacles while following a nominal path. However, it's not suitable in the racing case as it is not a general solution for trajectory planning and is not optimal due to the high-level planner based on a simple model. The bicycle model in the thesis [6] results in a nonlinear optimisation function and the real-time implementation suffers the issue of local minima, especially when the algorithm needs to make discrete decisions, for example choosing on what side to pass an obstacle. Liniger et al. [4] solved this by using a high level planner creating a corridor where there is only one possible way of passing the obstacles. Further, this paper compares the performance of a two-level path planner with the Nonlinear Model predictive control (NMPC). The nonlinear optimization problem was solved by an approach

called Real Time Iteration Sequential Quadratic programming (RTI) and is a computationally efficient way of solving the NMPC. This is done by linearizing the dynamics and form a second order approximation of the cost function using the solution from previous iteration[4]. The use of NMPC resulted in better performance than the two-level planner. Drawbacks of using the NMPC include the computational cost and it's sensitivity to measurement errors, due to the dependency on the previously planned path. If the error between the planned and traveled path is large, the previous solution might not be a good initial guess and can cause the planner to get stuck in local minima. An issue with RTI when adding online adaptation to the model is that the time-varying constraints can cause the input sequence from the previously planned path to become unfeasible. To model traction variations as time-varying constraints and to address the issue of local minima, without loss of optimality, Svensson et al. [1], proposed RTI scheme called Sampling Augmented Real Rime Iteration (SAARTI), based on sequential quadratic programming for real-time implementation. Using trajectory rollout of additional candidates for initial guess is added at each iteration and selected by checking the conditions and evaluating the cost function, the risk of getting stuck in local minima is reduced [1]. In this thesis, the SAARTI based planning algorithm will be used to combine friction estimation and longitudinal load transfer into the dynamic model, and performance will be evaluated by batch simulations for different model configurations.

# Chapter 3

# Theoretical Background

This chapter presents a comparison between the dynamic model with fixed constraints that were used as part of the research in [6], and the dynamic model with constraints adapting to the online estimation of tire-road friction coefficient $\mu$ and varying axle loads. The dynamic model from the work of Ionescu and Jonsson [6] serves as a baseline to compare the performance of the other models. Four instances of the dynamic bicycle model were considered, one with static constraints, the Non-adaptive model (NA), one with constraints updated by friction estimation, the Friction adaptive model (FA), one with constraints updated by pitch dynamics, the Load adaptive model (LA) and one with constraints updated by both friction estimation and pitch dynamics, the Traction adaptive model (TA). A table presenting the four models is seen in Table 3.1, the naming of models in this table will be used to describe these models henceforth.

| Vehicle model: | NA | LA | FA | TA |
|---|---|---|---|---|
| $\mu$ estimation | No | No | Yes | Yes |
| Load adaption | No | Yes | No | Yes |

Table 3.1: Table showing the difference between the four models

The chapter describes the theory for these models and the optimisation based planner implemented for autonomous racing during this research.

## 3.1 Vehicle model

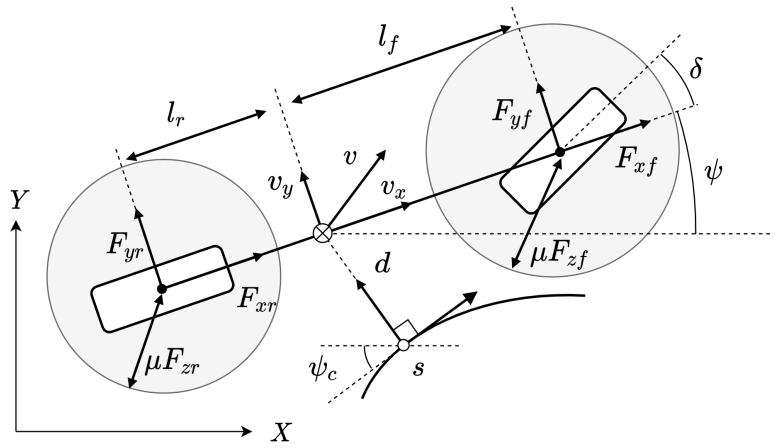



Figure 3.1: Illustration of the bicycle model showing the relation between Global X-Y frame and Frenet frame. $s$ denotes the progress along the centerline of the lane

### 3.1.1 Frenet frame

The Frenet frame is an alternate coordinate system to the Cartesian system, that simplifies the representation of a system moving along a continuous and differentiable curve in three-dimensional Euclidean space, by the use of variables $s$ and $d$, shown in Figure 3.1. $s$ represents the distance traveled along the curve and $d$ represents the perpendicular distance to the curve. Using the frenet frame to describe the vehicle motion along the road centerline makes it possible to formulate the optimization problem for racing by maximizing progression along $s$ while staying inside max and min values of $d$. All parameters measured as part of the perception system are passed in terms of global X-Y coordinates and is transformed to the local frenet frame, this permits the car dynamics to be described relative to the centerline.

### 3.1.2 Bicycle model

The vehicle is modeled by representing left and right wheels at the front and rear axles, as single wheels at each of the axle centers. This model is valid by the assumption that the roll dynamics on the left are canceled out by those on the right. This way of modeling a car is called a bicycle model. Equal steering angle for both left and right wheel is assumed, however, this is not always

the case. Although, in general, these angles will be approximately equal [27]. The bicycle model has previously been used in the field of autonomous driving [28], [29], [1], [6], and has become a standard of sorts.
The representation of a bicycle model driving in a track is shown in Figure 3.1. $\delta$ denotes the steering angle, while $\psi$ and $\psi_c$ denote the vehicle heading angle and the angle of path tangent at $s$ measured relative the X-axis, respectively. $v$ is the vehicle velocity, and distance from the center of gravity (CG) to the front and rear axles are given by $l_f$ and $l_r$. Lateral forces are denoted as $F_{yf}$ and $F_{yr}$, longitudinal forces represented as $F_{xf}$ and $F_{xr}$, and rear-wheel is denoted by the subscript *r*, while the front wheel is denoted by *f*. The coordinates of the vehicle are represented in both the traditional X-Y system and the frenet frame.

### 3.1.3 Dynamic model

Experiments as part of this thesis were conducted using dynamic vehicle models. Consider the dynamic bicycle model represented in Figure 3.1, the vehicle orientation relative to the centerline tangent at $s$ is given as

$$\Delta\psi = \psi - \psi_c \tag{3.1}$$

whereby the vehicle dynamics can be described using the frenet frame as

$$\dot{s} = \frac{v_x \cos(\Delta\psi) - v_y \sin(\Delta\psi)}{1 - d\kappa_c} \tag{3.2a}$$

$$\dot{d} = v_x \sin(\Delta\psi) + v_y \cos(\Delta\psi) \tag{3.2b}$$

$$\Delta\dot{\psi} = \dot{\psi} - \kappa_c \frac{v_x \cos(\Delta\psi) - v_y \sin(\Delta\psi)}{1 - d\kappa_c} \tag{3.2c}$$

$$\ddot{\psi} = \frac{1}{I_z}(l_f F_{yf} - l_r F_{yr}) \tag{3.2d}$$

$$\dot{v}_x = \frac{1}{m}(F_{xf} + F_{xr} - D_a v_x^2) \tag{3.2e}$$

$$\dot{v}_y = \frac{1}{m}(F_{yf} + F_{yr}) - v_x\dot{\psi} \tag{3.2f}$$

where $\dot{\psi}$, $v_x$ and $v_y$ denotes yaw rate, longitudinal velocity and lateral velocity, $m$ and $I_z$ represent the mass and moment of inertia of the vehicle and $\kappa_c$ is the curvature of the centerline at $s$. The term $v_x\dot{\psi}$ represents the centripetal acceleration and $D_a v_x^2$ represents the drag force component acting on the vehicle.

Note that the road is flat during this work, therefore bank angle and inclination of the road are assumed to be zero. State and control vectors are selected as

$$x = [s, d, \Delta\psi, \dot{\psi}, v_x, v_y] \tag{3.3}$$

and

$$u = [F_{yf}, F_{xf}, F_{xr}]^T \tag{3.4}$$

which allows the state space equation with the control and state vectors can be written as

$$\dot{x} = f(x, u) \tag{3.5}$$

### Slip Angle

The lateral force on the wheel depends on the characteristics of the tire like the deflection of the treads in the contact between road and wheel. As a result the lateral force on a tire is proportional to the slip angle, when the slip angle is small[27]. Hence, in order to model the lateral force acting on the rear wheel, the slip angle needs to be defined.

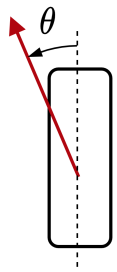



Figure 3.2: An illustration of the velocity vector (represented by the red arrow) and the angle relative the rear wheel

The vehicle slip angle is given by

$$\alpha_r = \theta_r \tag{3.6}$$

where $\alpha_r$ is the slip angle at the rear wheel and $\theta_r$ is the angle of the velocity vector at the rear wheel, shown in Figure 3.2. The direction of the velocity vector at the rear wheel can be described as

$$\theta_r = \arctan(\frac{v_y - l_r\dot{\psi}}{v_x}) \tag{3.7}$$

The variation of lateral tire force with slip angle is shown in the figure below.

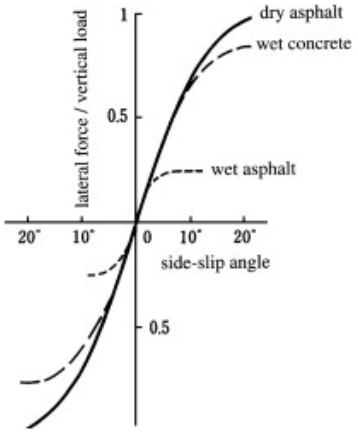



Figure 3.3: Lateral tire force as a function of slip angle for different road conditions. [30]

As seen in Figure 3.3 for small slip angles, the lateral force is proportional to the slip angle [27], if small slip angles are assumed the lateral force can be described as

$$F_y = C_\alpha \alpha \tag{3.8}$$

where $C_\alpha$ is the cornering stiffness of the wheel. Given equations (3.6),(3.7) and (3.8), the lateral force on the rear wheels is described as

$$F_{yr} = 2C_\alpha F_z \arctan(\frac{l_r\dot{\psi} - v_y}{v_x}) \tag{3.9}$$

a factor 2 is included to account for both tires.

**Force constraints**

The longitudinal and lateral tire forces $F_{yf}$, $F_{xf}$ and $F_{xf}$, and the relation between slip angles and tire forces exist as previously seen. Since these are used as the control parameters, basic Newtonian mechanics is sufficient to get the upper bound to set constraints on these control parameters. Upper bound for the horizontal force acting on the front and rear wheels are given by

$$F_i \leq \mu F_{zi}, i \in [f, r] \tag{3.10}$$

with

$$F_i = \sqrt{F_{xi}^2 + F_{yi}^2} \tag{3.11}$$

$f$ and $r$ stands for front respectively rear wheels, $F_{zi}$ represents the normal force on each of the wheels, and $\mu$ is the tire-road friction coefficient. The condition in (3.11) provides a boundary called the friction circle since it forms a circle with radii $\mu F_{zi}$, this is illustrated by the circles around the tires in figure 3.1. Both $\mu$ and $F_{zi}$ is dynamically changing with time, the friction varies as a result of the vehicle driving on different surfaces and $F_{zi}$ is dependent on the vehicle's longitudinal acceleration. In order to adapt to changes in the friction, a friction estimation algorithm needs to be implemented to predict the conditions ahead of the vehicle. For the purpose of this project, the estimation of the friction will be given by the simulation environment for the models using friction estimation, models without friction estimation will take an assumed value $\mu_a$ for the friction coefficient.

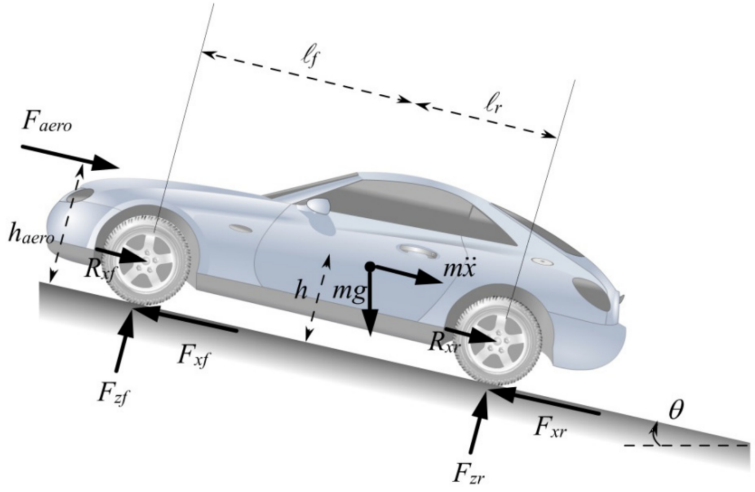



Figure 3.4: Force distribution - side view [27]

The time-varying axle loads can be modeled by moment balance about the two contact points, seen in Figure 3.4, gives the normal forces $F_{zf}$ and $F_{zr}$ at the front and rear wheels as

$$F_{zf} = \frac{1}{l_f + l_r}(mgl_r) \tag{3.12}$$

$$F_{zr} = \frac{1}{l_f + l_r}(mgl_f) \tag{3.13}$$

and gives maximum value on $F_{zi}$ for the models without pitch dynamics and therefore constant load on the axes. To account for the longitudinal load trans-

fer at high accelerations the following relationships are used to calculate the normal force on the front and rear axles taking into account the longitudinal load transfer effects

$$F_{zf} = \frac{1}{l_f + l_r}(m\dot{v}_x h - mghsin\theta + mgl_r cos\theta) \tag{3.14}$$

$$F_{zr} = \frac{1}{l_f + l_r}(-m\dot{v}_x h + mghsin\theta + mgl_f cos\theta) \tag{3.15}$$

and sets the maximum value on $F_{zi}$ for the models with pitch dynamics.

## 3.2 Optimisation based planning

The thesis utilizes the optimisation based planning algorithm from paper [1]. Optimisation based motion planning involves solving a Constrained Finite Time Optimal Control Problem (CFTOC), at each planning iteration in a receding horizon manner [31] [1]. The friction variation, motion planning CFTOC problem, is as follows:

$$\min_{u_{0|t},..,u_{N-1|t}} J(x_{k|t}, u_{k|t}) = p(x_{N|t}) + \sum_{k=0}^{N-1} q(x_{k|t}, u_{k|t}) \tag{3.16}$$

$$\begin{aligned} \text{such that,} \quad & x_{k+1|t} = f(x_{k|t}, u_{k|t}) \\ & u_{k|t} \in U_{k|t} \\ & x_{k|t} \in X_{k|t} \\ & \forall k = 0, ...., N-1 \\ & x_{0|t} = x_t, x_{N|t} \in X_{k|t} \end{aligned}$$

$[u_{0|t}, ...., u_{N-1|t}]^T$ is the input applied to vehicle model $f$, to generate N+1 states $[x_{0|t}, ...., u_{N|t}]^T$. Functions $p(\cdot)$ and $q(\cdot,\cdot)$ denote the positive definite running cost and terminal costs respectively. $J(\cdot)$ represents the overall quadratic cost function. Each of the dynamic vehicle models as discussed earlier has its' own set of states $x$ and inputs $u$. The predicted trajectory is written as $T_t = (x_{x|t}, u_{k|t}), k \in (0, ...., N)$. The predicted states are real at each step and represents collision free paths for the vehicle to follow. $T_t^* = (x_{k|t}^*, u_{k|t}^*, k \in (0, ..., N))$.

### 3.2.1 RTI algorithm

A common method of solving the previously defined optimisation problem in real-time is Real Time Iteration Sequential Quadratic programming (RTI). In

RTI, an initial guess is obtained from $T_t$ and dynamics are linearised by the solution obtained from the previous iteration $T^*_{t-1}$. Finally, the state and input constraints are expressed as linear inequalities.

The solution is unsolvable in cases when $T^*_{t-1}$ is infeasible and not optimal, at iteration step t and this leads to some drawbacks of RTI. Unsolvable optimisation problem points to the lack of a control signal $U_{k|t}$ at this iteration step. The optimal solution from previous time iteration $T^*_{t-1}$ may not be collision free due to a dynamic environment. RTI algorithm is prone to local sensitivity as it involves discrete decision making (e.g, whether to go left or right of an obstacle) and at rapid changes to the problem constraint (relevant for traction adaption).

### 3.2.2 sampling augmented real time iteration (SAARTI)

SAARTI overcomes the above mentioned drawbacks of RTI, by the introduction of sampling. SAARTI solves the quadratic cost function using the same planning horizon N and discretization step $T_s$ [1], the cost function is described as

$$J(x_{k|t}, u_{k|t}) = x^T_{N|t} Q_N x_{N|t} + \sum_{k=0}^{N-1} x^T_{k|t} Q x_{k|t} + u^T_{k|T} R u_{k|t} \tag{3.17}$$

with $Q_N > 0$, $Q > 0$, and $R > 0$, representing the tuning matrices. A dynamically feasible initial guess is generated from $T^*_{t-1}$, by projecting onto current time step $U_{k|t}$ and forward integration of dynamics. Additional guesses created from trajectory rollout [32]. The best guess is selected by evaluating the constraints and the cost function. The QP is then formulated around the initial guess and solved with the constraints. The process is repeated with closed loop control. The algorithm is summarised in the figure below,

**Algorithm 1** The SAA-RTI Algorithm

**Input:** $x_t$, $X^\star_{t-1}$, $M$, $O$
**Output:** $X^\star_t$

$\mu_t \leftarrow$ identifyFrictionCoefficient($x_t$)
$\mathcal{U}_t(\mu_t) \leftarrow$ computeAdaptiveConstraints($\mu_t$)
$\mathcal{T}_t \leftarrow$ sampleStateTrajectories($x_t$, $M$)
**for** each trajectory $\hat{X}^i_t$ in $[\mathcal{T}_t, X^\star_{t-1}]$ **do**
  **if** (chkConstr($\hat{X}^i_t$,$\mathcal{U}_t(\mu_t)$) $\wedge$ chkColl($\hat{X}^i_t$,$O$)) **then**
    $J(\hat{X}_t) \leftarrow$ evaluateCost($\hat{X}^i_t$)
  **end if**
**end for**
$\hat{X}^\star_t \leftarrow$ selectLowestCost(arg $J(\hat{X}^\star_t)$)
$A_{k|t}, B_{k|t} \leftarrow$ linearizeDynamicModel($\hat{X}^\star_t$)
$\hat{\mathcal{X}}_{k|t} \leftarrow$ computeStateConstraints($\hat{X}^\star_t$, $O$, $M$)
$X^\star_t = (x^\star_{k|t}, u^\star_{k|t}) \leftarrow$ opti($\hat{X}^\star_t$,$\mathcal{U}_t(\mu_t)$, $\hat{\mathcal{X}}_{k|t}$, $A_{k|t}$, $B_{k|t}$)
**return** $X^\star_t$

Figure 3.5: Sampling augmented adaptive RTI algorithm [1]

# Chapter 4

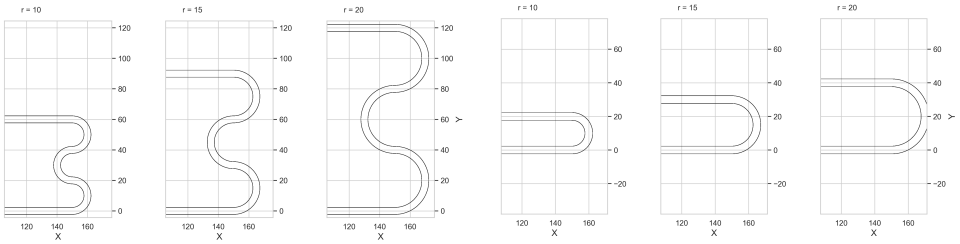
# Implementation

The implementation utilized for batch simulations and setup of experiments is described in this chapter. The initial sections deal with the reasoning for choosing the different parameter configurations and the latter end of the chapter deals with the implementation.

## 4.1 Experimental Setup

### 4.1.1 Selection of cases

Different track sections that affected the planning performance were identified. Adaption of friction and pitch dynamics were hypothesised to permit better vehicle handling in cornering at higher speeds. U and S sections of track provided a way of testing the vehicle handling under extreme cornering conditions with different complexities. U-section in real races, would start typically with a long straight stretch to accelerate, followed by quick cornering and then some accelerating stretch to follow up. S-section, is similar to U-section, except that it has two back to back cornering sections. These sections permit some racing action, given equivalent car and driving parameters, and hence are of interest. S- curve, formed by means of consecutive U turns, was opportune to investigate, if there were any problems due to build up of errors from earlier sections of the track and reasons for parameters like model complexity and planning horizon, affecting it. The effect of planning horizon on the performance and the effect of different models on the physical limits of speed, were tested by the use of tracks of different radii, for both U- and S-sections. The tracks used for testing is shown in Figure 4.1




(a) S-shape (b) U-shape

Figure 4.1: The S- and U- curves with different radius which form the cases of the case study, the radius of the curves is denoted as r

### 4.1.2 Selection of input parameters

The performance of TA, FA, and LA were evaluated individually and also compared with a generic model, to conclude how the added dynamics contributed to the performance. NA provided a baseline for performance measurement, which is a widely used vehicle model in the research field. Simulations were done for the different models, using equivalent parameter sets to ensure a fair trial for comparison of results on common grounds.

By identifying different sources of variations the input parameters of interest and the levels of variation needed were defined. The purpose was to find what input parameters affected the performance of the path planner. Parameters were selected in considerations of previous research and the models of interest. Previous work has shown that the planning horizon has an impact on the planning performance[6], and using standard deviation of the tire-road friction coefficient was chosen to study the effect of the friction estimation. The standard deviation values was used to generate friction values for different parts of the track by draw random samples from a normal distribution. Some questions that were investigated at this stage was, what is the maximum and minimum value, and what step size should be used to ensure that interesting behavior isn't missed. The possible levels of variations are also limited by the number of simulations that were possible to achieve under the restricted time frame.

Values for the planning horizon was decided by running test simulations for TA. The planning horizon denoted as N is the number of states to be generated by the planner. Figure 4.2 shows the failure for TA using different planning horizons and the blue patch represents a 68% confidence interval of failure.

At the horizon N equal to 30, a sudden increase of the failure can be observed, and running simulations with a failure of 100% would not generate any useful data to analyze. Therefore, the maximum value of N was set to 30. At N equal to 15 a small increase in failure is observed and is therefore chosen as the minimum value of N. The difference between consecutive N values decided to be five, to capture the entire bandwidth of values between the upper and lower bounds while ensuring a feasible number of simulations.

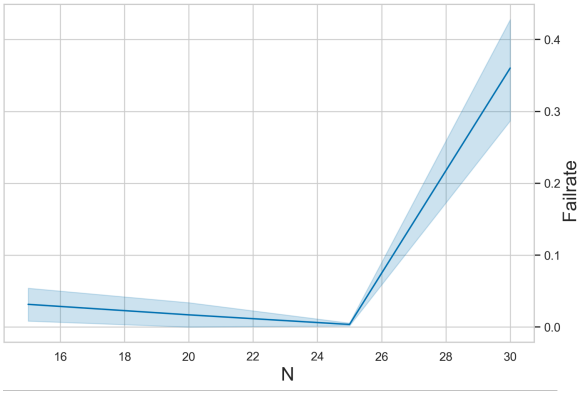



Figure 4.2: Failure for different N values using TA

NA was hypothesized to be most affected by changes in the tire-road friction coefficient. Hence, the values of the standard deviation of the road friction coefficient, denoted as $\mu$-$sd$, was selected by running simulations for NA. The variation of the friction coefficient as the car progresses along the track is shown in Figure 4.7. Figure 4.3 shows difference in failure for the chosen values on $\mu$-$sd$.

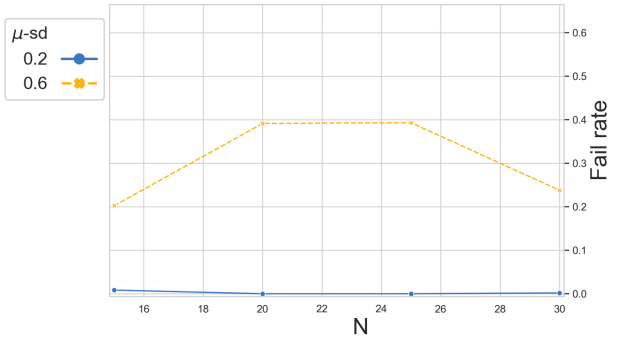



Figure 4.3: Effect on NA when varying $\mu$-$sd$

The failure for tracks with friction coefficient deviation of 0.2 is close to zero and low enough to see a good performance for NA. The failure increases as

the variance increases at a value of 0.6 a distinct increase in failure was observed and about 40%. To limit the number of parameter configurations while observing interesting behavior and ensure timely completion, the variances in friction values were fixed to 0.2 and 0.6.

The models being compared and the cases consisting of the tracks is also input parameters to the system. A table of the selected Input variables and their levels of variations is shown in Table 4.1

| Model | $\mu$-$sd$ | $N$ | Curve | radius |
|---|---|---|---|---|
| Non adaptive (NA) | 0.2 | 15 | S | 10 |
| Friction adaptive (FA) | 0.6 | 20 | U | 15 |
| Load adaptive (LA) | | 25 | | 20 |
| Traction adaptive (TA) | | 30 | | |

Table 4.1: Table showing the input variables and their levels

We evaluated the effect of these input parameters on the planning performance by simulating all combinations of these parameter sets. In total 32 sets of parameters, combinations are simulated for each case in terms of the tracks to explore the effect by individual and combination of parameters, on the planning performance. For each set of parameters, 80 laps were computed.

### 4.1.3 Identifying Experimental units

Experimental units define the conditions of the environment which decide the conclusions. The number of samples $N_s$ in the planning algorithms using sampling, assumption of tire-road friction coefficient $\mu_a$ used by models without friction estimation, and sample time $T_s$ were identified as the experimental units. These parameters were set to give each model as good conditions as possible, to make a fair comparison. $\mu_a$ was set to the mean value of $\mu$ for the whole track, while $N_s$ was set by testing to find a value giving good performance. $T_s$ decides how often planning of a new trajectory is done and was set to the same value used by [1] and is the same for all models. The values of the experimental units are seen in Table 4.2

| $\mu_a$ | $N_s$ | $T_s[ms]$ |
|---|---|---|
| 1.6 | 50 | 100 |

Table 4.2: Table showing the experimental units

### 4.1.4 Noise factors

Noise factors are factors that are not considered in the result but affect the result. When running simulations one noise factor identified was the placement of different $\mu$ values, meaning if a low value is placed in a sharp corner the risk of failing is higher than if the value was higher. Another factor was the initial conditions on speed and vehicle placement on the road. An important part of the research was to find the different generalisations possible from testing on specific sections. Therefore noise factors were dealt with by introducing randomness into the experiments and running batch simulations, meaning the result is based on the mean values of several laps for each set of parameters. This ensured that the various possible scenarios are accounted for and the most likely output was obtained. Hence permitted the results to be generalised for most sections of the track and answer the research question.

### 4.1.5 Measurements

The laptime and planning time was measured to conclude performance for different models. Data collection and output visualization were integrated into the simulation environment to get an agile understanding of the relations between various parameters.

## 4.2 Software

### 4.2.1 ROS

ROS [33] was chosen for handling the functionality aspect of the simulations due to simplicity and modularity. Robot Operating System (ROS) is a middleware platform that provides libraries and tools that help software developers create robot applications. It simplifies the integration of different components on the same system and over the network. A basic ROS system consists of a master, nodes, topics, services, and actions. ROS nodes are executables, that that can exchange information through topics, services, or actions. ROS master ensures that all the nodes are running smoothly and asynchronously. ROS handles the communication of different nodes to run the package as a single system. The entire setup is modular and permits different modules to be swapped in or out. The system of simulation is also close to reality, hence reduce setup time when switching to hardware implementation.

### 4.2.2 FSSIM

FSSIM was chosen as the racing simulator. FSSIM is a vehicle simulator dedicated for Formula Student Driverless Competition. It was developed for autonomous software testing purposes. A version of this simulator was used to predict laptime of gotthard car at the Formula student Germany 2018 trackdrive competition with one percent accuracy[3]. The simulator allows agile real-time visualisation of closely modeled (to reality) cars to be run on tracks defined by cones. The simulation is run in Gazebo environment with the RVIZ package, on ROS.

### 4.2.3 ACADO Toolkit

The ACADO toolkit was chosen for handling the real-time optimisation of the different variables while planning the path. ACADO Toolkit is a software environment and algorithm collection for automatic control and dynamic optimization. It provides a general framework for using a variety of algorithms for direct optimal control, including model predictive control, state and parameter estimation and robust optimization. The toolkit has an object oriented design, hence permits for easy modifications and optimisations.

## 4.3 Hardware in the loop

Jetson TX2 is a microcontroller and is used as a standard for hardware implementation in the industry. Jetson TX2 has a powerful GPU with parallel computing capabilities, and support for multiple hardware and software applications. This makes it widely used in the field of autonomous driving and robotics. We utilize the Jetson for implementation of the planning algorithm, as the main purpose is to evaluate the planning performance. The other aspects of the simulation like the racing track and it's associated parameters are implemented on another host machine, and this mimics the real scenario. The host communicates over ethernet with the Jetson via SSH. Ethernet is used to minimize the communication delay between the different nodes. The control node for the car runs on the Jetson.

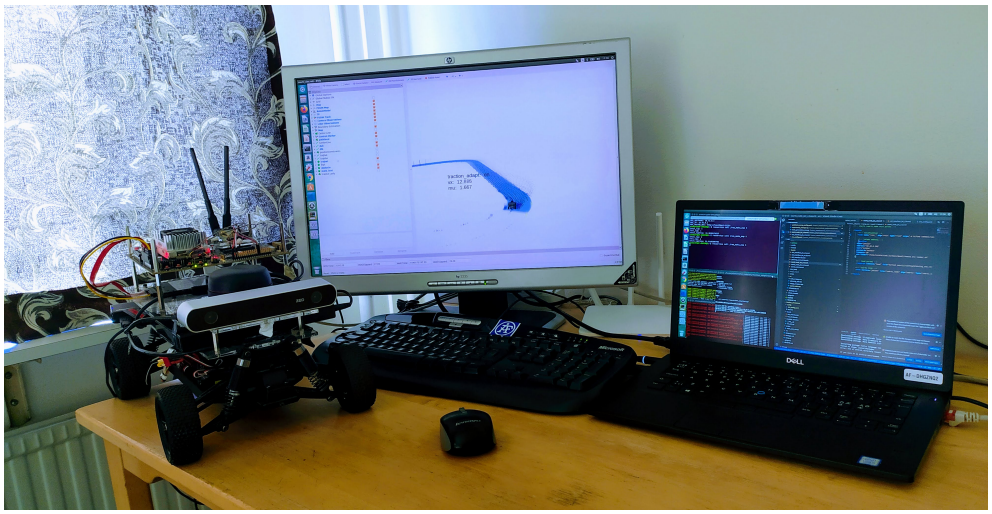


Figure 4.4: Hardware in the loop setup for batch simulations

## 4.4 System architecture

The simulation was implemented with the system architecture as shown in Figure 4.5. The simulation handler is responsible for handling the entire simulation and is the master node. Simulation handler is also responsible for resetting the simulations in case of failure. Experiments need to be run with a different combination of parameters and this is given as input from the update ROS parameters node. This node publishes the different input parameters to run the simulation. Run simulation runs one iteration of simulations, this could be single or multiple laps, depending on the input configuration for a single combination of parameters. Experiment manager receives inputs from various nodes like perception, state estimation, and planning, to give output commands to the control interface. The commands to the control interface tell the car how it should be driven in the track. The track, the car and its motion are visualized using RVIZ, figure . The FSSIM node provides the state of the car as feedback for both the experiment manager and planning nodes. The planning node takes input from the previous state of the car and the sample set from sampling augmentation, calculates the optimal path by using differently weighted parameters and constraints, and finally, output path is given to the experiment manager. Planning node runs on the Jetson, while the remaining nodes run remotely, allowing the planning performance to be evaluated as independently as possible on actual hardware.

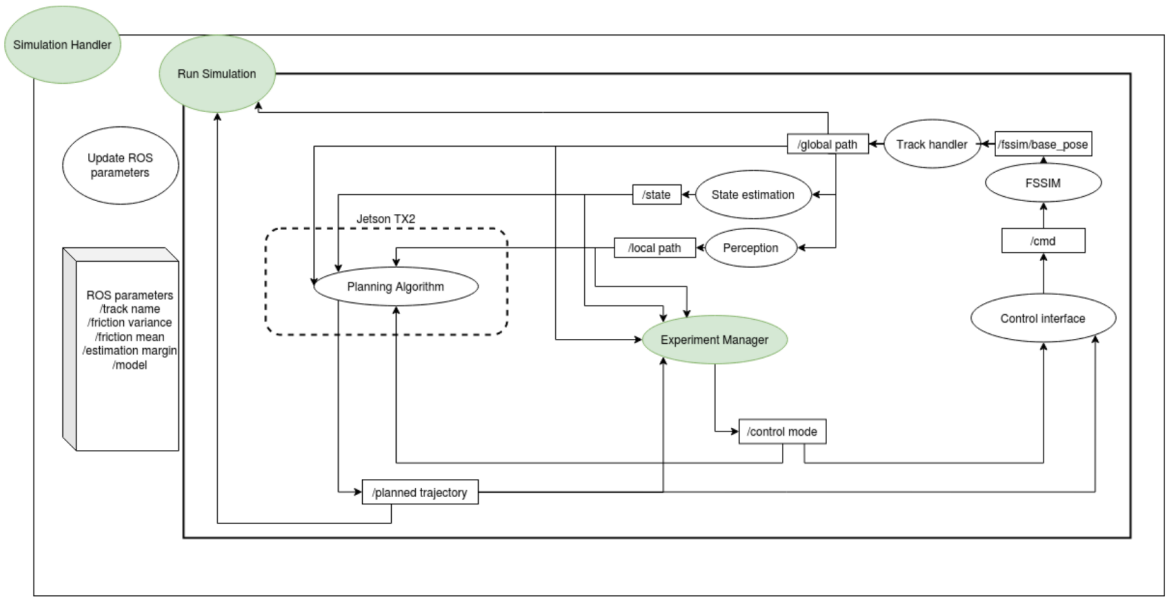



Figure 4.5: System architecture

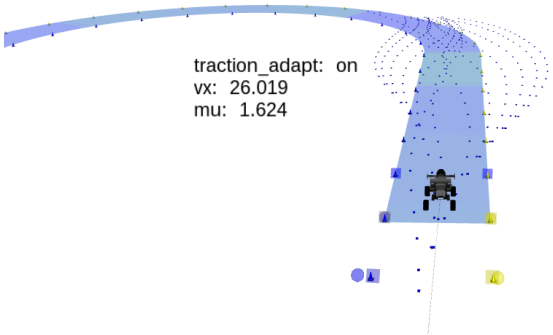



Figure 4.6: Visualisation of the planned path and the sample trajectories using RVIZ

## 4.5 Simulation environment

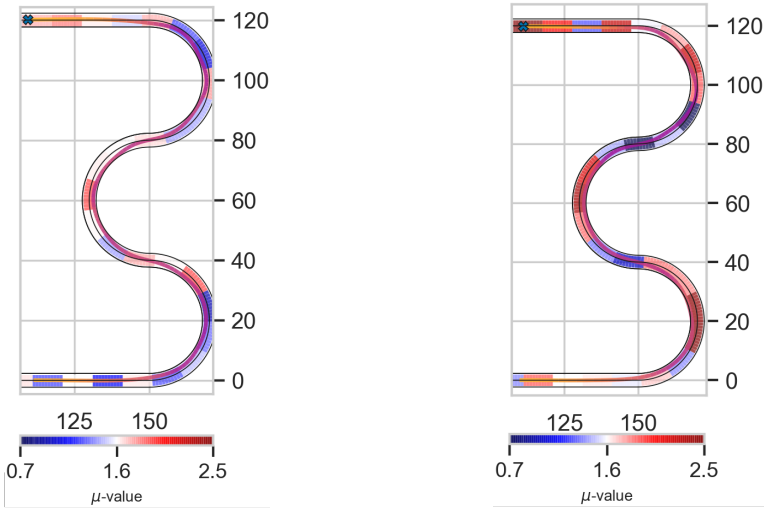
The visualization of the simulation environment is seen in figure 4.6. The sampled trajectories and the planned path are seen as the blue dotted curves. The speed of the car and the friction estimate is displayed on the screen as the car progresses along the track.

### 4.5.1 Obstacles at beginning of race

Racing dictates that the car travels the maximum distance in a minimum time interval and we are interested in randomizing the initial conditions only. Obstacles were introduced at the beginning of the race at random locations for each lap, to randomize the initial conditions and ensure a representative sample for analysis. The obstacles appear at a location such that it is far from the start point and also gets sufficient distance to accelerate after passing the obstacle, before entering critical testing regions of the race track. The existing architecture permitted running multiple laps continuously while in the results only data for the critical section of the track were considered.

### 4.5.2 Randomising $\mu$ throughout the track

Testing the traction adaptive planning algorithm required varying friction throughout the track. This was achieved by variable friction coefficients at different sections of the track. The coefficients were modeled as a normal distribution function with a mean and standard deviation. The simulations were run with constant mean value and standard deviations of 0.2 and 0.6, a visualization of the varying friction can be seen in 4.7. The coefficients for the Pacejka Tire Model [19] used to calculate the cornering stiffness $C_\alpha$ in the simulation environment were experimentally identified for a friction coefficient of 1.6[3]. Hence, the mean value of the friction was set to 1.6. The values were randomly generated while following the probability distribution function. The randomly generated values were assigned to smaller sections of the track. This later permitted testing the effect of different models and friction variations.




(a) $\mu$-$sd$ of 0.2

(b) $\mu$-$sd$ of 0.6

Figure 4.7: A visualization of an s-shaped track with varying friction sections. The colorbar in the bottom shows the colormapping for the friction coefficient sections in the track

### 4.5.3 Tracks

Critical sections of the tracks in racing include sections with roundabout turns, quick sharp corners, and consecutive corners after long stretches of acceleration. The tracks needed to be modeled to permit these sections to be tested while allowing us to run a large number of simulations. Generalisation of the results formed a large part of the test environment and simulations to answer the research question. It was decided to keep the majority of the track a constant while only swapping the critical sections to be tested. This would ensure that the results would be due to the critical sections of the track. Sufficient acceleration lengths were provided to ensure the car was going into the curves at high speed and test the adaptability of the algorithm. Tracks generated in frenet frame to allow the car to follow the curve path $s$ and variations along the width represented by $d$.

### 4.5.4 Tuning weights for optimisation

The weights of the cost function were tuned to give good performance in racing. A tune process was done where the weights are tuned one after the other. The weights were increased/decreased as long as laptime was consistent or

decreasing. When the failure or laptime increased the previous value of the weight was used and then the next weight was tuned. This process was repeated for all weights until the tuning was not changing the weights. Many sets of weights gave good results since the setup is insensitive to tuning, therefore a large step size could be used when tuning and the same weights could be used for all models.

## 4.6 Data Post-processing

Extensive experimentation with the different parameter combinations resulted in large data sets and required systematic methods for analysis and visualisation. The data is saved as numpy files, with all the details about path traversed, path planned, planning samples, vehicle parameters, track parameters, and any other parameter that required investigating. Python pandas were used for collecting all the data and storing the parameters that required investigation. Sometimes failed runs occurred due to factors outside the planner (the object of evaluation) and was filtered out during post-processing.

# Chapter 5

# Results and Discussion

During this thesis, it was hypothesized that models with dynamic constraints would perform better than the one with the static constraints up to the point of failure where all computational resources are used. By varying the input parameters that were expected to have the most effect on performance, we evaluate the effect of these parameters on the planning performance for different models. Suitable input parameters include the planning horizon and the friction coefficient variance. To generalize the evaluation, the algorithm was tested on several S and U shaped tracks and curves of varying radii. The resulting trade-offs between model complexity and planning horizon and the consequences of variation of input parameters are presented in this chapter. Data gathered by performing batch simulations of 80 laps for each of the combinations of input parameters which resulted in a total number of 15360 laps.

We utilize NA as the baseline for evaluating the planning performance. A single lap on an S-track with low variance on the friction coefficient is plotted in Figure 5.1 for different planning horizons. The vertical color bar represents the variation in speed and the path traced by the car is plotted with the same color coding. Variation in friction coefficient along the track is represented by the colors in the horizontal bar. The four N values represent the planning horizon for each lap. It can be seen that NA manages to plan feasible trajectories to compensate for the curve and handles small errors between assumed and real value on the friction coefficient for different planning horizons.

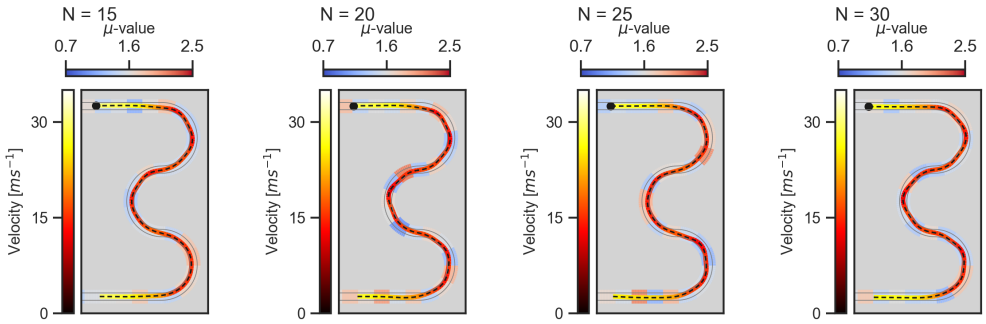
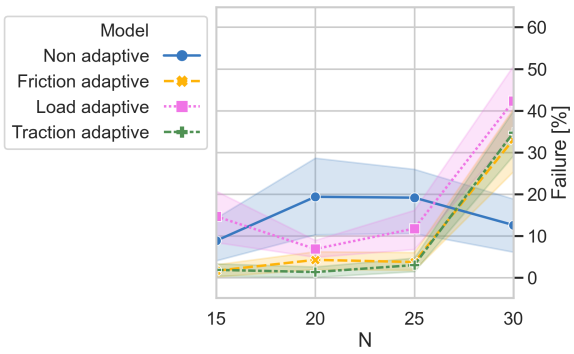



Figure 5.1: A sample of succeeded laps for NA on an S shaped track with radii$=$ 20 and $\mu$-$sd$ $=$ 0.2. Color mapping on the vehicle path shows the longitudinal velocity in m/s and color mapping on the track shows the real tire-road friction coefficient $\mu$, NA uses an values of $\mu = 1.6$ for planning

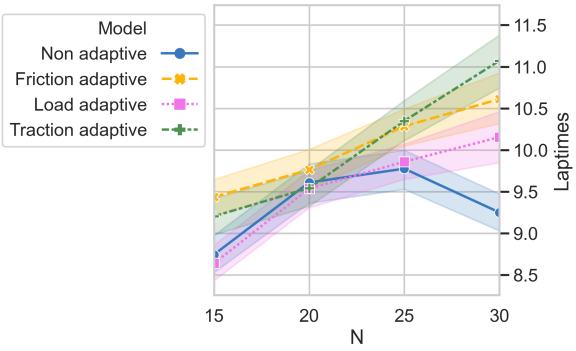
The following Overview section provides a generic view of the results obtained for all models. To study the effect different input parameters have on planning performance the result is then presented for varying planning horizon, friction variance, and track shapes in separate sections. The effect each of these parameters on output is presented as a contrast between using models with static constraints and models with dynamic constraints. A table of the mean results for all track shapes can be seen in appendix A.

## 5.1 Overview

This section presents the results and analysis of the data obtained by running 80 laps for each combination of input parameters. The input parameters used is summarized in Table 4.1 from earlier section. To get an overall view of the differences in performance for different models, plots in Figure 5.2, shows how the interaction between models and planning horizon affects failure and laptime. Results for all track shapes and $\mu$-$sd$ values are combined and visualized by mean value and 68% confidence interval of laptime and failure.




(a) Failure




(b) Laptimes

Figure 5.2: Plots showing failure and laptime vs planning horizon N. Lines illustrate mean value to combine the results for all input parameters and the colored area around the lines illustrates the 68% confidence interval for on each value of N

As seen in Figure 5.2, TA and FA have the lowest failure for planning horizons lower than 25 while NA allows for a higher planning horizon. For a planning horizon of 30, NA has the lowest failure, although it is higher compared to the adaptive models when using lower N. LA was hypothesized to have a higher failure and lower laptime compared to FA and TA, the figure shows that the results are in agreement with this hypothesis. Figure 5.2 shows that FA and TA are close in performance except for a planning horizon of 20, where TA has lower failure and laptime. TA and FA, show the most consistent result in failure which is shown by the shaded area around the curves in Figure 5.2a. The areas for NA and LA is both wider, and LA has lower spread for planning horizons between 20 and 25.

Table 5.1: Table presenting results at point of failure for each model, the result is an average for all track shapes and radius

| Model | N | $\mu$-sd | Avg. Laptime [s] | Failure [%] |
|---|---|---|---|---|
| **Friction adaptive** | **25** | **0.2** | 9.96 | 0.00 |
| | | **0.6** | 10.72 | 7.50 |
| **Load adaptive** | **25** | **0.2** | 9.69 | 0.00 |
| | | **0.6** | 10.10 | 22.92 |
| **Non adaptive** | **30** | **0.2** | 9.60 | 0.62 |
| | | **0.6** | 9.69 | 24.38 |
| **Traction adaptive** | **25** | **0.2** | 9.77 | 0.21 |
| | | **0.6** | 10.89 | 6.67 |

The results for all input parameters combined shows that TA has the best performance until a certain planning horizon. In the following sections the impact of planning horizon, varying friction coefficient and track shape will be further investigated, to better understand what influences the results seen in Figure 5.2.

## 5.2 Effect of planning horizon

The planning horizon limits the number of steps to which the car plans ahead in time. Higher planning horizons require higher numbers of computations, while a lower horizon meaning, the car sees a shorter length of the track to follow. Given this, it was hypothesised that all models will first get decreased failure with higher planning horizons and that there will be a trade off when the planning time gets close to point of failure, and failure will start increase with higher planning horizons. Since NA is a less complex model, it is assumed to be less affected by the increased planning time. Consider Figure 5.3 showing laptime and failure for each model.

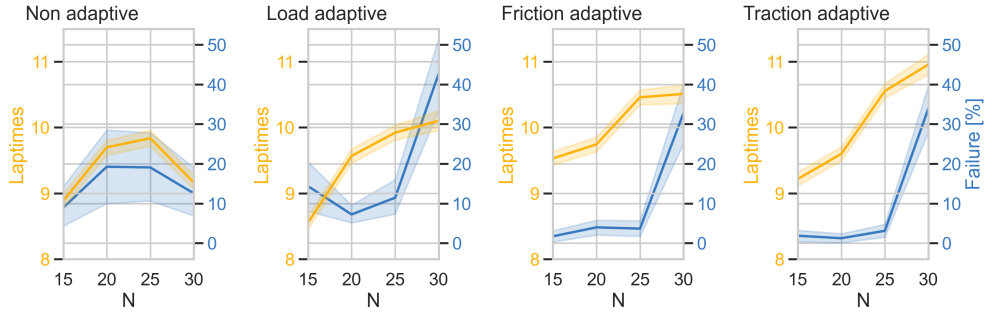



Figure 5.3: Laptime and failure visualized in the same graph to view the change in both laptime and failure

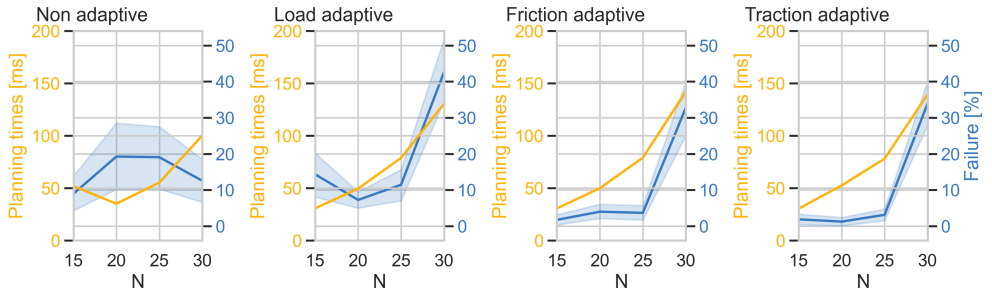



Figure 5.4: Planning time and failure. Planning time is defined as the average planning time for a lap, the lines defined the mean result for all combinations of input parameters, and the shaded area around the curves shows the standard deviations in the results

As mentioned, planning time is increasing with the planning horizon and planning time in turn affects failure. Planning time and failure was plotted for each planning horizon in Figure 5.4, to analyze the relation between parameters. Sample time $T_s$ used in the planning algorithm is set to $100ms$. Point of failure is defined by the lack of computational resources. Hence, point of failure can be seen for each model in Figure 5.4 by looking at the point where the average planning time is $100ms$. NA has a point of failure at N = 30, while LA, TA and FA have failure points at N = 25, Table 5.1 shows a summary of the results at these points of failures.

### 5.2.1 Static constraints

The increase in planning time is lower for NA compared to the models with dynamic constrains as can be seen in figure 5.4, which is in agreement with our initial hypothesis. The graph shows that the failure is decreasing for high

and low values on N, creating a concave curve with a maximum value at N = 20, contrary to our hypothesis.

Lower failure for lower planning horizon could be explained by looking at the planning time which has a lower value for N = 20 than N = 15. Planning time far from the sample time, could have an adverse effect on the performance when planning time is smaller than sample time. In the current setup the planner runs continuously, therefore the actual planning time decides the real sample time, while sample time used in calculations is $T_s$.

NA was run for 80 laps, however, the number of $\mu$ configurations is lesser in comparison to the other models. Even though the $\mu$ configurations is randomized the simulations was setup to run around the track until failure, where after a new $\mu$ configuration was generated. If the first randomized tracks happens to be in favour for the NA model this could cause a possible bias, and can also be the reason for the lower failure at horizons 15. As mentioned point of failure occurs at N = 30 for NA, therefore the increased failure is not seen in Figure 5.3, using a planning horizon higher than 30 should result in increased failure.

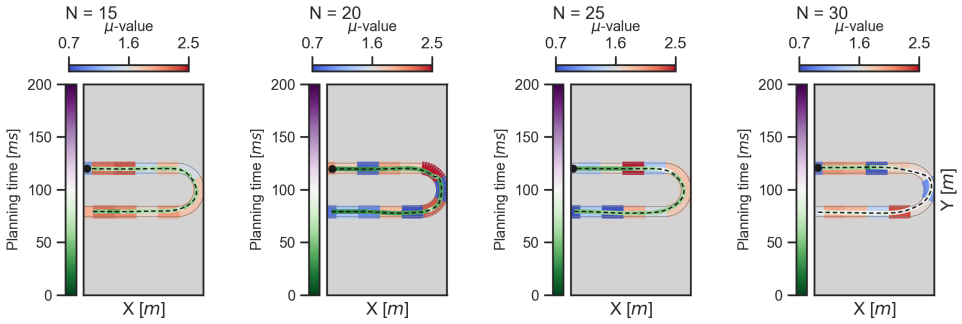



Figure 5.5: Successful laps for different planning horizons using NA model

In order to understand how laptime is affected by planning horizon, observe Figure 5.5 which shows paths and planning time for successful laps using different planning horizons. The car manages to compensate the track for different planning horizons, with planning time under 100ms, while driving along the U tracks. Performance is mostly affected by a higher N allowing for better planning. Figure 5.5 shows how the higher planning horizons allows to plan for the coming turn by making the car drive closer to the outer edge of the road before taking the turn. For NA, as seen in 5.4 mean planning time is kept

below 100ms, as a result increased planning horizon has a low negative effect on performance.

### 5.2.2 Dynamic constraints

For LA and TA, the failure initially decreases with increased N as hypothesized, while this is not seen for FA which is seen in Figure 5.3. Implying that a low planning horizon has a bigger negative effect on models adapting to varying normal load. The previous section on NA, showed that a low planning time negatively affected the performance. Looking at the mean planning time at N = 15 in Figure 5.4, it can be seen that there is not a big difference between the TA, FA and LA. In order to get a more detailed look at the differences in planning time a box plot of the planning time is shown in Figure 5.6. Here it can be seen that although the mean value is close between the models, the outliers of FA is higher than the ones for TA and LA and could be a reason to why FA is less sensitive to low N.

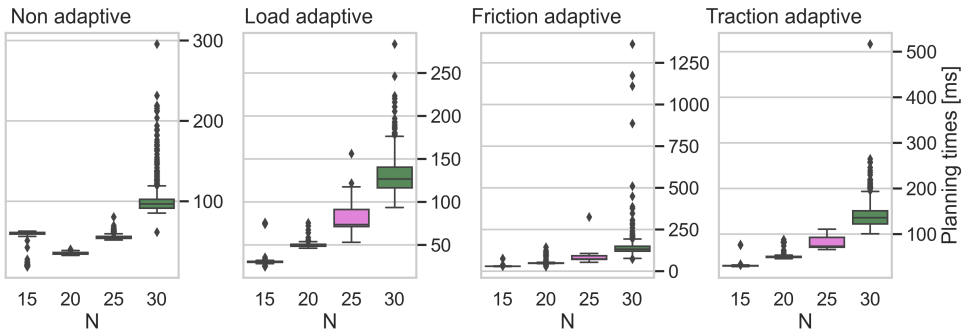



Figure 5.6: Boxplot showing the distribution of planning time for different planning horizons denoted as N

Using dynamic constraints with varying normal load allows the car to drive closer to the dynamic limits and is therefore more sensitive to errors. This could be another reason to why TA and LA is more sensitive to a low planning horizon and planning time than FA model.

The initial hypothesis was that failure will increase with increasing planning horizon after point of failure. The plot in Figure 5.3 shows that failure increases with increased N for the dynamic models using a planning horizon higher than 25. FA and TA with friction estimation has lowest failure for a planning horizon below 25, while LA has a higher failure as hypothesised.

The effect of an increased planning time causing slower control is seen in Figures 5.7, 5.8, and 5.9, where one can see how the vehicle path gets more unstable with high N. The path taken by the vehicle for LA, FA and TA is not optimal at higher horizon lengths and results in higher laptime, and can be seen in Figures 5.7, 5.8 and 5.9.

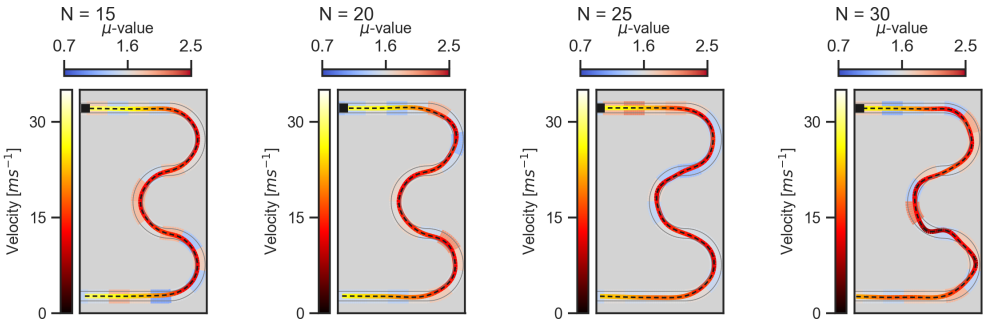



Figure 5.7: Successful laps for different planning horizons using LA

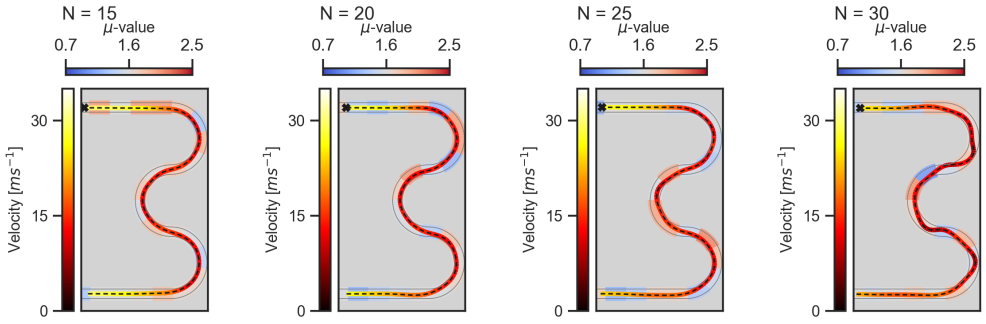



Figure 5.8: Successful laps for different planning horizons using FA

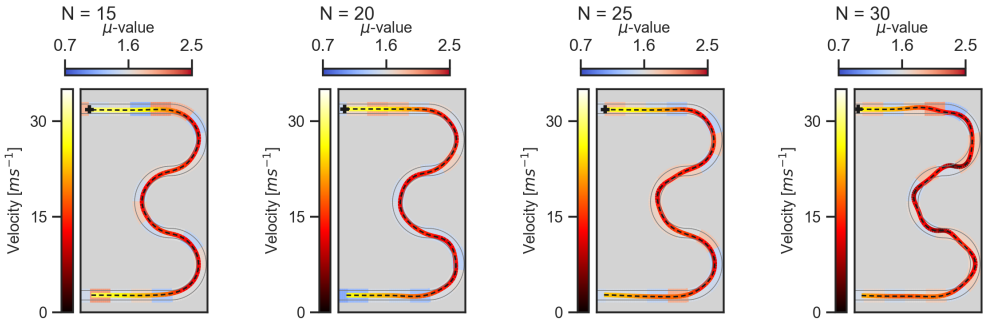



Figure 5.9: Successful laps for different planning horizons using TA

### 5.2.3 Takeaways

The trends in the advanced dynamic models for planning horizon against failure curve, are consistent. Uniform and consistent performance upto the point of failure for TA and FA. LA is seen to be the most sensitive to change in horizon. The advantages of variable normal force is improved vehicle handling. However, faster dynamics also needs faster sample time and hence makes it sensitive to increased planning time. NA shows higher failure in comparison to models with dynamic constraints and is insensitive to change in horizon. Laptime increases as the horizon is increased for dynamic models, while laptime is increasing with failure for NA.

## 5.3 Effect of friction coefficient variance

During the thesis, we hypothesised earlier that the models with friction estimate would perform better than the ones without it. Better performance hypothesised while running on tracks with lower friction variance. Figure 5.10 shows a higher failure while running on tracks with a higher friction coefficient, for all the models.

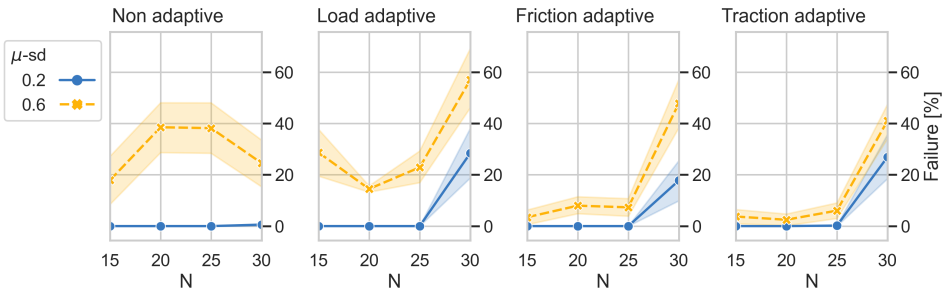



Figure 5.10: Failure graph filtered by friction coefficient variance

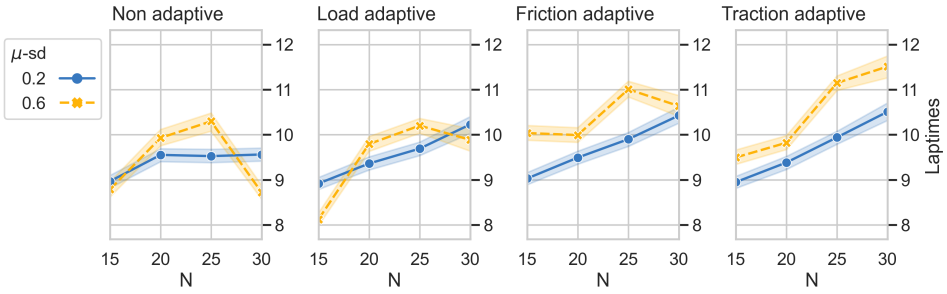



Figure 5.11: Laptime graph filtered by friction coefficient variance

Laptime is slightly higher while running on tracks with friction variance of 0.6, is shown in Figure 5.11. Higher value due to slower speeds essential to compensate the larger variations in consecutive friction coefficient values. The difference in planning time isn't as large, because the number of variables for optimisation remains the same in both cases of friction variance of 0.2 and 0.6. This is quantified in the form of similar planning time in the two cases and can be seen in Figure 5.12.

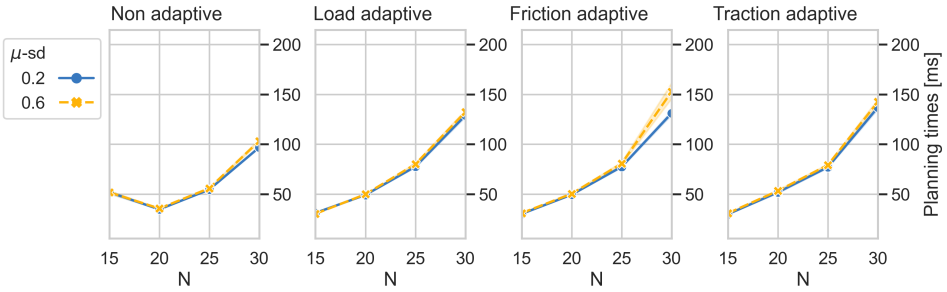
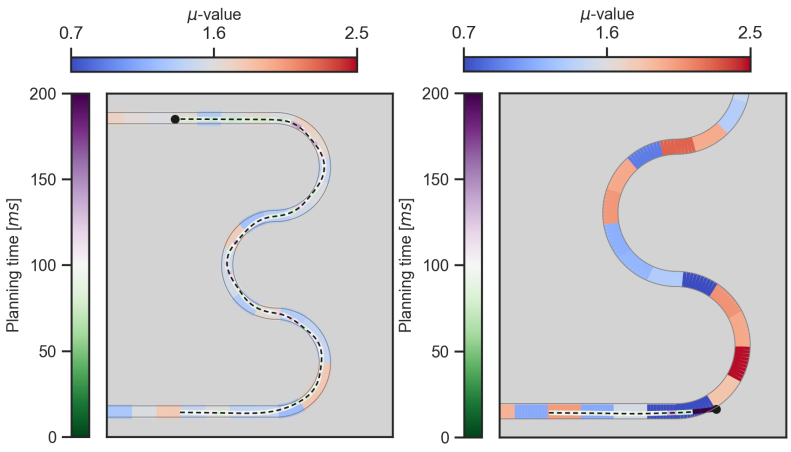



Figure 5.12: Planning time graph filtered by friction coefficient variance

### 5.3.1 Static constraints

NA experiences the highest failure while running on tracks with friction coefficient variance of 0.6, upto planning horizon 25. The model, however, shows lower failure while running on tracks with friction variance of 0.2. At a friction variance of 0.2, with the known mean value and sufficient computational resources, the model manages to compensate the track with same failure as its counterparts. While running on tracks with higher variance, the computational resources is not sufficient to overcome the larger error in friction estimation

and results in the worst performance. In the lower cases of friction variance, the performance is uniform which visualised in the form of colored patches, surrounding the mean curves, for a higher value of variance the spread in results is higher. Lack of friction estimate and constraint adaption ensures insensitivity to change in parameters when the variance in friction is low. Figure 5.13 shows NA compensating track with 0.2 variance of friction coefficient and a failed path while running along 0.6 friction variance track.




(a) Succsessful lap on $\mu$-$sd$ of 0.2 (b) Failed lap on $\mu$-$sd$ of 0.6

Figure 5.13: Differences between different variances in friction

The different cases of failure for NA is visualised in Figure 5.14. In subplot 1, the friction coefficient changes from high value to low value (red to blue, just before the point of failure). The car coming out of the corner, experiences loss in grip, oversteers and fails to stay in the track. In subplot 2, friction coefficient changes from high to low. Car experiences understeer due to loss in grip at the low friction section and is unable to change direction of velocity vector in time. In subplot 3, consecutively low friction sections are followed by high friction section. The model manages to compensate the lower friction sections, experiences oversteer immediately after entering high friction area and fails to stay in the track. In subplot 4, the change in friction region from low to high, is too late into the corner and fails to keep the car in the track. The lack of friction estimate for NA causes the loss in grip during the different cases presented here.

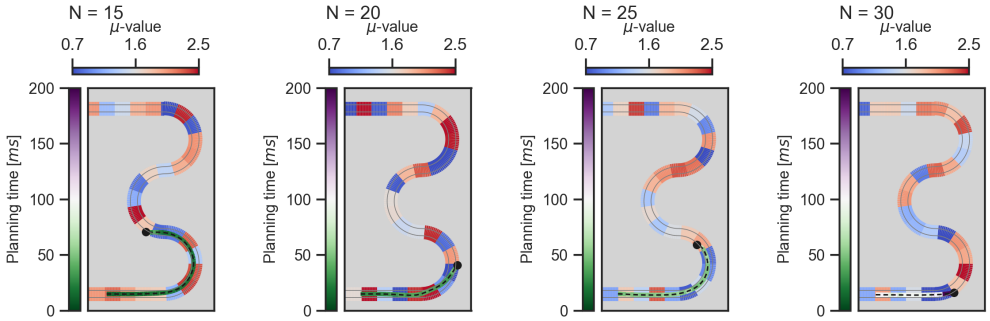



Figure 5.14: Failed path visualised on varying friction track for NA model on track with $\mu$-$sd$ = 0.6

### 5.3.2 Dynamic constraints

The trend of higher failure while running on tracks with friction variance of 0.6 in comparison 0.2, holds for all the models with dynamic constraint adaption. The models with friction estimate showed better performance in agreement with our earlier hypothesis. All models with dynamic constraints seen to outperform the model with static constraints, while running with sufficient computational resources.

TA shows the best performance at higher friction variance of 0.6 compared to the other models. When running on tracks with friction variance of 0.2 all models is close in performance although TA has a slightly higher laptime. The model's success at tracks with variance 0.6 can be owed to the combined friction estimation and pitch adaption. The cumulative effect of these individual models, make the traction adaptive constraints robust to changes in parameters.

Let's examine few of the failed laps for TA, shown in Figure 5.15 around the region of failure. In subplots 1 and 2, the change in friction coefficient from high to low happens, just before the car crashes in the cornering section. Although the planning time is under 100ms in both cases, the model attempts a tight corner and the slightest oversteer in the low friction region prevents the car from staying in the track. In subplot 3 and 4, planning time is maxed out and results in oscillatory motion of the car, before failing to stay in track. Though its important to note that TA is able to compensate the variations in friction from low to high and vice versa, due to the friction estimate. TA manages tight corners due to pitch adaptive constraints.

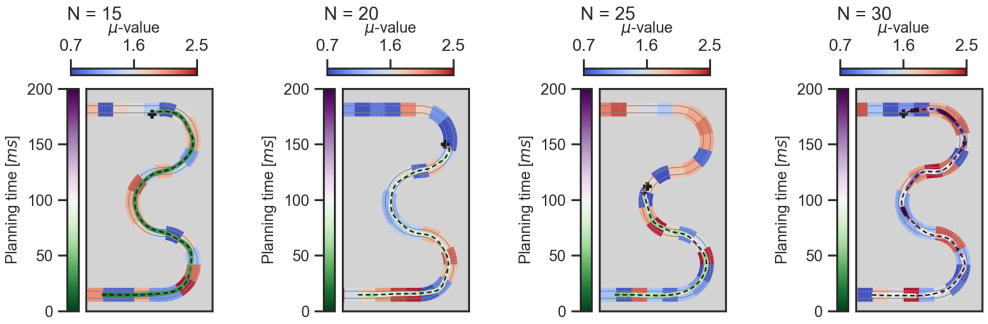



Figure 5.15: Failed path visualised on varying friction track for TA

FA shows good uniform performance, for different parameter combinations. The correct estimation of friction for different sections of the track, ensure that the performance is consistent as long as the available computational resources are sufficient.

Consider the visualisation of failed laps for FA, shown in Figure 5.16 around the fail region. In subplots 2 and 4, it can be seen that the model manages to compensate the long low friction region (represented by the blue region), due to the friction estimate, for some length of the track before failure. However, the path is not smooth during these laps and car fails to make tight corners. In subplot 3, the car fails due to higher planning time towards the end of the S section, despite the friction estimate.

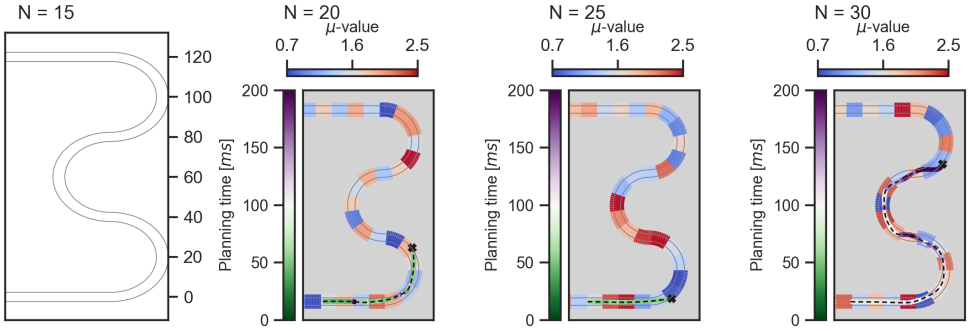



Figure 5.16: Failed path visualised on varying friction track for FA

LA shows good performance. The performance, however, is inconsistent and is quantified in the form of standard deviation, seen as yellow and blue patches for friction variances of 0.6 and 0.2 in Figure 5.10, respectively. The underestimation in friction values causes large deviation in performance for this model. The model still manages to outperform the static constraints model.

Pitch adaption ensures better vehicle handling and hence the better performance.

Let's examine the failed path for LA, shown in Figure 5.17 around the region of failure. The pitch adaption permits tight corners and can be seen in all the subplots. The lack of friction estimation while driving on the low friction sections, prevents the car from staying in the track. In subplot 4, the two reasons, in combination with high planning time ensures that the car goes off the track.

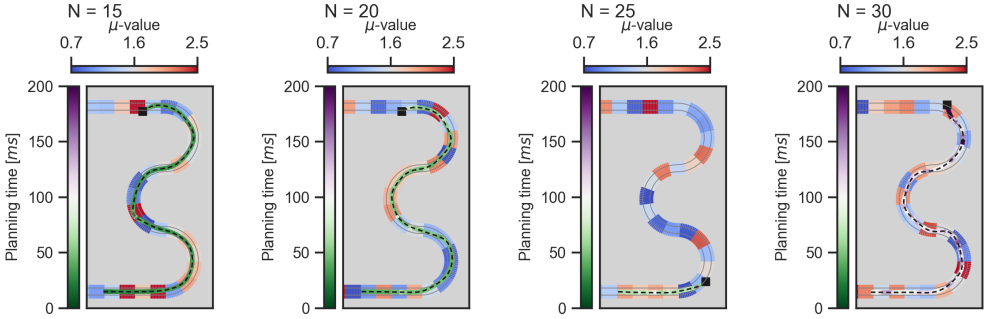



Figure 5.17: Failed path visualised on varying friction track for LA

### 5.3.3 Takeaways

The friction coefficient plays a huge role in the planning performance of each of the models. Models with the friction estimate had lower failure compared to models without the estimate. The performance was inconsistent for NA. TA with the pitch adaption and friction estimate, performed the most consistently and showed robustness to changes in parameters. Higher changes in friction coefficient values in between consecutive track sections causes failure for models without friction estimate. Slightly slower laps seen while running on tracks with variance of 0.6, because of slower speeds. FA without the pitch adaption has an oscillatory path towards the end of the S curve.

## 5.4 Effect of Track shape

During the thesis, we evaluate the previously stated factors by the use of S and U shaped tracks. These tracks were hypothesised to highlight the effect of each of these parameters on the planning performance. The U sections are simpler tracks in comparison to the S tracks, and observe lower failure overall for the different configurations. Figure 5.18 shows the failure for the different

models while running on S and U tracks. The S section permits errors to be carried forward, on account of being longer and complex tracks, in comparison to the U tracks. Driving through S tracks therefore causes sensitivity for latency created of high planning time, in comparison with the U tracks and is quantified by the higher failure for higher N seen in Figure 5.18. By looking at the failure plot for LA an increase in failure is seen for N = 15 for the S track, meaning that running on the S track require a higher N.

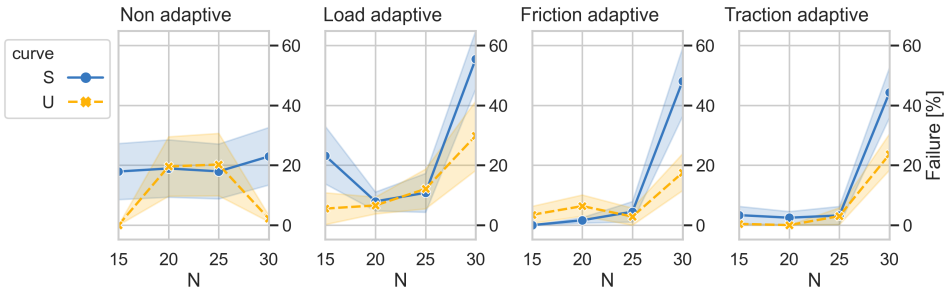



Figure 5.18: Failure graph filtered by track shape

The visualisation of failure for different models filtered by track radius is shown in Figure 5.19. The figure shows how the radius changes optimal trade-off between model accuracy and N, a higher radius requires a higher N in order to achieve good performance. The bigger radius permits the car to run at higher speeds. Inability to replan at higher speeds, results in the car failing to stay on track.

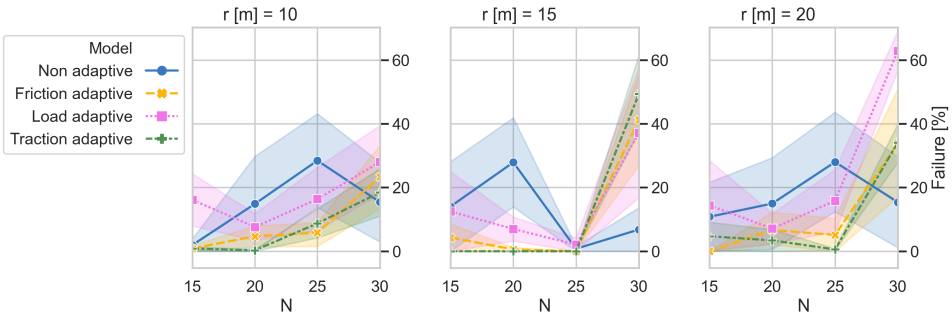



Figure 5.19: Failure graph filtered by track radius

The visualisation of laptime for different models filtered by track radius is shown in Figure 5.20. Laptime increases as the radius of curvature increases

for the different models. Increased laptime on account of longer tracks for bigger radius of curvature.

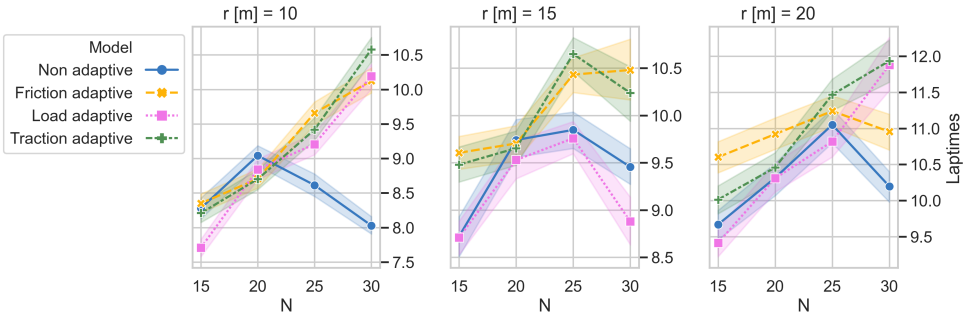



Figure 5.20: Laptime graph filtered by track radius

### 5.4.1 Static constraints

NA model has the highest failure, in comparison to models with dynamic constraints. The performance of the model is inconsistent and can be seen as deviation patch in Figure 5.19, remains unaffected by the radius of curvature. The simple nature of model due to static constraints, not accounting for traction variations, ensure that the model has the fastest laptime and the trend holds even at higher planning horizons.

Figure 5.21 represents the path of the car during the failed laps on the U-curves for NA. Vehicle fails to stay in track towards the end of the cornering section in most of the failed attempts.

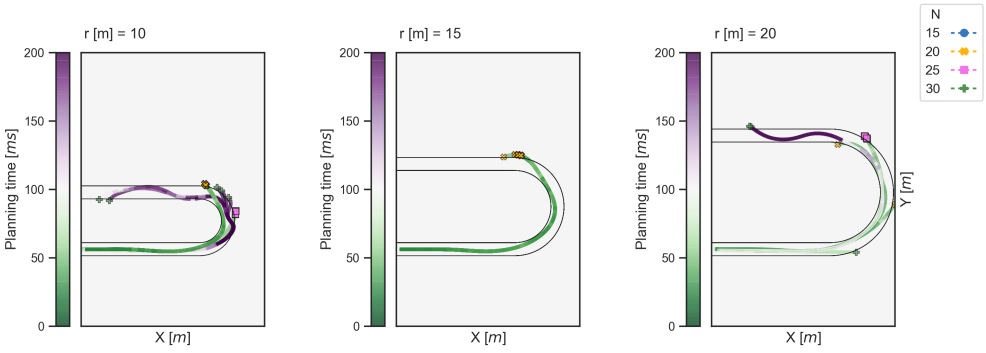



Figure 5.21: Lap plot showing failed laps for NA model on U curve with $\mu$-$sd$ = 0.6

The failed paths while driving on S track for NA is shown in Figure 5.22. The failures are still concentrated either towards the beginning or the end of smaller U cornering sections under consideration. The trend is repeating for the consecutive U sections, for NA. The increase in planning time along the length of the consecutive cornering sections, reduces chances to compensate the curvature and results in failure.

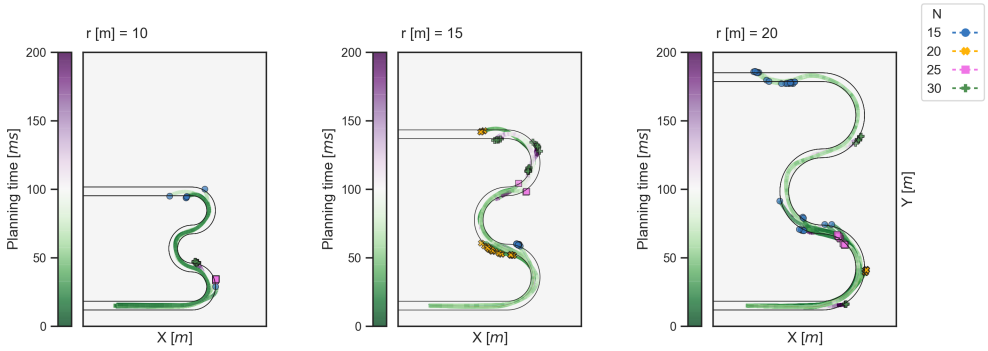



Figure 5.22: Lap plot showing failed laps for NA model on S curve with $\mu$-$sd$ = 0.6

The trends in terms of concentration of failure points for NA is recurring and hence permit generalisation of trends, to certain extent. Longer consecutive cornering sections, increases the model's chances of failure.

### 5.4.2 Dynamic constraints

All models with dynamic constraints outperform NA, as long as computational resources are sufficient. TA is observed to be the most robust to changes in radius of curvature of the track and shape of track (seen in Figure 5.18). This behaviour is attributed to pitch adaption contributing to better vehicle handling and friction adaption ensuring consistency. TA has laptime comparable with that of NA at lower horizons, seen in Figure 5.20. The model complexity results in higher planning time, and causes slower laptime at higher horizons.

The failed paths for TA while running on the U tracks are visualised in Figure 5.23. The points of failure concentrated towards the end of the U cornering section. The planning time is much higher compared to NA and can be observed by the purple colour of the paths for the different laps.

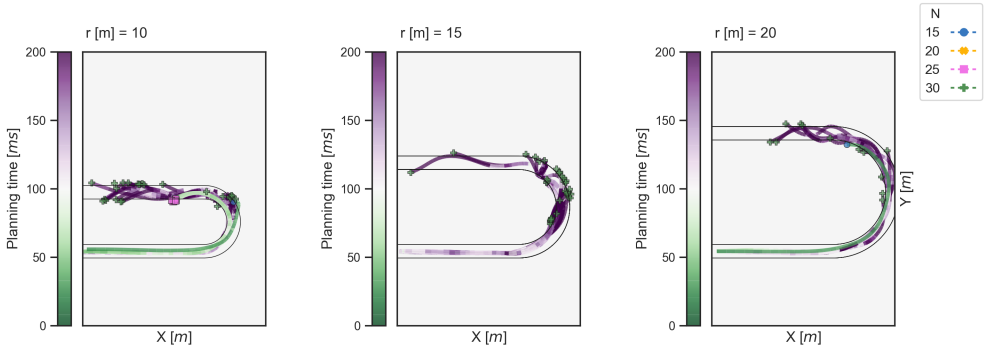



Figure 5.23: Lap plot showing failed laps for TA model on U curve with $\mu$-$sd$ = 0.6

The trend of higher concentration of fail points towards the beginning and end of smaller U sections, can be be seen in Figure 5.24 for TA. The planning time increases as the car progresses along the curve and exhaustion of resources, results in failure at different parts of S curve. Similar trends can be seen for LA and FA, in Figures 5.25 and 5.26 respectively. Lack of friction estimate causes failure points to be spread out for LA. Models with friction estimate have higher success upto the point of failure.

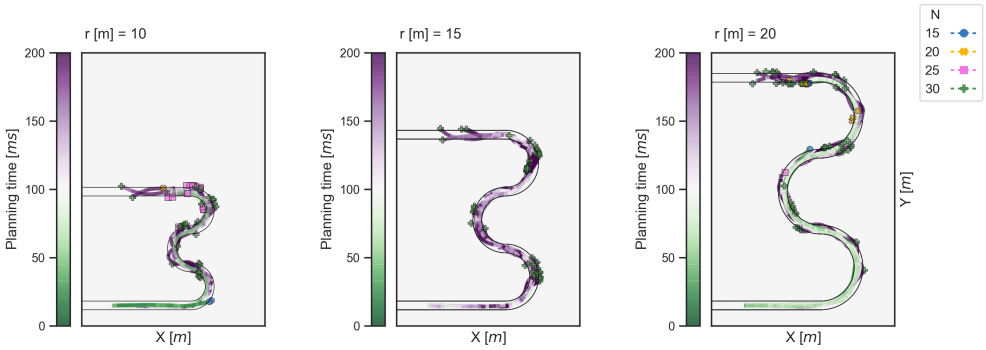



Figure 5.24: Lap plot showing failed laps for TA model on S curve with $\mu$-$sd$ = 0.6

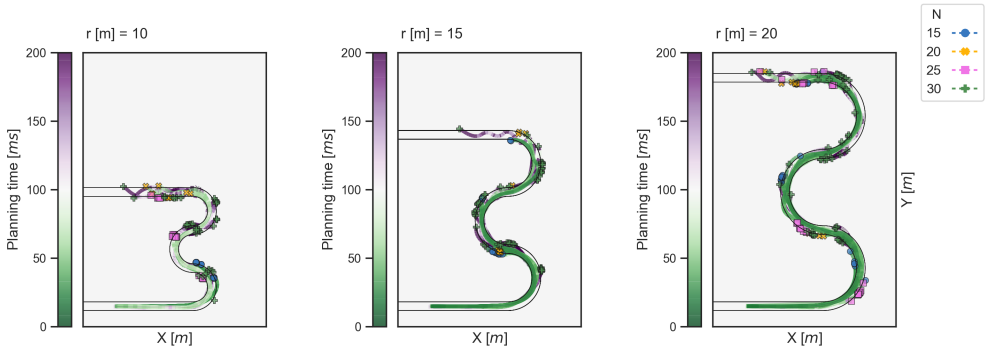



Figure 5.25: Lap plot showing failed laps for LA model on S curve with $\mu$-$sd$ = 0.6

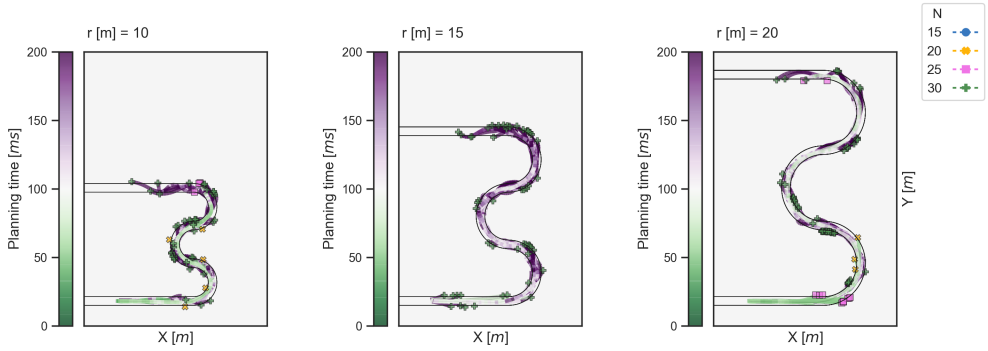



Figure 5.26: Lap plot showing failed laps for FA model on S curve with $\mu$-$sd$ = 0.6

### 5.4.3 Takeaways

The S and U curves of different radius proved to be good testing grounds for the different models. Failure points concentrated either towards the end or beginning of the smaller U sections of the S and U tracks, under consideration. Models performed better while running on the tracks with only U sections and the radius changes the optimal planning horizon for each model. Different constraints for the models affect the planning time and consequently affect, failure and laptime. This is seen in S tracks as the car progressed along the curve, for models with higher number of constraint adaption, the planning time increased and in turn the chances of failure rose.

## 5.5 Conclusions

During this thesis, we set out to evaluate the planning performance for dynamic model with static constraints and dynamic models with dynamic constraints. In addition, the thesis evaluated the generalisability of results for different sections of the track, at the point of failure. The point of failure was dictated by the increased planning time and lack of computational resources. Statistical approach of batch simulations, was adopted to evaluate these parameter configurations.

Models with dynamic constraints showed improved performance upto the point of failure. This is quantified by lower failure and faster laptime, in comparison to NA. TA with the friction estimate and pitch adaption showed consistent and robust performance. TA also manages to make tight corners while driving on the different track sections. FA managed to compensate the tracks with an oscillatory path, due to the lack of pitch dynamics adaption. The absence of friction estimate affects LA's ability to compensate consecutive sections undergoing large differences in friction coefficient. LA manages to have the fastest laptime and shows good performance while running on tracks with lower variation of friction coefficient. NA with static constraints showed robustness to changes in planning horizon and track shape but performed badly in comparison with the other models.

Failure points are concentrated either towards the beginning or end of the smaller U sections under consideration for the S and U tracks. Large differences in friction coefficient values between consecutive sections at these regions of the track, increased the failure chances for models without friction estimate. Models required higher planning horizon lengths to perform better on the S tracks. The S tracks permitted errors to be carried forward, in comparison to the U tracks and that negatively affected the planning performance for models with dynamic constraints at high planning horizons.

The results shows that friction adaption is crucial in a planner used for racing scenarios where the car is exposed for local variations. The planner is further improved by including adaption to varying normal load due to increased ability to handle critical scenarios like varying surfaces and complex track shapes. Adding load adaption enable driving closer to limits but also leads to a higher sensitivity to model errors. Hence, adapting constraints to varying load will further increase the need for friction estimation. In the real racing scenario

the car would be exposed for local variations affecting the tire-road friction coefficient, in addition a real race includes other cars which increases the need for more aggressive maneuvers. The results has shown that the load adaptive model allows for better handling of the vehicle while friction estimation reduces errors due to local variations, therefore we believe that the traction adaptive model is needed to achieve good performance in the racing case.

The tradeoffs between the accuracy of the model and planning horizon lengths, for models with static and dynamic constraints was brought out during this thesis. The parameters affecting the performance for each of these models was studied. The large number of simulations for different parameter configurations provided data on generalisation trends for different critical track sections. The shortcomings of the thesis will be addressed in the section to follow.

## 5.6 Future Work

- The current setup enabled testing of the traction adaptive algorithm with the hardware in the loop. The next logical step would be to implement and test, the algorithm on a scaled car with all the modules running on the host.
- Investigate what local variations needs to be included in the friction estimation for the racing case.
- The friction estimation module needs to be implemented for complete hardware run, as the current setup evaluates planning performance when the friction estimate is given.
- Implement all modules with parallel processing abilities of Jetson GPU, as it has a more powerful GPU in comparison to it's CPU.
- Further investigations on the trade-off between planning horizon and numbers of samples

# Bibliography


[1] Lars Svensson et al. “Adaptive Trajectory Planning and Optimization at Limits of Handling”. eng. In: (2019).

[2] Jesse Levinson et al. “Towards fully autonomous driving: Systems and algorithms”. eng. In: IEEE, 2011, pp. 163–168. ISBN: 1457708906.

[3] Juraj Kabzan et al. “AMZ Driverless: The Full Autonomous Racing System”. In: (2019).

[4] Alexander Liniger, Alexander Domahidi, and Manfred Morari. “Optimization-based autonomous racing of 1:43 scale RC cars”. In: *Optimal Control Applications and Methods* 36.5 (July 2014), pp. 628–647. ISSN: 0143-2087. DOI: 10.1002/oca.2123. URL: http://dx.doi.org/10.1002/oca.2123.

[5] B. Paden et al. “A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles”. In: *IEEE Transactions on Intelligent Vehicles* 1.1 (Mar. 2016), pp. 33–55. ISSN: 2379-8858. DOI: 10.1109/TIV.2016.2578706.

[6] Stefan Ionescu and Kaj Jonsson. “Design Trade-offs in Optimisation Based Trajectory Planning for Autonomous Racing”. MA thesis. KTH, Mechatronics, 2019.

[7] J. Kabzan et al. “Learning-Based Model Predictive Control for Autonomous Racing”. In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 3363–3370. ISSN: 23773766.

[8] Robert K Yin. *Case study research : design and methods*. eng. 4. ed.. Applied social research methods series, 5. London: SAGE, 2009. ISBN: 978-1-4129-6099-1.

[9] Per Runeson et al. *Case Study Research in Software Engineering: Guidelines and Examples*. eng. 1. Aufl. Hoboken: Wiley, 2012. ISBN: 1118104358.

[10] Angela Dean. *Design and Analysis of Experiments*. eng. 2nd ed. 2017.. Springer Texts in Statistics. 2017. ISBN: 3-319-52250-7.

[11] Geoff Keeling. "Why Trolley Problems Matter for the Ethics of Automated Vehicles". In: *Science and Engineering Ethics* 26.1 (2020), pp. 293–307. DOI: 10.1007/s11948-019-00096-1.

[12] Noa Kallioinen et al. "Moral Judgements on the Actions of Self-Driving Cars and Human Drivers in Dilemma Situations From Different Perspectives". In: *Frontiers in Psychology* 10 (2019). DOI: 10.3389/fpsyg.2019.02415.

[13] F. Alaieri and A. Vellino. "A decision making model for ethical (ro)bots". In: *2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*. Oct. 2017, pp. 203–207. DOI: 10.1109/IRIS.2017.8250122.

[14] A. Censi et al. "Liability, Ethics, and Culture-Aware Behavior Specification using Rulebooks". In: *2019 International Conference on Robotics and Automation (ICRA)*. May 2019, pp. 8536–8542. DOI: 10.1109/ICRA.2019.8794364.

[15] Gang Liu et al. "The 3-DoF bicycle model with the simplified piecewise linear tire model". In: Dec. 2013, pp. 3530–3534. ISBN: 978-1-4799-2565-0. DOI: 10.1109/MEC.2013.6885617.

[16] Jiechao Liu et al. "The Role of Model Fidelity in Model Predictive Control Based Hazard Avoidance in Unmanned Ground Vehicles Using LIDAR Sensors". In: vol. 3. Oct. 2013. DOI: 10.1115/DSCC2013-4021.

[17] John K Subosits and J. Christian Gerdes. "From the Racetrack to the Road: Real-Time Trajectory Replanning for Autonomous Driving". eng. In: *IEEE Transactions on Intelligent Vehicles* 4.2 (2019), pp. 309–320. ISSN: 2379-8858.

[18] Jiechao Liu et al. "A study on model fidelity for model predictive control-based obstacle avoidance in high-speed autonomous ground vehicles". eng. In: *Vehicle System Dynamics* 54.11 (2016), pp. 1629–1650. ISSN: 0042-3114. URL: http://www.tandfonline.com/doi/abs/10.1080/00423114.2016.1223863.

[19] Hans B Pacejka and Egbert Bakker. "THE MAGIC FORMULA TYRE MODEL". eng. In: *Vehicle System Dynamics* 21.sup001 (1992), pp. 1–18. ISSN: 0042-3114.

[20] Florent Altche, Philip Polack, and Arnaud de La Fortelle. "High-speed trajectory planning for autonomous vehicles using a simple dynamic model". eng. In: IEEE, 2017, pp. 1–7. ISBN: 9781538615263.

[21] B. Shyrokau and Diwakar Harsh. "Tire Model with Temperature Effects for Formula SAE Vehicle". eng. In: *Applied Sciences* 9.24 (2019).

[22] Meysam Khaleghian, Anahita Emami, and Saied Taheri. "A technical survey on tire-road friction estimation". In: *Friction* 5 (May 2017). DOI: 10.1007/s40544-017-0151-0.

[23] Max Ahlberg. "Optimization Based Trajectory Planning for Autonomous Racing". PhD thesis. Jan. 2018. DOI: 10.13140/RG.2.2.23680.38402.

[24] Robert Davison. "Dynamic programming: a practical introduction, by D. K. Smith. Pp l60. £15. 1991. ISBN 0-13-221805-4 (Ellis Horwood)". In: *The Mathematical Gazette* 76.476 (1992), pp. 314–314. DOI: 10.2307/3619177.

[25] Yiqi Gao. *Model Predictive Control for Autonomous and Semiautonomous Vehicles*. eng. 2014. URL: https://escholarship.org/uc/item/8xd0b56h.

[26] P. Falcone et al. "Predictive Active Steering Control for Autonomous Vehicle Systems". In: *IEEE Transactions on Control Systems Technology* 15.3 (May 2007). ISSN: 2374-0159. DOI: 10.1109/TCST.2007.894653.

[27] Rajesh Rajamani. *Vehicle Dynamics and Control*. eng. 2nd ed. 2012.. Mechanical Engineering Series. 2012. ISBN: 1-283-44402-X.

[28] Philip Polack et al. "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" In: June 2017, pp. 812–818. DOI: 10.1109/IVS.2017.7995816.

[29] J. Kong et al. "Kinematic and dynamic vehicle models for autonomous driving control design". In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. June 2015, pp. 1094–1099. DOI: 10.1109/IVS.2015.7225830.

[30] M Abe. *Vehicle handling dynamics theory and application*. eng. Oxford: Butterworth-Heinmann, 2009. ISBN: 1-282-16963-7.

[31] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017. DOI: 10.1017/9781139061759.

[32] Thomas Howard et al. “State Space Sampling of Feasible Motions for High-Performance Mobile Robot Navigation in Complex Environments”. In: *Journal of Field Robotics* 25.7 (June 2008), pp. 325–345.

[33] *Robotic Operating System*. URL: https://www.ros.org/.

# Appendix A

# Result Table

| Model | N | $\mu$-$sd$ | Avg. Laptime [s] | Failure [%] |
|---|---|---|---|---|
| **Friction adaptive** | **15** | **0.2** | 9.01 | 0.00 |
| | | **0.6** | 9.75 | 3.33 |
| | **20** | **0.2** | 9.42 | 0.00 |
| | | **0.6** | 9.96 | 8.33 |
| | **25** | **0.2** | 9.96 | 0.00 |
| | | **0.6** | 10.73 | 7.08 |
| | **30** | **0.2** | 11.05 | 18.33 |
| | | **0.6** | 12.60 | 47.71 |
| **Load adaptive** | **15** | **0.2** | 8.93 | 0.00 |
| | | **0.6** | 9.01 | 30.42 |
| | **20** | **0.2** | 9.33 | 0.00 |
| | | **0.6** | 9.83 | 13.96 |
| | **25** | **0.2** | 9.69 | 0.00 |
| | | **0.6** | 10.11 | 21.46 |
| | **30** | **0.2** | 11.04 | 28.54 |
| | | **0.6** | 11.42 | 56.88 |
| **Non adaptive** | **15** | **0.2** | 8.94 | 0.00 |
| | | **0.6** | 9.43 | 17.92 |
| | **20** | **0.2** | 9.50 | 0.00 |
| | | **0.6** | 10.13 | 37.29 |
| | **25** | **0.2** | 9.54 | 0.00 |
| | | **0.6** | 9.93 | 37.92 |
| | **30** | **0.2** | 9.61 | 0.62 |
| | | **0.6** | 9.72 | 25.00 |
| **Traction adaptive** | **15** | **0.2** | 8.95 | 0.00 |
| | | **0.6** | 9.62 | 3.54 |
| | **20** | **0.2** | 9.36 | 0.00 |
| | | **0.6** | 9.84 | 2.29 |
| | **25** | **0.2** | 9.76 | 0.21 |
| | | **0.6** | 10.94 | 5.62 |
| | **30** | **0.2** | 11.17 | 26.88 |
| | | **0.6** | 12.59 | 40.21 |

TRITA TRITA-ITM-EX 2020:444