University of Southampton

Faculty of Engineering and Physical Sciences

Electronics and Computer Science

RC Car Modelling and Trajectory Tracking Control

by

Arthur RODRIGUEZ

September 2, 2021

Supervisor: Matthew Turner

Second Examiner: Abhinav Kumar Singh

A dissertation submitted in partial fulfilment of the degree

of MSc Systems, Control and Signal Processing

Abstract* — *The best race driver is the fastest on track. However, "to finish first you must first finish" [1]. These objectives remain unchanged when dealing with remote control car racing. The objective of a race driver model is therefore to be as fast as possible on track while staying between the track limits.

This dissertation describes the development of a non-linear remote-controlled (RC) car model and its control system in order to test different trajectory optimisation strategies.

This project makes three contributions. Firstly, a high-fidelity non-linear vehicle model has been developed based on a four-wheel vehicle dynamics model and the Dugoff tyre model with adapted parameters to match the size and physics of a $1/10$***th remote-controlled car. Secondly, control systems for steering angle and speed have been developed to mimic the behaviour of a race driver. The steering controller is based on the pure pursuit algorithm. Classical control is then used to regulate the car speed to the speed target generated based on a friction circle model. Thirdly, trajectory optimization has been studied as a compromise between car speed and distance to be covered. The first approach to this issue consists in realising comparative runs on pre-defined optimized trajectories. On-line optimization has then been implemented.***

The vehicle model used in this project has been tuned to match the behaviour of a real RC car and therefore provide a fair comparison basis for controller tuning and trajectory optimization. Then, the combination of the steering angle and speed controllers enabled to keep the car on the desired trajectory with a mean accuracy of a dozen of centimetres. Finally, trajectory optimization enabled a 17% ***lap time gain compared with the track centerline which has been used a measurement standard.***

Index Terms* — *RC car; Vehicle dynamics model; Tyre model; Steering control; Pure pursuit algorithm; Friction model; Model linearisation; Trajectory optimization

CONTENTS

List of Figures

List of Tables

I. Introduction

In motorsports, the objective is to be the fastest. Depending on the series, being the fastest either means completing a fixed distance over the shortest amount of time, or being the first to complete the maximum distance over a fixed amount of time plus one lap. However, in order to reach this objective, the driver has to keep the car between the track limits while trying to get the best out of the car based on the power and grip conditions. Scaled remote-controlled (RC) car racing obeys similar rules and has the same objectives as full-scale racing. Despite their size, $1/10$th remote controlled cars are mechanically very closed to full-scale cars. As far as the drivers are concerned, the major difference relies on the fact they are all standing on a driving stand instead of being inside the vehicle.
In this quest for optimality, developing both vehicle and driver models can provide valuable information from car design to its exploitation. The main objective of a race driver model is therefore to mimic the behaviour and performances of a real driver while being as repeatable as possible. This driver model is then used to provide a fair basis during comparative runs either during the car development or for setup purposes. Vehicle and driver models are thus intrinsically linked and inseparable. Moreover, besides the ability to find the adherence limit to maximise the car performance, another racecraft of a driver is his ability to find the optimal trajectory between the track limits.

This project aims at developing both vehicle and driver mathematical models for a $1/10$th scale RC car. The vehicle model will be developed using dynamic models from the literature together with adapted parameters. The driver model is then developed using different control strategies to provide steering and throttle inputs for the vehicle model. These models are then used to provide a fair comparison basis to find the optimal trajectory in terms of lap-time. The outcome of the project are non-linear models of a RC car, driver and trajectory optimiser implemented in the *Matlab/Simulink* environment, together with various simulation results obtained from these models.

In this project, the first task consists in implementing a vehicle model. The latter is based on the vehicle dynamics model presented in [7], with adapted parameters to match the size and physics of a $1/10$th scaled car. In addition to the vehicle dynamics model, a tyre model based on [8] is implemented to model the tyre-road interaction, which is responsible for both the car lateral and longitudinal behaviour. Both model parameters are then matched, using either measured or estimated information, to coincide with the behaviour of a real remote control car.
Then, once the vehicle model is ready, the following part consists in developing a race driver model. The latter can be divided into two interdependent components: the steering controller and the speed controller. The steering controller considered in this project is a pure pursuit controller inspired from [5]. Once implemented, the main duty relies on the tuning of this algorithm to adapt its parameters to remote control cars. The speed controller is implemented in two steps: a first one to generate a target speed, and a second one to regulate the car speed to the speed target.
Finally, when both the vehicle and driver model are implemented and tuned, the next step consists in feeding these with a trajectory. As described in [6], finding the optimal trajectory can be seen as an optimization problem to solve the compromise between the shortest and the minimum curvature path. A further extension of this optimization problem is to propose an online implementation to cope with eventual changes in the run scheme.

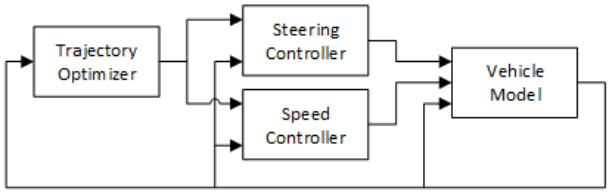

Fig. 1: Model structure.

Section II describes the vehicle dynamics model based on a generic mathematical model with adapted parameters. Section III describes different tyre models and justifies which are appropriate for this project. Section IV explains the overall vehicle and presents initial simulation results. Section V describes the pure pursuit algorithm used in this project for steering control. Section VI describes the speed control scheme used in this project. Section VII defines the lap time measurement setup as well as the track centerline tracking result which is used as a measurement standard for trajectory optimization. Section VIII introduces different trajectory optimization strategies and the resulting lap times. Section IX presents the concept of online trajectory optimization and the resulting lap times. The final three sections consists in discussion and conclusion of this project.

II. Vehicle Dynamics Model

In this project, the vehicle considered is an $1/10$th remote-controlled (RC) touring car depicted on the Fig 2. This car is $43cm$ long, $19cm$ wide (with bodywork) and its top speed is about $25m/s$. As far as its controls are concerned, this is a four-wheel drive car with front steering.

Fig. 2: Xray T4'19. Image from [2].

As far as the electronics is concerned, the car is powered by a Lithium-Polymer (LiPo) battery. The latter supplies a brushless motor through a speed controller receiving the throttle input for the transmitter and feedback directly from the motor sensor.

The main objective of this section is to describe the vehicle dynamics model used to replicate the behaviour of this RC car.

A. Vehicle model selection

As described in [9], the vehicle models are sorted into three main categories: geometric, kinematic and dynamic.
First, the geometric model only considers the parameters and states related to the vehicle dimension and position. This model describes the different manoeuvres only based on the vehicle characteristics with no regard to the dynamics (velocity or acceleration). With such a model, it would be possible to follow any admissible turn radius (depending on the vehicle geometry) at any speed, which is obviously not the case in reality. A geometric model is therefore not suitable for this project.
Contrary to geometric models, kinematic models describe the vehicle motion in a Cartesian frame based on its speed and geometry. However, kinematic models do not consider the internal dynamics of the system (mass, inertia...). Such a model can therefore not be used to describe mass or inertia related phenomena such as centrifuge effect in turns or mass transfer. A kinematic model is therefore not suitable for this project.
Finally, dynamic vehicle models consider internal forces, energy and momentum to describe the vehicle notion in terms of position, velocity and acceleration. This project will be based on such a vehicle model.
A comparison summary of these vehicle models is given in table I.

Then as mentioned in [9], there exist two main vehicle models: bicycle model (Fig. 3) or full vehicle model (Fig. 4).
Because of its configuration (combined wheels at the front and the rear), the bicycle model does not consider the effects of inner and outer wheels and suspension (Ackermann steering, roll...).
The model used in this project is therefore a four wheels model to capture these behaviours.

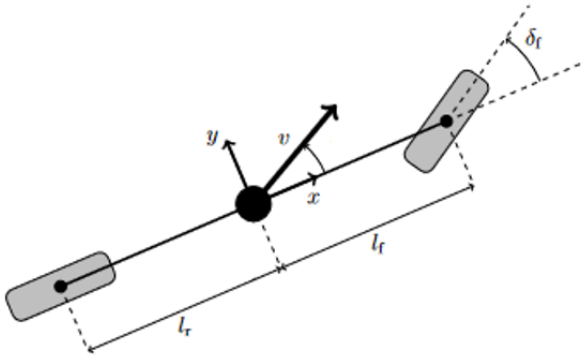

Fig. 3: Bicycle model geometry. Adapted from [3].

B. Vehicle positioning

In this project, it is assumed that the vehicle has a planar motion, i.e. the track is considered flat with no elevation or banking. The vehicle notion can therefore be described by its longitudinal speed v_x, lateral speed v_y and yaw rate r. In an inertial frame, the vehicle's position is obtained using the following equations of motion

$$\dot{x} = v_x cos(\psi) - v_y sin(\psi) \tag{1a}$$
$$\dot{y} = v_x sin(\psi) + v_y cos(\psi) \tag{1b}$$
$$\dot{\psi} = r \tag{1c}$$

where (x, y) are the vehicle's position in a Cartesian frame and ψ is its orientation.
It can also be noted that the combined velocity of the car is given by

$$v_{Car} = \sqrt{v_x^2 + v_y^2} \tag{2}$$

C. Vehicle dynamics model

As mentioned in part II-A, the model used is a four wheels dynamic model mainly based on the one given in [4].
From now on, the considered vehicle axis system is depicted in Fig. 4.

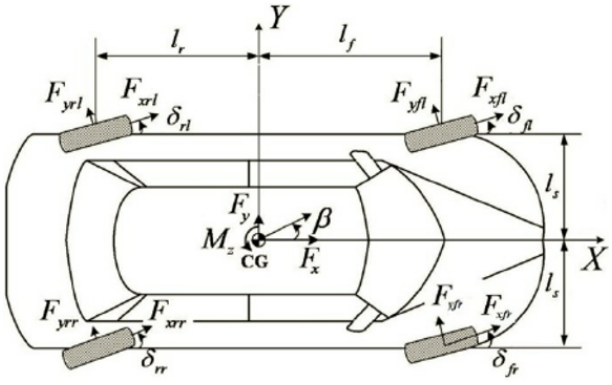

Fig. 4: Vehicle axis system and parameters. Adapted from [4].

It can be noted that the parameters and forces applied to the wheels will be indexed with the subscript $i \in \{FL, FR, RL, RR\}$ with FL, FR, RL, RR meaning Front Left, Front Right, Rear Left, Rear Right respectively.

TABLE I: Vehicle models comparison.

Model	Parameters	Comments
Geometric	Only geometric (dimensions)	Simple Model Only vehicle position information No dynamic considerations
Kinematic	Only geometric (dimensions)	Simple Model Information regarding the vehicle speed and heading relative to both a local and global coordinates No dynamic considerations
Dynamic	Geometric and dynamic (dimensions, mass, inertia…)	Complex Model Consider both the internal and external forces applied on the vehicle Complete handling model

According to [4], simplified vehicle dynamics equations of motion are

$$m_{Car}(\dot{v_x} - rv_y) = F_x \tag{3a}$$
$$m_{Car}(\dot{v_y} + rv_x) = F_y \tag{3b}$$
$$I_{z,Car}\dot{r} = M_z \tag{3c}$$

where m_{Car} is the vehicle mass, $I_{z,Car}$ is the moment of inertia about the Z axis. F_x, F_y and M_z are the generalized forces and moment applied on the vehicle and can be decomposed as following

$$F_x = F_{x,Tyres} + F_{x,Ext} \tag{4a}$$
$$F_y = F_{y,Tyres} + F_{y,Ext} \tag{4b}$$
$$M_z = M_{z,Tyres} + M_{z,Ext} \tag{4c}$$

As far as the parameter identification in this section is concerned, the car mass can be directly measured (with battery and body, ready to run): $m_{Car} = 1.32kg$.

In order to be able to define the different lengths depicted in Fig. 4, the centre of gravity of the car needs to be located in the (X, Y) plane. Given that the target setup of such a car is to have the same four corner weights, it can be assumed that the centre of gravity of the car is equidistant to the four wheels, leading to $l_f = l_r = \frac{0.26}{2} = 0.13m$ and $l_s = \frac{0.165}{2} = 0.0825m$. Then, as far as the height of the centre of gravity is concerned, it can be approximated by the height of the centre of gravity of the battery (weighing $300g$), i.e. $h_{CG} = 0.02m$.

The moment of inertia of the car about the Z axis $I_{z,Car}$ is not straightforward to measure. Assuming that the vehicle can be modelled by a solid rectangular shape with a uniform mass repartition (coherent with the previous assumption about the centre of gravity), $I_{z,Car}$ can be approximated by

$$I_{z,Car} = \frac{1}{12}m_{Car}((l_f + l_r)^2 + (2l_s)^2) = 0.0104kg.m^2 \tag{5}$$

D. External forces

The external forces mainly consist in environmental effects. The two external forces considered in this project are the aerodynamics drag and the tyre rolling resistance

$$F_{x,Ext} = -F_{x,Drag} - F_{x,Roll} \tag{6}$$

1) Drag force:

The drag force applied on the car is given by

$$F_{x,Drag} = \frac{1}{2}\rho_{Air}Sv_{Car}^2C_x \tag{7}$$

where S is the frontal area of the car and C_x is its drag coefficient.

The frontal area of the car can easily be computed as the product of the car width and height: $S = 0.2 * 0.115 = 0.023m^2$. As the drag coefficient is not straightforward to measure, it is estimated to be the same as full-scale motorsport cars: $C_x = 0.3$.

Note that the direction of this force is opposed to the motion of the car.

2) Tyre rolling resistance:

The tyre rolling resistance of the car is given by

$$F_{x,Roll} = m_{Car}gC_{RR}v_{Car} \tag{8}$$

where C_{RR} denotes the tyre rolling resistance coefficient of the tyres. This parameter is chosen to match the tyre rolling of full-scale cars: $C_{RR} = 0.01$.

Similarly to the drag force, the roll resistance is opposed to the motion of the car.

III. Tyre Model

A. Tyre model selection

RC touring cars are equipped with rubber tyres. The behaviour of such tyres is similar to the one of full-scale cars, both in terms of grip and wear. This justifies the use of a full-scale tyre model with adapted parameters.

Tyres have been and remain one of the most challenging vehicle components to model. This complexity led to the appearance of a wide range of models for different usage.

Among these models, the most simple tyre one is the linear one. However, its validity range is rather limited. This model will only be valid when considering simple and moderated driving schemes such as straight-line dynamics with moderated longitudinal acceleration (moderated slip ratio and zero slip angle), constant speed cornering with moderated lateral acceleration (zero slip ratio and moderated slip angle)... This is used when dealing with straight line vehicle dynamics linearisation in section VI-C

Then, in order to be able to model more complex (combined) driving schemes, a more advanced tyre model needs to be used. The reference model is the "Magic Formula" developed by Hans B. Pacejka. However, because of its complexity, both in terms of formulas and parameters, this model is not suitable for this project.

Another model that is often used for both simple and combined driving schemes simulation is the Dugoff model. The article [8] presents a comparison between the Dugoff model, the modified Dugoff model and the Magic Formula. It appears that the Dugoff model provides a first approximation of the Magic Formula. Thanks to its relative simplicity (only three tyre parameters are required), this model will be used for the simulations in this project.

A comparison summary of these tyre models is given in table II.

B. Tyre slip

1) Slip angle:

The tyre slip angle is responsible for the transmission of the lateral effort on the wheel (for turning purposes). It can be expressed as

$$\alpha_{FL} = \delta_{FL} - tan^{-1}\left(\frac{v_y + rl_f}{v_x - rl_s}\right) \tag{9a}$$

$$\alpha_{FR} = \delta_{FR} - tan^{-1}\left(\frac{v_y + rl_f}{v_x + rl_s}\right) \tag{9b}$$

$$\alpha_{RL} = \delta_{RL} - tan^{-1}\left(\frac{v_y - rl_f}{v_x - rl_s}\right) \tag{9c}$$

$$\alpha_{RR} = \delta_{RR} - tan^{-1}\left(\frac{v_y - rl_f}{v_x + rl_s}\right) \tag{9d}$$

where δ_i is the steering angle of the wheel i.

As this car is a front steering car only, the rear steering angles δ_{RL} and δ_{RR} are constant, set to zero in this project. Moreover, the front steering angles δ_{FL} and δ_{FR} are set to be equal within the range $[-26deg, 26deg]$. Ackermann steering (different steering angle for the inner and outer wheel to avoid slip induced by the difference of turning radius between the inner and outer wheel) is not considered here given the car trackwidth ($0.165m$) and the turn radius (from $2.5m$). Regarding the implementation in the non-linear model, four-wheel steering has been implemented even if the rear wheel steering angles have been set to zero. This would indeed enable to study the impact of toe-in and toe-out of both the front and rear wheels.

2) Slip ratio:

The tyre slip ratio enables the transmission of the longitudinal effort on the wheel for both accelerating and braking. According to [7], the longitudinal slip ratio is given by

$$s_i = \frac{\omega_i R_i - v_{x,i}}{v_{x,i}} \tag{10}$$

where ω_i is the rotational speed of the wheel i, R_i is the effective tyre radius and $v_{x,i}$ is the longitudinal speed of the centre of the wheel i.

The rotational speed of each wheel is obtained by combining both the driving/braking torque applied to the wheel T_i and the ground reaction

$$I_{y,i}\dot{\omega}_i = T_i - R_i F_{x,i} \tag{11}$$

where $I_{y,i}$ is the rotational moment of inertia of the wheel.

This time, the rotational moment of inertia of a wheel can be computed assuming the fact a wheel can be modelled by a thick-walled hollow cylinder leading to

TABLE II: Tyre models comparison.

Model	Parameters	Comments
Linear	Longitudinal and lateral tyre stiffnesses (K_s, K_α)	Simple Model Decoupled longitudinal and lateral responses Limited dynamics amplitude
Dugoff	Friction coeffiencient μ Longitudinal and lateral tyre stiffnesses (K_s, K_α)	Simple Model Coupled longitudinal and lateral responses Moderated dynamics amplitude
"Magic Formula" by Hans B. Pacejka	Empirical coefficients	Reference model Coupled longitudinal and lateral responses Wide validity domain

$$I_{y,i} = \frac{1}{2} m_{Wheel}(R_{Inner}^2 + R_{Outer}^2)) = 2.076e-05 kg.m^2 \tag{12}$$

By combining equations 10 and 11, we obtain the derivative of the tyre slip used for implementation

$$\dot{s_i} = \frac{\dot{\omega_i} R_i}{v_{x,i}} - \frac{\omega_i R_i \dot{v_{x,i}}}{v_{x,i}^2} \tag{13}$$

C. Powertrain model

The motor model used in this project is a constant torque - constant power model. The motor torque T_M can be expressed as a function of the motor rotational speed ω_M as following

$$T_M = \gamma \frac{\eta_i P_{Max}}{max(\omega_L, \omega_M)} \tag{14}$$

where γ is the throttle percentage, P_{Max} is the battery maximum power and η_i is the inverter efficiency. ω_L is the rotational speed limit between the constant torque and the constant speed domain.

The throttle percentage is saturated within the range $[-1, 1]$ with $\gamma = 1$ being the full throttle and $\gamma = -1$ being the maximum braking command.
The battery maximum power is set to be its nominal voltage $7.6V$ with a discharge current of $100A$. The inverter efficiency here is chosen to be $\eta_i = 0.8$.
RC car motors can reach a rotational speed between $20000RPM$ and $25000RPM$. The speed threshold between the two behaviour domains can be set to be $\omega_L = 1000RPM$.
The torque at each wheel (four wheel drive) is then given by

$$T_i = \frac{1}{4} \eta_d G_j T_M \tag{15}$$

where η_d is the drive train efficiency and G_j is the jth gear ratio.

Similarly to the inverter efficiency, the drive train efficiency is set to be $\eta_d = 0.8$.
In this project, the gear ratio is set to be $G_1 = 3.325$ (the car considered has only one gear ratio).

D. Tyre normal load

One of the dominant parameters to compute the tyre grip is their normal load. This force can be decomposed as the sum of a static component $F_{z,i,0}$ and a dynamic one $F_{z,i,d}$.
The static components are given by

$$F_{z,FL,0} = \frac{m_{Car} g l_r}{2(l_f + l_r)} \tag{16a}$$

$$F_{z,FR,0} = \frac{m_{Car} g l_r}{2(l_f + l_r)} \tag{16b}$$

$$F_{z,RL,0} = \frac{m_{Car} g l_f}{2(l_f + l_r)} \tag{16c}$$

$$F_{z,RR,0} = \frac{m_{Car} g l_f}{2(l_f + l_r)} \tag{16d}$$

The central position of the centre of gravity leads to the following simplification

$$F_{z,i,0} = \frac{m_{Car} g}{4} \tag{17}$$

The dynamic load is due to the vehicle sprung mass $m_{Car,S}$ (being the mass of the car without its wheels) subject to acceleration and is given by

$$F_{z,FL,d} = -\frac{m_{Car,S} \dot{v_x} h_{CG}}{2(l_f + l_r)} - \frac{m_{Car,S} \dot{v_y} h_{RC} \kappa_f}{2 l_s} \tag{18a}$$

$$F_{z,FR,d} = -\frac{m_{Car,S} \dot{v_x} h_{CG}}{2(l_f + l_r)} + \frac{m_{Car,S} \dot{v_y} h_{RC} \kappa_f}{2 l_s} \tag{18b}$$

$$F_{z,RL,d} = \frac{m_{Car,S} \dot{v_x} h_{CG}}{2(l_f + l_r)} - \frac{m_{Car,S} \dot{v_y} h_{RC} \kappa_r}{2 l_s} \tag{18c}$$

$$F_{z,RR,d} = \frac{m_{Car,S} \dot{v_x} h_{CG}}{2(l_f + l_r)} + \frac{m_{Car,S} \dot{v_y} h_{RC} \kappa_r}{2 l_s} \tag{18d}$$

where h_{CG} is the height of the centre of gravity of the car, h_{RC} is the height of the roll centres (considered equal at the front and the rear) and κ_f, κ_r are the roll stiffness factor of the front and rear suspension and $\kappa_f + \kappa_r = 1$.

The height of the roll centres is a predominant parameter when dealing with vehicle setup. The lower the roll centre is (but keeping it above the ground), the less the car will roll. After running some simulation, it appears that a good value for the roll centres height is $h_{RC} = 0.01m$.
In order to keep the relative symmetry of the car, the roll stiffnesses are set to be equal at the front and rear i.e. $\kappa_f = \kappa_r = 0.5$.

E. Linear tyre model

The linear tyre model can be expressed as

$$F_{x,i} = K_s s_i F_{z,i} \tag{19a}$$

$$F_{y,i} = K_\alpha \alpha_i F_{z,i} \tag{19b}$$

where K_s and K_α denotes the tyre longitudinal and lateral stiffness respectively.

F. Dugoff tyre model

The Dugoff model described in [8] is given by

$$F_{x,i} = K_s \frac{s_i}{1 + s_i} f(\lambda_i) \tag{20a}$$

$$F_{y,i} = K_\alpha \frac{tan(\alpha_i)}{1 + s_i} f(\lambda_i) \tag{20b}$$

where K_s and K_α are the same the tyre longitudinal and lateral stiffnesses as in the linear model.

As these two parameters are not available for RC car tyres, these were estimated by trial and error in simulation. The main observation in these simulations is that the lateral stiffness should be larger than the longitudinal one. This led me to choose the following parameters values : $K_s = 500 rad^{-1}$ and $K_\alpha = 1000 rad^{-1}$.

Finally, λ_i and the function $f(\lambda_i)$ are given by

$$\lambda_i = \frac{\mu_i F_{z,i}(1 + s_i)}{2\sqrt{(K_s s_i)^2 + (K_{\alpha_i} tan(\alpha_i))^2}} \tag{21}$$

$$f(\lambda_i) = \begin{cases} (2 - \lambda_i)\lambda_i & \lambda_i < 1 \\ 1 & \lambda_i \geq 1 \end{cases} \tag{22}$$

where $F_{z,i}$ denotes the normal load on the tyre and μ_i the maximum friction coefficient.

The value of this tyre friction coefficient is chosen to match the one of full-scale motorsport car in dry weather: $\mu_i = 1.7$

Fig. 5 and Fig. 6 respectively depict the normalized tyre longitudinal and lateral forces as a function of the slip ratio and slip angle based on the Dugoff model.

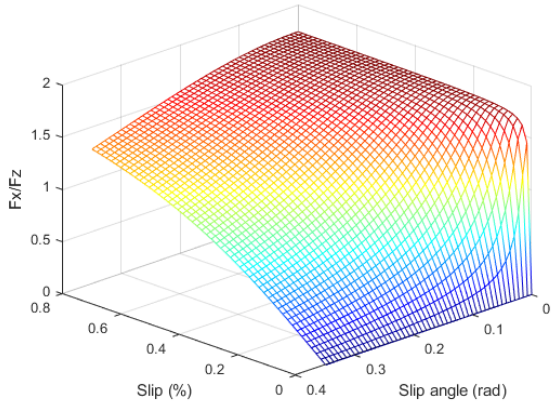

Fig. 5: Normalized tyre longitudinal force as a function of the slip ratio and slip angle.

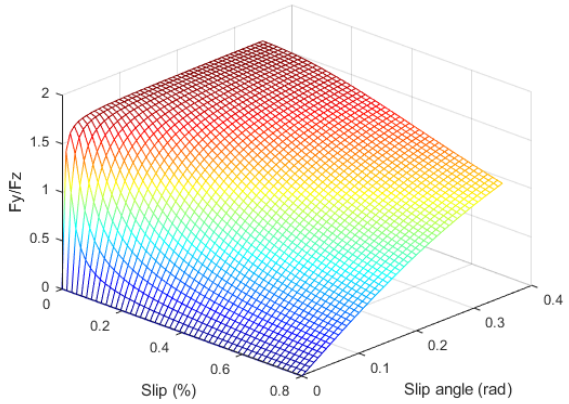

Fig. 6: Normalized tyre lateral force as a function of the slip ratio and slip angle.

G. Resulting tyre forces

Finally, the generalised tyre-related forces and moment are obtained by combining the forces on each of the four tyres. According to [4], these are given by

$$\begin{aligned} F_{x,Tyres} = \; & F_{x,FL}cos(\delta_{FL}) - F_{y,FL}sin(\delta_{FL}) \\ & + F_{x,FR}cos(\delta_{FR}) - F_{y,FR}sin(\delta_{FR}) \\ & + F_{x,RL}cos(\delta_{RL}) - F_{y,RL}sin(\delta_{RL}) \\ & + F_{x,RR}cos(\delta_{RR}) - F_{y,RR}sin(\delta_{RR}) \end{aligned} \tag{23a}$$

$$\begin{aligned} F_{y,Tyres} = \; & F_{x,FL}sin(\delta_{FL}) + F_{y,FL}cos(\delta_{FL}) \\ & + F_{x,FR}sin(\delta_{FR}) + F_{y,FR}cos(\delta_{FR}) \\ & + F_{x,RL}sin(\delta_{RL}) + F_{y,RL}cos(\delta_{RL}) \\ & + F_{x,RR}sin(\delta_{RR}) + F_{y,RR}cos(\delta_{RR}) \end{aligned} \tag{23b}$$

$$\begin{aligned} M_{z,Tyres} = \; & l_s(-F_{x,FL}cos(\delta_{FL}) + F_{y,FL}sin(\delta_{FL})) \\ & + l_s(F_{x,FR}cos(\delta_{FR}) - F_{y,FR}sin(\delta_{FR})) \\ & + l_s(-F_{x,RL}cos(\delta_{RL}) + F_{y,RL}sin(\delta_{RL})) \\ & + l_s(F_{x,RR}cos(\delta_{RR}) - F_{y,RR}sin(\delta_{RR})) \\ & + l_f(F_{x,FL}sin(\delta_{FL}) + F_{y,FL}cos(\delta_{FL})) \\ & + l_f(F_{x,FR}sin(\delta_{FR}) + F_{y,FR}cos(\delta_{FR})) \\ & + l_r(-F_{x,RL}sin(\delta_{RL}) - F_{y,RL}cos(\delta_{RL})) \\ & + l_r(-F_{x,RR}sin(\delta_{RR}) - F_{y,RR}cos(\delta_{RR})) \end{aligned} \tag{23c}$$

IV. Vehicle Model

In this project, the vehicle model has been implemented on *Matlab/Simulink* for simulation purposes.

The vehicle dynamics model is the one presented in section II and the tyre model considered is the Dugoff one presented in section III-F.

As far as the simulation sample time is concerned, the one used for the vehicle model is $T_{s,VM} = 0.001s$.

A. Vehicle model flowchart

The vehicle model is implemented following the flowchart described in Fig. 7.

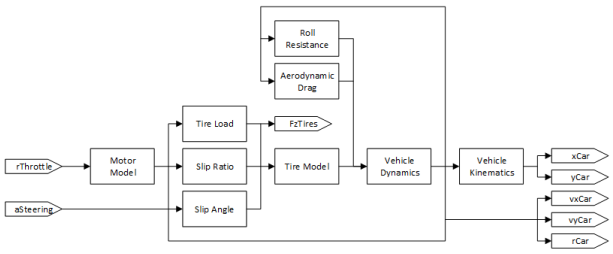

Fig. 7: Vehicle model flowchart.

B. First simulation results

1) Straight line acceleration:

For this first test, let's consider a throttle command step at time $2s$. The evolution of the car speed v_x is given in Fig. 8.

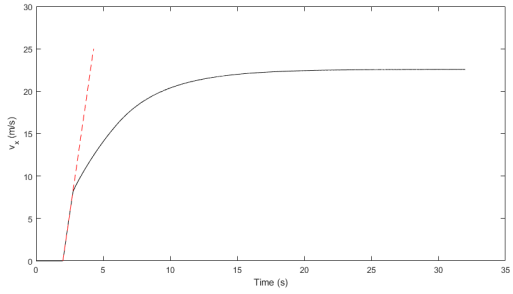

Fig. 8: Car speed v_x during a straight line acceleration.

As it can be seen, the maximum reachable speed is around $22.5m/s$ which is coherent with the maximum speed of a real RC car.

Moreover, the two motor behaviour domains (constant torque and constant power) can clearly be identified in this figure. The constant torque domain corresponds to a linear acceleration of the car (depicted by the red dashed line).

2) Constant speed turning:

In this test, the car speed is regulated to be $2m/s$. The steering command implemented is the one depicted in Fig. 9

The resulting car position is depicted on the Fig. 10.

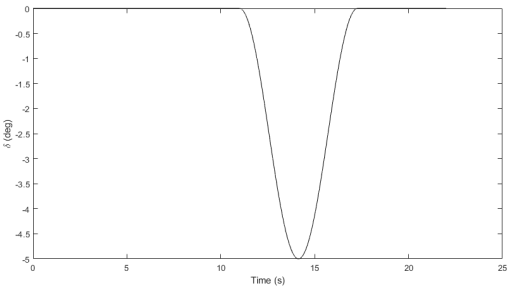

Fig. 9: Steering command δ.

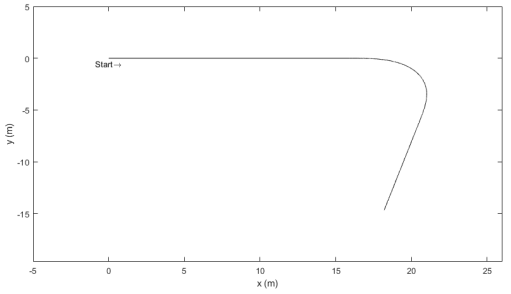

Fig. 10: Car position (x, y).

3) Variable speed turning:

In this test, the steering angle is set to $\delta = -5deg$ and the speed is regulated to follow a ramp as depicted in Fig. 11.

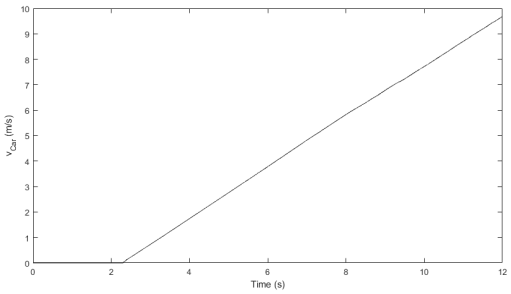

Fig. 11: Car speed v_{Car}.

The resulting car position is depicted in Fig. 12.

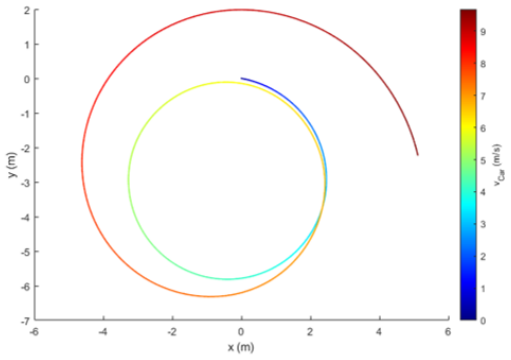

Fig. 12: Car position (x, y).

As it can be seen, the increase of the car speed implies lateral deviation from the trajectory. This deviation results from the action of the centrifugal force applied on the car when turning.

V. Vehicle Steering Control

This section aims at presenting the vehicle steering controller. Different control techniques have been developed in the recent years. [5] has classified the path tracking system in three approaches: geometric, kinematic and dynamic model-based. A comparison summary of these steering controllers is presented in table III. Among these, the pure pursuit algorithm appears to be one of the most commonly used and will be considered in this project. The description of the pure pursuit algorithm is based on the article [5].

A. Pure pursuit algorithm

For simplification purposes, the vehicle is represented in this section as a bicycle (Fig. 13). This model is often used when describing steering controllers in order to avoid any Ackermann steering related considerations.

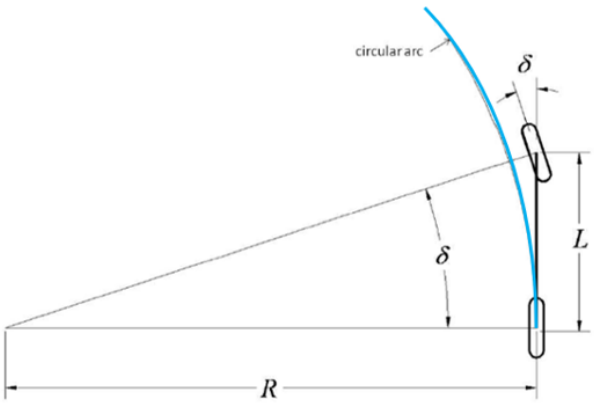

Fig. 13: Geometric bicycle model. Adapted from [5].

On this figure, the wheelbase L corresponds to the sum $l_f + l_r$ on the Fig. 4.

As it can be seen, the geometric relation between the steering angle δ and the circular arc radius R is given by

$$\tan(\delta) = \frac{l_f + l_r}{R} \tag{24}$$

The pure pursuit algorithm then uses geometric consideration to determine the required steering angle.

The first step of the pure pursuit algorithm is to target a goal point (denoted (g_x, g_x) on the Fig. 14) located on the desired trajectory. This goal point is defined to be the point on the trajectory at the look-ahead distance l_d from the vehicle.

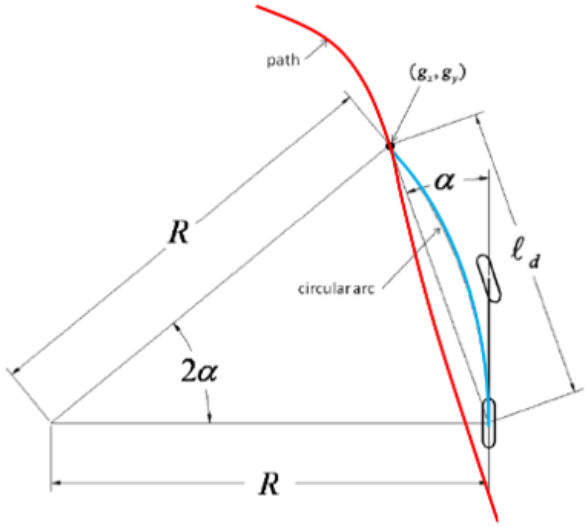

Fig. 14: Pure pursuit geometry. Adapted from [5].

It is then possible to define the heading error α of the car as well as the target turning radius R to bring the car back on the trajectory. This turning radius can be expressed as

$$R = \frac{l_d}{2\sin(\alpha)} \tag{25}$$

Finally, combining the equations 24 and 25 leads the steering angle δ

$$\delta = \tan^{-1}\left(\frac{2(l_f + l_r)\sin(\alpha)}{l_d}\right) \tag{26}$$

B. Pure pursuit interpretation

Let κ denote the circular arc curvature. The steering angle δ (equation 24) can be rewritten

$$tan(\delta) = \kappa(l_f + l_r) \tag{27}$$

Then, the equation 25 can be rewritten in terms of turning curvature as

$$\kappa = \frac{2\sin(\alpha)}{l_d} \tag{28}$$

Then, the pure pursuit algorithm interpretation requires the introduction of the lateral distance e_{l_d} between the heading vector and the goal point

$$\sin(\alpha) = \frac{e_{l_d}}{l_d} \tag{29}$$

TABLE III: Steering controllers comparison.

Model	Parameters	Comments
Geometric	Pure pursuit Stanley method	Simple and robust approaches Cutting or overshooting corners at high speed Steady state error in curves when speed increases
Kinematic	Kinematic controller	Poor robustness Heavy requirement on the path geometry Similar performance compared to geometric controllers
Dynamic	Optimal controller Optimal preview controller	Vehicle model linearisation is required Poor to fair robustness Cutting or overshooting corners in changing condition (speed or curvature) Small steady state error in curves

Combining the equation 28 and 29 leads to

$$\kappa = \frac{2}{l_d^2} e_{l_d} \tag{30}$$

The circular arc curvature (and therefore the steering angle δ) can be interpreted as a proportional controller with a gain of $2/l_d^2$. In practice, the look-ahead distance is not constant but tuned to be speed dependant.

C. Pure pursuit implementation

The pure pursuit algorithm is implemented following the flowchart described in Fig. 15.

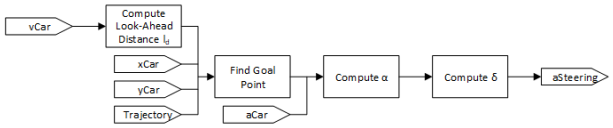

Fig. 15: Pure pursuit algorithm flowchart.

The sampling time for the steering controller is set to $T_{s,PP} = 0.01s$. It would indeed not have any sense to make it faster or as fast as the vehicle model which is supposed to be a continuous-time model.

D. Pure pursuit tuning

As far as the tuning of the pure pursuit algorithm is concerned, it is a matter of finding the correct look-ahead distance. The Fig. 16 depicts the tracking result for different look-ahead distances at a fixed speed of $v_{Car} = 10m/s$. The tracking precision is measured thanks to two indicators: the peak and the mean errors. These tracking errors are measured as the lateral deviation from the target trajectory (the centerline of the track in this case). These tracking errors are presented in table IV.

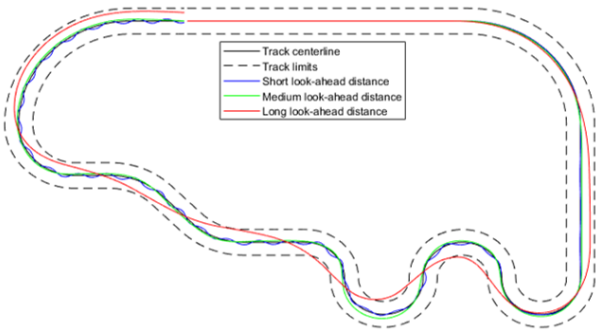

Fig. 16: Pure pursuit tracking result for different look-ahead distances on a test track.

TABLE IV: Pure pursuit tracking errors for different look-ahead distances on an oval track.

Look-ahead distance	Peak error	Mean error
Short ($l_d = 2m$)	0.57m	0.14m
Medium ($l_d = 4m$)	0.86m	0.21m
Long ($l_d = 10m$)	3.34m	1.03m

As it can be seen in Fig. 16, a too short look-ahead distance l_d (in blue) results in oscillation around the target trajectory which can be interpreted as a lack of anticipation. This can also the interpreted as a controller with too high gain, since from equation 30 the gain is inversely proportional to l_d. On the contrary, a too long look-ahead distance l_d (in red) makes the car cut the corners as a result of excessive anticipation. Finally, a medium look-ahead distance (in green) achieves good tracking results.

As mentioned before, the optimal look-ahead distance also depends on the car velocity. In order to see the influence of the speed on the look-ahead distance, the Fig. 17 depicts the trajectory tracking results for different speeds with a fixed look-ahead distance $l_d = 2m$. These tracking errors are presented in table V.

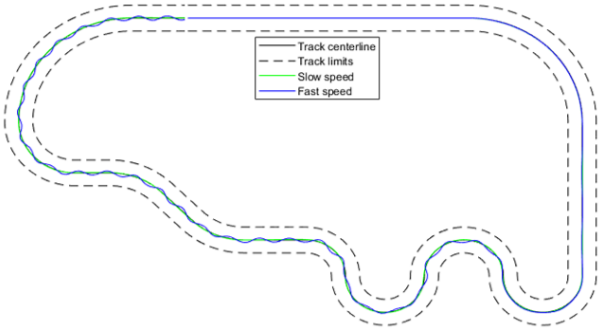

Fig. 17: Pure pursuit tracking result for different look-ahead distances on a test track.

TABLE V: Pure pursuit tracking errors for different look-ahead distances on an oval track.

Speed	Peak error	Mean error
Slow ($v_{Car} = 5m/s$)	0.15m	0.04m
Fast ($v_{Car} = 10m/s$)	0.57m	0.15m

As it can be seen in Fig. 17, the look-ahead distance achieving a good tracking at a slow speed does not perform as well with a higher speed. This shows the importance to tune a speed-dependent look-ahead distance.

Let's now consider that the look-ahead distance has a saturated linear dependency on the car speed.
The lower bound for the saturation can be set to $1m$. Indeed, a shorter look-ahead distance, even at a slow speed, results in an underdamped behaviour with oscillations around the desired trajectory. As far as the upper limit is concerned, its value is based on the track width. Indeed, consider the track depicted in Fig. 18.

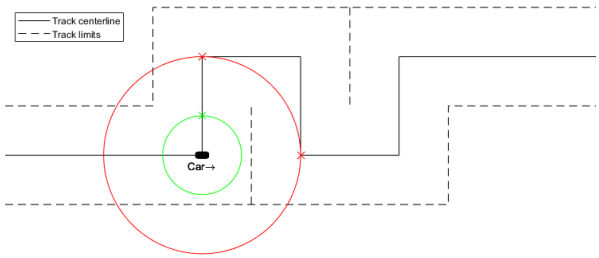

Fig. 18: Look-ahead distance on a tight track (indoor).

As it can be seen (when considering the red circle), if the look-ahead distance is greater or equal than the track width,

there exists more than one possible goal point (red markers). Following this reasoning, the upper look-ahead distance limit is set to $5m$.
This can be interpreted as an application of Shannon's theorem. In this case, the wavelength is $2L$. The critical sampling distance is therefore the half of this wavelength, i.e. L.

Denote t the linear gain between the look-ahead distance and the car speed ($l_d = tv_{Car}$). The gain t is chosen to scale a car speed of $v_{Car} = 20m/s$ with the maximum look-ahead distance, i.e. $t = 0.25$.
Finally, the look-ahead distance is given by

$$l_d = \begin{cases} 1 & \text{if} \quad v_{Car} \leq 5m/s \\ 0.25v_{Car} & \text{if} \quad 5m/s < v_{Car} < 20m/s \\ 5 & \text{if} \quad v_{Car} \geq 20m/s \end{cases} \tag{31}$$

VI. Vehicle Speed Control

Contrary to the trajectory tracker, the speed controller requires two steps: to generate the speed target and to regulate the car speed to reach this target.

In a certain sense, the target speed computation can be thought of as a non-linear feedforward controller.

A. Target speed computation

The generation of the speed target is based on the friction circle model. As mentioned in [10], the idealized tyre-force circle is described by

$$\frac{F_y^2}{\mu^2 F_z^2} + \frac{F_x^2}{\mu^2 F_z^2} \leq 1 \tag{32}$$

As an approximation, it can be considered that the car turns at a constant speed, i.e. $F_x \simeq 0$. This means that the speed regulation happens before the corner entry and after the corner exit. In this case, the equation 32 becomes

$$F_y \leq \mu F_x \tag{33}$$

Interpreting F_y as the centrifugal force applied on the car gives a relationship between the speed and the radius of curvature of the trajectory. Combining this with the equation 33, we obtain the following speed limit $v_{x,l}$

$$v_{x,l}^2 \leq \frac{\mu F_z}{m\kappa} \tag{34}$$

where κ is the track curvature.
Finally, let's introduce a margin coefficient η_v chosen between 0 and 1 to define the target speed $v_{x,t}$ as

$$v_{x,t} = \eta_v \sqrt{\frac{\mu F_z}{m\kappa}} \tag{35}$$

As it can be intuited, braking requires anticipation. Indeed, braking when the required curvature exceeds the turning ability of the car is often too late. If l_a denotes the anticipation distance, κ will be the maximum curvature of the track over the anticipation distance l_a. As for the trajectory controller, the degree of anticipation depends on the current speed of the car.

B. Target speed tuning

As far as the speed controller tuning is concerned, the task consists in tuning the anticipation distance l_a.
This tuning requires the simulation of test runs in order to determine the distance to full stop from a given speed with a given braking command. In this case, only two test runs are considered, from an initial speed of $22m/s$, with a braking command of $\gamma = -1$ and $\gamma = -0.8$ respectively. Note that in this case, the test runs are done in straight line. The braking results are depicted in Fig. 19.
As it can be seen in Fig. 19, the braking distance from any initial speed between $0m/s$ and $22m/s$ with one of the considered braking commands can easily be extracted from the related test run.
A first possibility would be to set the anticipation distance to be the distance to full stop from $22m/s$ with a braking command $\gamma = -0.8$ (and not $\gamma = -1$ to keep some margin), i.e. $l_a = 48.5m$. However, such an approximation would be largely

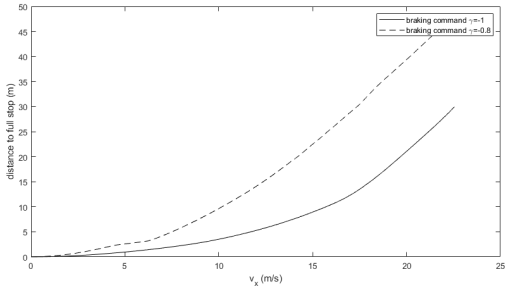

Fig. 19: Distance to full stop (m) as a function of the car speed (m/s) for different braking commands.

suboptimal since it would consist in far too early braking when the initial speed is lower than $22m/s$.

In order to avoid this excess of anticipation, the anticipation distance is set to be speed dependent. The anticipation distance is therefore set to approximate the distance to full stop based on the initial speed, for a braking command $\gamma = -0.8$ (and not $\gamma = -1$ to keep some margin). Moreover, a minimum anticipation distance is set to $5m$ to avoid any slow speed related issues.

C. Vehicle model linearisation

The friction circle model provides a speed target. A controller needs to be designed to regulate the car speed as close as possible to this speed target. Because of the complexity and non-linearities of the vehicle and tyre models, designing a speed controller is not straightforward. Deriving a linear model would provide a basis to apply the classical control theory.

1) Equations simplification:

When simplifying the friction circle equation to obtain the speed target, we assumed that any acceleration or braking is mainly done in straight line (braking before the corners and accelerating after). The main focus in this section will therefore be to linearise the vehicle model in the straight line case. This assumption can be expressed as

$$\dot{v_x} = \frac{1}{m_{Car}} F_x \tag{36a}$$
$$\dot{v_y} = 0 \tag{36b}$$
$$\dot{r} = 0 \tag{36c}$$

Moreover, when considering straight-line dynamics, it can be assumed the four wheels have the same longitudinal and rotational speed. Indexing the different parameters and states on the wheels is therefore not required in this section.

Then, for simplification purposes, the tyre model used in this section is the linear one described in section III-E. As far as the tyre normal load is concerned, only its static component will be used here ($F_z = F_{z,0}$).

The first and most obvious state to consider in this situation is the longitudinal speed v_x

$$\begin{aligned}
\dot{v_x} &= \frac{1}{m_{Car}} F_x + r v_y \\
&= \frac{1}{m_{Car}} F_x \\
&= \frac{1}{m_{Car}} \left(F_{x,Tyres} - F_{Drag} - F_{x,Roll} \right) \\
&= \frac{1}{m_{Car}} \left(\sum_i \left(F_{x,i} \right) - F_{Drag} - F_{x,Roll} \right) \\
&= \frac{1}{m_{Car}} \left(4F_x - F_{Drag} - F_{x,Roll} \right) \\
&= \frac{1}{m_{Car}} \left(4K_s s F_z - \underbrace{\frac{1}{2} \rho_{Air} S C_x}_{\overline{C_x}} v_x^2 - \underbrace{m_{Car} g C_{RR}}_{\overline{C_{RR}}} v_x \right) \\
&= \frac{1}{m_{Car}} \left(4K_s \left(\frac{\omega R}{v_x} - 1 \right) F_z - \overline{C_x} v_x^2 - \overline{C_{RR}} v_x \right)
\end{aligned} \tag{37}$$

As it can be seen in the equation 37, another state needs to be considered to induce a non zero slip ratio responsible for acceleration and braking. Let's then consider the wheel rotational speed ω

$$\begin{aligned}
\dot{\omega} &= \frac{1}{I_y} \left(T_i - R F_x \right) \\
&= \frac{1}{I_y} \left(\frac{1}{4} \eta_d G_j T_M - R K_s s F_z \right) \\
&= \frac{1}{I_y} \left(\frac{1}{4} \eta_d G_j \gamma \frac{\eta_i P_{Max}}{max(\omega_L, \omega_M)} - R K_s \left(\frac{\omega R}{v_x} - 1 \right) F_z \right)
\end{aligned} \tag{38}$$

For simplification purposes, let's assume that $\omega_L = 0$. This leads to

$$\begin{aligned}
\dot{\omega} &= \frac{1}{I_y} \left(\frac{1}{4} \eta_d G_j \gamma \frac{\eta_i P_{Max}}{\omega_M} - R K_s \left(\frac{\omega R}{v_x} - 1 \right) F_z \right) \\
&= \frac{1}{I_y} \left(\frac{1}{4} \eta_d G_j \gamma \frac{\eta_i P_{Max} R}{v_x G_j} - R K_s \left(\frac{\omega R}{v_x} - 1 \right) F_z \right) \\
&= \frac{1}{I_y} \left(\underbrace{\frac{1}{4} \eta_d \eta_i P_{Max} R}_{\overline{P_M}} \frac{\gamma}{v_x} - R K_s \left(\frac{\omega R}{v_x} - 1 \right) F_z \right)
\end{aligned} \tag{39}$$

As it can be seen from the equation 39, no additional state is required for this simplified system.

2) Equilibrium points:

The equilibrium points of the system 39 and 37 satisfy

$$\dot{\omega} = 0 \tag{40a}$$
$$\dot{v_x} = 0 \tag{40b}$$

Using the equation 39, the equation 40a leads to

$$
\begin{aligned}
\dot{\omega} = 0 &\Rightarrow \overline{P_M}\frac{\gamma}{v_x} - RK_s\left(\frac{\omega R}{v_x} - 1\right)F_z = 0 \\
&\Rightarrow \overline{P_M}\gamma - RK_s\left(\omega R - v_x\right)F_z = 0 \\
&\Rightarrow R^2K_s\omega F_z = \overline{P_M}\gamma + RK_s v_x F_z \\
&\Rightarrow \omega = \frac{\overline{P_M}}{R^2K_sF_z}\gamma + \frac{1}{R}v_x \\
&\Rightarrow \omega = \frac{\eta_d\eta_i P_{Max}}{4RK_sF_z}\gamma + \frac{1}{R}v_x
\end{aligned} \tag{41}
$$

Using the equation 37, the equation 40b leads to

$$
\begin{aligned}
\dot{v_x} = 0 &\Rightarrow 4K_s\left(\frac{\omega R}{v_x} - 1\right)F_z - \overline{C_x}v_x^2 - \overline{C_{RR}}v_x = 0 \\
&\Rightarrow 4K_s\left(\omega R - v_x\right)F_z - \overline{C_x}v_x^3 - \overline{C_{RR}}v_x^2 = 0 \\
&\Rightarrow 4K_s\left(\frac{\eta_d\eta_i P_{Max}}{4K_sF_z}\gamma\right)F_z - \overline{C_x}v_x^3 - \overline{C_{RR}}v_x^2 = 0 \\
&\Rightarrow \eta_d\eta_i P_{Max}\gamma - \overline{C_x}v_x^3 - \overline{C_{RR}}v_x^2 = 0 \\
&\Rightarrow \eta_d\eta_i P_{Max}\gamma = \overline{C_x}v_x^3 + \overline{C_{RR}}v_x^2
\end{aligned} \tag{42}
$$

Finding the equilibrium speed for a given throttle command γ therefore consists in finding the real positive root of the equation 42. Given that both $\overline{C_x}$ and $\overline{C_{RR}}$ are positive real numbers, this guarantee the uniqueness of the real positive root of the equation 42. This unique solution is given in Fig. 20

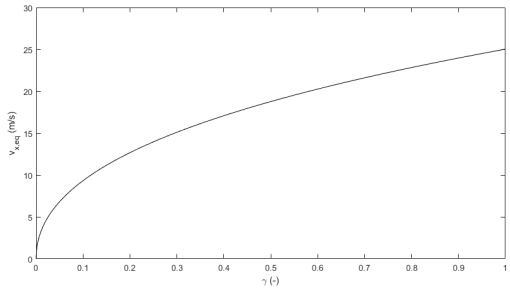

Fig. 20: Equilibrium longitudinal speed as a function of the throttle command γ.

It can be noted that this result is coherent with the straight-line acceleration test depicted in Fig. 8.
The equilibrium rotational speed of the wheel can then be computed thanks to the equation 41. The result is depicted in Fig. 21.

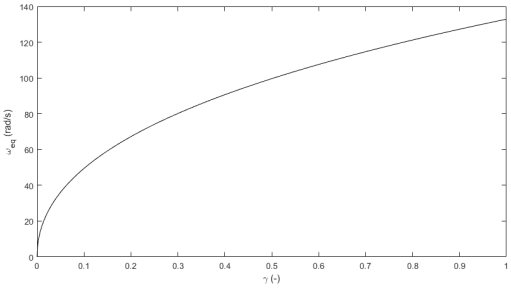

Fig. 21: Equilibrium rotational speed of the wheel as a function of the throttle command γ.

3) Space state model:

Denote $\chi = \begin{bmatrix}\omega & v_x\end{bmatrix}^T$ the state of this system. The state matrix of this system is the Jacobian matrix associated with the equations 39 and 37

$$
A(\chi) = \begin{bmatrix}
-\frac{R^2K_sF_z}{I_y}\frac{1}{v_x} & \frac{R^2K_sF_z}{I_y}\frac{\omega}{v_x^2} - \frac{\eta_d\eta_i P_{Max}R}{4}\frac{\gamma}{v_x^2} \\
\frac{4RK_sF_z}{m_{Car}}\frac{1}{v_x} & -\frac{4RK_sF_z}{m_{Car}}\frac{\omega}{v_x^2} - \frac{\rho_{Air}SC_x}{m_{Car}}v_x - gC_{RR}
\end{bmatrix} \tag{43}
$$

The input matrix of this system is given by

$$
B(\chi) = \begin{bmatrix}\frac{\eta_d\eta_i P_{Max}R}{4I_y}\frac{1}{v_x} \\ 0\end{bmatrix} = \frac{\eta_d\eta_i P_{Max}R}{4I_y}\frac{1}{v_x}\begin{bmatrix}1\\0\end{bmatrix} \tag{44}
$$

D. Speed controller design

This section aims at designing a controller based on the previous model linearisation.
The controller design setup is depicted in Fig. 22.

Fig. 22: Speed controller design setup.

The initial speed is set to $v_{x,0} = 5m/s$. Indeed, except from the start of a run, the car speed only rarely goes below $5m/s$, even in tight corners. Then, as far the speed target is concerned, it has been chosen to reproduce a speed target in a tight corner section and is depicted in Fig. 23.

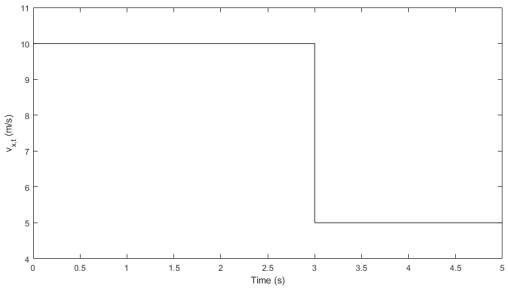

Fig. 23: Speed controller design setup.

Given that the car speed in this design setup should remain between $5m/s$ and $10m/s$, the state-space matrices are the system linearisation corresponding to equilibrium speeds of $v_{x,eq} \in \{5, 6, 7, 8, 9, 10\}$.

1) Proportional controller:

Denote $K_{p,SC}$ the proportional controller gain. Setting $K_{p,SC} = 1$ (unity gain) gives the results depicted on the Fig. 24 and Fig. 25.
As it can be seen on the zoomed control result, there is some steady-state error. A simple proportional controller is therefore not sufficient to perform accurate speed control.
Moreover, even if increasing the proportional gain $K_{p,SC}$ would reduce the steady-state error, it will introduce high gain feedback issues such as the amplification of noise. This is not desirable given that the velocity target is rapidly evolving (Fig. 26).

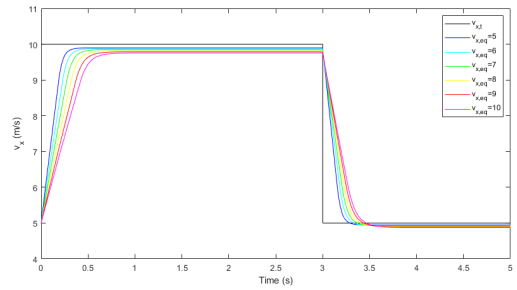

Fig. 24: Proportional speed control (global) result.

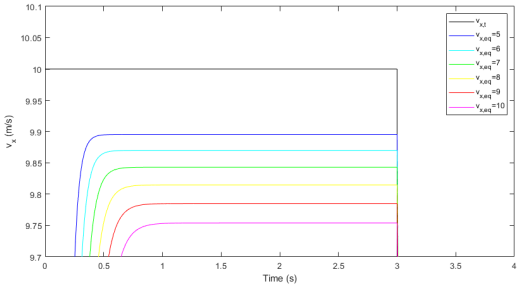

Fig. 25: Proportional speed control (zoomed) result.

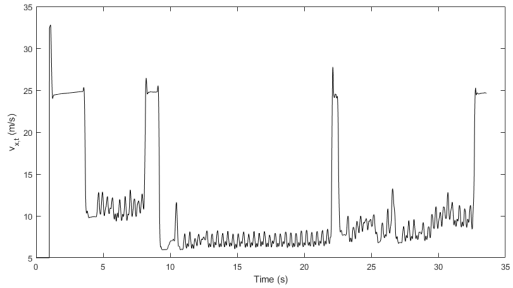

Fig. 26: Speed target $v_{x,t}$ over one lap on the test track.

2) Proportional integral (PI) controller:

In order to get rid of the steady-state error seen with the proportional controller, let's introduce an integral component denoted $K_{i,SC}$. Setting $K_{i,SC} = 1$ gives the results depicted on the Fig. 27 and Fig. 28.

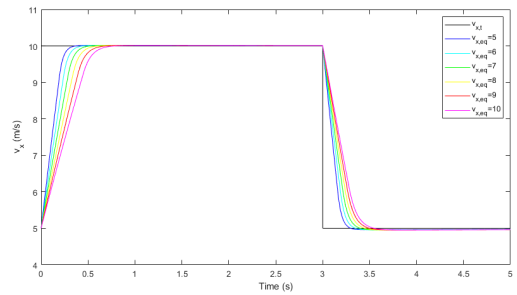

Fig. 27: Proportional integral speed control (global) result.

Introducing an integral component has the property to delete the steady-state error. This might not be obvious in Fig. 28. However, this is a consequence of the presence of an integrator with a saturation. As it can be seen in Fig. 29, the integrator value winds up and then slowly decrease.

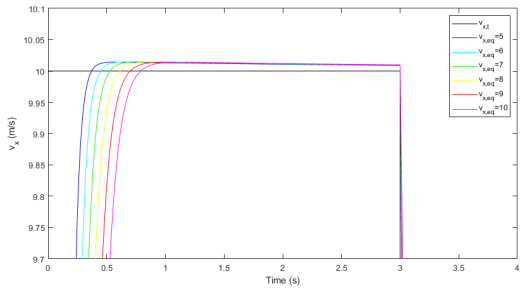

Fig. 28: Proportional integral speed control (zoomed) result.

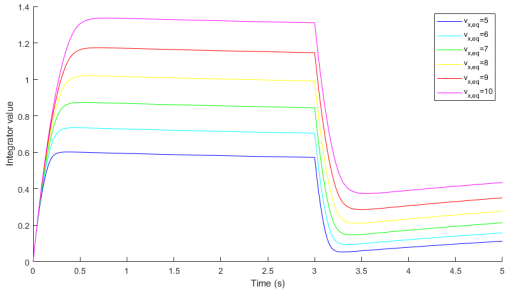

Fig. 29: Integrator wind up phenomenon.

3) Proportional integral with anti-windup (PI+AW) controller:

In order to avoid this integrator windup issue, let's introduce an anti-windup strategy. The latter is depicted in Fig. 30, with $K_{aw,SC}$ being the anti-windup gain. The purpose of this anti-windup gain is to enable to large integrator state to decay more quickly. Such an anti-windup scheme is rather simple, but quite effective in this instance.

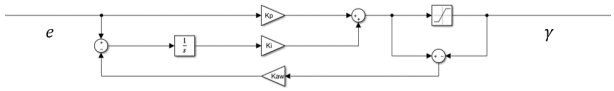

Fig. 30: Anti-windup sturcture.

Setting $K_{aw,SC} = 0.11$ gives the results depicted on the Fig. 31 and Fig. 32.

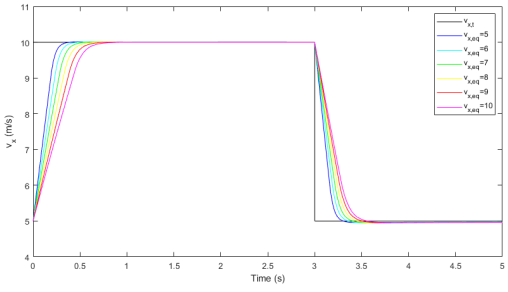

Fig. 31: Proportional integral with anti-windup speed control (global) result.

As it can be seen in Fig. 32, there is no steady-state error and the target speed is reached within $1.5s$.

However, given the rapidly evolving velocity target (Fig. 26), the speed controller mainly works in the transitional regime.

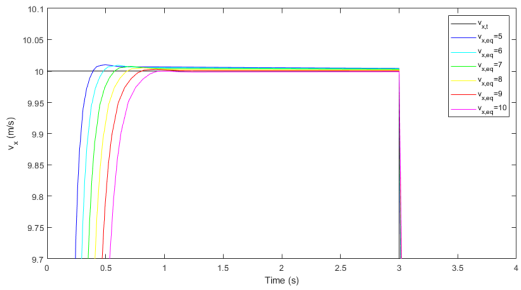

Fig. 32: Proportional integral with anti-windup speed control (zoomed) result.

Therefore, the improvement between the proportional controller and the proportional integral with anti-windup one is only marginal.

E. Speed controller implementation

The speed control is implemented following the flowchart described in Fig. 33.

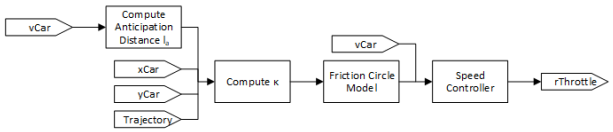

Fig. 33: Speed controller flowchart.

Similarly to the pure pursuit steering control, the sampling time of the speed controller is set to $T_{s,SC} = 0.01s$.

VII. Lap Time Measurement

A. Race track

The test track used for performance measurement is given in Fig. 34.

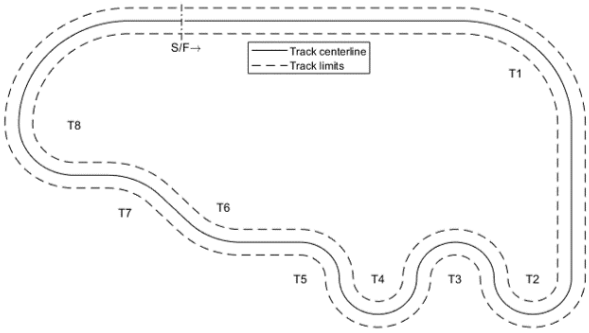

Fig. 34: Test track.

This test track is a prototype track designed to contain different types of corners. The first turn (T1) is a high-speed turn. Turns T2 to T5 is a sequence of slow speed turns. Hard braking is required before turn T2 to avoid understeer. Turns T6 and T7 consist in a medium speed chicane. This chicane can nearly be taken as a straight line from the exit of turn T5 to the entry of turn T8. Finally, the turn T8 is a curvature changing curve (curvature decrease through the turn) allowing acceleration within the turn.
As far as the dimensions are concerned, this prototype track is $291m$ long and $5m$ wide which is in agreement with the track regulation for $1/10$th RC car competition.

B. Measurement protocol

From now on, the vehicle model considered is the one described in section II with the Dugoff tyre model and the steering and speed controllers are the one described in section V and VI respectively.

As far as the performance measurement is concerned, the main indicator considered here is the lap times.
The testing protocol is a sequence of four laps. The sequence begins with a standing start on the middle of the start/finish line (denoted S/F in Fig. 34). This first lap is then followed by three rolling starts laps.

C. Track centerline tracking results

In this section, the trajectory considered is the track centerline. The related tracking results are presented in Fig. 35 and in table VI.

TABLE VI: Track centerline tracking results.

Time	
Lap 1	33.586s
Lap 2	32.214s
Lap 3	32.152s
Lap 4	32.168s
Total	130.120s

Error	
Peak	0.90m
Mean	0.15m

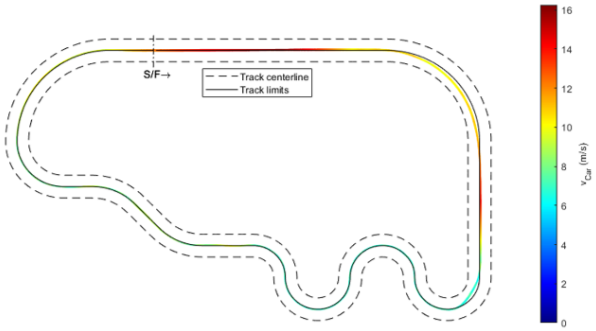

Fig. 35: Track centerline tracking results.

The Fig. 36 and Fig. 37 depict the steering and throttle input respectively.

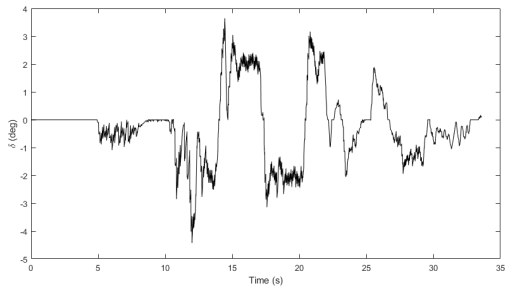

Fig. 36: Track centerline tracking results.

As it can be seen in Fig. 36, the steering angle control input is quite irregular in many places with the maximum steering rate in tight turns.

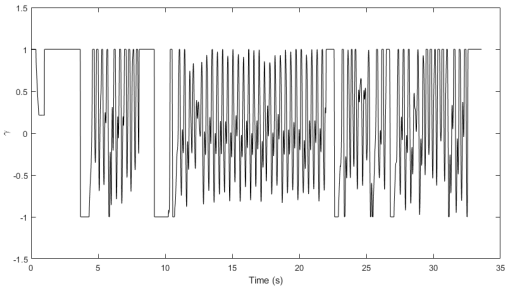

Fig. 37: Track centerline tracking results.

Note also that the throttle command (Fig. 37) is often saturated, and corresponds to a quite aggressive speed control system.

VIII. Trajectory Optimization

When driving on a racetrack, everything can be seen as a matter of optimisation to minimize the resulting lap times. For a given driving strategy (steering and throttle), choosing the right trajectory is the next optimization step.

A. Geometric problem formulation

In this section, the aim is to translate the trajectory optimisation problem into a geometric optimisation problem.
The parametrisation for this geometric problem is given in Fig. 38.

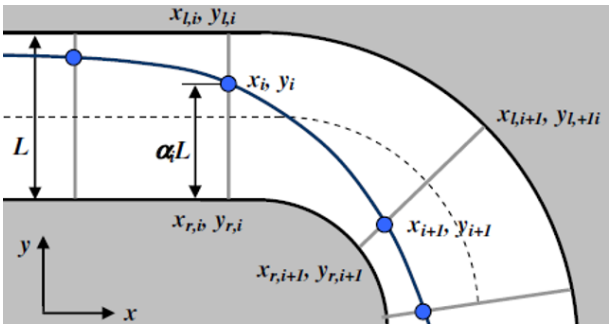

Fig. 38: Trajectory discretization. Extracted from [6].

As it can be seen in Fig. 38, each trajectory point can be written

$$\begin{aligned}\vec{P}_i &= x_i\vec{i} + y_i\vec{j} \\ &= (x_{r,i} + \tau_i \underbrace{(x_{l,i} - x_{r,i})}_{\Delta x_i})\vec{i} + (y_{r,i} + \tau_i \underbrace{(y_{l,i} - y_{r,i})}_{\Delta y_i})\vec{j} \end{aligned} \quad (45)$$

where τ_i is the optimization parameter. In order to keep the trajectory between the track limits, the parameters τ_i lives in the range $[0, 1]$. These optimization parameters can be collected into a vector $\bar{\tau}$

$$\bar{\tau} = \begin{bmatrix}\tau_1 & \tau_2 & \cdots & \tau_N\end{bmatrix} \quad (46)$$

where N is the number of points on the trajectory.

B. Optimization constraints

The main constraint for this trajectory optimizer is to keep the trajectory between the track limits. This constraint can be rewritten $\forall i \in [\![1, N]\!], \tau_i \in [0, 1]$. Moreover, given the possibility of deviation of the car from the desired trajectory, this constraint is reduced to $\forall i \in [\![1, N]\!], \tau_i \in [0.05, 0.95]$ which correspond to a $25cm$ margin.
The second constraint corresponds to the initial condition. As mentioned in section VII-B, the start of the performance measurement sequence takes place in the middle of the start/finish line. This initial condition can be written $\tau_1 = 0.5$. Note that this initial condition is in agreement with the previous constraint.

C. Shortest path trajectory

A first trajectory optimization strategy is to find the shortest path between the track limits. A simple way to estimate the trajectory length is to approximate it with discrete line segments between the trajectory points $\vec{P}_i$. The length of the ith segment can be written

$$\overrightarrow{P_{i+1}} - \vec{P_i} = \Delta P_{x,i}\vec{i} + \Delta P_{y,i}\vec{j} \tag{47}$$

where

$$\begin{aligned}\Delta P_{x,i} &= \tau_{i+1}\Delta x_{i+1} - \tau_i \Delta x_i + x_{r,i+1} - x_{r,i} \\ &= \begin{bmatrix}-\Delta x_i & \Delta x_{i+1}\end{bmatrix}\begin{bmatrix}\tau_i \\ \tau_{i+1}\end{bmatrix} + \Delta x_{i,0}\end{aligned} \tag{48a}$$

$$\begin{aligned}\Delta P_{y,i} &= \tau_{i+1}\Delta y_{i+1} - \tau_i \Delta y_i + y_{r,i+1} - y_{r,i} \\ &= \begin{bmatrix}-\Delta y_i & \Delta y_{i+1}\end{bmatrix}\begin{bmatrix}\tau_i \\ \tau_{i+1}\end{bmatrix} + \Delta y_{i,0}\end{aligned} \tag{48b}$$

Then, the squared total length of the trajectory can be expressed as

$$\begin{aligned}S^2 &= \sum_{i=1}^{N} \Delta P_{x,i}^T \Delta P_{x,i} + \Delta P_{y,i}^T \Delta P_{y,i} \\ &= \sum_{i=1}^{N} \bar{\tau_i}^T H_{S,i} \bar{\tau_i} + B_{S,i}\bar{\tau_i} + cost\end{aligned} \tag{49}$$

where

$$\bar{\tau_i} = \begin{bmatrix}\tau_i \\ \tau_{i+1}\end{bmatrix} \tag{50}$$

$$\begin{aligned}H_{S,i} = &\begin{bmatrix}\Delta x_i^2 & -\Delta x_i \Delta x_{i+1} \\ -\Delta x_i \Delta x_{i+1} & \Delta x_{i+1}^2\end{bmatrix} \\ &+ \begin{bmatrix}\Delta y_i^2 & -\Delta y_i \Delta y_{i+1} \\ -\Delta y_i \Delta y_{i+1} & \Delta y_{i+1}^2\end{bmatrix}\end{aligned} \tag{51}$$

$$\begin{aligned}B_{S,i} = &\begin{bmatrix}-2\Delta x_i \Delta x_{i,0} & 2\Delta x_{i+1}\Delta x_{i,0}\end{bmatrix} \\ &+ \begin{bmatrix}-2\Delta y_i \Delta y_{i,0} & 2\Delta y_{i+1}\Delta y_{i,0}\end{bmatrix}\end{aligned} \tag{52}$$

Finding the shortest path is therefore a quadratic problem where $\bar{\tau}$ is the optimization parameter.
Given that this problem is written in a standard form, it can easily be solved by using a quadratic programming solver. In this project, such a problem is solved using the *quadprog* command of *Matlab*.

The shortest path on the test track is given in Fig. 39.

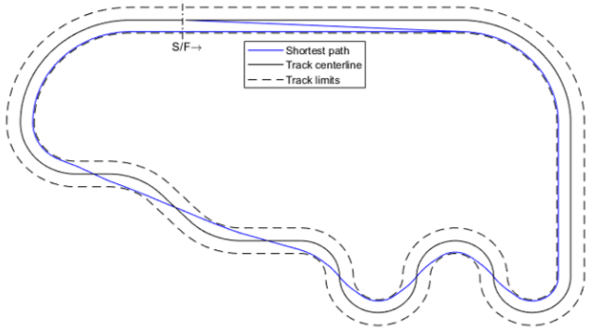

Fig. 39: Shortest path on the test track.

Following the lap time measurement protocol described in section VII, we obtain the results presented in Fig. 40 and in table VII.

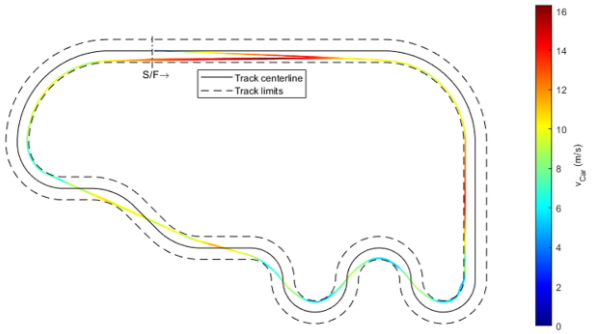

Fig. 40: Shortest path tracking result.

TABLE VII: Shortest path tracking results.

Time	
Lap 1	29.885s
Lap 2	28.588s
Lap 3	28.611s
Lap 4	28.496s
Total	115.580s

Error	
Peak	0.55m
Mean	0.11m

D. Minimum curvature trajectory

As described in the equation 35, the target velocity is inversely proportional to the future track curvature. It is therefore of interest to compute the minimum curvature trajectory between the track limits in order to maximise the speed of the car everywhere on the track.
For this problem, the target is to estimate and minimize the curvature at each trajectory point $\vec{P_i}$. A solution to estimate the curvature at $\vec{P_i}$ is to consider the second-order derivative of a polynomial (denoted $\tilde{P_i}$) fitted through the point $\overrightarrow{P_{i-1}}$ to $\overrightarrow{P_{i+1}}$ (a second-order polynomial can indeed be fully described by three points). The following parametrisation can therefore be used

$$\tilde{P_i} = a_i + b_i t + c_i t^2 \tag{53}$$

$$t = \frac{s - s_i}{s_{i+1} - s_{i-1}} \tag{54}$$

where s is in the range $[s_{i-1}, s_{i+1}]$ and denotes the curvilinear abscissa of a point on the trajectory (s_i being the curvilinear abscissa of the point P_i). t is therefore a dimensionless variable living in the range $[-1, 1]$.

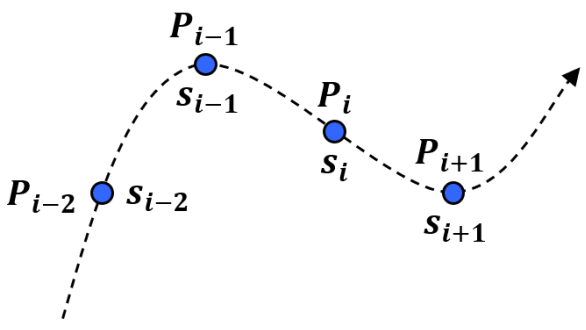

Fig. 41: Curvilinear abscissa example.

With this notation, finding the curvature at $\vec{P_i}$ consists in finding the c_i which is the curvature of $\tilde{P_i}$ at $t = 0$.

In this case, c_i is the second derivative of the polynomial $\tilde{P}_i$ at $t = 0$. Given the discrete nature of the problem, this involves using finite differences.
Then, as mentioned in [11], this problem can be written in the form of a quadratic problem

$$\begin{aligned} C^2 &= \sum_{i=1}^{N} c_i^2 \\ &= \sum_{i=1}^{N} \bar{\tau_i}^T H_{C,i} \bar{\tau_i} + B_{C,i} \bar{\tau_i} + cost \end{aligned} \tag{55}$$

where

$$\bar{\tau_i} = \begin{bmatrix} \tau_{i-1} & \tau_i & \tau_{i+1} \end{bmatrix}^T \tag{56}$$

$$\begin{aligned} H_{C,i} = &\begin{bmatrix} \Delta x_{i-1}^2 & -2\Delta x_{i-1}\Delta x_i & \Delta x_{i-1}\Delta x_{i+1} \\ -2\Delta x_{i-1}\Delta x_i & 4\Delta x_i^2 & -2\Delta x_i \Delta x_{i+1} \\ \Delta x_{i+1}\Delta x_{i-1} & -2\Delta x_{i+1}\Delta x_i & \Delta x_{i+1}^2 \end{bmatrix} \\ &+ \begin{bmatrix} \Delta y_{i-1}^2 & -2\Delta y_{i-1}\Delta y_i & \Delta y_{i-1}\Delta y_{i+1} \\ -2\Delta y_{i-1}\Delta y_i & 4\Delta y_i^2 & -2\Delta y_i \Delta y_{i+1} \\ \Delta y_{i+1}\Delta y_{i-1} & -2\Delta y_{i+1}\Delta y_i & \Delta y_{i+1}^2 \end{bmatrix} \end{aligned} \tag{57}$$

$$\begin{aligned} B_{C,i} =&2(\Delta x_{i,0} - \Delta x_{i-1,0}) \begin{bmatrix} \Delta x_{i-1} & -2\Delta x_i & \Delta x_{i+1} \end{bmatrix} \\ &+ 2(\Delta y_{i,0} - \Delta y_{i-1,0}) \begin{bmatrix} \Delta y_{i-1} & -2\Delta y_i & \Delta y_{i+1} \end{bmatrix} \end{aligned} \tag{58}$$

Finding the minimum curvature path is therefore a quadratic problem where $\bar{\tau}$ is the optimization parameter.

The minimum curvature path on the test track is given in Fig. 42.

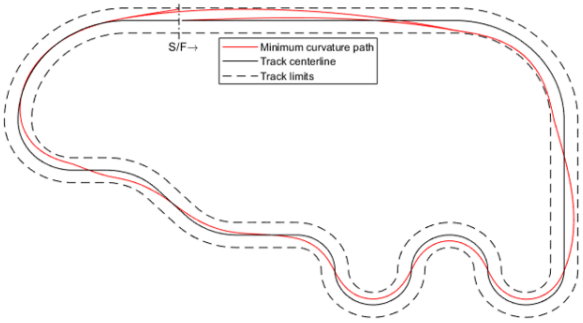

Fig. 42: Minimum curvature path on the test track.

Following the lap time measurement protocol described in section VII, we obtain the results presented in Fig. 43 and in table VIII.

TABLE VIII: Minimum curvature path tracking results.

Time	
Lap 1	29.394s
Lap 2	27.821s
Lap 3	27.811s
Lap 4	27.744s
Total	112.770s

Error	
Peak	0.36m
Mean	0.10m

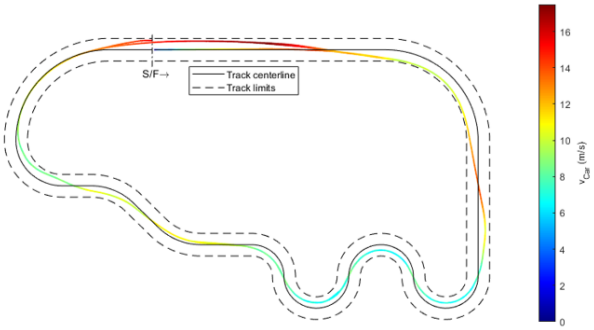

Fig. 43: Minimum curvature path tracking result.

E. Optimal trajectory

Depending on track geometry and the vehicle considered, the optimal trajectory may vary. It can be assumed that the optimal trajectory is a mixture between the shortest and the minimum curvature path. This new problem can be formulated based on the previous ones. Introduce the now optimisation function O defined by:

$$\begin{aligned} O^2 =&(1-\epsilon)C^2 + \epsilon S^2 \\ =&\sum_{i=1}^{N} \bar{\tau_i}^T ((1-\epsilon)H_{C,i} + \epsilon H_{S,i})\bar{\tau_i} \\ &+ \sum_{i=1}^{N} ((1-\epsilon)B_{C,i} + \epsilon B_{S,i})\bar{\tau_i} + cost \\ =&\bar{\tau}^T \underbrace{((1-\epsilon)H_C + \epsilon H_S)}_{H_O} \bar{\tau} + \underbrace{((1-\epsilon)B_C + \epsilon B_S)}_{B_O} \bar{\tau} + cost \end{aligned} \tag{59}$$

where ϵ is a weighting parameter between the two previous problems. This new problem satisfies the same constraints on $\bar{\tau}$ as the two previous ones.

The Fig. 44 presents the shortest path, the minimum curvature path as well as the intermediate path corresponding to $\epsilon = 0.005$.

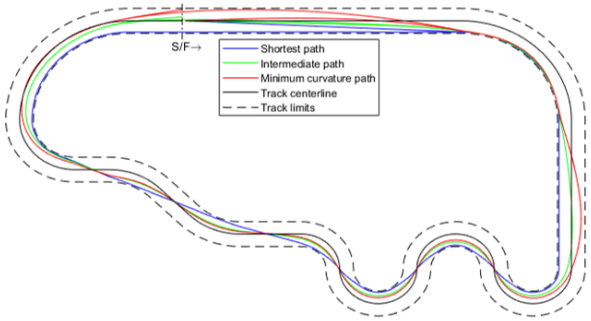

Fig. 44: Intermediate path on the test track.

Finding the optimal trajectory now consists in finding the value of ϵ minimizing the lap times of the vehicle. The lap time measurement protocol is applied for different trajectories corresponding to values of epsilon between 0 and 1. The Fig. 45 depicts the best lap for each sequence as well as the mean lap time of the rolling start laps.

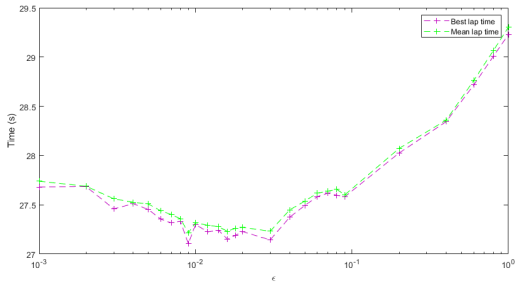

Fig. 45: Lap times as a function of ϵ.

As it can be seen, neither the shortest path nor the minimum curvature path gives the minimum lap time. In this case, the optimal value of ϵ is $\epsilon = 0.009$. The related tracking results presented in Fig. 46 and in table IX.

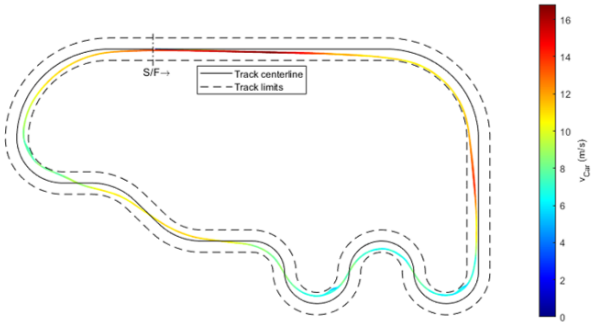

Fig. 46: Optimal path tracking result.

TABLE IX: Optimal path tracking results.

Time	
Lap 1	28.812s
Lap 2	27.345s
Lap 3	27.334s
Lap 4	27.478s
Total	110.969s

Error	
Peak	0.33m
Mean	0.09m

This trajectory optimization enabled a $19.151s$ time reduction on four laps and a best lap time improvement of $4.818s$ compared to the track centerline.

IX. Online Trajectory Optimization

Offline optimization is a first solution to define the target trajectory. However, this involves defining in advance the trajectory for the full test run, without any possible changes during the run. A solution to counter this issue is to consider online trajectory optimization.

A. Online problem formulation

The online optimization problem itself is very similar to the offline one. The optimisation function only differs from the offline one by the number of points N considered. For the offline trajectory optimizer, the number of points considered is the number of points defining the track. For the online case, the number of points considered is set to provide sufficient anticipation in the different corner sequences (such as turn T2 to T5 or T5 to T6).
As far as the optimization constraints are concerned, these remain the same as for the offline optimization problem, only the initial condition varies. Consider the scenario depicted in Fig. 47. The initial condition (represented by the blue point) is chosen to match as closely as possible the current vehicle position (but remaining on track even if the car is not). α_1 is then set to be the abscissa of the blue point.

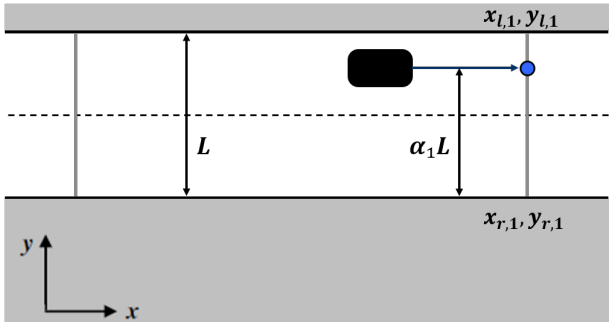

Fig. 47: Online initial condition definition.

B. Online trajectory optimization implementation

The online trajectory optimization is implemented following the flowchart described in Fig. 48.

Fig. 48: Online trajectory optimization flowchart.

Given that this trajectory optimization optimizes the trajectory on a distance way longer than the look-ahead distance (for the trajectory controller) or the anticipation distance (for the speed controller), the optimization protocol is not required to be run as fast as the vehicle controllers. The sampling time for the trajectory optimizer is therefore set to $T_{s,TO} = 0.1s$.

C. Local optimization results

The Fig. 49 and Fig. 50 respectively depict the online optimization through turn T1 and turn T2 to T5 with the optimization parameter $\epsilon = 0.009$.
As it can be seen, the optimization result at different time steps only differs at the end. This can be explained by the fact that

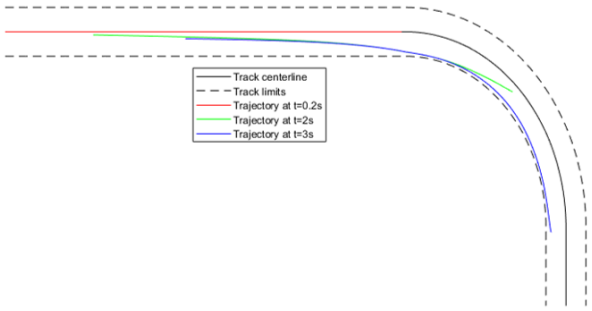

Fig. 49: Online trajectory optimization through turn T1.

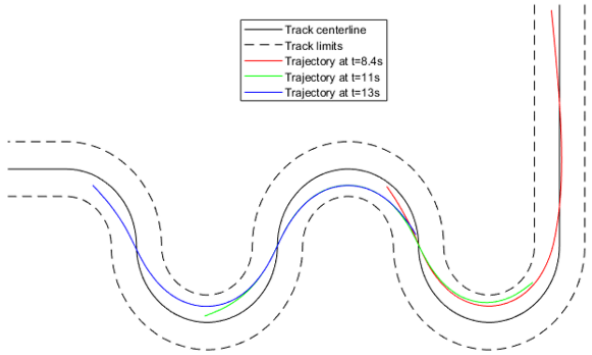

Fig. 50: Online trajectory optimization through turn T2 to T5.

the optimization constraints are not strong enough toward the end of the optimized segment. Indeed, the minimum curvature component of the optimization problem results in a straight line at the end of the optimized segment.

D. Lap time minimization

Given that the trajectory optimization protocol changed compared to the section VIII-E, the optimal value of ϵ might have changed. The Fig. 51 presents the lap times as a function of ϵ with the online trajectory optimization setup.

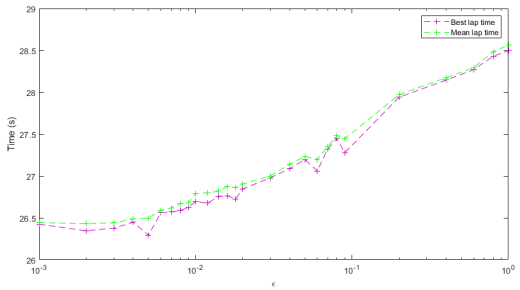

Fig. 51: Lap times as a function of ϵ.

It can be noticed that the lap times resulting from the online optimisation problem are better than the ones obtained with the offline problem. This can be explained by the fact the online problem provides a smoother response in the case of divergence from the target trajectory.

In this case, the optimal trajectory is obtained for $\epsilon = 0.005$. The related tracking is presented in Fig. 52 and in table X. With this setup, the best lap time is $26.291s$ which is an improvement of $1.043s$ compared with the best lap time resulting from the offline trajectory optimization.

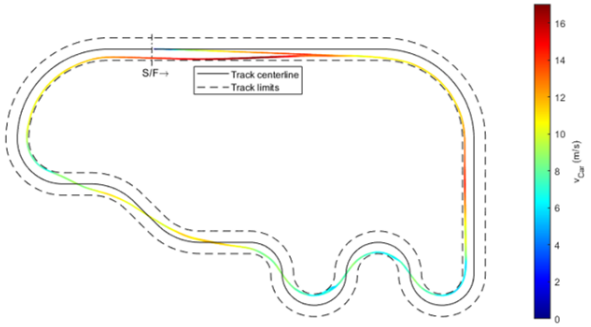

Fig. 52: Online optimal path tracking result.

TABLE X: Online optimal path tracking results.

Time	
Lap 1	27.975s
Lap 2	26.619s
Lap 3	26.291s
Lap 4	26.584s
Total	107.469s

Note that in this case, the tracking error is not indicated since the optimal trajectory is not predefined beforehand.

X. Discussion

A. Controllers performance

As far as the steering controller is concerned, the latter is able to keep the car on the desired trajectory with a mean lateral deviation of a dozen of centimetres. Even if there exists differents enhancement of the pure pursuit algorithm (such as the one presented in [12]), the resulting lap time improvement would be marginal.
Similarly, the speed controller enables to keep the car on the trajectory without inducing either understeer or oversteer. However, given the non-linearities of the model, the usefulness of the linear analysis was quite limited.

B. Vehicle class influence

The major difference between cars is their type of chassis. In this project, the chassis considered belongs to the touring class. In terms of its mechanical conception, this class is the closest to full-size car, with an equal mass repartition on the four wheels, independent suspensions and a GT like bodywork.
Other classes have their own specificities. For instance, the formula class is characterized by a rear-wheel drive propulsion, with a fixed rear axle and a formula 1 kind bodywork (with high aerodynamic downforce).

Touring cars are equipped with rubber tyres. Their behaviour is similar to the one of full-scale cars. On the contrary, pancar cars are equipped with foam tyres. These are characterized by their high wear resulting in diameter reduction and ride height evolution.

These differences, both in terms of chassis and tyres, influence both the models and the controllers. Chassis differences impact the vehicle dynamics and tyre differences impact the tyre model. These differences can be on the models themselves and/or their parameters. This results in controllers tuning differences to find the optimal parameters for each vehicle class.

C. Track influence

RC car tracks are really different one from the other both in terms of layout and surface. Outdoor tracks are usually fast on an asphalt surface. On the contrary, indoor tracks are tight, technical with a carpeted surface.

Similarly to chassis differences, track differences have an impact on the controllers tuning. For instance, the look-ahead and anticipation distances for the steering and speed controller respectively may change as well as the optimal trajectory parameter ϵ.

D. Simulation accuracy

In this project, models have been used to develop a simulation environment. The different assumptions and simplifications impact the model accuracy and also narrow down the validity domain of the model.
Moreover, as explained in sections II and III, some parameters can be directly measured but others need to be estimated. These estimations induce inaccuracy that impacts the car behaviour and performances.

This model can therefore not be used for performance prediction. The use of such a model is rather to perform parameters comparison and optimization.

XI. Conclusion

As far as the vehicle model is concerned, the main task consisted in adapting a full-scale vehicle model to match the size and physics of a remote-controlled car. The main aim of this model was to provide a simulation environment for the development of a racing driver model.
In this project, the driver skills considered are its ability to control the car (to keep it on track) while pushing it to its limits. This mainly consists in finding the optimal steering and throttle input to keep the car on the racing line. The second driving skill considered is the ability to optimized the racing line.
As mentioned in section V, the steering controller implemented is the pure pursuit algorithm. The latter is able to keep the car on the desired trajectory with a mean lateral deviation of dozen of centimetres. This accuracy is sufficient to keep the car close enough to the trajectory to stay on track and to avoid getting stuck on the gravel or grass.
The speed controller can be decomposed into two parts: a first one to generate the target speed and a second one to regulate the car speed to the target. The elaboration of the speed target is based on the friction circle model which gives a speed limit based on the tyre grip and the track curvature. A linearised vehicle dynamics model is then used in order to develop and tune a speed controller to regulate the car speed to the target.
Finally, finding the optimal trajectory consists in finding the optimal compromise between speed and distance. This optimization problem is solved by running comparative runs. In comparison with the track centerline, offline trajectory optimization enabled a time gain of $19.151s$ on four laps and a best lap improvement of $4.818s$. Then, the online optimization led to a further improvement of $3.500s$ on four laps and a best lap improvement of $1.043s$.
As described above, the project fulfilled all of its initial goals and enabled the design of several advanced control and optimization strategies for an RC car. The overall project can therefore be judged as successful. Moreover, as with any project on modelling and controller design, further improvements can always be made to either increase the model fidelity or refine the control system. Some of these improvements will be considered in the next section.

XII. Future Work

A. Parameters measurement experiments

As mentioned in sections II and III, some parameters can directly be measured on the vehicle. However, some others need to be estimated, with more or less accuracy depending on the parameter. An improvement would therefore be to perform additional measurements or experiments in order to improve the accuracy of the parameters. For instance, an experimental setup is described in [13] to measure the tyre parameters.

B. Real track implementation

In this project, simulations have been conducted on a prototype track. Such tracks are mainly used for development purposes. Indeed, these have the particularity to contain the different kinds of corners. This avoids the need of running the simulations on different tracks during the first tuning phase. The next step is then to implement real tracks to test and verify the algorithm robustness and fine-tune them.

C. Car setup optimization

As mentioned in section II, most of the parameters have been set to be as neutral as possible and considered fixed during the whole project to provide a fair comparison basis for controllers tuning and trajectory optimization. Another use of this model could now be to fix the controllers value in order to study the influence of the different car parameters. This would provide an initial optimized setup to be refined on track.

D. Obstacle avoidance and overtake

An important point in car racing is the ability to avoid obstacles and to overtake opponents. As far as static obstacle avoidance is concerned, it can be interpreted as an additional constraint to be applied to the trajectory optimization problem. On the other hand, overtaking requires a separate strategy. Overtaking is indeed a complex manoeuvre with various possible patterns. Choosing the right pattern is part of the racecraft of the driver.

E. Real world implementation

In order to test and validate the controllers' design in a virtual environment, these need to be implemented in a real remote control car. The main challenge when dealing with real-world implementation consists in the availability of feedback information. The most challenging information to measure accurately is the car position. The accuracy required is of the order of $0.1m$. However, the scale of the vehicle adds constraints in terms of sensor size and mass. The other required information for the car controllers is its speed. Contrary to the car position, its speed can more easily be estimated thanks to a sensor mounted on the spur gear.

References

[1] C. Rouelle, "Formula student 101," pp. 0–15, 2013. [Online]. Available: www.racecar-engineering.com

[2] "Gallery — XRAY T4'19," 2019. [Online]. Available: https://teamxray.com/t4/2019/en/gallery/

[3] T. Brüdigam, K. Ahmic, M. Leibold, and D. Wollherr, "Legible Model Predictive Control for Autonomous Driving on Highways," vol. 51, no. 20, pp. 215–221, 2018.

[4] W. Junmin, J. Steiber, and B. Surampudi, "Autonomous ground vehicle control system for High-Speed and safe operation," *International Journal of Vehicle Autonomous Systems*, vol. 7, no. 1-2, pp. 18–35, 2009.

[5] J. M. Snider, "Automatic Steering Methods for Autonomous Automobile Path Tracking," *Work*, no. February, pp. 1–78, 2009. [Online]. Available: http://www.ri.cmu.edu/pub_files/2009/2/Automatic_Steering_Methods_for_Autonomous_Automobile_Path_Tracking.pdf

[6] F. Braghin, F. Cheli, S. Melzi, and E. Sabbioni, "Race driver model," *Computers and Structures*, vol. 86, no. 13-14, pp. 1503–1516, 2008.

[7] J. Wang and R. Longoria, "Combined tire slip and slip angle tracking control for advanced vehicle dynamics control systems," 2006.

[8] M. Bian, L. Chen, Y. Luo, and K. Li, "A dynamic model for tire/road friction estimation under combined longitudinal/lateral slip situation," *SAE Technical Papers*, vol. 1, no. May 2015, 2014.

[9] N. H. Amer, H. Zamzuri, K. Hudha, and Z. A. Kadir, "Modelling and Control Strategies in Path Tracking Control for Autonomous Ground Vehicles: A Review of State of the Art and Challenges," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 86, no. 2, pp. 225–254, 2017. [Online]. Available: http://dx.doi.org/10.1007/s10846-016-0442-0

[10] R. Brach and M. Brach, "The tire-force ellipse (Friction Ellipse) and tire characteristics," *SAE 2011 World Congress and Exhibition*, 2011.

[11] "Dry technology: the shortest path and minimum curvature path algorithm in racing games." [Online]. Available: https://www.programmersought.com/article/66151083841/

[12] E. Cocconi, "Enhanced Pure Pursuit Algorithm And Autonomous Driving Edoardo Cocconi," no. May, 2020.

[13] P. Hoblet, R. T. O'Brien, and J. A. Piepmeier, "Scale-model vehicle analysis for the design of a steering controller," *Proceedings of the Annual Southeastern Symposium on System Theory*, vol. 2003-January, no. May 2014, pp. 201–205, 2003.

APPENDIX

A. Model parameters

TABLE XI: Sample times.

Symbol	Name	Value	Unit
$T_{s,VM}$	tVehicleModel	$0.001s$	s
$T_{s,PP}$	tPurePursuit	0.01	s
$T_{s,SC}$	tSpeedController	0.01	s
$T_{s,TO}$	tTtrajectoryOptimizer	0.1	s

TABLE XII: Physics parameters.

Symbol	Name	Value	Unit
g	gGravity	9.81	m/s^2
ρ_{Air}	DAir	1.2	kg/m^3

TABLE XIII: Car parameters.

Symbol	Name	Value	Unit
	lCarWidth	0.2	m
	hCarHeight	0.115	m
C_x	CCarDrag	0.3	m
$2l_s$	lTrackWidth	0.165	m
$l_f + l_r$	lWheelbase	0.26	m
	mWheel	0.03	kg
	rWheel	0.03	m
$I_{y,i}$	IWheel	$2.076e-5$	$kg.m^2$
R_i	rWheelEffective	0.1885	m
m_{Car}	mCar	1.32	kg
$mCar, S$	mCarSprung	1.198	kg
$I_{z,Car}$	IzCar	0.0104	$kg.m^2$
h_{CG}	hCenterGravity	0.02	m
h_{RC}	hRollCenterF	0.01	m
h_{RC}	hRollCenterR	0.01	m
κ_f	KRollF	0.5	N/m
κ_r	KRollR	0.5	N/m
G_1	rGearRatio1	3.325	$-$
G_2	rGearRatio2	2.9	$-$

TABLE XIV: Tyre parameters.

Symbol	Name	Value	Unit
μ	muTire	1.75	$-$
K_s	KTireLong	500	rad^{-1}
K_α	KTireLat	1000	rad^{-1}
C_{RR}	CRollResistance	0.01	$-$

TABLE XV: Motor parameters.

Symbol	Name	Value	Unit
P_{Max}	PMotorMax	760	W
ω_L	nLimit	1000	RPM
η_i	eInverter	0.8	$-$
η_d	eDrivetrain	0.8	$-$
G_1	rGearRatio1	3.325	$-$
G_2	rGearRatio2	2.9	$-$

TABLE XVI: Track parameters.

Symbol	Name	Value	Unit
	NTrack		$-$
	xTrack		
	yTrack		
$L/2$	lHalfTrackWidth	2.5	m
	xRightSide		
	yRightSide		
	xLeftSide		
	yLeftSide		
	xStartLine		
	yStartLine		
	lTrack		m
ϵ	rTrajOpt	0.005	$-$

B. Simulink model

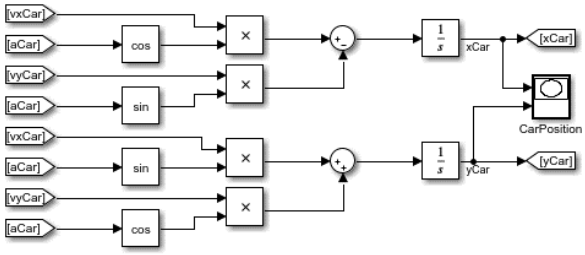

Fig. 53: Car position equations implementation.

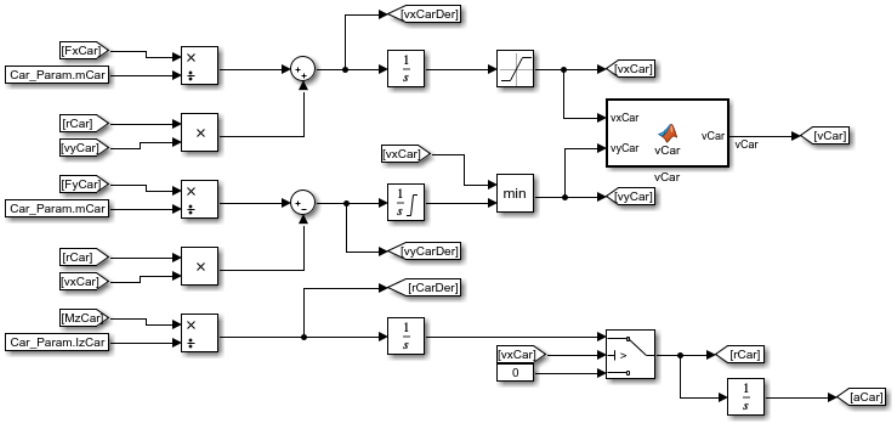

Fig. 54: Equations of motion implementation.

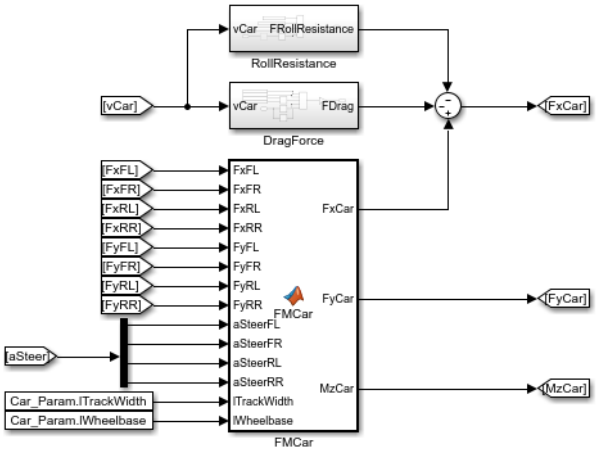

Fig. 55: Generalized forces computation.

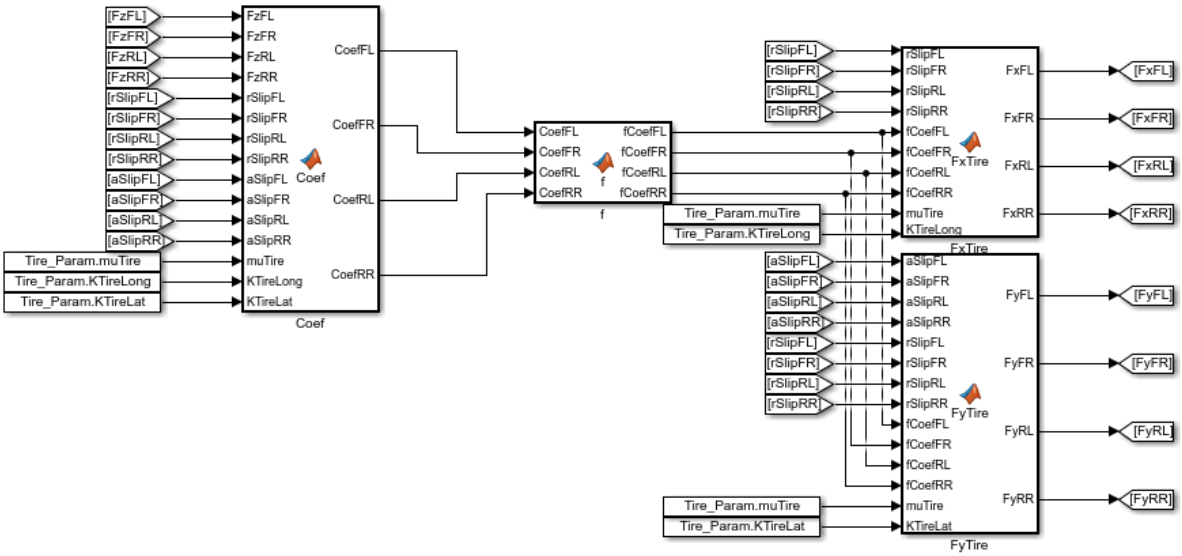

Fig. 56: Dugoff tyre model computation.

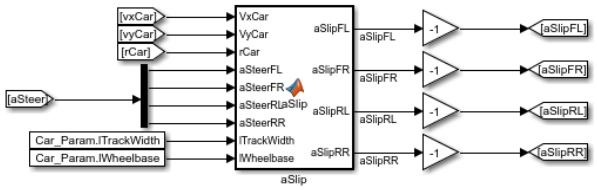

Fig. 57: Tyre slip angle computation.

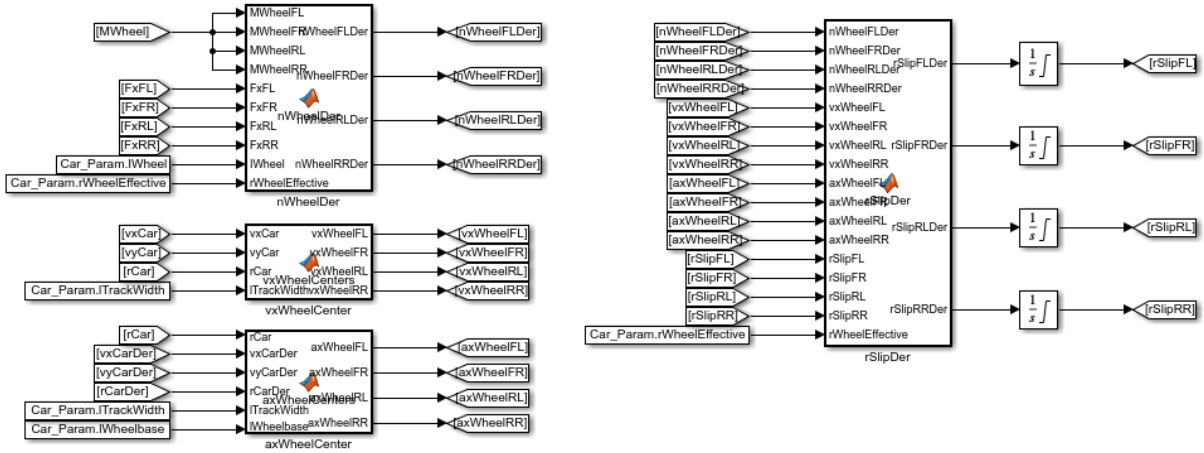

Fig. 58: Tyre slip ratio computation.

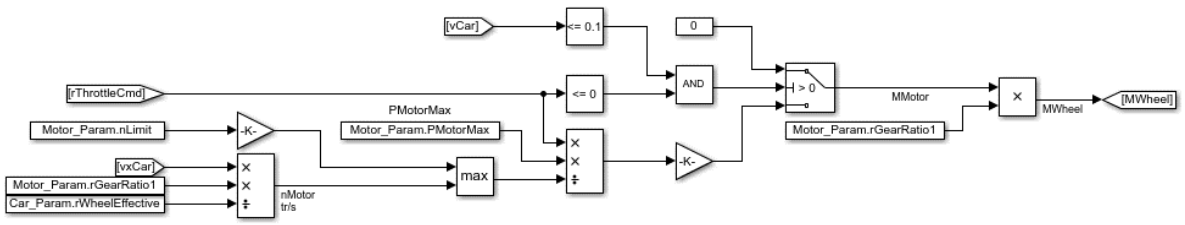

Fig. 59: Powertrain model implementation.

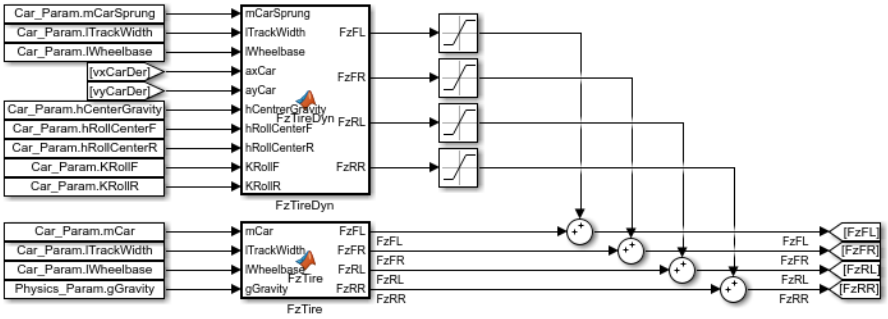

Fig. 60: Tyre normal load computation.

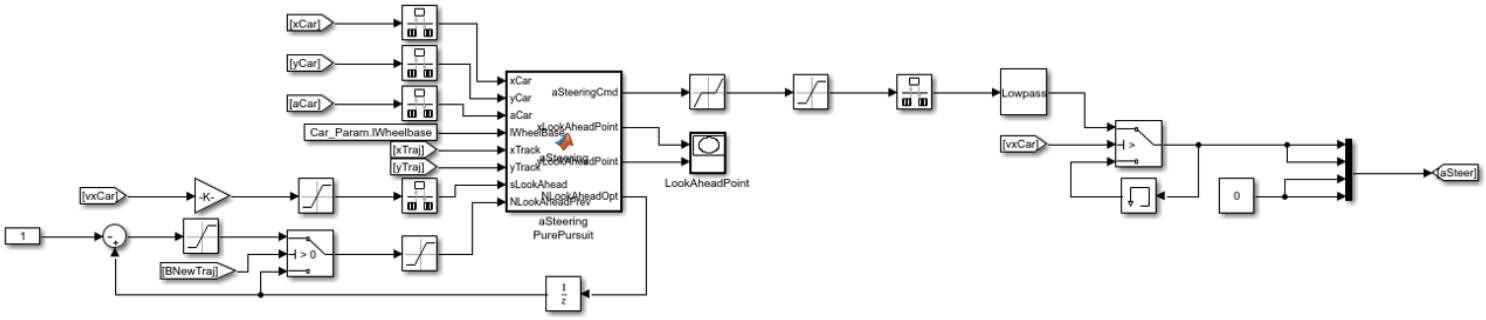

Fig. 61: Pure pursuit algorithm implementation.

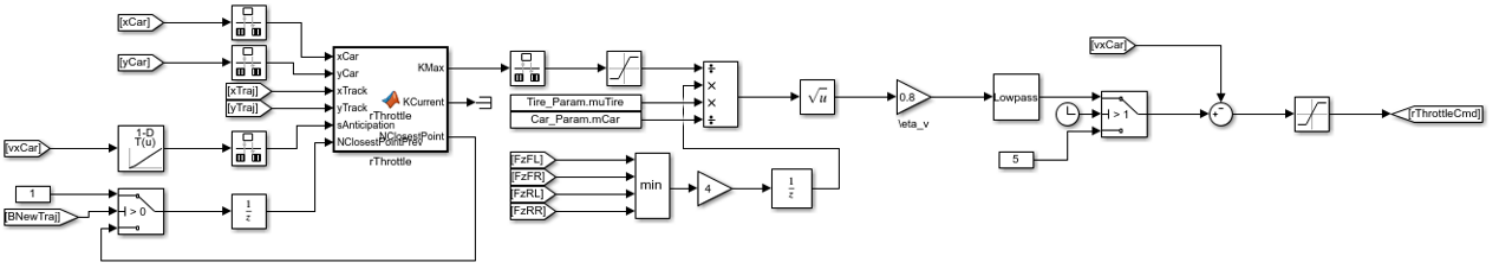

Fig. 62: Speed target and controller implementation.

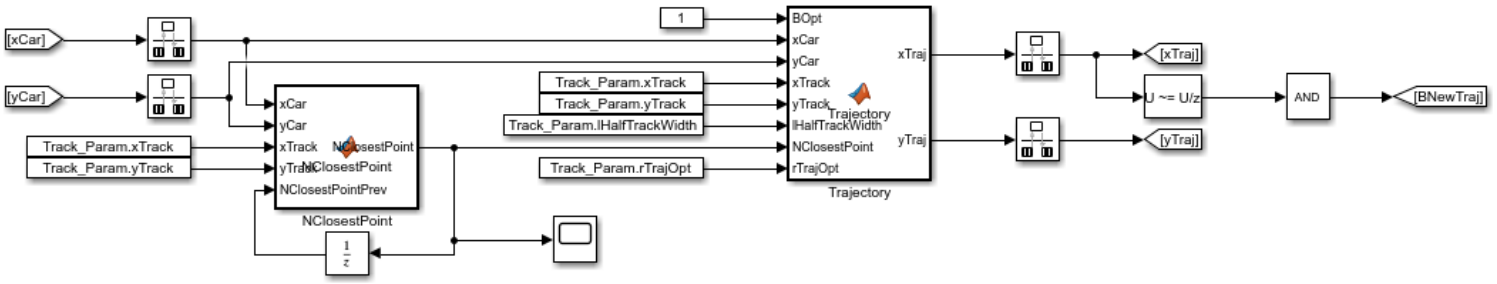

Fig. 63: Online trajectory optimization implementation.

C. Gantt diagram

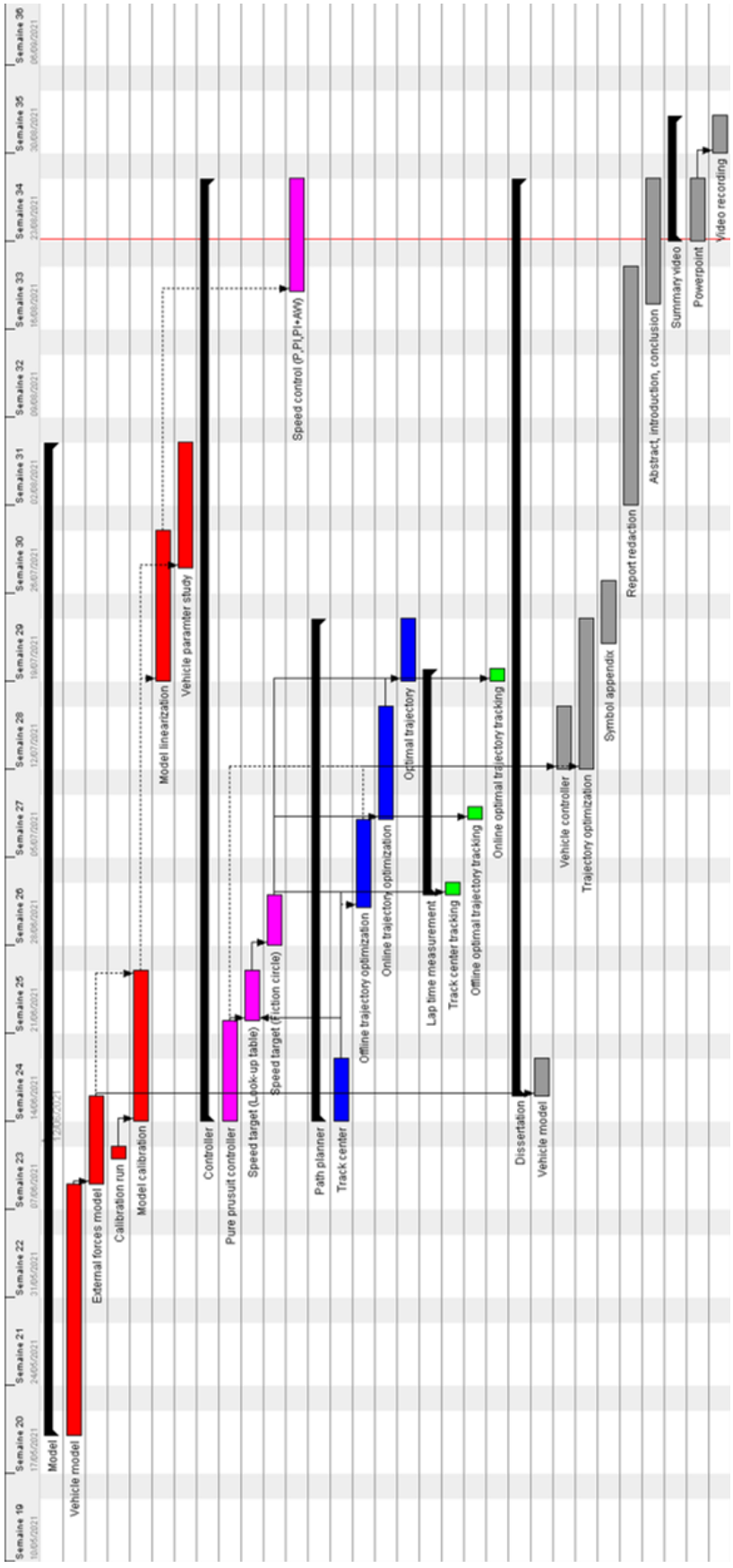

Fig. 64: Gantt diagram.