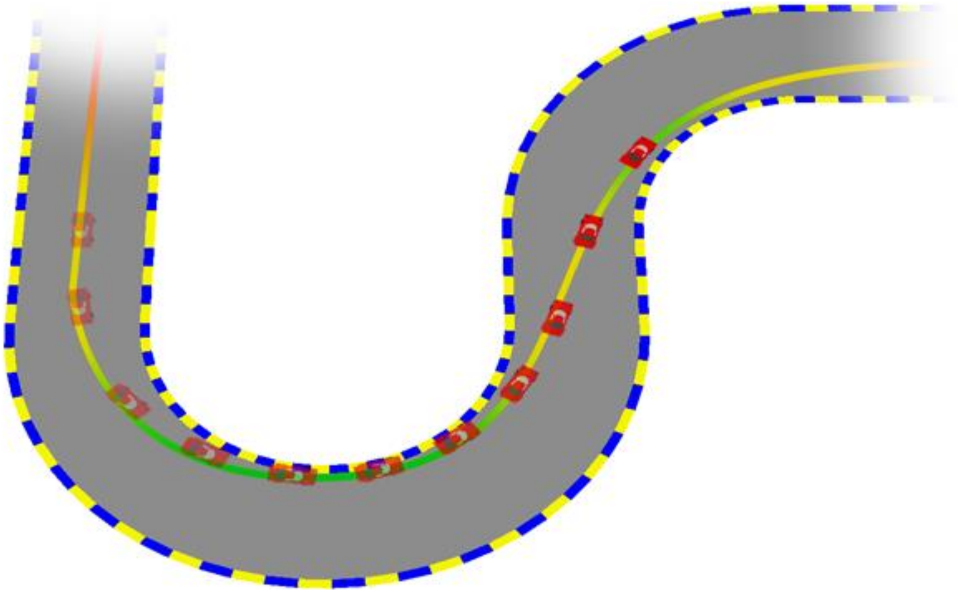


Optimal Lap Time for a Race Vehicle

Luís Henrique Vieira Brites Marques

Thesis to obtain the Master of Science Degree in

Mechanical Engineering

Supervisor: Prof. Jorge Alberto Cadete Ambrósio

Examination Committee

Chairperson: Prof. João Orlando Marques Gameiro Folgado

Supervisor: Prof. Jorge Alberto Cadete Ambrósio

Member of the Committee: Prof. Luís Alberto Gonçalves de Sousa

October 2016

Acknowledgements

Ao Professor Jorge Ambrósio, o meu orientador, quero agradecer por me ter sugerido um tema que me permitiu tirar enorme prazer da realização deste trabalho e por todo o apoio que ofereceu durante o seu desenvolvimento. A sua ajuda, orientação e entusiasmo foram inexcedíveis e contribuíram muito positivamente para a minha formação tanto a nível académico como pessoal.

Aos meus colegas do Projecto FST Novabase, em especial ao Diogo Silva, pela ajuda e sugestões oferecidas e ainda ao Pedro Antunes, ao Mário Viegas e ao Hugo Magalhães pelo suporte e partilha de informação durante este trabalho.

A todos os meus amigos, um agradecimento pelo apoio, motivação e paciência.

Finalmente, agradeço à minha família todo o apoio dado. Ao meu pai, mãe e irmã um grande Obrigado por todo o suporte ao longo da minha formação académica. O exemplo por eles dado foi a grande inspiração para que tenha cumprido com sucesso este ciclo, que termina com a entrega deste trabalho.

Resumo

O tempo de volta ótimo para um veículo de competição num circuito automóvel representa o mínimo tempo que determinado veículo tem a capacidade de realizar enquanto negoceia uma volta completa a determinada pista. Neste trabalho, um programa multicorpo de análise dinâmica é desenvolvido de forma a permitir modelar e simular o comportamento de um veículo numa pista através da implementação de todos os constrangimentos cinemáticos necessários, nos quais se inclui um constrangimento de direção, 'steering constraint', para um automóvel de 4 rodas com sistema de direção a atuar nas rodas da frente. As várias forças envolvidas num sistema automóvel são implementadas, nomeadamente as forças de tração e travagem e o contacto pneu-estrada.

A otimização da trajetória, numa dada pista e com geometria pré-definida, é efetuada usando a combinação da trajetória de menor caminho, i.e., trajetória mais curta possível dentro dos limites da pista, com a trajetória de menor curvatura. O perfil de velocidades é otimizado usando acelerações máximas e mínimas, características de cada veículo.

Um controlador é desenvolvido com o objetivo de assegurar que o veículo segue uma trajetória pré-definida e o perfil de velocidades ótimo. Este controlador usa uma distância de visualização, 'preview distance', a qual permite que o veículo encontre trajetória correta, mesmo quando está fora de pista. O funcionamento do controlador e o programa de análise dinâmica é demonstrado num cenário em que o comportamento de um Lancia Stratos numa pista real é analisado. A mesma metodologia de controlo é em seguida aplicada para um modelo de veículo especial que é analisado utilizando o programa de análise dinâmica DAP3D, disponível para dinâmica de multicorpo espacial.

Keywords: Veículo de competição, Otimização de tempo de volta, Dinâmica multicorpo, Dinâmica de veículo, Controlo ótimo

Abstract

The optimal lap time for a race vehicle in a race track represents the minimum possible time for a determined vehicle to negotiate a complete round about the race track. In this thesis, a 2D multibody dynamic analysis program is developed to allow modelling and simulating the vehicle and racetrack scenario by implementing all the necessary kinematic constraints, which includes a steering constraint for a 4-wheel vehicle with a front steering axle and the necessary force elements including traction and braking and the tyre-road contact.

A trajectory optimization, on a given track with a prescribed geometry, is performed using a mix of the shortest path and the least curvature criteria being the speed profile optimized by using maximum a minimum accelerations for the vehicle as it negotiates the curves.

A controller is developed in order to make the vehicle follow the predefined path and the optimal speed profile. This controller uses a preview distance, which allows for the vehicle to find its way even when it starts or goes off-track. The controller and the dynamic analysis program are demonstrated in a scenario in which the behavior of a Lancia Stratos in a real race track is analyzed. The same control methodology is then applied in a spatial vehicle model analyzed in the Dynamic Analysis Program (DAP3D) available for the spatial multibody dynamics.

Keywords: Race Vehicle, Lap Time Optimization, Multibody Dynamics, Vehicle Dynamics, Optimal Control

Contents

List of figures

List of tables

List of symbols

Convention

a, A, α	Scalar;
$\mathbf{a}$	Vector;
$\mathbf{A}$	Matrix;

Overscript

$\dot{\mathbf{a}}$	First time derivative;
$\ddot{\mathbf{a}}$	Second time derivative;

Superscript

$\mathbf{a}^{\mathrm{T}}$	Matrix or vector transpose;
$\mathbf{a}'$	Vector expressed in the body-fixed reference frame;
a^{i}	Referred to body *i;*
${}^{sp}a$	Quantity interpolated by a cubic spline*;*
a^{P}	Quantity associated with point P*;*

Subscript

$\mathbf{a}_{i}$	Referred to body *i*;
$\mathbf{a}_{j}$	Referred to body *j*;
a_{t}	Referred to the tangent component;
a_{n}	Referred to the normal component;
a_{b}	Referred to the binormal component;
a_{c}	Quantity associated with the vehicle;
a_{ct}	Quantity associated with the vehicle, relative to the track;
a_{f}	Quantity associated with the front of the vehicle;
a_{lc}	Quantity associated with the least curvature track;
a_{pt}	Quantity associated with the preview trajectory;
a_{r}	Quantity associated with the rear of the vehicle;
a_{st}	Quantity associated with the shortest track;
a_{t}	Quantity associated with the track reference;
a_{w}	Quantity associated with the wheel;

Latin Symbols

$\mathbf{A}$	Generic transformation matrix;
$\mathbf{a}$	Acceleration vector;
a	Distance from vehicle center of mass to front axle;
$\mathbf{B}$	Control input matrix; Matrix used in the trajectory optimization;
b	Distance from vehicle center of mass to rear axle;
$\mathbf{C}$	Control system matrix;
C_s	Longitudinal slip coefficient;
C_α	Cornering stiffness;
C_γ	Camber stiffness;
$\mathbf{d}$	Control disturbance vector; Distance vector;
d	Half vehicle track;
$\mathbf{E}$	Identity matrix;
e_1, e_2, e_3, e_4	Euler parameters;
$\mathbf{f}$	Vector of generic forces;
F	Tyre force;
$\mathbf{g}$	Vector of generalized forces;
$\mathbf{H}$	Matrix used in the trajectory optimization;
$\mathbf{J}$	Inertia tensor;
J'	Inertia;
J	Moment of inertia; Performance index;
$\mathbf{K}_{FB}$	Feedback gain;
$\mathbf{k}_t$	Vector with the curvatures of track segments;
k	Curvature;
$\mathbf{l}_t$	Vector with the lengths of track segments;
l	Generic length;
l_{tw}	Track width;
$\mathbf{M}$	Global mass matrix;
M	Torque or moment;
m	Body mass;
$\mathbf{N}$	Diagonal mass matrix; Control input matrix;
n_b	Total number of bodies in the system;
n_p	Total number of trajectory points;
n_{ps}	Total number of preview trajectory points;

n_s	Total number of track segments;
n_{sp}	Total number of preview track segments;
$\mathbf{p}$	Vector with the Euler parameters;
P	Generic point on body *i;*
$\mathbf{Q}$	Weighting matrix;
$\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}$	Vector of generalized coordinates, velocities and accelerations;
$\mathbf{R}$	Weighting matrix;
$\mathbf{r}$	Cartesian coordinates of generic point;
R	Curvature radius;
R_g	Tyre geometric radius;
R_w	Tyre effective radius;
s	Longitudinal slip ratio; Curvature length;
$\mathbf{T}_t$	Track centerline data matrix;
t	Time;
$\mathbf{u}$	Unit vector; Control vector;
U	Longitudinal vehicle velocity;
$\mathbf{v}$	Velocity vector;
V	Lateral vehicle velocity;
V_x	Wheel forward velocity;
V_y	Wheel lateral velocity;
v	Velocity;
x,y,z	Global coordinates;
$\mathbf{y},\dot{\mathbf{y}}$	Auxiliary vectors used in the integration process;
$\mathbf{z}$	State vector;

Greek Symbols

α	Baumgarte penalty factor; Tyre slip angle;
β	Baumgarte penalty factor;
δ	Steering angle;
$\varepsilon,\dot{\varepsilon},\ddot{\varepsilon}$	Vehicle yaw angle, yaw rate and angular acceleration;
ϕ	Body angular position;
$\varphi_1,\varphi_2,\varphi_3$	Bryant's angles;
φ	Vehicle roll angle;
$\boldsymbol{\gamma}$	Right-hand-side vector of acceleration equation;
γ	Tyre camber angle;

$\boldsymbol{\lambda}$	Vector of Lagrange multipliers;
$\boldsymbol{\nu}$	Right-hand-side vector of velocity equation;
θ	Vehicle pitch angle;
σ	Relative distance between trajectory and track inner limit;
τ	Combination factor;
ω	Angular velocity;
ξ,η,ζ	Body fixed reference frame;
$\boldsymbol{\Phi}$	Vector of kinematic position constraints;
$\boldsymbol{\Phi}_{\mathbf{q}}$	Jacobian matrix of kinematic constraints.

1 Introduction

1.1. Motivation

The optimal time lap problem has great interest, not only in racing but also in the design of different transportation activities. Knowing what a vehicle is capable of, at its maximum level of performance, allows companies to understand the behavior of the vehicle negotiating any racetrack, even before it is built. Having reliable simulations permits saving resources by simulate testing scenarios, before doing real tests and thus being a tool for test design, vehicle design and road design.

This problem is challenging of great complexity that brings together multibody dynamic analysis and optimal control. In this thesis, an approach for the optimal time lap problem using 2D and 3D multibody approaches for modelling and analysis of automotive vehicles is presented.

The first step in this work consists in developing a 2D multibody dynamics analysis program able to analyze simple models of automobiles, by allowing to use a limited number of bodies with revolute joints between the wheels and chassis. A steering constraint for the front wheels of the vehicle is developed and a tyre model is implemented to compute the lateral and longitudinal forces acting on each wheel of the vehicle.

In the second step, a controller is developed using a Linear Quadratic Regulator based on the work of Antos and Ambrósio [1], which is supported in a simplified vehicle to follow a pre-determined trajectory the vehicle is acted upon exclusively by the steering angle and by the wheels torques. A preview distance for the control is implemented in this work in order not only to stabilize its action even when the car is off-track but also to better represent the realistic driver attitude. Furthermore, the controller is upgraded with the ability to follow a determined speed profile by acting on the traction and braking of the wheels.

Apart from the controlling methodology, an optimization of the trajectory and speed profile is carried based on Meier’s work [2]. A mixed shortest path and least curvature approach is used to find the optimal path while the speed profile is computed by using upper and lower bounds of the lateral and longitudinal accelerations that can be developed by the tyres due to the tyre contact forces.

1.2. Literature Review

1.2.1. Multibody Dynamics

A multibody system is defined as a set of bodies or links, rigid or not, which can have constrained relative motion between each being acted upon by internal and external forces [3]. The bodies are connected by kinematic joints which constrain the relative motion between bodies. Examples of kinematic joints are pins, sliders, gears and cam followers. Force elements may be present in any mechanical system and they can be active or passive elements. In this work both active, such as engines, and passive elements, such as springs and dampers, are used and have great importance in the representation of the behavior of the system which is being modelled. Here only rigid bodies are considered, which means that any body deformations are neglected or considered small such that they have no impact on the body motion and in the dynamics analysis [4].

In a dynamic analysis, the main objective is to predict the motion of the system knowing its initial conditions, i.e. positions, orientations, velocities and external forces, and the applied loads over time. The multibody equations of motion are made into a set of Ordinary Differential Equations and their solution is obtained by integrating these equations in time. By using multibody models meaningful results can be obtained provided that proper numerical methods are used. Nevertheless there is an important limitation of using this kind of analysis, when the models are complex with complex interactions between bodies and environment, which is the computer power needed to run simulations.

1.2.2. Vehicle Dynamics

Vehicle dynamics is an important subject in the engineering world as the vehicles are the foundation of human mobility and there is a demand for their efficient and safe operation at reasonable cost. Knowing how a vehicle, particularly an automobile, behaves on a road is of great importance in order to improve either the passengers' comfort or the vehicle's performance, or both, with respect to the type of vehicle is being studied and for the purpose for which it was built. Many works have been published in this area, some of them specifically addressing auto racing purposes [5] [6]. In most cases of racing applications the objective is to promote maximum tyre-ground contact. Therefore, tyres and suspensions play a major role in the vehicle dynamics when safety, maneuverability and stability are the concern.

Different types of suspensions are proposed in past years, being less likely that completely new suspension configurations appear in the future. The double wishbone and McPherson suspension configurations dominate the market [7]. The vehicle model used in this work has front double wishbone and rear McPherson suspension systems. It is important to refer that there is not such thing as a perfect suspension being this always a compromise between parameters that may have greater or lower importance for the designer, such as comfort, maneuverability, cost, compactness, etc.

It is no surprise that tyres may be considered among the most important aspects of a car with regard to vehicle dynamics. Tyres provide the contact between the road and vehicle and produce the necessary contact forces that allow controlling the vehicle. The tyre mechanics is a very complex subject being the focus of many studies that have been carried in order to understand their behavior. Many models for the tyre behavior have been proposed, but Pacejka's Magic Formula [8] is accepted as one of the most accurate and simple to use. Other models have been proposed, as that by Gim and Nikravesh [9] [10] [11]. In order to use this kind of tyre models, experimental data is necessary to estimate the formula coefficients.

Steering dynamics is also a subject which cannot be discarded. Vehicles may be classified into two different categories: guided vehicles and piloted vehicles [7]. In this work piloted vehicles are of major importance and this kind of vehicles is known to follow its path by using a guidance system controlled by a human driver or by an electric surrogate.

1.2.3. Optimal Lap Time

A large amount of research have been developed regarding the optimal lap time problem as it is a problem that not only allows designers to understand the vehicle behavior at its maximum performance potential but also if the results are trustable, it allows saving resources by using simulation results. As referred by [12] many computational strategies are being developed in order to solve the optimal lap time problem which can be divided into quasi-steady state methods, as for example in the work by Brayshaw and Harrison [13], and transient optimal methods. The former ones provide good results at a reasonable computer cost, while the latter gives better results but with much higher computational effort. For the transient optimal methods there are also different approaches regarding the objective function to minimize during the optimization process. Maximizing the distance traveled for a defined time interval [14] and minimizing lap time [15] [16] are two different methods for solving the same problem that can be applied.

Some work focus the development of virtual car drivers with the model predictive control assuming an important role. As shown by Tommyppillai et al. [17] it became obvious that driving a vehicle effectively is only possible by using information on positions ahead of the vehicle actual location, because that is how a human driver also drives a car. For that reason, model predictive control using previewing seems to be the most effective method for developing a virtual driver [17].

Differently from any of the aforementioned approaches, simpler methods are also developed that can provide reasonable results with much less computational power. The method proposed by Meier [2] is based on a compromise between the shortest possible path within the track boundaries and the least curvature track. In this approach the optimal trajectory is not found, but it provides a good approximation of the best trajectory using a much easier method. Meier also develops a speed profile optimization procedure for any given track that, based on the vehicle limit to sustain lateral forces, leads to the smallest lap time.

1.3. Thesis Outline

The objectives of this work are the development of a virtual driver for a sports car, described by a multibody system using a simplified linearized controller which allows the vehicle to follow a predefined trajectory and also a predefined speed profile along a race track. In the process another objective consists in finding the optimal trajectory and speed profile that allows the vehicle to perform the best lap time, deemed here as optimal lap time.

In chapter 2 an overview of the multibody dynamics methodology used is presented with the description of the kinematic constraints and the equations of motion which support the 3D and 2D dynamic analysis programs, used and developed here, respectively. Vehicle dynamics fundamentals are presented in chapter 3 with some insight in the automotive vehicle and a detailed description of the tyre mechanics and tyre models used as modules in both dynamic analysis programs. Also in chapter 3 the Lancia Stratos model is described with detail and a demonstration case is developed. Chapter 4 presents the control strategy pursued in this work. Control strategy and algorithms are presented, including the development of the preview distance used in this controller. Some demonstrative cases are developed and explored. The minimal lap time problem is presented and developed in chapter 5, where the different approaches for optimal trajectory are explained. Furthermore, the speed profile optimization and some applications are developed to demonstrate how the best lap time can be identified. Chapter 6 addresses the 3D analysis, where the spatial vehicle model is described and the differences between planar and spatial analysis are referred. Chapter 7 presents the conclusions and recommendations for future developments.

1.4. Novel aspect of the work

The work presented here uses programs and methods previously developed. The controller which ensures that the vehicle follows a prescribed trajectory from Antos and Ambrósio [1] is used as well as the methodology used by Meier [2] for the optimal trajectory and speed profile. Finally, the DAP3D program developed by Ambrósio and Pombo is also used for the spatial dynamic analysis.

The novel aspect of the work includes a 2D multibody dynamic analysis program fully developed here, using the Pacejka Tyre Model [8]. Based on the work by Antos [1] a new version of the controller is implemented with preview distance and speed following features.

2 Multibody Dynamics Overview

In this chapter an insight of multibody dynamics is given to introduce the methodology used throughout the work. A detailed description of the representation of rigid bodies is presented followed by the explanation of kinematic constraints that are necessary for developing the models to be used and also the dynamic analysis programs. Apart from the most common kinematic constraints, a steering constraint is also presented here. Equations of motion and their integration are also explained together with the solution algorithm.

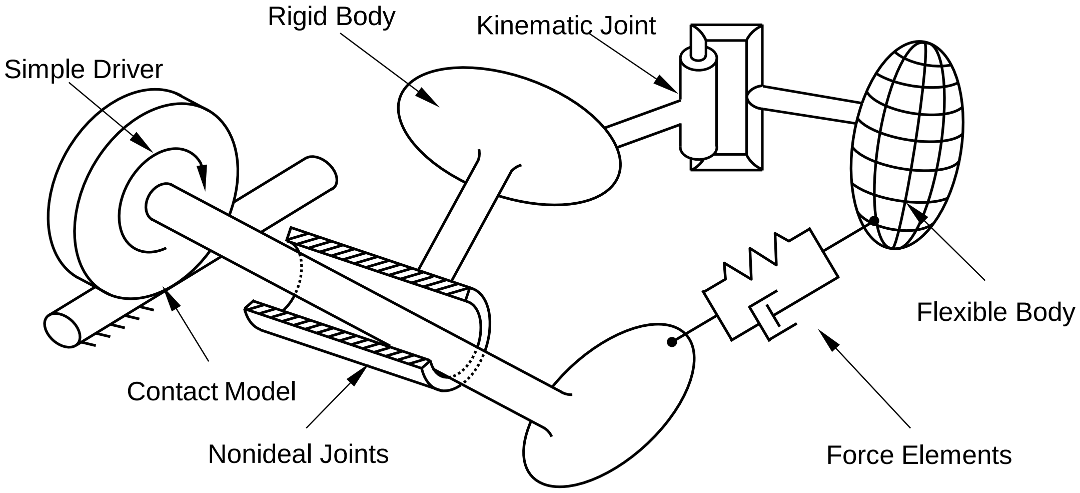


Figure 2.1 - Generic multibody system

A multibody system, as that shown in Figure 2.1, is composed by several types of bodies, rigid and flexible, connected by mechanical joints, or kinematic joints, and acted upon by force elements, passive and active. For the simulation of specific scenarios other, non-standard, kinematic joints may have to be developed. The interaction forces, as for instance between tyres and road constitute, often, the most demanding task when developing multibody dynamics analysis tools.

2.1. Cartesian Coordinates

The position and orientation of a rigid body is defined by that of a body fixed frame $(\xi,\eta,\zeta)_i$. When using Cartesian coordinates a rigid body *i* position is defined by a set of translational coordinates, $\mathbf{r}_i = \{x, y, z\}_i^T$, while its orientation is defined by a set of Euler parameters, $\mathbf{p}_i = \{e_0, e_1, e_2, e_3\}_i^T$ which represent the body orientations, with respect to a global reference frame (x, y, z) [18].

The coordinates for a rigid body *i* can be grouped into one vector $\mathbf{q}_i$, depicted in Figure 2.2, deemed as the rigid body in Cartesian coordinates,

$$\mathbf{q}_i = \left\{\mathbf{r}_i^T, \mathbf{p}_i^T\right\}^T = \{x, y, z, e_0, e_1, e_2, e_3\}_i^T \tag{2.1}$$

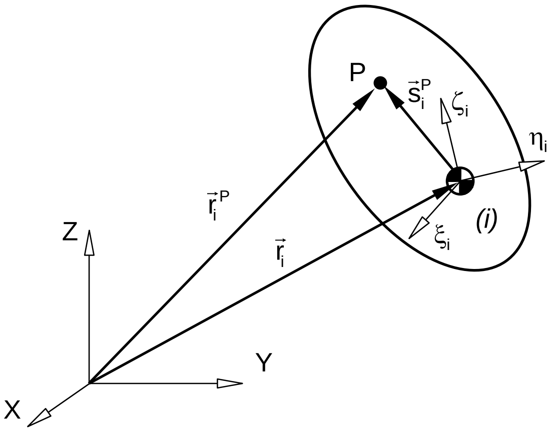


Figure 2.2 - Representation of a rigid body position with body fixed coordinate frame attached to the centre of mass

The position of a point *P* in body *i* with respect to the global reference frame $\mathbf{r}_i^P$ is defined by:

$$\mathbf{r}_i^P = \mathbf{r}_i + \mathbf{s}_i^P = \mathbf{r}_i + \mathbf{A}_i \mathbf{s}_i'^P \tag{2.2}$$

where $\mathbf{s}_i^P$ is the vector position for point *P*, with respect to the body fixed frame, and $\mathbf{A}_i$ is the transformation matrix from local to global coordinates, given by [18]:

$$\mathbf{A} = \mathbf{2}\begin{bmatrix} e_0^2 + e_1^2 - \frac{1}{2} & e_1e_2 - e_0e_3 & e_1e_3 + e_0e_2 \\ e_1e_2 + e_0e_3 & e_0^2 + e_2^2 - \frac{1}{2} & e_2e_3 - e_0e_1 \\ e_1e_3 - e_0e_2 & e_2e_3 + e_0e_1 & e_0^2 + e_3^2 - \frac{1}{2} \end{bmatrix} \tag{2.3}$$

With the basic Cartesian coordinates for all bodies involved in a model it is possible to define the kinematic constraints, all external and internal forces, and, afterwards, the dynamic equations of motion. When the system has no kinematic constraints the total number of coordinates equals the number of degrees of freedom.

2.2.Kinematic Constraints

A kinematic joint constrains the relative motion between two bodies. Some of the kinematic constraints used in both dynamic analysis programs – 2D and 3D – are here explained. They are all holonomic algebraic constraints and also lower-pair category, which means that not only they are developed in the position space but also no information about the shape of the bodies is needed [18]. A general constraint equation is denoted by [3]:

$$^{(a,b)}\mathbf{\Phi} = \mathbf{0} \tag{2.4}$$

where *a* states the constraint type and *b* is the number of equations that define the kinematic joint. By computing the first time derivative of equation (2.4) the velocities equation is obtained:

$$^{(a,b)}\dot{\mathbf{\Phi}} = \mathbf{0} \equiv \mathbf{\Phi}_q \dot{\mathbf{q}} = \mathbf{v} \tag{2.5}$$

where $\mathbf{\Phi}_q$ is the Jacobian matrix, $\dot{\mathbf{q}}$ is the velocity vector and $\mathbf{v}$ is a vector which contains the partial derivatives of constraints with respect to time. By computing the second time derivative of equation (2.4) the acceleration equation is obtained as:

$$^{(r,2)}\mathbf{\Phi} = \mathbf{0} \equiv \mathbf{\Phi}_q \ddot{\mathbf{q}} = \boldsymbol{\gamma} \tag{2.6}$$

where $\boldsymbol{\gamma}$ is a vector that contains all acceleration independent terms of the equations being written as:

$$\boldsymbol{\gamma} = -\frac{\partial^2 \mathbf{\Phi}}{\partial t^2} - \frac{\partial}{\partial t}(\mathbf{\Phi}_q)\dot{\mathbf{q}} \tag{2.7}$$

In what follows, 2D kinematic constraints are overviewed here first, being the 3D kinematic constraints overviewed afterwards. For the 2D dynamics analysis program only revolute joints and steering constraints are used as the vehicle models that are the focus of this work only require this type of kinematic constraints.

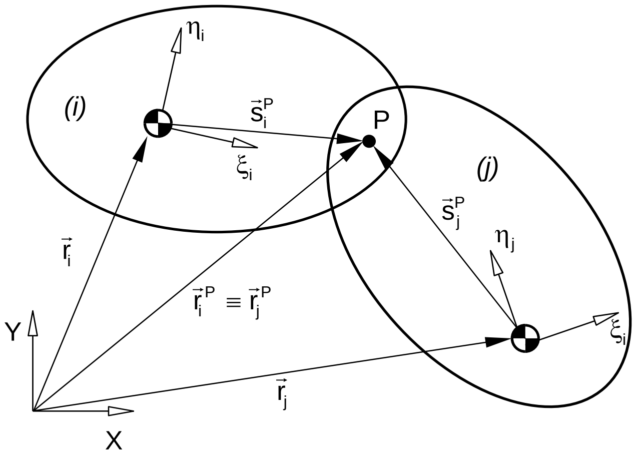


Figure 2.3 - 2D revolute joint representation

In a revolute joint, shown in Figure 2.3, the objective is to attach two points in different bodies P_i and P_j allowing the two bodies to rotate freely around this point. The translational coordinates of the two points of the two different bodies should always be coincident, which means $\mathbf{r}_i^P = \mathbf{r}_j^P$ [3]. Hence, the kinematic constraint equation is given by:

$$^{(r,2)}\mathbf{\Phi} = \mathbf{r}_j^P - \mathbf{r}_i^P = \mathbf{0} \tag{2.8}$$

Applying equation (2.2) and using the 2D transformation matrix, **A**$_i$, equation (2.8) is now expressed as:

$$^{(r,2)}\mathbf{\Phi} = \mathbf{r}_j + \mathbf{A}_j \mathbf{s}_j^{'P} - \mathbf{r}_i - \mathbf{A}_i \mathbf{s}_i^{'P} \tag{2.9}$$

or

$$\begin{Bmatrix} x_j \\ y_j \end{Bmatrix} + \begin{bmatrix} \cos\phi_j & -\sin\phi_j \\ \sin\phi_j & \cos\phi_j \end{bmatrix} \begin{Bmatrix} \xi_j^P \\ \eta_j^P \end{Bmatrix} - \begin{Bmatrix} x_i \\ y_i \end{Bmatrix} + \begin{bmatrix} \cos\phi_i & -\sin\phi_i \\ \sin\phi_i & \cos\phi_i \end{bmatrix} \begin{Bmatrix} \xi_i^P \\ \eta_i^P \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{2.10}$$

where ϕ_i and ϕ_j are the body-fixed reference frame orientation for bodies *i* and *j*, respectively. Equation (2.9) is now written in its expanded form as:

$$^{(r,2)}\mathbf{\Phi} = \begin{Bmatrix} x_j + \xi_j^P \cos\phi_j - \eta_j^P \sin\phi_j - x_i - \xi_i^P \cos\phi_i + \eta_i^P \sin\phi_i \\ y_j + \xi_j^P \sin\phi_j + \eta_j^P \cos\phi_j - y_i - \xi_i^P \sin\phi_i - \eta_i^P \cos\phi_i \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{2.11}$$

The velocity and acceleration constraints are also needed for the dynamic analysis. For a revolute joint the velocity constraint is expressed by its time derivative [3]:

$$^{(r,2)}\dot{\mathbf{\Phi}} = \dot{\mathbf{r}}_j^P - \dot{\mathbf{r}}_i^P = 0 \tag{2.12}$$

Which in its expanded form equation (2.12) becoming:

$$^{(r,2)}\dot{\mathbf{\Phi}} = \left[\begin{array}{ccc|ccc} -1 & 0 & (\xi_i^P \sin\phi_i + \eta_i^P \cos\phi_i) & 1 & 0 & -(\xi_j^P \sin\phi_j + \eta_j^P \cos\phi_j) \\ 0 & -1 & -(\xi_i^P \cos\phi_i - \eta_i^P \sin\phi_i) & 0 & 1 & (\xi_i^P \cos\phi_j - \eta_j^P \sin\phi_j) \end{array}\right] \begin{Bmatrix} \dot{x}_i \\ \dot{y}_i \\ \phi_i \\ - \\ \dot{x}_j \\ \dot{y}_j \\ \phi_j \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{2.13}$$

The $\boldsymbol{\gamma}$ vector is for the revolute joint given by:

$$^{(r,2)}\boldsymbol{\gamma} = \begin{Bmatrix} (\cos\phi_i \xi_i^P - \sin\phi_i \eta_i^P)\dot{\phi}_i^2 + (-\cos\phi_j \xi_j^P + \sin\phi_j \eta_j^P)\dot{\phi}_j^2 \\ (\sin\phi_i \xi_i^P + \cos\phi_i \eta_i^P)\dot{\phi}_i^2 + (-\sin\phi_j \xi_j^P - \cos\phi_j \eta_j^P)\dot{\phi}_j^2 \end{Bmatrix} \tag{2.14}$$

The steering constraint is aimed to control the steering angle, in this case between each front wheel and the chassis regarding which steering angle input is given. Note that both wheels, in Figure 2.4, are being steered equally, which is not what is required in vehicle applications in which the Ackerman geometry needs to be taken into account [7].

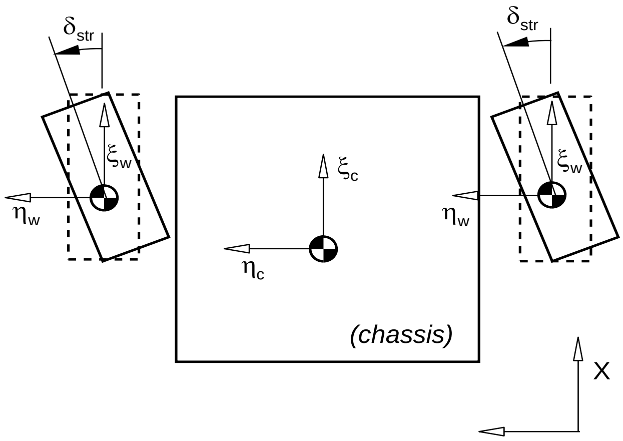


Figure 2.4 - Simplified steering constraint representation

Figure 2.4 shows the representation of the steering constraint, which is now described:

$$^{(str,1)}\mathbf{\Phi} = (\phi_w - \phi_c) - \delta_{str} = 0 \tag{2.15}$$

where ϕ_{wi} is the angular position of the wheel i, ϕ_c is the angular position of the chassis and δ_{str} is the steering angle. The velocity constraint equations are given by:

$$^{(str,1)}\dot{\mathbf{\Phi}} = \begin{bmatrix} 0 & 0 & 1 & | & 0 & 0 & -1 \end{bmatrix} \begin{Bmatrix} \dot{x}_w \\ \dot{y}_w \\ \dot{\phi}_w \\ - \\ \dot{x}_c \\ \dot{y}_c \\ \dot{\phi}_c \end{Bmatrix} = \dot{\delta}_{str} \tag{2.16}$$

The $\boldsymbol{\gamma}$ vector for the steering constraint is given by:

$$^{(str,1)}\boldsymbol{\gamma} = \ddot{\delta}_{str} \tag{2.17}$$

While the kinematic joints required for planar vehicle modelling and used in the 2D program are just the revolute and the steering constraints, more types of kinematic joints are required for spatial vehicle modelling and, therefore, are used in the 3D dynamic analysis program. In particular, it is necessary to describe spherical and translational joints. The spherical joint, now presented, is similar to the revolute joint for the 2D case, because the objective is the same, i.e. to attach two points from different bodies P_i and P_j allowing the two bodies to rotate freely around a point. The kinematic constraints, associated to the spherical joint represented in Figure 2.5 (a) are written as:

$$^{(s,3)}\mathbf{\Phi} = \mathbf{r}_j^P - \mathbf{r}_i^P \tag{2.18}$$

Which is now described in its expanded form as:

$$^{(s,3)}\mathbf{\Phi} = \mathbf{r}_j + \mathbf{A}_j\mathbf{s}_j^P - \mathbf{r}_i - \mathbf{A}_i\mathbf{s}_j^P \tag{2.19}$$

where $\mathbf{A}_i$ and $\mathbf{A}_j$ are the 3D transformation matrices for each body. The velocity constraint equations are given by:

$$^{(s,3)}\dot{\mathbf{\Phi}} = \dot{\mathbf{r}}_j - \mathbf{A}_j\tilde{\mathbf{s}}_j^P\,'\omega_j\,' - \dot{\mathbf{r}}_i + \mathbf{A}_i\tilde{\mathbf{s}}_i^P\,'\omega_i\,' \tag{2.20}$$

where $\tilde{\mathbf{s}}_j^P$ and $\tilde{\mathbf{s}}_i^P$ is the skew-symmetric matrix of vectors $\mathbf{s}_j^P$ and $\mathbf{s}_i^P$, respectively. The $\boldsymbol{\gamma}$ vector for the spherical joint is provided by:

$$^{(s,3)}\boldsymbol{\gamma} = -\mathbf{A}_j\tilde{\omega}_j\,'\tilde{\mathbf{s}}_j^P\,'\omega_j\,' + \mathbf{A}_i\tilde{\omega}_i\,'\tilde{\mathbf{s}}_i^P\,'\omega_i\,' \tag{2.21}$$

The spatial revolute joint, shown in Figure 2.5 (b), consists in adding the spherical constraint to a new set of equations to allow only a rotation between the bodies, i.e.,

$$^{(rev,5)}\mathbf{\Phi} = \left\{ \begin{array}{c} ^{(s,3)}\mathbf{\Phi} \\ ^{(n1,1)}\mathbf{\Phi} = \mathbf{u}_i^T\mathbf{h}_j \\ ^{(n1,1)}\mathbf{\Phi} = \mathbf{u}_i^T\mathbf{t}_j \end{array} \right\} = \mathbf{0} \tag{2.22}$$

where vectors $\mathbf{u}_i$, $\mathbf{h}_j$ and $\mathbf{t}_j$ are unit vectors fixed to bodies *i* and *j*, as shown in Figure 2.5 (b).

Finally, the spatial translational joint, shown in Figure 2.5 (c), is defined by the constraint equations as:

$$^{(tra,5)}\mathbf{\Phi} = \left\{ \begin{array}{c} ^{(n1,1)}\mathbf{\Phi} = \mathbf{u}_i^T\mathbf{h}_j \\ ^{(n1,1)}\mathbf{\Phi} = \mathbf{u}_i^T\mathbf{t}_j \\ ^{(n1,1)}\mathbf{\Phi} = \mathbf{d}^t\mathbf{h}_j \\ ^{(n1,1)}\mathbf{\Phi} = \mathbf{d}^t\mathbf{t}_j \\ ^{(n1,1)}\mathbf{\Phi} = \mathbf{h}_i^T\mathbf{t}_j \end{array} \right\} = \mathbf{0} \tag{2.23}$$

where all the vectors are represented in Figure 2.5 (c). Just as for the spherical joint, also the contributions to the Jacobian matrix and the acceleration equations of the revolute and translation joints must be evaluated. For the sake of conciseness these are not shown here, being the reader directed to the work of Nikravesh [18] for detailed information.

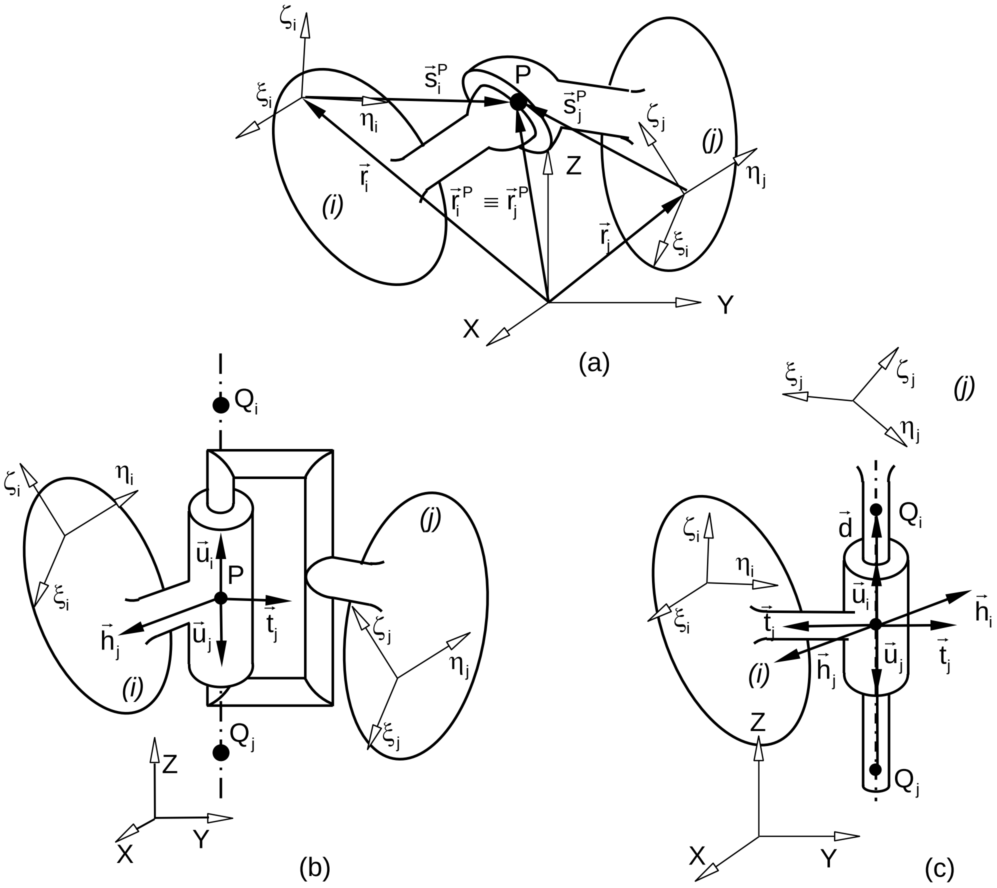


Figure 2.5 - Kinematic joints: (a) spherical joint; (b) revolute joint; (c) translational joint

2.3. Equations of motion

In order to perform the multibody dynamic analysis a set of equations of motion must be computed. For this purpose, the Jacobian matrix $\mathbf{\Phi}_q$ needs to be constructed as well as the $\boldsymbol{\gamma}$ vector. Furthermore, the mass matrix **M** and the external forces vector **g**, need to be computed. The mass matrix is written as:

$$\mathbf{M}_i = \begin{bmatrix} \mathbf{N}_i & \\ & \mathbf{J}_i^{'} \end{bmatrix} ; \ \mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & & & & \\ & \cdot & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \mathbf{M}_{nb} \end{bmatrix} \qquad (2.24)$$

being the contribution to the mass matrix $\mathbf{M}_i$ of each individual body given by:

$$\mathbf{N}_i = \begin{bmatrix} m_i & & \\ & m_i & \\ & & m_i \end{bmatrix} \; ; \; \mathbf{J}_i^{'} = \begin{bmatrix} J_{\xi\xi_i}^{'} & & \\ & J_{\eta\eta_i}^{'} & \\ & & J_{\zeta\zeta_i}^{'} \end{bmatrix} \tag{2.25}$$

where m_i is the mass of body i and $J_{\xi\xi_i}^{'}$, $J_{\eta\eta_i}^{'}$ and $J_{\zeta\zeta_i}^{'}$ are the moments of inertia with respect to the axes of inertia for the same body which are supposed to coincide with the body fixed frame axes. The global mass matrix **M** is built by assembling the mass matrix of each body, for all bodies *nb*. The external forces vector **g** is constructed with the different forces and moments applied to each body with respect to the global reference frame axes, as shown in detail in reference [18].

The system of equations which describe the multibody system constrained motion is now written as [18]:

$$\begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_{\mathbf{q}}^{\mathbf{T}} \\ \mathbf{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ \boldsymbol{\gamma} \end{bmatrix} \tag{2.26}$$

where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. The equations of motion, equation (2.26), have to be solved and integrated in time, which is prone to the accumulation of numerical errors in the positions and velocities. As the velocity and position equations are not used, explicitly, in the equations of motion the violations of constraints that develop, besides not being desirable, they require the use of methods to minimize them and help stabilizing the equations. One of the most common methods to control the constraint errors is the one proposed by Baumgarte [19] which leads to a slight modification in the equations of motion, by introducing a feedback term that includes the velocity and position constraint equations as:

$$\begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_{\mathbf{q}}^{\mathbf{T}} \\ \mathbf{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ \boldsymbol{\gamma} - 2\alpha\dot{\mathbf{\Phi}} - \beta^2\mathbf{\Phi} \end{bmatrix} \tag{2.27}$$

where α and β are constants, generally with values $\alpha=\beta=5.0$ [18]. Note that this method does not correct the violation errors but simply control them within workable limits.

2.4. Integration of equations of motion

The system of equations of motion (2.27) is solved and integrated in time. Both 2D and 3D dynamic analysis programs work using the same scheme, except with a small divergence when handling the spatial angular velocity. The algorithm for the solution and integration of the equations of motion has the following steps:

i. Input files are given to the program with information regarding the number of bodies, masses, moments of inertia, kinematic constraints and initial conditions, i.e., positions and velocities, therefore, with the complete model information;

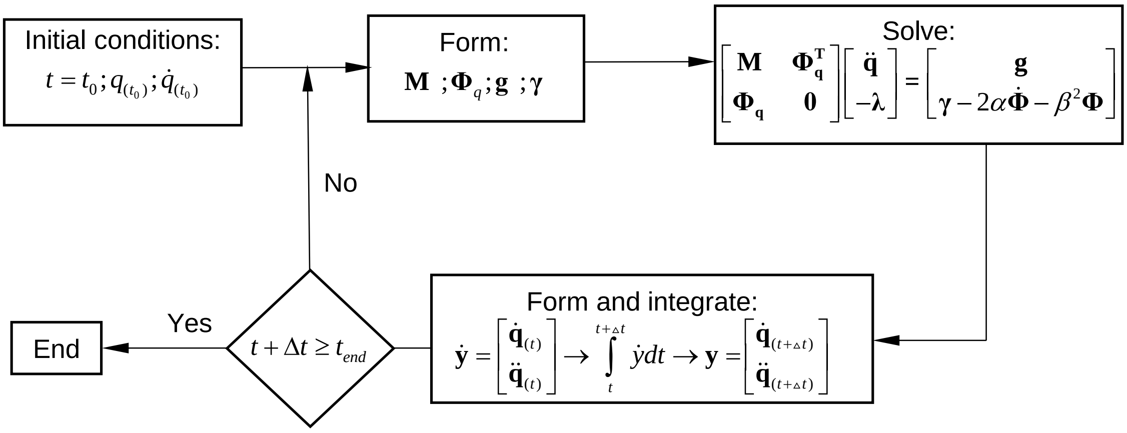
ii. The mass matrix **M**, Jacobian matrix $\mathbf{\Phi}_q$, $\boldsymbol{\gamma}$ vector and the external forces vector **g** are computed;

iii. The equations of motion are solved for the accelerations vector $\ddot{\mathbf{q}}$ and for the Lagrange multipliers vector $\boldsymbol{\lambda}$;

iv. The vector $\dot{\mathbf{y}} = \begin{bmatrix} \dot{\mathbf{q}}^T & \ddot{\mathbf{q}}^T \end{bmatrix}^T$ is formed and integrated in time, leading to a new vector with positions and velocities $\mathbf{y} = \begin{bmatrix} \mathbf{q}^T & \dot{\mathbf{q}}^T \end{bmatrix}^T$ that are required to compute the new equations of motion for the next time step $t=t+\Delta t$;

v. If $t+\Delta t > t_{end}$ the program stops, otherwise it goes back to step ii) and continues.

Figure 2.6 shows the schematic representations for the programs algorithms and allows a better visualization and understanding of the programs.



Initial conditions:
$t = t_0 ; q_{(t_0)} ; \dot{q}_{(t_0)}$
Form:
$\mathbf{M}\ ; \mathbf{\Phi}_q ; \mathbf{g}\ ; \boldsymbol{\gamma}$
Solve:
$\begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_\mathbf{q}^\mathbf{T} \\ \mathbf{\Phi}_\mathbf{q} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ \boldsymbol{\gamma} - 2\alpha\dot{\mathbf{\Phi}} - \beta^2\mathbf{\Phi} \end{bmatrix}$
No
Yes
End
$t + \Delta t \geq t_{end}$
Form and integrate:
$\dot{\mathbf{y}} = \begin{bmatrix} \dot{\mathbf{q}}_{(t)} \\ \ddot{\mathbf{q}}_{(t)} \end{bmatrix} \rightarrow \int_t^{t+\Delta t} \dot{y}dt \rightarrow \mathbf{y} = \begin{bmatrix} \dot{\mathbf{q}}_{(t+\Delta t)} \\ \ddot{\mathbf{q}}_{(t+\Delta t)} \end{bmatrix}$

Figure 2.6 - Schematic representation of the dynamic analysis programs algorithms

The only difference in the solution scheme between the planar and spatial Cartesian coordinates involves the angular velocities. The relation between time derivatives of Euler parameters and the angular velocity is [18]:

$$\dot{\mathbf{p}}_i = \frac{1}{2}\mathbf{L}_i^T \boldsymbol{\omega}_i^{'} \tag{2.28}$$

then, in a spatial system,

$$\dot{\mathbf{y}} = \begin{bmatrix} \dot{\mathbf{q}}^{*T} & \ddot{\mathbf{q}}^T \end{bmatrix}^T \tag{2.29}$$

which after the integration results in:

$$\mathbf{y} = \begin{bmatrix} \mathbf{q}^{*T} & \dot{\mathbf{q}}^T \end{bmatrix}^T \tag{2.30}$$

in which $\dot{\mathbf{q}}^* = \begin{bmatrix} \dot{\mathbf{q}}_1^{*T} & \ldots & \dot{\mathbf{q}}_n^{*T} \end{bmatrix}^T$ with $\dot{\mathbf{q}}_i^* = \begin{bmatrix} \dot{\mathbf{r}}_i^T & \dot{\mathbf{p}}_i^T \end{bmatrix}^T$ and $\dot{\mathbf{q}} = \begin{bmatrix} \dot{\mathbf{q}}_1^T & \ldots & \dot{\mathbf{q}}_n^T \end{bmatrix}^T$ with $\dot{\mathbf{q}}_i = \begin{bmatrix} \dot{\mathbf{r}}_i^T & \boldsymbol{\omega}_i^{'T} \end{bmatrix}^T$. Similar vector arrangements are written for the position and acceleration vectors.

3 Vehicle dynamics

Some of the fundamental issues in road vehicle dynamics are discussed in this chapter with particular emphasis in the automotive vehicle suspension and the tyre mechanics. As the work presented here is demonstrated with the Lancia Stratos its 2D model is described and a demonstrative case of its dynamics simulated with the 2D dynamic analysis program is developed.

3.1. The automotive vehicle

In order to better understand how an automobile is built and runs, which is essential for this work, some concepts are presented, especially with respect to suspensions and steering systems. The suspension is one of the most important mechanisms, regarding the dynamics of an automobile, linking the wheels to the vehicle body while allowing their relative motion [6]. The wheels should be able to traction, brake and steer the vehicle using the suspensions linkages to withstand all the tie-ground contact forces involved. The function of the suspension mechanism not only provides a filtering mechanism for the vibrations transmitted from the road, with all its irregularities, to the car body but also promotes the stability and maneuverability of the vehicle by ensuring the best possible contact between tyres and ground [7]. Before a detailed explanation about the suspension systems, it is necessary to define a reference system for the automobile, in which the vehicle suspension concepts and its dynamics are described. Such vehicle reference frame, shown in Figure 3.1, is fixed to the car body center of mass and has the axis aligned as shown in the figure.

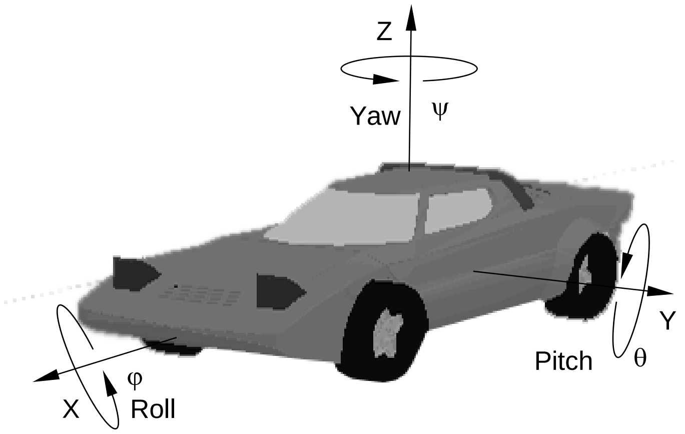


Figure 3.1 - Automobile axis system, fixed to the car body center of mass, with roll, pitch and yaw angles

Automotive suspension systems may be divided into three different groups: independent, dependent and semi-dependent suspensions [7]. Dependent and semi-dependent suspensions are only briefly described here, while a more detailed description of the independent suspensions is provided, due to their importance particularly in race and sport vehicles.

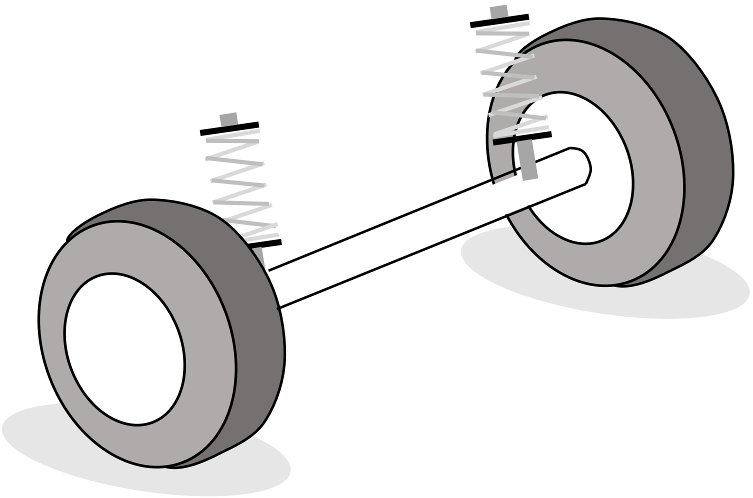


Figure 3.2 - Simplified representation of a type of dependent suspensions system

The dependent suspension systems also called rigid axle suspensions are schematically shown in Figure 3.2. Note that other types of dependent suspensions also exist. As the name suggests there is a rigid connection between the wheels of the axle, being most common in rear axles. In this type of suspensions the motion of one wheel also affects the coupled wheel [7]. A semi-independent suspension system has a behavior that puts them somewhere in between dependent and independent suspensions. Usually the two wheels are connected by the same axle but the flexibility of the axle cannot be neglected.

Independent suspensions are the type of mechanism which allows the two wheels of the same axle to move independently of each other, as for instance the double A-Arm and the McPherson suspensions shown in Figure 3.3. Many other types of independent suspensions, besides double A-Arm and McPherson, exist for particular applications.

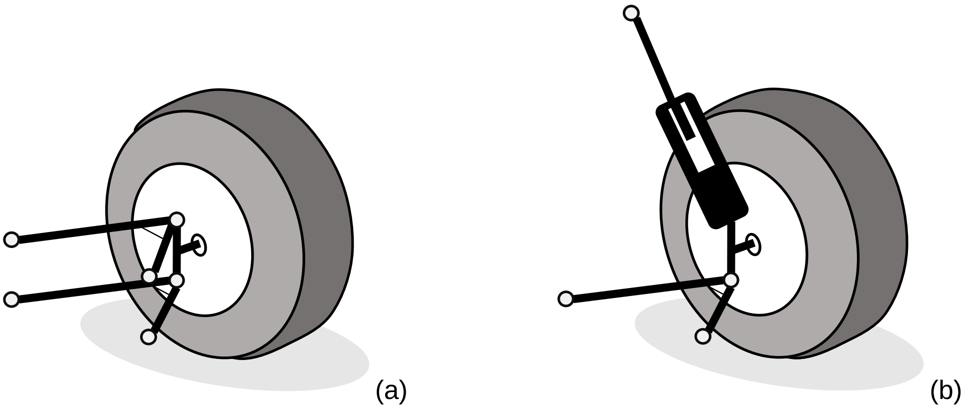


Figure 3.3 - Independent suspension types: (a) double A-Arm; (b) McPherson

The double A-Arm suspension is a four linkage suspension that uses the chassis as the base link. This type of suspension has, what is commonly named, the upper and lower triangles, being represented in Figure 3.3 (a). In a McPherson suspension the upper triangle is replaced by an inverted slider mechanism that also uses the chassis as the base link. Multilink suspensions are also used in some cars, being their complexity a problem to deal with [6]. The five point suspension systems, which provide extra steer ability in irregular roads, are used with success in high-end vehicles [7].

The linkages between the wheel and chassis must ensure that the wheels are always in the upright position and constrain all the degrees of freedom, except for the vertical displacement. The primary elastic components include the springs and anti-roll bars and they connect the struts to the chassis elastically. An anti-roll bar is a mechanism which increases the roll stiffness (about the X axis) of the vehicle in order to ensure a more even distribution of the normal contact force of the tyres in the same axle, being used in the majority of the automobiles. The anti-roll bar is also called stabilizer by some authors [6]. Finally the damping members of the suspension such as the shock absorbers or dampers are of fundamental importance to the vehicle stability as they are responsible for ensuring a better continuous contact between tyre and road. Note that the tyres also have elastic and damping properties, but these characteristics are unrelated to the suspension characteristics.

To maneuver an automobile a steering system is needed in order to rotate the wheels and support the control of the vehicle heading according to the driver requirements. A steering system is usually composed by three main parts: steering mechanism, steering box and steering column [7]. The steering mechanism makes the connection between the wheels and the steering box. The steering column connects the steering wheel, handled by the driver, to the steering box. Almost every car has a front steering system, although it can have a four wheel steering system mechanism or even a rear steering system only.

There are some considerations to make regarding the steering kinematics. When an automobile is negotiating a corner at a slow speed, there is a relationship between the inner wheel angle and the outer one, as the arc of circle traveled by each wheel implies a different steering angle for each wheel in order to have to turn slip-free [6]. Hence, there is a geometric condition, called Ackerman steering or Ackerman geometry, which provides the relation between the steering angles of the two wheels.

The Ackerman condition is written as [7]:

$$\tan(\delta_i) = \frac{w}{R_1 - \frac{t}{2}} \qquad ; \qquad \tan(\delta_o) = \frac{w}{R_1 + \frac{t}{2}} \tag{3.1}$$

where w is the vehicle wheelbase, distance between the two axles, t is the vehicle track, distance between wheel of the same axle, and δ_i and δ_o are the angle of the inner and outer wheels, respectively. R_1 is the distance between the turning center O and the point of the rear axle at the vehicle's half-track width. All quantities in equation (3.1) are defined in Figure 3.4.

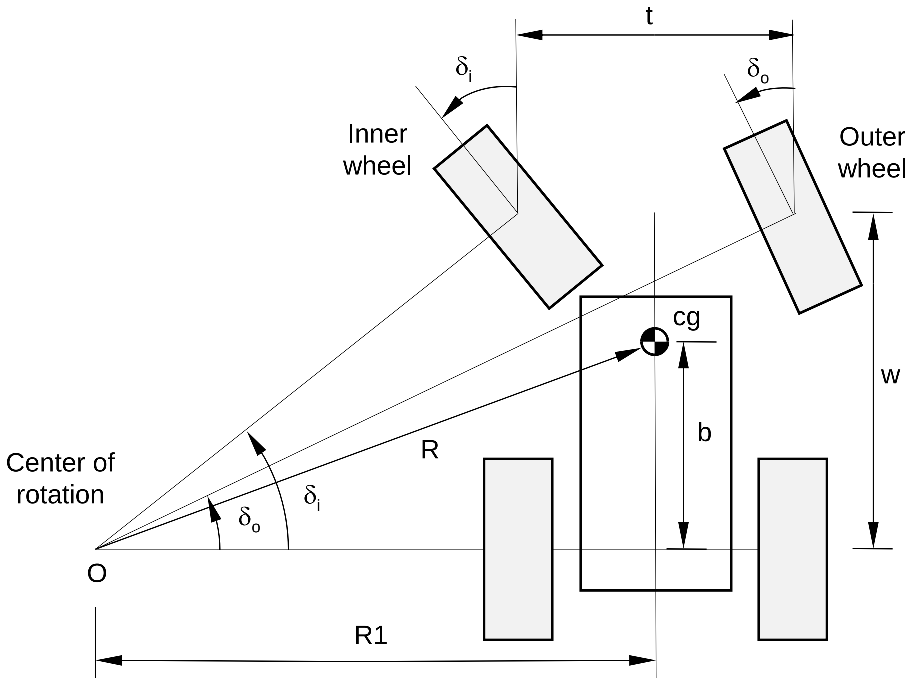


Figure 3.4 - Schematic representation of the Ackerman geometry

From equation (3.1) a direct relation between the two angles is found:

$$\cot(\delta_i) - \cot(\delta_o) = \frac{t}{w} \tag{3.2}$$

There are other concepts which have importance in vehicle dynamics, and in particular in race dynamics, such as aerodynamics. However in this work all aerodynamic forces are neglected in order to maintain the complexity of the models at a reasonable level. For the interest reader the work of Milliken and Milliken [5] provides the necessary insights into the particular problematic of a race car dynamics.

3.2. Tyre Mechanics

Some important concepts regarding tyre mechanics are presented in this section, leading to the representation of the Pacejka tyre model [8]. When an automobile is designed it is known that the tyres are the only means to transfer forces between the vehicle and the road [6] being the objective to maximize the lateral forces that the tyre can produce in order to control the behavior of the vehicle. Beyond that, and as aforementioned, the tyre also has an elastic and damping characteristics that influence the tyre-road contact forces and, consequently, the complete vehicle dynamics. Therefore, it is only natural that the tyre is one of the most important components of any automobile.

A tyre is made of rubber and a series of other synthetic materials. It is a complex composite structure made up of many rubber components and reinforcement steel chords, as well. The geometry of the several layers and reinforcement chords that make part of the tyre give each model a unique structure that is reflected in the tyre behavior. There are two different tyre groups: radial and non-radial tyres and they are divided depending on the metallic reinforcement chords orientation [6]. Radial tyres are the most used. Usually radial tyres have a better cornering ability, which means that they permit higher lateral forces. This is easily explained because in a radial tyre the sidewall is more flexible. Hence, there is more vertical deflection in a radial tyre which keeps the tread flat on the ground as it is shown in Figure 3.5. The geometry of the tyre tread has also great importance especially when a vehicle is negotiating a road on wet conditions.

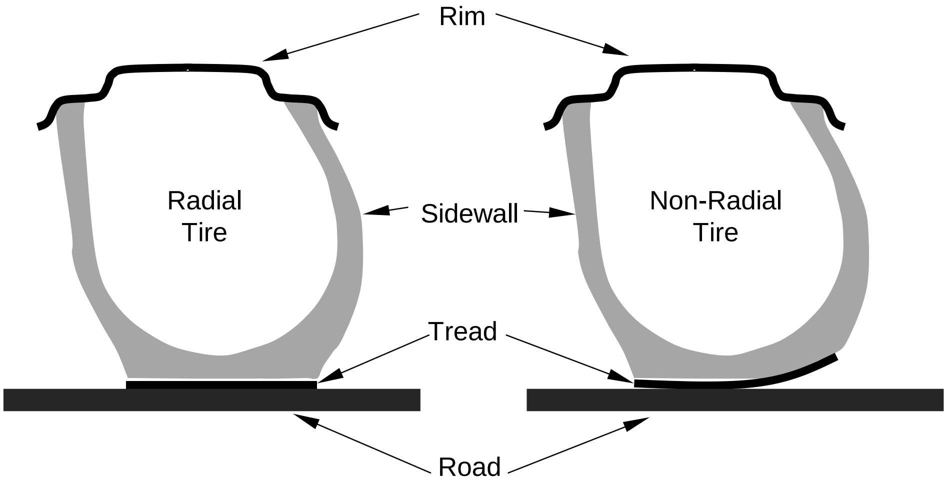


Figure 3.5 - Radial and Non-Radial tread deformation comparison with presence of a lateral force

It is now important to understand tyre dynamics as it influences greatly the behavior of the entire vehicle. Figure 3.6 shows the tyre coordinate system and the forces and moments applied in each wheel being the complete tyre mechanisms described in this referential. F_x is the tyre longitudinal force and it may be positive or negative as the vehicle is accelerating or braking, respectively, F_y is the tyre lateral force and F_z is the normal force, which is mostly dependent on vehicle weight and load transfers. M_x, M_y and M_z are the roll moment, pitch moment, also named rolling resistance moment, and yaw moment, which is most commonly called self-aligning torque or moment. The angles α and γ are the side slip and camber angles, respectively and they have great influence in the tyre behavior, **u** is the velocity of the wheel.

Both longitudinal and lateral forces are functions of the side slip angle, slip ratio, camber angle and the normal force F_z. The self-aligning torque M_z, also a very important component of the tyre handling characteristics, is also dependent on the same parameters.

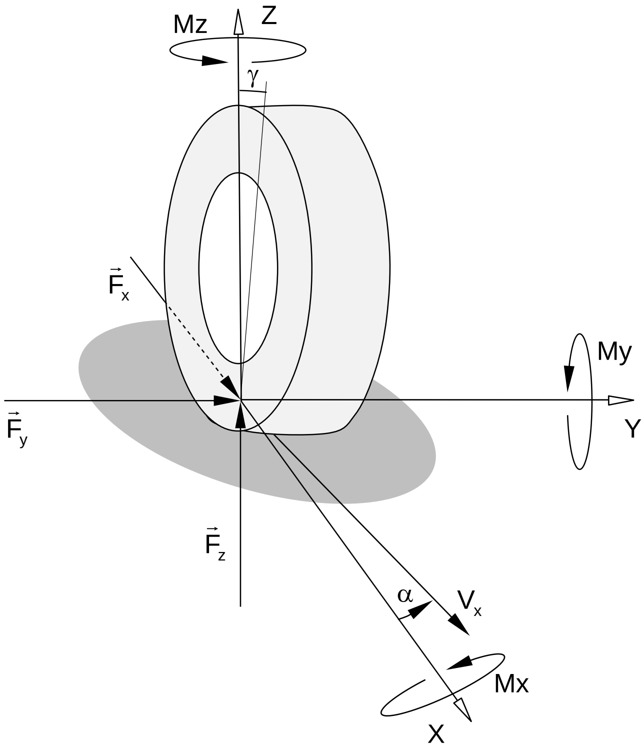


Figure 3.6 - Tyre coordinate system

The side slip angle *α*, has a large influence in the tyre lateral force. Usually the tyre lateral force grows until a certain peak value of slip angle, which varies from tyre to tyre, and then decreases with increasing slip angles. Of course it also depends greatly on the vertical load force or normal force of the tyre-road contact. The tyre side slip angle is calculated using the relation between the *x* and *y* components of the tyre velocity:

$$\alpha = \arctan\left(\frac{V_y}{V_x}\right) \tag{3.3}$$

where *x* and *y* refer to the tyre coordinate frame in Figure 3.6, V_x is the *x* component of the wheel velocity and V_y is the *y* component.

The longitudinal slip ratio *s* involves the tyre geometric radius R_g and the angular velocity of the tyre ω_w. The tyre geometric radius is the radius of the unloaded tyre, i.e., tyre without any deformation, and the angular velocity is simply the real angular velocity of the wheel. The tyre longitudinal slip ratio is now presented:

$$s = \frac{R_g \omega_w}{V_x} - 1 \tag{3.4}$$

The concept of tyre effective radius R_e, which is required for the tyre model, is introduced as the relation between the tyre longitudinal velocity V_x and the wheel's angular speed ω_w. The effective tyre radius is obtained for zero longitudinal slip conditions as:

$$R_e = \frac{V_x}{\omega_w} \tag{3.5}$$

It is important to explain that the slip ratio has a large contribution for the tyre longitudinal force F_x. In order to produce a longitudinal force, the tyre must always have a slip ratio other than zero, which means that the angular velocity of the wheel times its geometric radius is different from the actual longitudinal velocity of the tyre which means that in fact the tyre is slipping, no matter how low that slippage may be. Depending on the type of tyre used and on its normal load, the longitudinal force grows with the slip ratio until a peak value and from that point on the tyre starts to slide. The slip ratio is positive for traction and negative for braking.

The tyre's camber angle γ, which is schematically represented in Figure 3.7, is the angle between the tyre plane and the vertical plane of the vehicle measured in turn of the x axis. It is negative when the top of the tyre points towards the car and it is positive when it points to the outside of the vehicle.

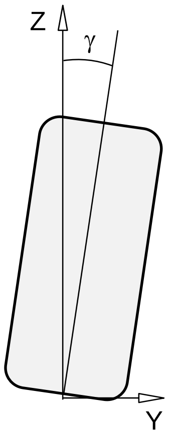


Figure 3.7 - Representation of tyre's camber angle

The linearized forces and aligning moment, for a small level of slip are [8]:

$$F_x = C_s s \tag{3.6}$$

$$F_y = C_\alpha \alpha + C_\gamma \gamma \tag{3.7}$$

$$M_z = -C_{M\alpha}\alpha + C_{M\gamma}\gamma \tag{3.8}$$

where C_s is the longitudinal slip coefficient, C_α is the cornering stiffness and C_γ is the camber stiffness. The cornering stiffness has great importance when computing the tyre's lateral force. It is given by the slope of the lateral force F_y versus the side slip angle α, near the origin. Regarding the self-aligning torque, $C_{M\alpha}$ is the aligning stiffness and $C_{M\gamma}$ is the camber contribution for the aligning stiffness. All these parameters are specific properties of each tyre and are very important to determine the vehicle global dynamics.

3.2.1. Pacejka tyre model

There is a large number of tyre models in the literature, each one with its own virtues and drawbacks. Some of the best known models have been proposed by Gim and Nikravesh [10] and by Rill [20]. However, one of the most known approaches to model the tyre mechanics is the Pacejka tyre model with the Pacejka magic formula [8]. This method uses a formula which describes the tyre contact forces, with all their components, as function of the side slip angle, slip ratio, camber angle and normal load.

Pacejka magic formula has several parameters which are obtained using experimental data. Each tyre has its own magic formula parameters. Magic formula started to be first developed in the mid-eighties and has been updated several times since then. This kind of modelling is considered as “semi-empirical” because the models are based in experimental data but also have some structures based in physical modelling [8].

Pacejka’s magic formula describes the lateral and longitudinal tyre forces for pure side and longitudinal slip, respectively, but also for combined slip, as well as the self-aligning torque that develops in each case.

The general form of the Magic Formula is given by [8]:

$$Y = D\sin[C\arctan\{B(X+S_H)-E(B(X+S_H)-\arctan B(X+S_H))\}]+S_V \tag{3.9}$$

where Y may assume the value for F_x, F_y or even for M_z. Furthermore the input variable X may be the slip angle α or the slip ratio s, depending on which force is being computed. The different coefficients are briefly described in Table 3.1.

Table 3.1 - Magic formula coefficients

B	Stiffness factor
C	Shape factor
D	Peak value
E	Curvature factor
S_H	Horizontal shift
S_V	Vertical shift

The camber angle and normal load are also used not only directly in the magic formula but also while computing the formula coefficients. Needless to say that the coefficients are calculated differently for each one of the forces, longitudinal or lateral, or for the aligning torque

For pure slip, i.e., the side slip angle does not affect it, the longitudinal force F_{xp} is given by:

$$F_{xp} = D_x \sin[C_x \arctan\left\{B_x(s+S_{Hx}) - E_x(B_x(s+S_{Hx}) - \arctan B_x(s+S_{Hx}))\right\}] + S_{Vx} \tag{3.10}$$

For the pure side slip, i.e., the slip ratio does not affect it, the lateral force F_{yp} is given by:

$$F_{yp} = D_y \sin[C_y \arctan\left\{B_y(\alpha+S_{Hy}) - E_y(B_y(\alpha+S_{Hy}) - \arctan B_y(\alpha+S_{Hy}))\right\}] + S_{Vy} \tag{3.11}$$

The pure side slip aligning torque M_{zp} is given by:

$$M_{zp} = M_{zp}^{'} + M_{zrp} \tag{3.12}$$

$$M_{zp}^{'} = -t_p F_{yp} \qquad ; \qquad M_{zrp} = D_r \cos[C_r \arctan(B_r \alpha_r)] \tag{3.13}$$

where F_{yp} is the pure side slip lateral force and t_p is given by:

$$t_p = D_t \cos[C_t \arctan\left\{B_t\alpha_t - E_t(B_t\alpha_t - \arctan(B_t\alpha_t))\right\}]\cos'\alpha \tag{3.14}$$

For the combined slip forces F_x, F_y and moment M_z, the pure slip forces and moment, given by equations (3.10) through (3.14) are calculated first. Then the new forces and moment for combined slip are given as:

$$F_x = G_{x\alpha} F_{xp} \tag{3.15}$$

$$F_y = G_{ys} F_{yp} + S_{Vys} \tag{3.16}$$

$$M_z = M_z^{'} + M_{zr} + C_s F_x \tag{3.17}$$

where:

$$M_z^{'} = -t(F_y - S_{Vys}) \tag{3.18}$$

$$t = D_t \cos[C_t \arctan\left\{B_t\alpha_{t,eq} - E_t(B_t\alpha_{t,eq} - \arctan(B_t\alpha_{t,eq}))\right\}]\cos'\alpha \tag{3.19}$$

$$\alpha_{t,eq} = \sqrt{\alpha_t^2 + \left(\frac{K_{xs}}{K_{y\alpha}^{'}}\right)^2 s^2}\,.\mathrm{sgn}(\alpha_t) \tag{3.20}$$

and:

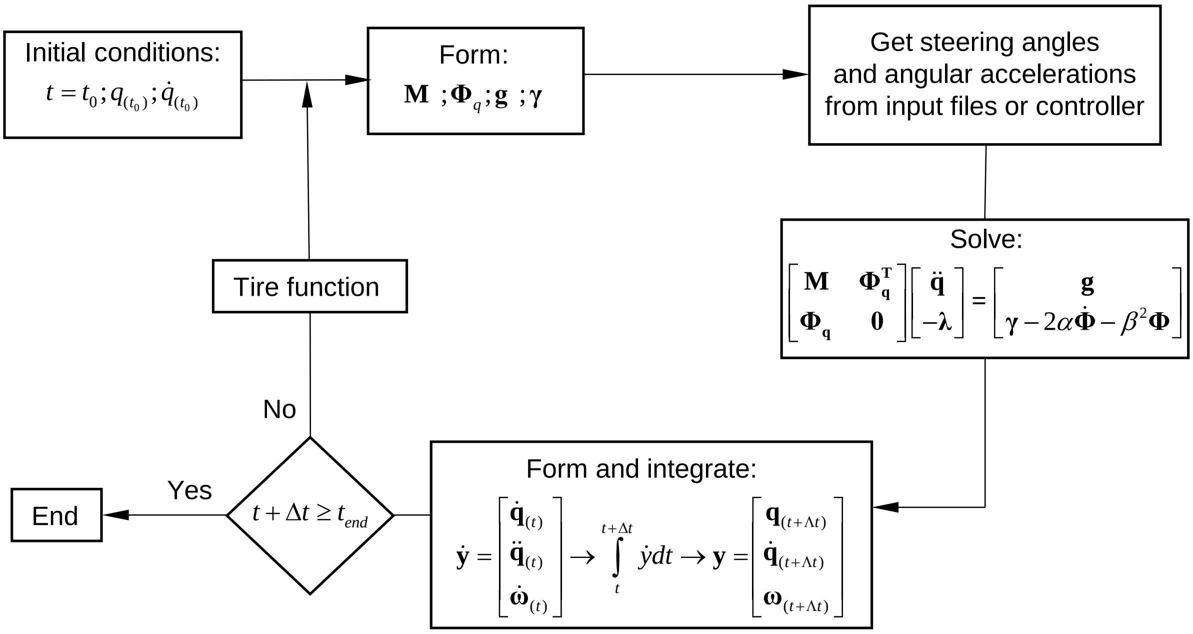
$$M_{zr} = D_r \cos[C_r \arctan(B_r \alpha_{r,eq})] \tag{3.21}$$

$$\alpha_{r,eq} = \sqrt{\alpha_r^2 + \left(\frac{K_{xs}}{K'_{y\alpha}}\right)^2 s^2}\,.\text{sgn}(\alpha_r) \tag{3.22}$$

Note that combined slip is the case when both side slip angle and slip ratio have a contribution in the calculation of the lateral and longitudinal forces and self-aligning moment. The complete Pacejka Magic Formula also includes shape parameters to adjust the tyre behavior to specific operational conditions that are not considered here. For a more detailed description of the model the reader is directed to the work of Pacejka [8].

3.2.2. Computational implementation

The tyre model is implemented in the 2D dynamics analysis program as a force that develops between the rigid bodies to which the tyre is attached and the road. Based on the dynamic analysis program described in section 2.4, the fluxdraw that describes the solution of the equations of motion is modified as shown in Figure 3.8. A new function is implemented to impose the steering mechanics and the tyre model is used for the evaluation of the force vector **g**. The tyre model function computes the longitudinal and lateral forces and the aligning moment for each tyre during each time step. First of all, the tyre coefficients are obtained in the literature, being those for the Hoosier 13x8 slick racing tyre used here for demonstration purposes.



Initial conditions:
$t = t_0; q_{(t_0)}; \dot{q}_{(t_0)}$
Form:
$\mathbf{M}\ ;\mathbf{\Phi}_q;\mathbf{g}\ ;\boldsymbol{\gamma}$
Get steering angles and angular accelerations from input files or controller
Solve:
$\begin{bmatrix}\mathbf{M} & \mathbf{\Phi}_\mathbf{q}^\mathbf{T}\\ \mathbf{\Phi}_\mathbf{q} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\ddot{\mathbf{q}}\\ -\boldsymbol{\lambda}\end{bmatrix} = \begin{bmatrix}\mathbf{g}\\ \boldsymbol{\gamma} - 2\alpha\dot{\mathbf{\Phi}} - \beta^2\mathbf{\Phi}\end{bmatrix}$
Tire function
No
Yes
End
$t + \Delta t \geq t_{end}$
Form and integrate:
$\dot{\mathbf{y}} = \begin{bmatrix}\dot{\mathbf{q}}_{(t)}\\ \ddot{\mathbf{q}}_{(t)}\\ \dot{\boldsymbol{\omega}}_{(t)}\end{bmatrix} \rightarrow \int_t^{t+\Delta t} \dot{y}dt \rightarrow \mathbf{y} = \begin{bmatrix}\mathbf{q}_{(t+\Delta t)}\\ \dot{\mathbf{q}}_{(t+\Delta t)}\\ \boldsymbol{\omega}_{(t+\Delta t)}\end{bmatrix}$

Figure 3.8 - Schematic representation of the 2D dynamic analysis program algorithm with the tyre model

In the planar multibody dynamics code the normal load F_z is considered constant for all 4 wheels, which means that no load transfer is considered here being the normal load in each wheel one fourth of the total weight of the car.

The equations of motion are solved for each time step allowing to get the positions, velocities and accelerations for each wheel, and using the 2D transformation matrix the local lateral and longitudinal velocities are computed. Using equation (3.3) the side slip angle α is known and using equation (3.4) the slip ratio s is calculated using the tyre geometric radius R_g, which is also a constant characteristic of the tyre. It is not possible to get angular velocities of the wheels ω_w, because it does not exist in the planar case, so an alternative method was implemented. For the wheel angular velocities ω_w, the program either assumes that the vehicle to have constant velocity while negotiating the track, i.e., the angular velocities are assumed constant and defined as a program initial condition, or allows the wheel to have angular accelerations which are either supplied in an input file or by a controller. The angular accelerations of the wheels, and the angular velocities, are then included in the bodies velocities and accelerations vector $\dot{\mathbf{y}}$ and integrated in time to obtain new angular velocities and tyre angular positions, to be used in the tyre model function in order to calculate the slip ratio. The camber angle γ, which is also considered constant, does not exist in the planar model being assumed null in what follows.

Having all the necessary parameters, the longitudinal and lateral forces are computed for all the wheels using equations (3.10) through (3.22). The forces are transformed to global coordinates in order to be included in the external forces vector **g** and applied in the next time step for each wheel of the vehicle, as depicted in Figure 3.8.

The automobile model is steered and accelerated either by input files, which gives steering angles and/or traction/braking torques for determined time intervals, or by the controller, as it is shown in Figure 3.8. The steering angles are used in the steering kinematic constraint where the Ackerman condition is neglected. In the case where the steering angles are controlled by an input file, this is written as a text file, .txt, where each line provides an incremental value. In each line the initial time, the end time and the steering angle variation during this time interval is provided. So at the input initial time the steering angle is the same as in the last time step and it starts varying, nonlinearly, until the input end time, when the steering angle will be the variation given by the input plus the angle at the initial time.

The steering angle variation during the input time interval is not linear so that the time derivatives of the steering angle are such that angle variation is smooth. In order to do that, a polynomial function is computed for each steering input [21]:

$$\delta_s(t) = \delta_i + 6\frac{\Delta\delta}{\Delta t^2}(t-t_i)^2 - 8\frac{\Delta\delta}{\Delta t^3}(t-t_i)^3 + 3\frac{\Delta\delta}{\Delta t^4}(t-t_i)^4 \tag{3.23}$$

where δ_s is the steering angle, δ_i is the input's initial steering angle, $\Delta\delta$ is the steering angle variation provided in the input file, Δt is the input time interval in which the complete variation must occur and t_i is the input initial time.

Figure 3.9 shows an example for the steering polynomial curve using an input initial time of 0.5 seconds and 1.5 seconds for the input end time. The initial steering angle comes from the previous steering angle and is equal to 0.5º in the example. The input steering variation is 2.0º.

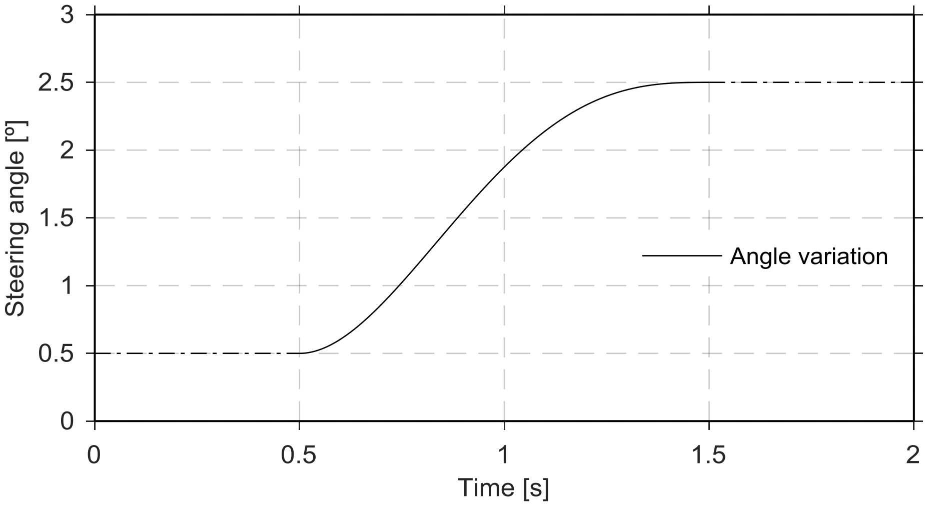


Figure 3.9 - Steering angle variation polynomial curve

In the case in which the angular velocities in each wheel are not assumed constant, i.e., when there are wheel torques provided by either an input file or by a controller, the angular accelerations of each wheel need to be calculated in order to get the angular velocities by integrating them in time. The angular accelerations of each wheel are computed by relating the total torque on the wheel with its polar moment of inertia. The total torque on the wheel is calculated as the balance between the moment produced by longitudinal tyre force and the acceleration or braking torque provided by the input or controller. The tyre rolling resistance moment is neglected.

3.3. Lancia Stratos model

The purpose of this thesis is not to evaluate any particular car model being Lancia Stratos chosen because all the necessary data to describe the model is available in the work of Ambrósio and Gonçalves [22]. The Lancia Stratos is an Italian sports car from the 1970's which became very popular because it served as the base vehicle for the successful Lancia Stratos that competed and won races in the world rally championship. Lancia Stratos is a rear wheel driving vehicle, which means that only the rear wheels can produce traction and is a front wheel steering car. Furthermore, it has front double A-Arm suspension and rear McPherson suspension. The vehicles main dimensions are described in Figure 3.10, being the most important for the purpose of this work the wheelbase, with 2.180 m, the front track, with 1.430 m, and the rear track, with 1.460 m.

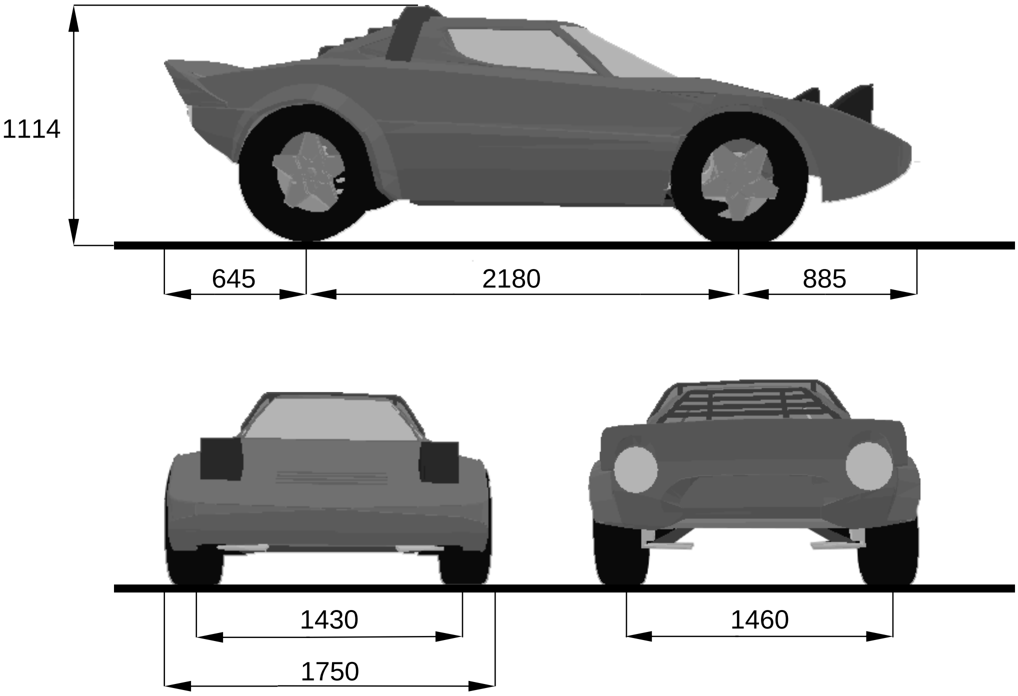


Figure 3.10 - Lancia Stratos model with all of its dimensions in millimetres

The simplified multibody system for the Lancia Stratos used in the 2D dynamics analysis program is adapted from its spatial counterpart. The multibody system is composed by only 3 bodies as shown in Figure 3.11. The first body includes every part of Lancia Stratos model except the two front wheels. Even the rear wheels are attached to the chassis as a single rigid body. This first body has all the vehicles mass and inertia minus the part that belongs to the front wheels.

The body fixed frames are all attached to the center of mass of each body and the distances between the centers of mass of each body, presented in millimeters, are needed to describe the kinematic constraints. Table 3.2 resumes the main characteristics of all the bodies that compose the simplified planar Lancia Stratos multibody system. In the front wheels bodies the suspension parts are all included, i.e., the knuckles are part of the wheels.

Table 3.2 - Mass and inertial characteristics of the system

Body	Description	Mass, *Kg*	Inertia, *Kg.m*^{2}
1	Chassis	783	750
2	Front Left Wheel	29	10
3	Front Right Wheel	29	10

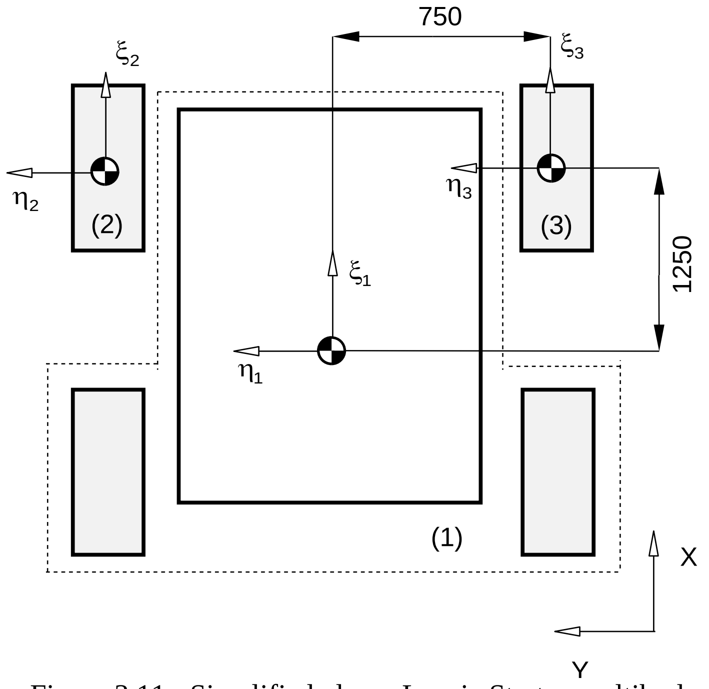


Figure 3.11 - Simplified planar Lancia Stratos multibody system

The multibody system contains four kinematic joints: two revolute joints and two steering constraints. Table 3.3 presents all the information for the kinematic constraints.

Table 3.3 - Kinematic joints description for the multibody system

Joint	Type	Body i	Body j	ξ_i^P	η_i^P	ξ_j^P	η_j^P
1	Revolute	1	2	1.25	0.75	0	0
2	Revolute	1	3	1.25	-0.75	0	0
3	Steering	2	1	-	-	-	-
4	Steering	3	1	-	-	-	-

3.4. Demonstration case

With the objective of demonstrating the features of the program developed here and to explore the performance of the vehicle model for a lane changing maneuver, a demonstration case is presented. For this demonstration case a 15 seconds simulation is performed being the steering constraint and the traction/braking property particularly used for the maneuver under analysis. Table 3.4 and Table 3.5 show the input values, for the traction/braking torque and for the steering, given to the model in this demonstration case.

Table 3.4 - Wheel's torques inputs for the demonstration case, where positive is traction and negative is braking

Input number	Initial time, [s]	End time, [s]	Wheel’s torque, [N.m]
1	1.0	3.0	100
2	4.0	6.0	-100

As Table 3.4 presents the initial time is the moment when the torques start to be applied to the wheels and it goes on until the end time is reached. It should be noted that the torque values shown are applied on each wheel, i.e. each wheel is submitted to the total torque input value. Positive torque values are used for traction, while braking torques assume negative values. When the vehicle is being accelerated, i.e., positive torques, the input torque is applied exclusively on the rear wheels, since the vehicle is rear wheel driven. Braking torques are applied on all 4 wheels, with the front/rear brake distribution, also called brake bias, being 50/50.

Table 3.5 - Steering angle inputs for the demonstration case

Input number	Initial time, [s]	End time, [s]	Steering angle, [º]
1	7.0	7.5	2.5
2	9.0	9.5	-2.5
3	11.0	11.5	-2.5
4	13.0	13.5	2.5

The steering angle inputs behave differently from the torque inputs. While in the previous case the torque value is being applied between the initial and end times and then return to zero, meaning that no torque is applied between torque inputs, in the steering angle input the time interval between initial and end times is for the steering angle to reach the final steering angle value, which means that when the end time is reached, the steering angle value will remain with the same as it reached in the final value. Having the first input of Table 3.5 as an example, the steering angle will grow from 0º to 2.5º in 0.5 seconds and is maintained at that value until new input is given. Then, the steering angle changes from the current value by the amount specified, reached at the end time. In the case depicted in Table 3.5, at the end of 9.5 s the steering angle is 0º.

The trajectory for the vehicle center of mass as a result of the traction and steering inputs for a motion that lasts 15 seconds is shown in Figure 3.12.

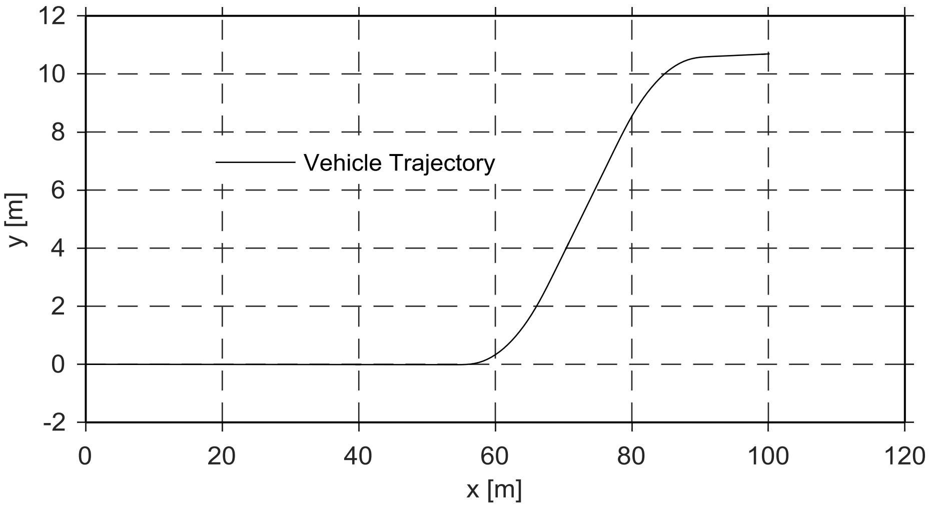


Figure 3.12 - Vehicle trajectory for the demonstration case

It is obvious that the vehicle successfully completes the lane change maneuver, which is further emphasized in the sequence of still shots of the vehicle motion in Figure 3.13.

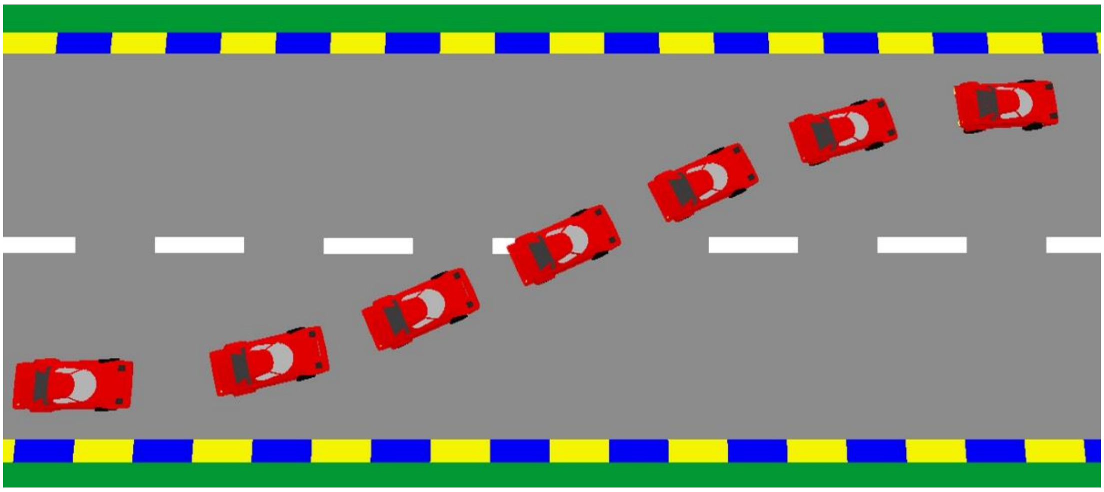


Figure 3.13 - Sequence of still shots of the vehicle performing its lane changing maneuver

The steering angles presented in Figure 3.14 are the angles between the vehicle's chassis and the left wheel during the complete lane changing maneuver. It is observed that not only the sequence of steering and counter-steering is properly defined, but also that the effective steering of the wheel is not linear and obeys the polynomial function described in section 3.2.2.

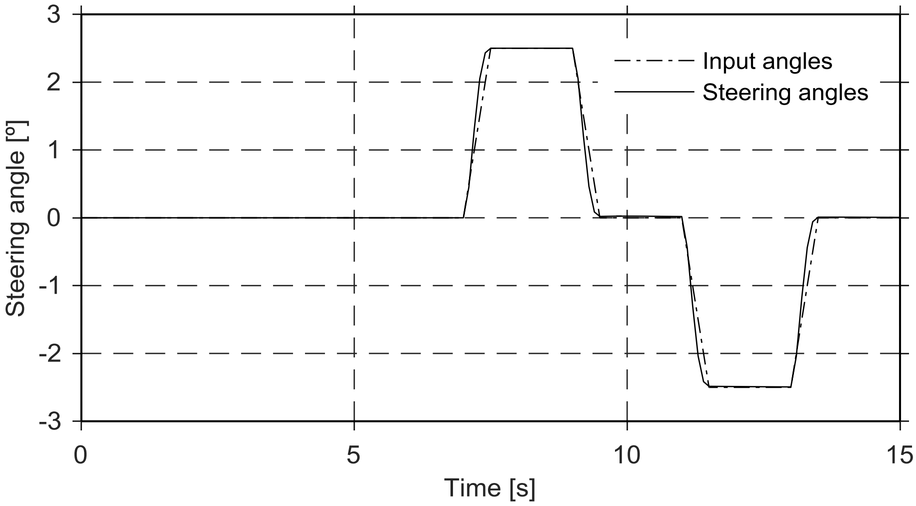


Figure 3.14 - Input steering angles vs. simulation steering angles (Left wheel)

Figure 3.15 represents the vehicle longitudinal velocity as function of time. The traction and the braking periods are clearly identified in Figure 3.15. It is also worth noting that when no torque is applied to the wheels, the vehicle velocity steadily reduces.

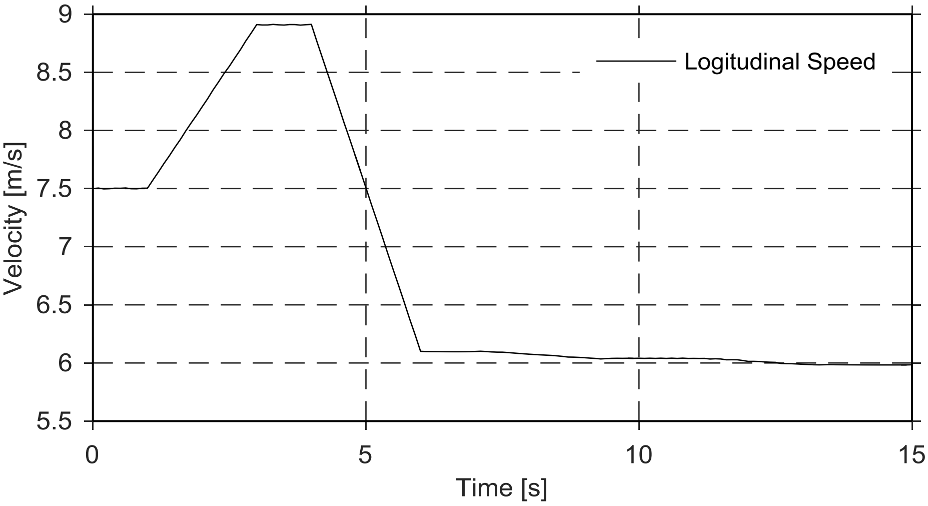


Figure 3.15 - Vehicle longitudinal velocity for the demonstration case

Figure 3.16 shows the front wheels angular velocity profiles during the maneuver, being possible to qualitatively observe the relative velocities, as the outer wheel has larger angular velocity than the inner one while negotiating a curve. This difference of the wheels angular velocity and steering angles leads the vehicle to turn left approximately between 7.0s and 9.5s and to turn right within the 11.0-13.5s time interval. Moreover it is clearly identified the acceleration and braking periods during the lane change maneuver.

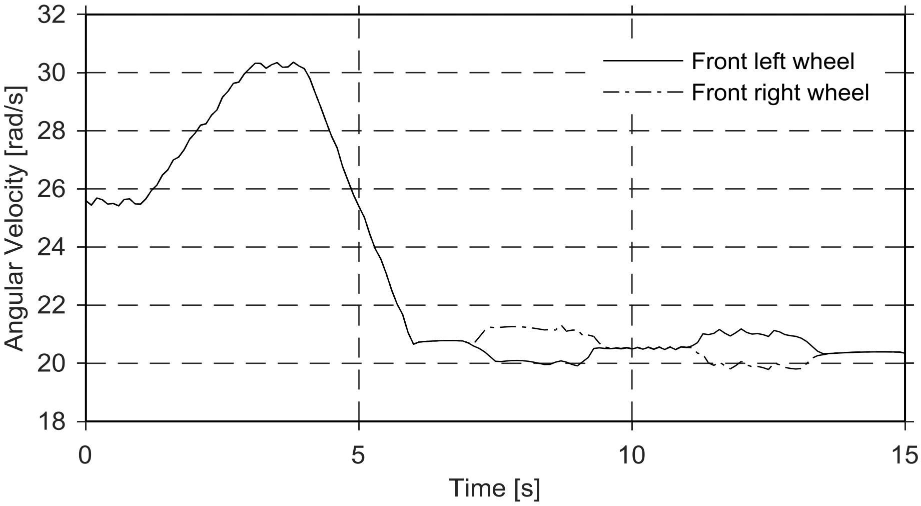


Figure 3.16 - Front wheels angular velocity for the demonstration case

All in all, the demonstration scenario in which the Lancia Stratos model negotiates a lane change maneuver clearly shows that all the main features for the vehicle planar dynamics are working properly. It is worth emphasizing that the state-variables that do not exist in the planar vehicle model, i.e., the wheels angular velocities, are handled and integrated independently of the model actually enabling to calculate the longitudinal slip ratio, which is fundamental for the vehicle tyre dynamics.

4 Vehicle Control

In this chapter a controller for the vehicle that is able to handle steering and traction/braking, is developed and implemented computationally. Particular importance is given to the previewing distance, required for a realistic steering control. Demonstrative cases featuring a scenario in which the vehicle runs around a track is presented to demonstrate the methods developed and implemented.

4.1. Control objectives and previewing distance

In order to have the minimal lap time around a race track, a controller should be able to take the vehicle from the beginning of the lap until its end as fast as possible and using the optimal trajectory. Another way to describe this objective is to state that the controller must be able to use the best trajectory with the highest speed while the vehicle tyres have enough grip to maintain the car in the track.

The controller developed and implemented here is based on the work of Antos and Ambrósio [1], who built a Linear Quadratic Regulator controller for steering purposes only. The objective in their work was the development of a virtual driver to enable a vehicle to follow a parameterized trajectory, without any previewing or traction or braking.

The previewing distance used to control the vehicle steering is not only a natural way of driving for a human driver but also a way to smooth the vehicle steering. A controller with previewing is generally more stable, especially when the vehicle is off-track or far from the ideal position. With reference to Figure 4.1, instead of trying to find the closest point in the trajectory, the controller looks for a point further ahead, O_t, using a predefined preview distance. Using that point and the current vehicle position O_c and orientation, the controller computes a new trajectory linking the two points. Hence, if the vehicle is off-track the controller brings it back in the desired trajectory using a smoother path as shown in Figure 4.1.

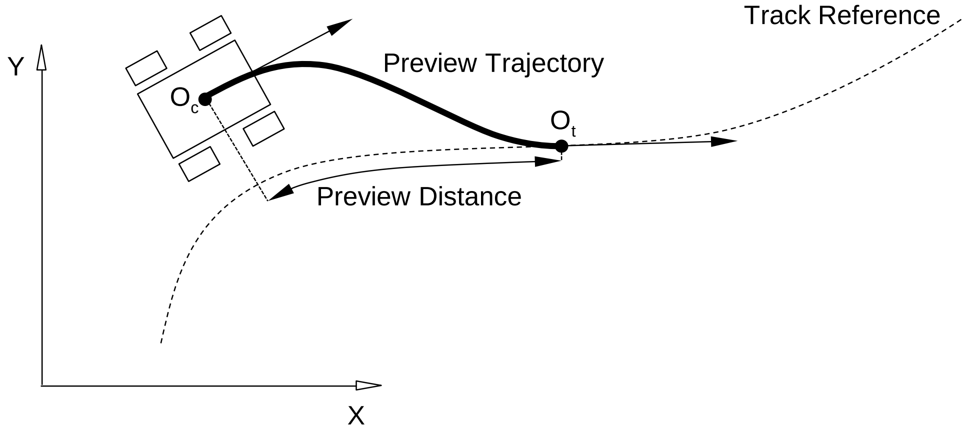


Figure 4.1 - Schematic representation of the preview distance and trajectory

The preview distance has to be predefined being dependent on the vehicle type and track geometry [23]. For instance race vehicles require smaller preview distances than family cars and motorcycles require even longer previewing distances [23]. Once the vehicle starts following one preview trajectory it will keep going until a certain point of this trajectory and then a new preview trajectory is computed using the same preview distance as before. In this way the preview trajectory is continuously updated.

The preview trajectory is represented by using a cubic polynomial. Knowing the current vehicle's point and its orientation and the position and vehicle orientation in the preview point, in the trajectory, it is possible to calculate the cubic polynomial constants. A general cubic polynomial is given by:

$$y = a + bx + cx^2 + dx^3 \tag{4.1}$$

using the coordinates of the two points and their derivatives, it is possible to get the polynomial constants by solving the system of equations defined as:

$$\begin{bmatrix} 1 & x_c & x_c^2 & x_c^3 \\ 1 & x_p & x_p^2 & x_p^3 \\ 0 & 1 & 2x_c & 3x_c^2 \\ 0 & 1 & 2x_p & 3x_p^2 \end{bmatrix} \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} = \begin{Bmatrix} y_c \\ y_p \\ y'|_{x=x_c} \\ y'|_{x=x_p} \end{Bmatrix} \tag{4.2}$$

where x_c and y_c are the vehicle current position coordinates and x_p and y_p are the coordinates of the preview point located in the track reference.

4.2. Control strategy and algorithms

4.2.1. Description of the vehicle reduced model

The development of the controller made by Antos and Ambrósio [1] is overviewed here. The controller uses a bilinear structure of the vehicle dynamics description in which the dynamics of the system are derived from a simplified planar vehicle model, represented in Figure 4.2 [1]. In this simplified vehicle model the distances *a*, *b* and *d* are respectively the distance between the car center of mass and the front axle, the distance between the car center of mass and the rear axle and the half-width of the vehicle track.

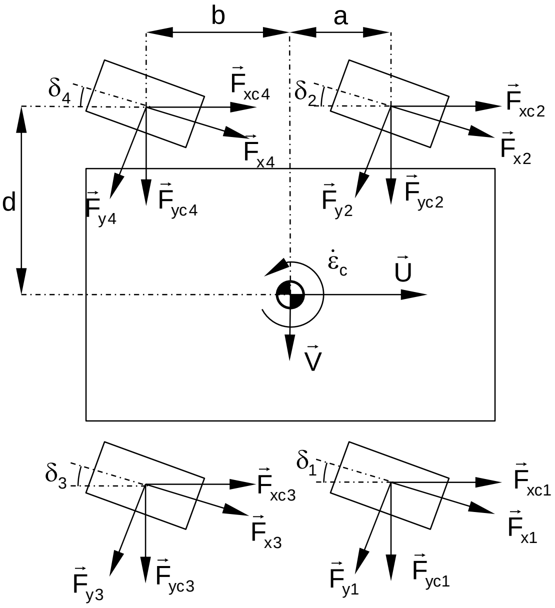


Figure 4.2 - Simplified planar vehicle model

The equations of transient motions for the planar vehicle model are given by Alleyne [24] and by Wong [25] and are written as:

$$m(\dot{U} - V\dot{\varepsilon}_c) = F_{xc1} + F_{xc2} + F_{xc3} + F_{xc4} \tag{4.3}$$

Equation (4.3) is set for the longitudinal motion and only the tyre forces of each wheel *i*, F_{xci} and F_{yci}, are taken into account in the dynamics of the model. The aerodynamic forces and any other interaction forces are neglected. The body mass is given by *m*, U and *V* are respectively the body fixed longitudinal and lateral velocities. The vehicle yaw rate is represented by $\dot{\varepsilon}_c$ and the lateral acceleration by $\dot{V}$. The lateral equation of motion is written as [24]:

$$m(\dot{V} + U\dot{\varepsilon}_c) = F_{yc1} + F_{yc2} + F_{yc3} + F_{yc4}s \tag{4.4}$$

Note that the lateral equation of motion takes into account all the wheels lateral component of the force. Finally the yaw motion is controlled by the equation of motion written as:

$$J_{Zc}\ddot{\varepsilon}_c = a(F_{yc1} + F_{yc2}) - b(F_{yc3} + F_{yc4}) + d(-F_{yc1} + F_{yc2} - F_{yc3} + F_{yc4}) \tag{4.5}$$

where J_{Zc} is the vehicle yaw moment of inertia, *a* is the distance between the vehicle's front axle and the line that is perpendicular to the chassis and crosses its center of mass, *b* is the distance between the vehicle's rear axle and the same chassis center of mass line and *d* is half of the vehicle's track width.

It is important to refer that F_{xci} and F_{yci} are not the longitudinal and lateral tyre forces but instead they are the x and y components of the tyre forces in the global reference frame. In the reduced vehicle model a simplified tyre model is used.

$$F_{xci} = F_{xi} - F_{yi}\delta_i \tag{4.6}$$

$$F_{yci} = F_{xi}\delta_i + F_{yi} \tag{4.7}$$

where here F_{xi} and F_{yi} are the tyre longitudinal and lateral forces, respectively. The longitudinal tyre force is directly proportional to the wheels' torques M_{wi} and is given by:

$$F_{xi} = \frac{M_{wi}}{R_{wi}} \tag{4.8}$$

where R_{wi} is the effective rolling radius of each wheel. Assuming the camber angle γ to be null, equation (3.7) becomes:

$$F_{yi} = C_{\alpha i}\alpha_i \tag{4.9}$$

where $C_{\alpha i}$ is the cornering stiffness of each tyre and α_i is the slip angle. The cornering stiffness for the front wheels $C_{\alpha f}$ is simply the sum of the front wheels cornering stiffness. The same applies to the rear wheels cornering stiffness $C_{\alpha r}$.

The front and rear tyre slip angles, α_{if} and α_{ir}, are calculated here using a small angle approximation being given by:

$$\alpha_{if} = \frac{V + a\dot{\varepsilon}_c}{U} - \delta_{if} \tag{4.10}$$

$$\alpha_{ir} = \frac{V - b\dot{\varepsilon}_c}{U} \tag{4.11}$$

The rear wheels do not have steering and, consequently, their slip angles are assumed to be null. Furthermore the front steering angles are assumed to be equal in both front wheels in this simplified car model. Using equations from (4.4) to (4.11) the simplified vehicle model equations of motion, also known as extended bicycle model, become:

$$m(\dot{V} + U\dot{\varepsilon}_c) = \frac{C_{\alpha f} + C_{\alpha r}}{U}V + \frac{aC_{\alpha f} - bC_{\alpha r}}{U}\dot{\varepsilon}_c + \left(\frac{M_{w1}}{R_{w1}} + \frac{M_{w2}}{R_{w2}} - C_{\alpha f}\right)\delta_f \tag{4.12}$$

$$J_{Zc}\ddot{\varepsilon}_c = \frac{aC_{\alpha f} - bC_{\alpha r}}{U}V + \frac{a^2C_{\alpha f} + b^2C_{\alpha r}}{U}\dot{\varepsilon}_c$$
$$+a\left(\frac{M_{w1}}{R_{w1}} + \frac{M_{w2}}{R_{w2}} - C_{\alpha f}\right)\delta_f + d\left(-\frac{M_{w1}}{R_{w1}} + \frac{M_{w2}}{R_{w2}} - \frac{M_{w3}}{R_{w3}} + \frac{M_{w4}}{R_{w4}}\right) \tag{4.13}$$

The equations of motion must now be transformed to the track coordinate system *(X_t,Y_t)*, as proposed by Peng and Tomizuka [26]. In Figure 4.3 some relationships between the vehicle's reference frame coordinates, the track coordinates and global coordinates are shown.

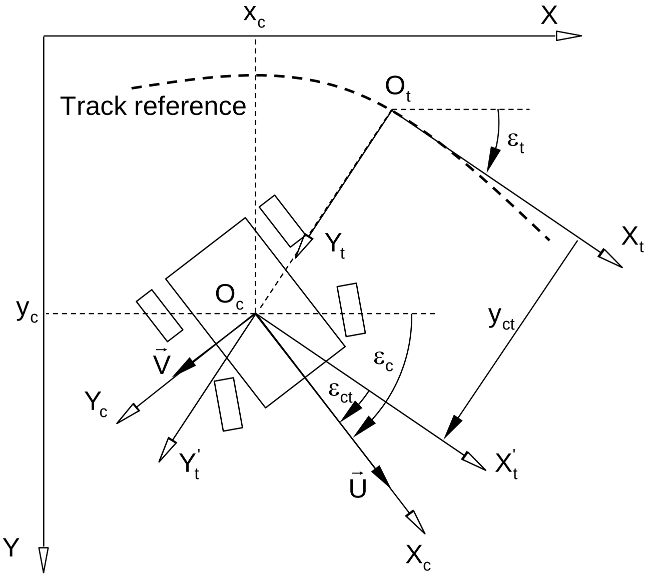


Figure 4.3 - Vehicle and track coordinate systems

The relationships between angular positions and velocities are given by:

$$\varepsilon_c = \varepsilon_t + \varepsilon_{ct} \tag{4.14}$$

$$\dot{\varepsilon}_c = \dot{\varepsilon}_t + \dot{\varepsilon}_{ct} \tag{4.15}$$

The lateral velocity of the vehicle with respect to the track reference frame is given by:

$$\dot{y}_{ct} = V\cos\varepsilon_{ct} + U\sin\varepsilon_{ct} \tag{4.16}$$

The longitudinal velocity of the track reference frame is approximated by:

$$U_t = U\cos\varepsilon_{ct} - V\sin\varepsilon_{ct} \tag{4.17}$$

Furthermore, the rotation rate of the track reference frame is:

$$\dot{\varepsilon}_t = \frac{U_t}{R_t} \tag{4.18}$$

where R_t is radius of the track.

Using the relationships from equation (4.12) to (4.18) in the first time derivative of equation (4.16), it is possible to compute the lateral acceleration of the vehicle in the track reference frame, written as:

$$\begin{aligned}\ddot{y}_{ct} &= \frac{C_{\alpha f}+C_{\alpha r}}{mU}\dot{y}_{ct} - \frac{C_{\alpha f}+C_{\alpha r}}{m}\varepsilon_{ct} + \frac{aC_{\alpha f}-bC_{\alpha r}}{mU}\dot{\varepsilon}_{ct} \\ &+ \left(\frac{M_{w1}}{mr_{w1}}+\frac{M_{w2}}{mr_{w2}}-\frac{C_{\alpha f}}{m}\right)\delta_f + \left(\frac{aC_{\alpha f}-bC_{\alpha r}}{m}-U^2\right)\frac{1}{R_t}\end{aligned} \tag{4.19}$$

Moreover, the same approach is now applied to equation (4.13) to write the yaw angular acceleration in the track reference frame:

$$\begin{aligned}\ddot{\varepsilon}_{ct} &= \frac{aC_{\alpha f}-bC_{\alpha r}}{J_{Zc}U}\dot{y}_{ct} - \frac{aC_{\alpha f}-bC_{\alpha r}}{J_{Zc}}\varepsilon_{ct} + \frac{a^2C_{\alpha f}+b^2C_{\alpha r}}{J_{Zc}U}\dot{\varepsilon}_{ct} \\ &+ \frac{a}{J_{Zc}}\left(\frac{M_{w1}}{r_{w1}}+\frac{M_{w2}}{r_{w2}}-C_{\alpha f}\right)\delta_f + \frac{d}{J_{Zc}}\left(-\frac{M_{w1}}{r_{w1}}+\frac{M_{w2}}{r_{w2}}-\frac{M_{w3}}{r_{w3}}+\frac{M_{w4}}{r_{w4}}\right)+\left(\frac{a^2C_{\alpha f}+b^2C_{\alpha r}}{J_{Zc}}\right)\frac{1}{R_t}\end{aligned} \tag{4.20}$$

The steering dynamics is now added to the model being described by a second order differential equation, given by:

$$J_f\ddot{\delta}_f + b_f\dot{\delta}_f = M_f \tag{4.21}$$

where J_f is the equivalent moment of inertia of the steering system, while b_f is an equivalent damping factor and M_f is the steering torque. From equations (4.19) to (4.21) the complete dynamics of the model are rewritten in the bilinear form as:

$$\dot{\mathbf{z}} = \mathbf{Cz} + \mathbf{Bu} + \delta_f\mathbf{N}_f\mathbf{u} + \mathbf{d} \tag{4.22}$$

Where the matrices and vector from equation (4.22) result from rearranging the equations (4.19) through (4.21) and are written as:

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -\frac{C_{\alpha f}+C_{\alpha r}}{m} & -\frac{C_{\alpha f}}{m} & \frac{C_{\alpha f}+C_{\alpha r}}{mU} & \frac{aC_{\alpha f}-bC_{\alpha r}}{mU} & 0 \\ 0 & -\frac{aC_{\alpha f}-bC_{\alpha r}}{J_{Zc}} & -\frac{aC_{\alpha f}}{J_{Zc}} & \frac{aC_{\alpha f}-bC_{\alpha r}}{J_{Zc}U} & \frac{a^2C_{\alpha f}+b^2C_{\alpha r}}{J_{Zc}U} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{b_f}{J_f} \end{bmatrix} \tag{4.23}$$

$$\mathbf{B}=\frac{d}{J_{Zc}}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{r_{w1}} & \frac{1}{r_{w2}} & \frac{1}{r_{w3}} & \frac{1}{r_{w4}} \\ \frac{J_{Zc}}{dJ_f} & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.24}$$

$$\mathbf{N}_f=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{mr_{w1}} & \frac{1}{mr_{w2}} & 0 & 0 \\ 0 & \frac{a}{J_{Zc}r_{w1}} & \frac{a}{J_{Zc}r_{w2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad , \quad \mathbf{d}=\frac{1}{R_t}\begin{bmatrix} 0 \\ 0 \\ 0 \\ \left(\frac{aC_{\alpha f}-bC_{\alpha r}}{m}-U^2\right) \\ \left(\frac{a^2C_{\alpha f}+b^2C_{\alpha r}}{J_{Zc}}\right) \\ 0 \end{bmatrix} \tag{4.25}$$

The state vector **z** and control vector **u** are given by:

$$\mathbf{z}=\begin{bmatrix} y_{ct} & \varepsilon_{ct} & \delta_f & \dot{y}_{ct} & \dot{\varepsilon}_{ct} & \dot{\delta}_f \end{bmatrix}^T \tag{4.26}$$

$$\mathbf{u}=\begin{bmatrix} M_f & M_{w1} & M_{w2} & M_{w3} & M_{w4} \end{bmatrix}^T \tag{4.27}$$

The bilinear model represented by equation (4.22) just includes steering via the angle δ_f but not traction or braking. These torques are presented in an updated version of the controller described in section 4.2.5.

4.2.2. Design of the controller

The design of the controller itself is based on the methodology proposed by Alleyne [27] for bilinear control. The continuous-time state-space regulator form of the system is deduced by simplifying equation (4.22), removing the vector **d**, which represents in fact the addictive disturbance:

$$\dot{\mathbf{z}}=\mathbf{Cz}+\mathbf{Bu}+\mathbf{Nu} \tag{4.28}$$

Matrix $\mathbf{N}\in\Re^{n\times m}$ includes the sum of the product of state variables by matrices such as $\mathbf{N}_f$, written as:

$$\mathbf{N}\equiv\sum_{j=1}^{n}z_j\mathbf{N}_j \tag{4.29}$$

In the controller of this work *j* is considered equal to 1, while $\mathbf{N}_j=\mathbf{N}_f$ and $z_j=\delta_f$. The initial values of **z** are required to initialize the controller model dynamics, being depicted as:

$$\mathbf{z}(0)=\mathbf{z}_0 \tag{4.30}$$

The state and control vectors, $\mathbf{z}\in\Re^n$ and $\mathbf{u}\in\Re^m$, are related by the state-feedback law:

$$\mathbf{u}=-\mathbf{K}_{FB}(\mathbf{z})\mathbf{z} \tag{4.31}$$

where $\mathbf{K}_{FB}$ is called the optimal gain and it is defined in such a way that the state-feedback law represented by equation (4.31) minimizes the performance index, *J*, given by:

$$J=\frac{1}{2}\int_{t_0}^{t_\infty}(\mathbf{z}^T\mathbf{Q}\mathbf{z}+\mathbf{u}^T\mathbf{R}\mathbf{u})dt \tag{4.32}$$

$\mathbf{Q}\in\Re^{n\times n}$ and $\mathbf{R}\in\Re^{m\times m}$ are the symmetric weighting matrices being positive semi-definite and definite, respectively. The values of both matrices are defined by the user and influence greatly the behavior of the controller by determining the relative importance of the energy associated to the state action, for the **Q** matrix, or to the control action, for the **R** matrix. Because there is no recommendation in the literature for the values of the coefficients for matrices **Q** and **R**, in this work only diagonal matrices are considered.

The optimal gain $\mathbf{K}_{FB}\in\Re^{m\times n}$ is calculated using the solution of the state-dependent form of the algebraic Riccati equation [28]:

$$\mathbf{C}^T\mathbf{S}(\mathbf{z})+\mathbf{S}(\mathbf{z})\mathbf{C}-\mathbf{S}(\mathbf{z})(\mathbf{B}+\mathbf{N})\mathbf{R}^{-1}(\mathbf{B}+\mathbf{N})^T\mathbf{S}(\mathbf{z})+\mathbf{Q}=\mathbf{0} \tag{4.33}$$

The relation between the solution of equation (4.33) **S**(**z**) and the optimal gain is written as:

$$\mathbf{K}_{FB}=\mathbf{R}^{-1}(\mathbf{B}+\mathbf{N})^T\mathbf{S}(\mathbf{z}) \tag{4.34}$$

The use of this approach requires the fulfillment of the following assumptions [27]:

a) The differential equation (4.28) has a solution defined on $[0,+\infty[$ for each admissible input **u** and $\mathbf{z}(\mathbf{t})\to\mathbf{0}$ as $t\to 0$, which means that the vehicle trajectory tends to follow the track reference trajectory shown in Figure 4.3;

b) The pair (**C**,**B**+**N**) in equation (4.28) is completely controllable and **z** stays in the controllability domain defined by:

$$\mathbf{Z}_c\equiv\left\{\mathbf{z}\in\Re^n\,|\,(\mathbf{C},\mathbf{B}+\mathbf{N})\ \ \mathit{controllable}\right\} \tag{4.35}$$

c) The full state vector **z** is always available for use in the control law.

Function lqr from Matlab provides the the gain matrix $\mathbf{K}_{FB}$ and the solution of the Riccati equation $\mathbf{S}(\mathbf{z})$ [28] being used here as an implementation of the control strategy described before.

4.2.3. Computer implementation and track reference

The control procedure is implemented in the Matlab programming environment together with the dynamic analysis program. At every integration step the dynamic analysis program calls the controller model described above and provides it with the integration time *t*, the vehicle position $\mathbf{q}_c$ as well as the vehicle velocities $\dot{\mathbf{q}}_c$. In turn, the controller provides the dynamic analysis program with the steering angle and its time derivatives.

When using the 3D program, DAP3D, the vehicle state-variables are the position and velocity vectors of the chassis of the vehicle, given by:

$$\mathbf{q}_c = \begin{bmatrix} x_c & y_c & z_c & e_{0_c} & e_{1_c} & e_{2_c} & e_{3_c} \end{bmatrix}^T \tag{4.36}$$

$$\dot{\mathbf{q}}_c = \begin{bmatrix} \dot{x}_c & \dot{y}_c & \dot{z}_c & \omega_{\xi_c} & \omega_{\eta_c} & \omega_{\zeta_c} \end{bmatrix}^T \tag{4.37}$$

with these vectors it is first necessary to find the vehicle rotation ε_c and rotation rate $\dot{\varepsilon}_c$, which are the rotation and rotation rate of the 3D vehicle multibody model about *z* axis. Considering equations (2.3) and (4.36) the transformation matrix $\mathbf{A}_c$ is seen as being built with Bryant angles, φ_1, φ_2 and φ_3, computed as [18]:

$$\left.\begin{aligned} \varphi_2 &= \arcsin(\mathbf{A}_{13_c}) \\ \varphi_1 &= \arcsin\left(-\frac{\mathbf{A}_{23_c}}{\cos(\varphi_2)}\right) \\ \varphi_3 &= \arctan\left(-\frac{\mathbf{A}_{12_c}}{\cos(\varphi_2)}, \frac{\mathbf{A}_{11_c}}{\cos(\varphi_2)}\right) \end{aligned}\right\} \mathit{for}\,|\varphi_2| < \frac{\pi}{2} \quad \mathit{and} \quad |\varphi_1| \le \frac{\pi}{2} \tag{4.38}$$

The angular orientation of the vehicle reduced model is approximately given by:

$$\varepsilon_c = \varphi_3 \tag{4.39}$$

Furthermore, the time derivatives of the Bryant angles are related to the angular velocities as:

$$\begin{Bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ \dot{\varphi}_3 \end{Bmatrix} = \frac{1}{\cos\varphi_2} \begin{bmatrix} \cos\varphi_3 & -\sin\varphi_3 & 0 \\ \sin\varphi_3\cos\varphi_2 & \cos\varphi_3\cos\varphi_2 & 0 \\ -\cos\varphi_3\sin\varphi_2 & \sin\varphi_3\sin\varphi_2 & \cos\varphi_2 \end{bmatrix} \begin{Bmatrix} \omega_{\xi_c} \\ \omega_{\eta_c} \\ \omega_{\zeta_c} \end{Bmatrix} \tag{4.40}$$

Therefore, the reduced model vehicle yaw rate is approximately given by:

$$\dot{\varepsilon}_c = \dot{\varphi}_3 \tag{4.41}$$

In the 2D dynamic analysis program no transformation is needed being these vectors written as:

$$\mathbf{q}_c = [x_c \quad y_c \quad \phi_c]^T \tag{4.42}$$

$$\dot{\mathbf{q}}_c = [\dot{x}_c \quad \dot{y}_c \quad \dot{\phi}_c]^T \tag{4.43}$$

and the reduced model vehicle rotation and rotation rate are directly written as:

$$\varepsilon_c = \phi_c \tag{4.44}$$

$$\dot{\varepsilon}_c = \dot{\phi}_c \tag{4.45}$$

The longitudinal and lateral velocities of the reduced model are also calculated being a transformation of coordinates performed in order to get the velocities with respect to the vehicle local reference frame *(ξηζ)c*, i.e., $\dot{x}_c^{'}$, $\dot{y}_c^{'}$ and $\dot{z}_c^{'}$, as:

$$\begin{bmatrix} \dot{x}_c^{'} \\ \dot{y}_c^{'} \\ \dot{z}_c^{'} \end{bmatrix} = \mathbf{A}^T \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{z}_c \end{bmatrix} \tag{4.46}$$

Then, the longitudinal and lateral velocities are simply:

$$U = \dot{x}_c^{'} \tag{4.47}$$

$$V = \dot{y}_c^{'} \tag{4.48}$$

For the 2D program the same approach is taken, with only two components of the velocity, and with the 2D transformation matrix **A** used.

The required vehicle trajectory deemed as track reference in Figure 4.3, is described as a collection of consecutive line segments, which may be either linear or curve arc segments. This path is built using an input file that contains the curvature k_i and length l_i information of each track segment for the total number of segments n_s.

$$\mathbf{T} = [\mathbf{k}_t \quad \mathbf{l}_t] \tag{4.49}$$

Considering a collection of n_p points on the path centerline obtained from equation (4.49), the coordinates of each point P_j in the path centerline is defined as function of parameter *s* as:

$$(x_{tj}, y_{tj}) = (x(s_j), y(s_j)), \qquad j = 1,..., n_p \tag{4.50}$$

where the parameter s_j is the total length of the arc of the curve until the point P_j, i.e. the arc length from the origin of the path to the point. In order to have a continuous description of the track reference, the n_p points are interpolated by using a cubic spline $({}^{sp}x_t(s), {}^{sp}y_t(s))$, in which the cubic spline interpolates satisfy:

$$({}^{sp}x_t(s_j), {}^{sp}y_t(s_j)) = (x_j, y_j), \qquad j = 1, \ldots, n_p \tag{4.51}$$

The Matlab `spline` function is used to implement the cubic spline in the controller.

The track reference angle for any given point is given by:

$$\varepsilon_t(s) = \arctan\left(\frac{\delta^{sp}y_t(s)}{\delta_s}, \frac{\delta^{sp}x_t(s)}{\delta_s}\right) \tag{4.52}$$

While the curvature of the track reference, at a given point, is provided by the relation:

$$k_t(s) = \frac{1}{R_t(s)} = \frac{{}^{sp}x_t^{'}(s){}^{sp}y_t^{''}(s) - {}^{sp}y_t^{'}(s){}^{sp}x_t^{''}(s)}{\left(\sqrt{({}^{sp}x_t^{'}(s))^2 + ({}^{sp}y_t^{'}(s))^2}\right)} \tag{4.53}$$

where ${}^{sp}y_t^{'}(s)$ and ${}^{sp}y_t^{''}(s)$ are the first and second parametric derivatives of the parametric curve, being written as:

$${}^{sp}y_t^{'}(s) = \frac{\partial^{sp}y_t(s)}{\partial s} \qquad ; \qquad {}^{sp}y_t^{''}(s) = \frac{\partial^{2sp}y_t(s)}{\partial^2 s} \tag{4.54}$$

The track curvature $k_t(s)$ is the inverse of the track radius $R_t(s)$ at the point defined by s.

Since the control objective is to keep the vehicle in the track reference, and knowing that the vehicle center of mass O_c may not always coincide with the track reference, it is necessary to find the track reference point O_t, shown in Figure 4.3, which is the point where the vehicle should be if it had followed the track reference perfectly. Note that the point O_t is the closest track reference point to the vehicle position O_c. This is essential in order to calculate the distance y_{ct}. The track reference point is determined by finding the parameter s that minimizes the distance $\overline{P(s)O_c}$:

$$O_t \equiv P(s_t) \wedge s_t = s : \min\sqrt{(x_c - {}^{sp}x(s)_t)^2 + (y_c - {}^{sp}y(s)_t)^2} \tag{4.55}$$

where s_t is the parameter that minimizes the distance function, using the Matlab function `fminbd`, which is based on the quadratic fit method and the golden section search proposed by Atkinson [29]. Hence, the O_t point coordinates are given by:

$$(x_t, y_t) \equiv ({}^{sp}x(s_t)_t, {}^{sp}y(s_t)_t) \tag{4.56}$$

Generally there is no assurance of the uniqueness of the solution of the track reference point position identified. However, the assumption that $R_t >> y_{ct}$ used as an approximation in equation (4.17) ensures that the O_t point found is unique.

In order to prevent a steady state controller error Alleyne [24] suggest to redefine the dynamics of the system by adding the integral of the lateral path error y_{ct} as a new state:

$$\bar{\mathbf{z}} = \begin{bmatrix} \int y_{ct} & y_{ct} & \varepsilon_{ct} & \delta_f & \dot{y}_{ct} & \dot{\varepsilon}_{ct} & \dot{\delta}_f \end{bmatrix}^T \tag{4.57}$$

Hence, the continuous-time state-space regulator needs to be redefined and is now written as:

$$\dot{\bar{\mathbf{z}}} = \bar{\mathbf{C}}\bar{\mathbf{z}} + [\bar{\mathbf{B}} + \delta_f \bar{\mathbf{N}}_f]\bar{\mathbf{u}} + \bar{\mathbf{d}} \tag{4.58}$$

The time derivative of the new state vector is now given by:

$$\dot{\bar{\mathbf{z}}} = \begin{bmatrix} y_{ct} & \dot{y}_{ct} & \dot{\varepsilon}_{ct} & \dot{\delta}_f & \ddot{y}_{ct} & \ddot{\varepsilon}_{ct} & \ddot{\delta}_f \end{bmatrix}^T \tag{4.59}$$

The control vector remains the same as the one defined by equation (4.27) and it is re-written as:

$$\bar{\mathbf{u}} = \begin{bmatrix} M_f & M_{w1}^b & M_{w2}^b & M_{w3}^b & M_{w4}^b \end{bmatrix}^T \tag{4.60}$$

The new matrices $\bar{\mathbf{C}}$, $\bar{\mathbf{B}}$, $\bar{\mathbf{N}}_f$ and **d** are based on equations from (4.23) to (4.25) but with null rows added in the top of all of them and a null column inserted in the left of matrix **C**. The only non-null coefficient added is $\bar{C}_{12} = 1$.

The computer algorithm for the controller has two different phases: the first one is the initialization and the second one is for every time step while the vehicle is moving. The detailed computer algorithm is as:

Initialization:

Step 1: Read the track centerline data $\mathbf{T}_t$, from the input file, in the form of equation (4.49) and generate track reference points (x_{tj},y_{tj}) and angles ε_{tj} for $j=1,\ldots,n_p$;

Step 2: Initialize the steering dynamics by defining $\delta_f=\delta_f(0)$ and $\dot{\delta}_f = \dot{\delta}_f(0)$;

For every time step:

Step 3: Get positions $\mathbf{q}_c$ and velocities $\dot{\mathbf{q}}_c$ from the dynamics analysis program;

Step 4: Calculate the position (x_c,y_c) and orientation ε_c of the reduced model by using equations (4.36), (4.38) and (4.39) for 3D and equations (4.42) and (4.44) for 2D;

Step 5: Calculate the vehicle velocities (U,V) and angular velocity $\dot{\varepsilon}_c$ of the reduced model by using equations (4.37), (4.40), (4.41) and from (4.46) to (4.48) for 3D. For the 2D program equations (4.43), (4.45), (4.47) and (4.48) are used instead.

Step 6: Find the track reference point O_t by finding the parameter s that minimizes the distance between O_c and the track reference, using equation (4.55);

Step 7: Find the O_t point coordinates, the path tangent angle ε_t and the track curvature k_t by using equations from (4.51) to (4.53);

Step 8: Calculate the state variables ε_{ct}, $\dot{\varepsilon}_{ct}$ and $\dot{y}_{ct}$ by using equations from (4.14) to (4.18). The state variable y_{ct} is the minimum distance given by equation (4.55);

Step 9: Integrate the time derivatives of the steering angle of the previous step, $\dot{\delta}_f$ and $\ddot{\delta}_f$ and the vehicle lateral deviation y_{ct} to get the state variables δ_f, $\dot{\delta}_f$ and $\int y_{ct}$ of the current time step;

Step 10: Build the state-variable vector $\bar{\mathbf{z}}$ by using equation (4.57) and the matrices $\bar{\mathbf{C}}$, $\bar{\mathbf{B}}$ and $\bar{\mathbf{N}}_f$ as specified by equations (4.23) to (4.25) with the modifications needed to accommodate the redefined state-variable vector;

Step 11: Calculate the optimal gain $\mathbf{K}_{FB}$, using equation (4.34), and evaluate the control vector $\mathbf{u}$ by using equation (4.31), but using matrices $\bar{\mathbf{C}}$, $\bar{\mathbf{B}}$ and $\bar{\mathbf{N}}_f$, instead of $\mathbf{C}$, $\mathbf{B}$ and $\mathbf{N}_f$;

Step 12: Find the time derivative of the state-variable vector $\dot{\bar{\mathbf{z}}}$ by using equation (4.59);

Step 13: Exit Matlab and return to the dynamics analysis program with the new steering angle and its time derivatives δ_f, $\dot{\delta}_f$ and $\ddot{\delta}_f$ and the new wheel torques M_{w1}, M_{w2}, M_{w3} and M_{w4} to be applied in the multibody model;

In step 9 the controller needs to provide the steering angle and its first and second time derivatives. However, the control theory only provides the second time derivative. Hence, to get the steering angle and its first time derivative, the second derivative has to be integrated in time. This integration is done inside the controller and for that purpose the variable step size Adams-Bashford integration method is applied [30], because the integration is controlled by the dynamics analysis program integrator variable time steps. The Antos and Ambrósio controller [1], which is the base of the controller used in this work, is now described.

4.2.4. Preview distance implementation

The preview distance allows the vehicle to follow smoother trajectories, especially when it is off-track, which leads to a more stable behavior. Furthermore, it reduces the effect of the track curvature discontinuities as the vehicle lateral and angular errors reach maximum values when there is a change of the track curvature when using the controller without preview distance.

The controller design remains the same as explained in section 4.2.2. However, there are some changes in the trajectory to follow. Instead of following the trajectory provided by the input file, new trajectories are computed as the car moves on.

To compute the preview trajectory, first of all it is necessary to have the vehicle present coordinates and the vehicle rotation angle, *(x_c,y_c)* and ε_c. Secondly O_t is found, the track reference point which is the closest point in the track reference to the present position, defined by O_c, by finding the parameter s_t that minimizes the distance between O_c and the track reference. Using the s_t parameter from equation (4.55), it is possible to define the preview point O_p coordinates which are given by:

$$(x_p, y_p) \equiv ({}^{sp}x(s_t + l_p)_t, {}^{sp}y(s_t + l_p)_t) \tag{4.61}$$

where l_p is the preview distance. Furthermore the track tangent angle at the preview point ε_p is given by:

$$\varepsilon_p = \arctan\left(\frac{\partial^{sp} y_t(s_t + l_p)}{\partial s}, \frac{\partial^{sp} x_t(s_t + l_p)}{\partial s}\right) \tag{4.62}$$

The vehicle and preview coordinates and angles are transformed into a new reference frame which is coincident with the car reference frame and has its origin at O_c. This prevents the tangent of the angles to assume very high values.

The preview trajectory is built using a cubic polynomial. In order to construct the trajectory, the system of equations (4.2) has to be solved with:

$$y'|_{x=x_c} = \tan(\varepsilon_c) \qquad ; \qquad y'|_{x=x_p} = \tan(\varepsilon_p) \tag{4.63}$$

The solution of the equation (4.2) gives the constants that allow the construction of the cubic polynomial from equation (4.1), which defines the preview trajectory curve. A mesh of n_{pp} points between $O_c{}'$ and $O_p{}'$, i.e., the car and preview points defined in the car reference frame, is then defined from the polynomial function, being their coordinates are transformed to the global reference frame *(xyz)*. The coordinates of each point P_{ptj} in the preview trajectory as function of parameter s are given by:

$$(x_{ptj}, y_{ptj}) = (x(s_{pj}), y(s_{pj})), \qquad j = 1,...,n_{pp} \tag{4.64}$$

Similarly to the procedure adopted to the track reference, the points of the preview trajectory are also interpolated by a cubic spline *(${}^{sp}x_{pt}(s)$, ${}^{sp}y_{pt}(s)$)*. The cubic spline interpolation satisfies:

$$({}^{sp}x_{pt}(s), {}^{sp}y_{pt}(s)) = (x_{ptj,} y_{ptj}), \qquad j = 1,...,n_{pp} \tag{4.65}$$

The preview trajectory tangent angle for any given point is written as:

$$\varepsilon_{pt}(s) = \arctan\left(\frac{\delta^{sp} y_{pt}(s)}{\delta_s}, \frac{\delta^{sp} x_{pt}(s)}{\delta_s}\right) \tag{4.66}$$

The curvature of the preview trajectory, at a given point, is provided by the relation:

$$k_{pt}(s)=\frac{1}{R_{pt}(s)}=\frac{{}^{sp}x_{pt}^{'}(s)^{sp}y_{pt}^{"}(s)-^{sp}y_{pt}^{'}(s)^{sp}x_{pt}^{"}(s)}{\left(\sqrt{({}^{sp}x_{pt}^{'}(s))^{2}+({}^{sp}y_{pt}^{'}(s))^{2}}\right)}s \tag{4.67}$$

Where ${}^{sp}y_{pt}^{'}(s)$ and ${}^{sp}y_{pt}^{''}(s)$ are the first and second parametric derivatives of the parametric curve being written as:

$${}^{sp}y_{pt}^{'}(s)=\frac{\partial^{sp}y_{pt}(s)}{\partial s} \quad ; \quad {}^{sp}y_{pt}^{"}(s)=\frac{\partial^{2sp}y_{pt}(s)}{\partial^{2}s} \tag{4.68}$$

The track curvature $k_{pt}(s)$ is the inverse of the trajectory radius $R_{pt}(s)$ at the point defined by s.

Having the preview trajectory, the controller uses it as a reference track and ensures that the vehicle to follow it until a new trajectory is computed. The preview distance may vary regarding the vehicle type and track geometry and it has to be predefined. Likewise, the update distance l_{pu} also need to be provided. The update distance is a parameter that defines when a new preview trajectory has to be computed being its value defined in between 0 and l_p. A new preview trajectory is computed when the current parameter s reaches the value of the s_t parameter at the beginning of the current preview trajectory, represented by s_{ti}, plus the update distance. Hence, the next condition has to be fulfilled:

$$s_t = s_{ti} + l_{pu} \tag{4.69}$$

Figure 4.4 permits a better understating of the preview update distance. It shows that a new preview trajectory is created when the preview update distance is reached.

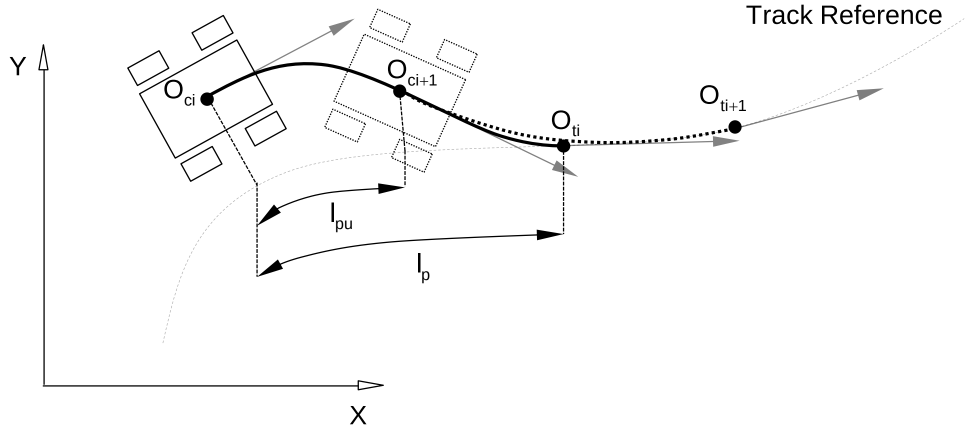


Figure 4.4 - Schematic representation of the preview and preview update distances

Figure 4.5 shows a representation of the controller algorithm with the preview approach. An analogy with the previous controller algorithm is made as all the steps mentioned are from the algorithm described in 4.2.3.

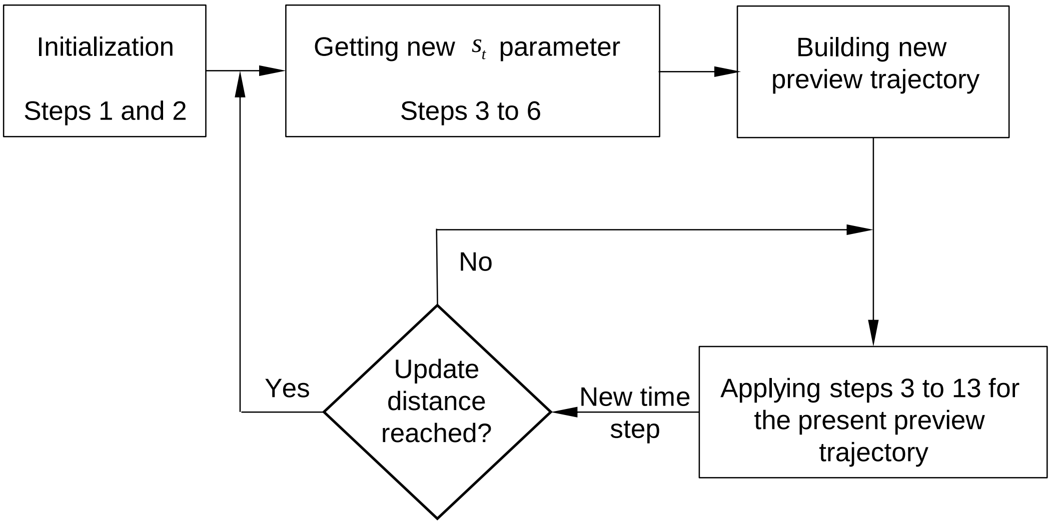

Initialization
Steps 1 and 2
Getting new s_t parameter
Steps 3 to 6
Building new preview trajectory
No
Yes
Update distance reached?
New time step
Applying steps 3 to 13 for the present preview trajectory

Figure 4.5 - Schematic representation of the controller algorithm with previewing

4.2.5. Controller with speed profile following

Besides the control of the trajectory it is also important to control the vehicle velocity, called here as the speed profile following feature. Unlike the previewing method described in the previous section, the speed profile controller changes de design of the controller by Antos and Ambrósio [1], even if it continues to be based that.

To perform the best time in a race track there is the need to accelerate and brake de car, which leads to the controller which ensures that the vehicle follows a predefined speed profile for the longitudinal velocity of the vehicle. During the initialization process of the controller it reads a speed profile, which is given by a cubic spline, similarly to what happens with the trajectories. The spline is also function of the parameter *s*.

At every time step the controller reads the longitudinal velocity v_t, from the speed profile. A new state variable is added to the state-variables vector $\bar{\mathbf{z}}$, which is now written as:

$$\bar{\mathbf{z}} = \left[\int y_{ct} \quad y_{ct} \quad \varepsilon_{ct} \quad \delta_f \quad \dot{y}_{ct} \quad \dot{\varepsilon}_{ct} \quad \dot{\delta}_f \quad v_{ct} \right]^T \tag{4.70}$$

While the time derivative of the new state vector is given by:

$$\dot{\bar{\mathbf{z}}} = \left[y_{ct} \quad \dot{y}_{ct} \quad \dot{\varepsilon}_{ct} \quad \dot{\delta}_f \quad \ddot{y}_{ct} \quad \ddot{\varepsilon}_{ct} \quad \ddot{\delta}_f \quad \dot{v}_{ct} \right]^T \tag{4.71}$$

where v_{ct} is the velocity error between the car present longitudinal velocity and the expected velocity by the speed profile, whilst $\dot{v}_{ct}$ is its time derivative.

$$v_{ct} = v_t - U \tag{4.72}$$

$$\dot{v}_{ct} = \dot{v}_t - \dot{U} \tag{4.73}$$

Considering the equation of motion:

$$m(\dot{v}_t - \dot{v}_{ct}) = \sum_{i=1}^{4} F_{xci} \tag{4.74}$$

Taking into account equation (4.74) the velocity error time derivative is approximated to:

$$\dot{v}_{ct} = -\left(\frac{M_{w1}}{mr_{w1}} + \frac{M_{w2}}{mr_{w2}} + \frac{M_{w3}}{mr_{w3}} + \frac{M_{w4}}{mr_{w4}} \right) \tag{4.75}$$

The matrices $\bar{\mathbf{C}}$, $\bar{\mathbf{B}}$ and $\bar{\mathbf{N}}_f$ of the updated controller are now defined as:

$$\bar{\mathbf{C}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{C_{\alpha f} + C_{\alpha r}}{m} & -\frac{C_{\alpha f}}{m} & \frac{C_{\alpha f} + C_{\alpha r}}{mU} & \frac{aC_{\alpha f} - bC_{\alpha r}}{mU} & 0 & 0 \\ 0 & 0 & -\frac{aC_{\alpha f} - bC_{\alpha r}}{J_{Zc}} & -\frac{aC_{\alpha f}}{J_{Zc}} & \frac{aC_{\alpha f} - bC_{\alpha r}}{J_{Zc}U} & \frac{a^2C_{\alpha f} + b^2C_{\alpha r}}{J_{Zc}U} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{b_f}{J_f} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.76}$$

$$\bar{\mathbf{B}} = \frac{d}{J_{Zc}} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{r_{w1}} & \frac{1}{r_{w2}} & \frac{1}{r_{w3}} & \frac{1}{r_{w4}} \\ \frac{J_{Zc}}{dI_f} & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{mr_{w1}} & -\frac{1}{mr_{w2}} & -\frac{1}{mr_{w3}} & -\frac{1}{mr_{w4}} \end{bmatrix} \tag{4.77}$$

$$\bar{\mathbf{N}}_f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \dfrac{1}{mr_{w1}} & \dfrac{1}{mr_{w2}} & 0 & 0 \\ 0 & \dfrac{a}{J_{Zc}r_{w1}} & \dfrac{a}{J_{Zc}r_{w2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.78}$$

The new, and final, version of the controller is applied to a demonstration case in which a vehicle has to go around a track for which the speed profile is defined. It is left for chapter 5 the definition of the speed profile for the optimal lap time track trajectory.

4.3. Demonstration cases

Three different scenarios are used to demonstrate the three different versions of the controller: the standard version of the controller proposed by Antos and Ambrósio [1], that guides the vehicle through an entire lap; the controller that includes the previewing; the controller which includes previewing and speed profile following.

For the two first demonstration cases the controller setting, with respect to the weighting matrices is given by:

$$\mathbf{Q} = diag([1 \quad 50000 \quad 10000 \quad 200 \quad 1 \quad 1 \quad 1]) \tag{4.79}$$

$$\mathbf{R} = ([50 \quad 1 \quad 1 \quad 1 \quad 1]) \tag{4.80}$$

For the speed profile demonstration case, the following weighting matrices are used:

$$\mathbf{Q} = \left([1 \quad 50000 \quad 10000 \quad 200 \quad 1 \quad 1 \quad 1 \quad 200000]\right) \tag{4.81}$$

$$\mathbf{R} = ([50 \quad 1 \quad 1 \quad 1 \quad 1]) \tag{4.82}$$

Furthermore, the steering dynamics applied to all the cases are such that:

$$J_f = 0.3 \quad , \quad b_f = 1 \tag{4.83}$$

The same trajectory, consisting in a closed race track is used for all demonstration cases. The track trajectory is not optimized here to be the shortest trajectory or the least curvature trajectory.

4.3.1. Lancia Stratos performing an entire lap

The first controller demonstration case uses the standard controller, proposed by Antos and Ambrósio [1], with the vehicle following the reference track without preview distance or speed control. The vehicle is intended to follow the centerline of an imaginary track, represented in Figure 4.6.

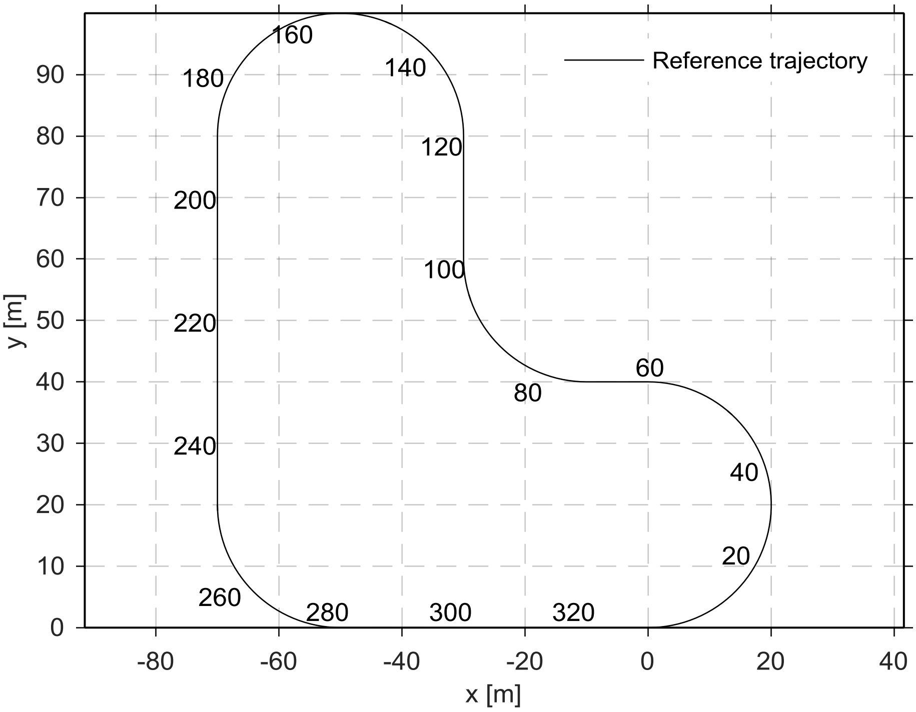


Figure 4.6 - Track centerline used for the demonstration cases, with the track arc length shown at the side of the trajectory

The values displayed alongside the reference trajectory represent the total curvature length, i.e., the parameter *s*, in meters. The track input file used by the 2D program for this case has the following geometry information:

Table 4.1 - Track properties for the demonstration cases

Segment number, *j*	Segment radius of curvature, R_t [m]	Segment length, l_t [m]
1	20	62.83
2	0	10.00
3	-20	31.42
4	0	20.00
5	20	62.83
6	0	60.00
7	20	31.42
8	0	50.00

The positive and negative values of the segment radius of curvature differentiates the left and right curves. The segment curvature is found by:

$$k_j = \frac{1}{R_j} \tag{4.84}$$

The input information is then rearranged and placed in the form of equation (4.49). The total track centerline length is approximately 328.5 m, the vehicle initial velocity is 7.5 m/s, which is kept almost constant during the motion. The initial position of the car center of mass is placed at the origin of the global reference frame, which coincides with the track starting point. The simulation performed has a total time of 60 seconds which allows the vehicle to run a little bit over one entire lap.

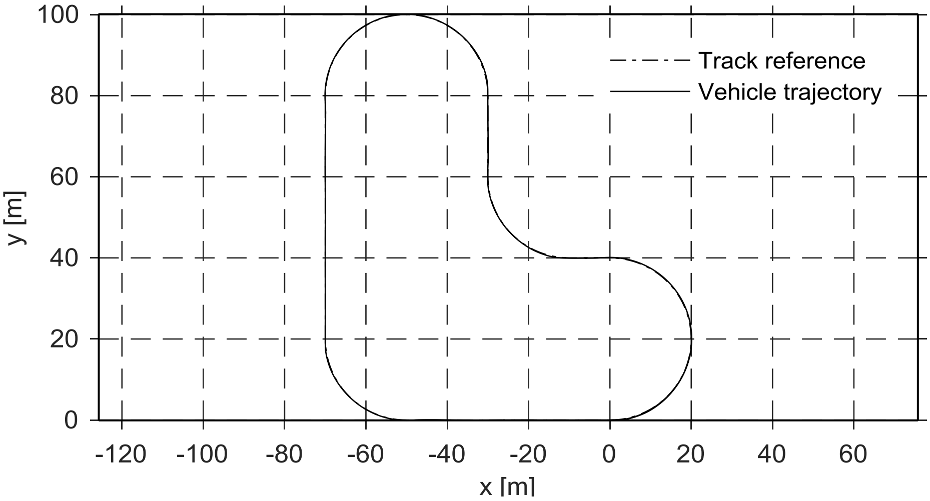


Figure 4.7 - Vehicle trajectory to be tracked by the controller during a complete lap

Figure 4.7 shows the vehicle trajectory while performing an entire lap using the controller. It suggests that the vehicle follows adequately the track reference. Nevertheless, it is needed to analyze a segment of the track to have a more detailed idea if the car is correctly following the track reference.

In fact, Figure 4.9 shows that the vehicle does not follow the trajectory perfectly. A small segment of the track at the begging of a right hand corner shows small differences between the targeted and real vehicle trajectories. It is clear that the vehicle is following the exact target trajectory before the curve starts. However, when it starts the turning maneuver the vehicle is not able to keep following the trajectory without error which is, nevertheless, corrected. The controller has the ability to achieve a steady state and bring the car back to the trajectory after a few meters. Even if the controller can be improved, the results are well aligned with its objectives.

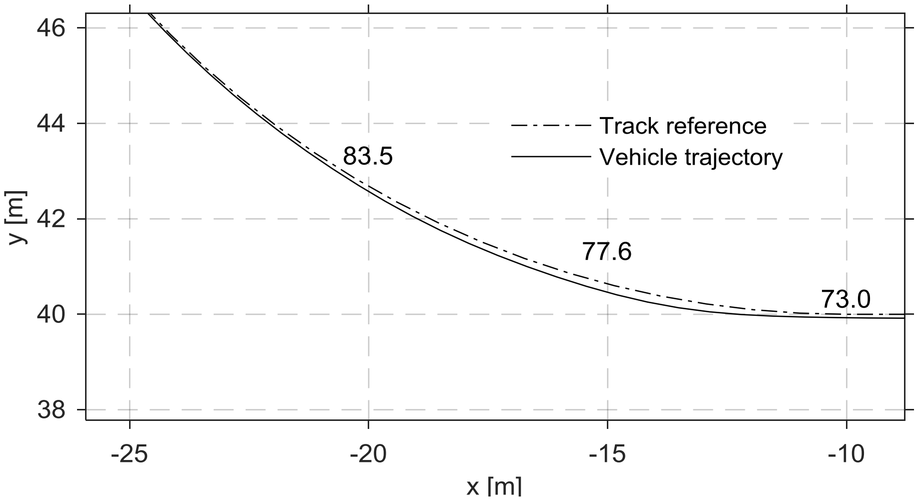


Figure 4.9 - Detail of a track segment for the demonstration case

Figure 4.8 represents the steering angle as function of time during the entire lap, or as a function of the arc length parameter *s* to better allow the understanding of the results and their relation to the trajectory geometry. These results show that the controller is able to find a steady state, even if some time is needed. The change of curvature induces some oscillations in the system, which are damped in a reasonable short period of time, as it can be seen in the steering angle graph. This is due to an overshooting of the steering angle when starting a segment with a different curvature. This overshooting may be due to excessive tyre lateral slip angle or due to the controller action. Moreover the variations of steering angle can be easily related to the changes of the track curvature, by analyzing the reference trajectory in Figure 4.6.

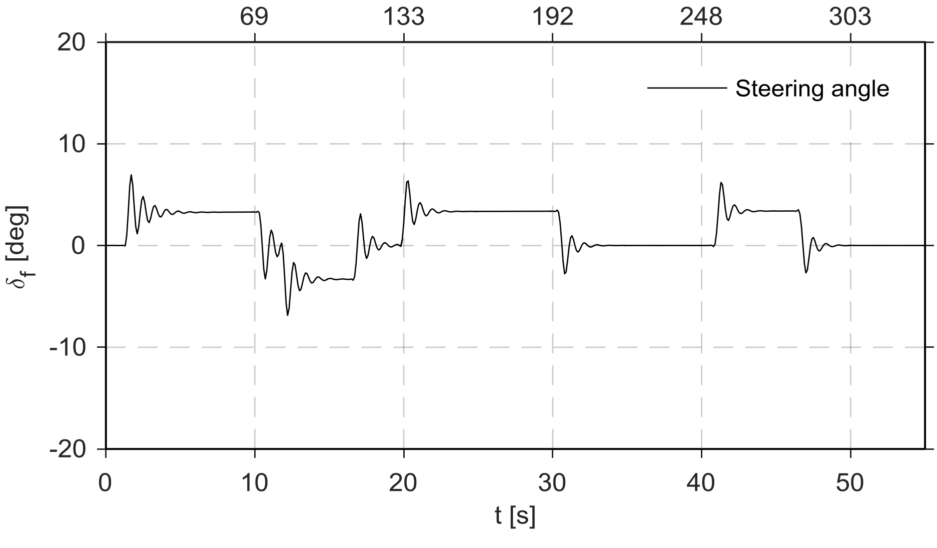


Figure 4.8 - Steering angle during the entire lap demonstration case

Figure 4.10 depicts the error in the lateral position of the vehicle along the track. The controller tries to minimize the lateral error which is attained when it reaches a steady state. Due to the several changes in the trajectory curvature, the control steering overshooting of the tyre lateral slippage never achieves to get the error to zero. Nevertheless the maximum lateral error for this demonstration case was of 0.18 meters, which is acceptable.

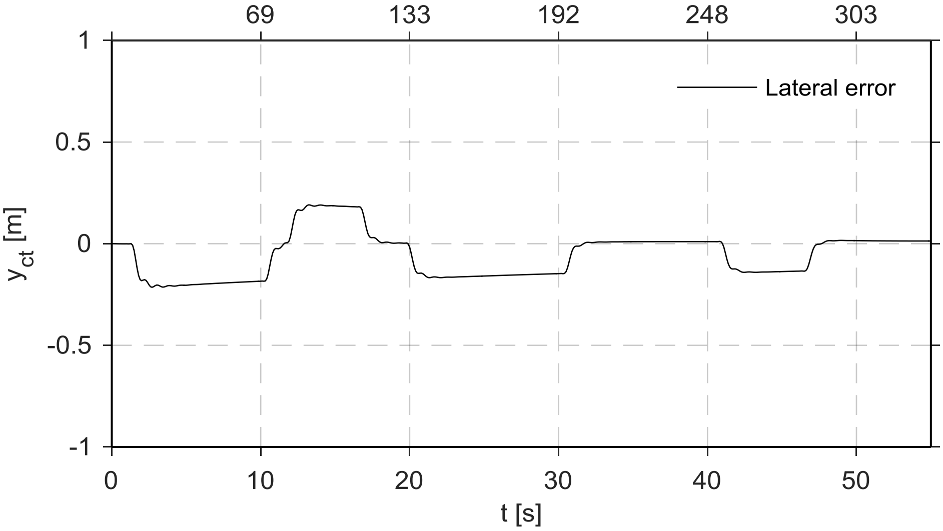


Figure 4.10 - Lateral relative position between the vehicle and the reference trajectory

As observed in Figure 4.11, the angular orientation of the vehicle exhibits the same general behavior of the lateral position error being the maximum angular error equal to 4.1 deg.

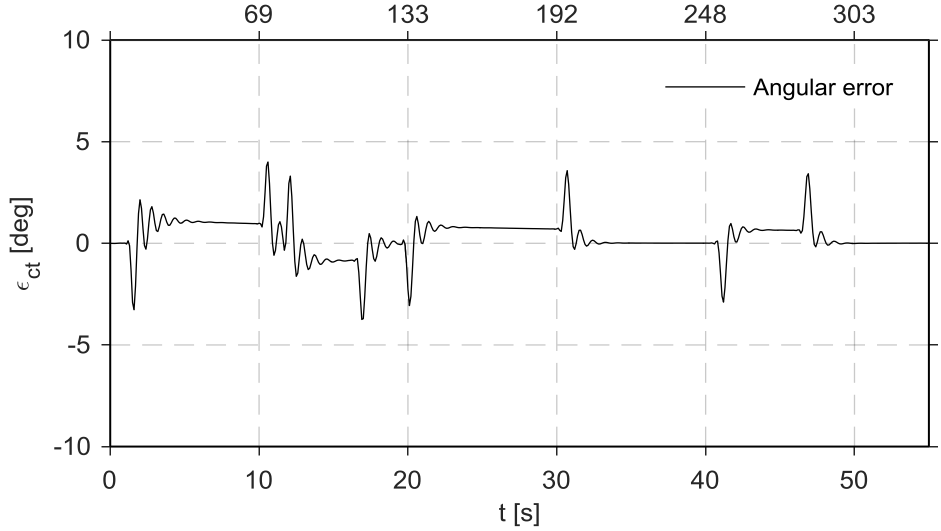


Figure 4.11 - Angular relative position between the vehicle and the reference trajectory

4.3.2. Demonstration case with previewing

This second demonstration case is meant to show how the controller behaves when using the preview distance. Firstly a simulation with the car starting off-track is presented to show the ability to get back to the track. Afterwards, in a second case the car performs all the track starting at the origin. This second case allows to compare the vehicle behavior when controlled with and without the previewing. The track centerline is defined using the same properties shown in Table 4.1.

In this first demonstration case the vehicle center of mass initial position is set off-track, i.e., instead of being placed at the origin, the car is moved 2 meters away in the *y* direction. The car initial velocity is also 7.5 m/s.

As shown in Figure 4.12 the vehicle initial position is off-track being a new preview trajectory from the vehicle starting point to a point in the prescribed trajectory at a preview distance. The vehicle tries to follow the preview trajectory instead of going directly to the reference track. In this simulation the preview and update distances used were respectively 15 m and 10 m. When the car reaches the update distance, a new trajectory is computed. However, the second trajectory is not represented in Figure 4.12 for the sake of having a clear graphic visualization.

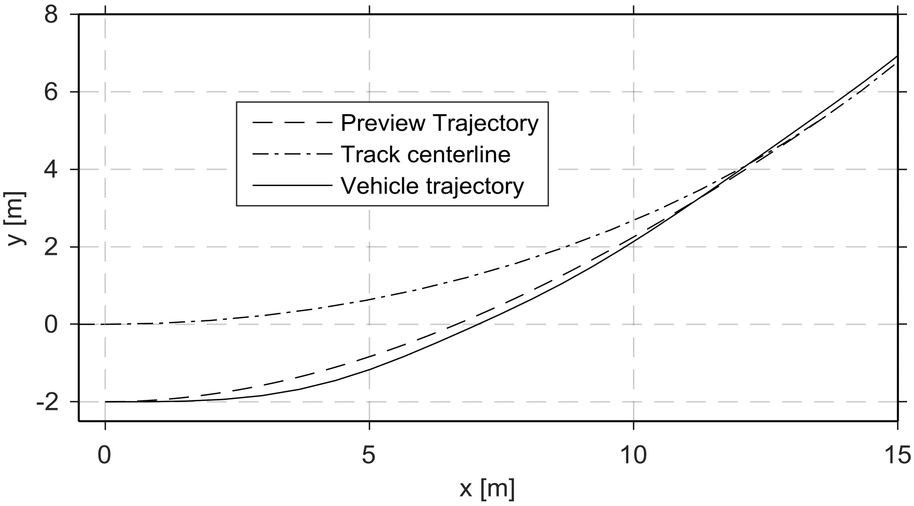


Figure 4.12 - Demonstration case for the vehicle control with previewing. The vehicle starts off-track.

Figure 4.13 shows a case in which the vehicle has the same initial conditions as the previous case, but where the controller has no previewing and, therefore, no preview trajectory is computed which leads the vehicle to have a more aggressive behavior towards the predefined trajectory. The controller with the preview trajectory promotes, as expected, a smoother vehicle path that seems closer to what a human driver would attempt.

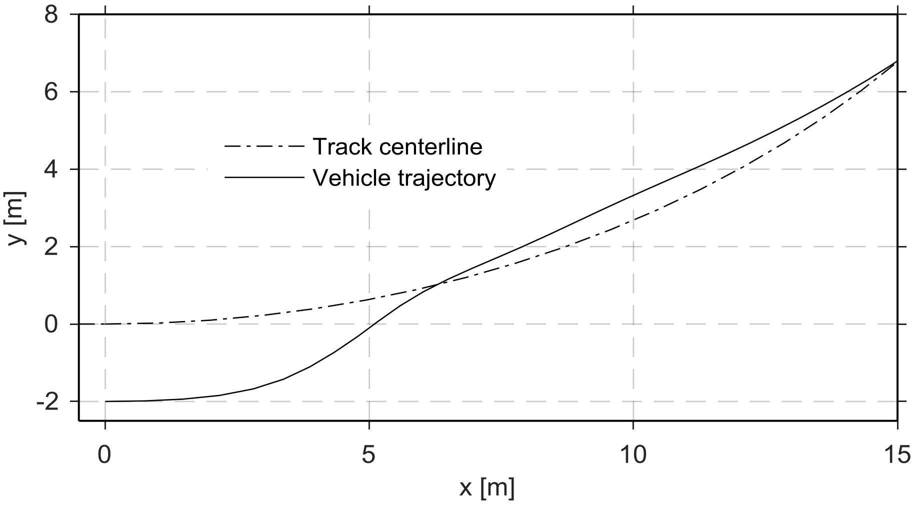


Figure 4.13 - Demonstration case with the vehicle starting off-track without previewing

The demonstrative case for the vehicle performing an entire lap with the previewing controller implemented is also analyzed here. Some comparisons between results, for controller with and without previewing, are presented. Similarly to what happens in the demonstration case for the controller without preview, the vehicle trajectory is very close to the predefined trajectory, as observed in Figure 4.14. It is clear that the vehicle runs the entire track following close to the reference trajectory but, as for the controller without previewing, a closer look on the results is required.

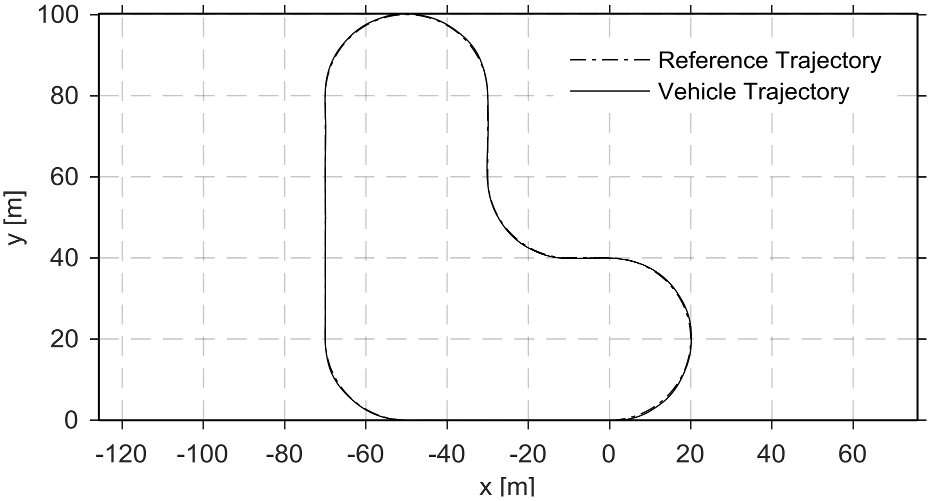


Figure 4.14 - Vehicle trajectory for the entire lap demonstration case with preview

Figure 4.15 shows that, using the controller with preview, the relative distance between the vehicle’s center of mass and the reference trajectory is different from zero during the majority of the lap. However, the controller is still able to lead the vehicle to follow the trajectory with this associated error being very small. As the vehicle has a new trajectory to follow every time it reaches the preview update distance, it means that the curvature is constantly changing and the controller has greater difficulty in leading the vehicle to reach a steady state. Using greater preview and update distances could minimize the error, as the controller would have more time to stabilize. However, greater preview distances could bring new issues. On one hand, it would became unreal, as usually a driver cannot see much beyond the curve entry, for instance. On the other hand, when the vehicle is off-track it would take too much time for the vehicle return to the reference trajectory. Moreover, and also very important, a large preview distance could lead to new preview trajectories that may lead the vehicle to go out of the track limits.

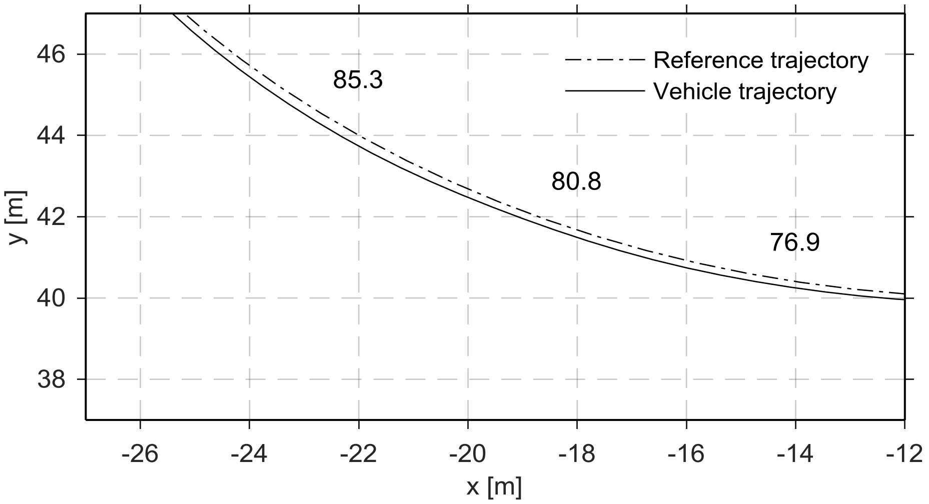


Figure 4.15 - Detailed view for a part of the trajectory of the vehicle with a controller with preview

As discussed before, the fact that a new trajectory is computed constantly does not allow the vehicle to reach a steady state and the oscillations cannot be fully damped, unless the curvature is kept constant for a long time. Figure 4.16 depicts the steering angle. It seems that the steering angle changes are associated to instants in time in which a new preview trajectory is being followed.

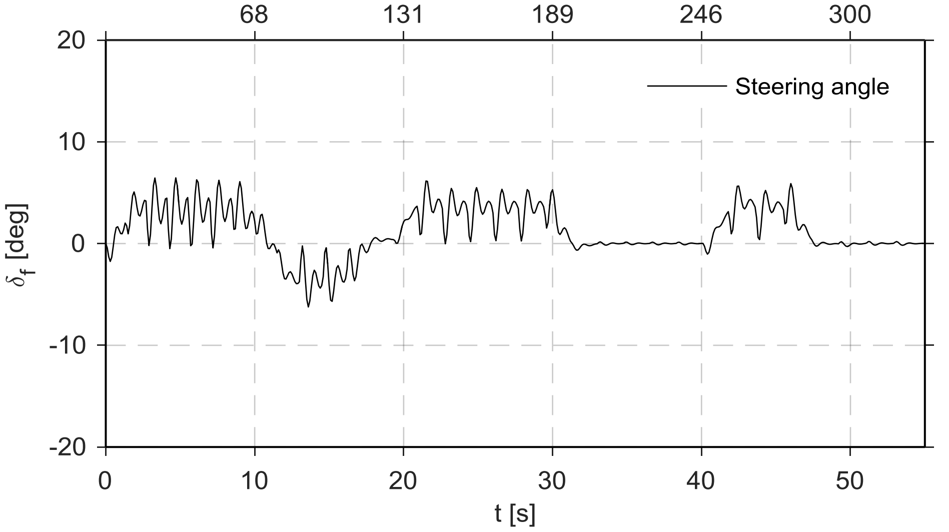


Figure 4.16 - Steering angle along the entire lap for the vehicle with the controller with preview

The relative lateral distance between the vehicle and the reference track are shown in Figure 4.17. The controller tries to minimize the distance error, but the continuous curvature change makes the task difficult. Nevertheless, the maximum lateral error is 0.21 meters, which is not much larger than the lateral error for the controller without preview.

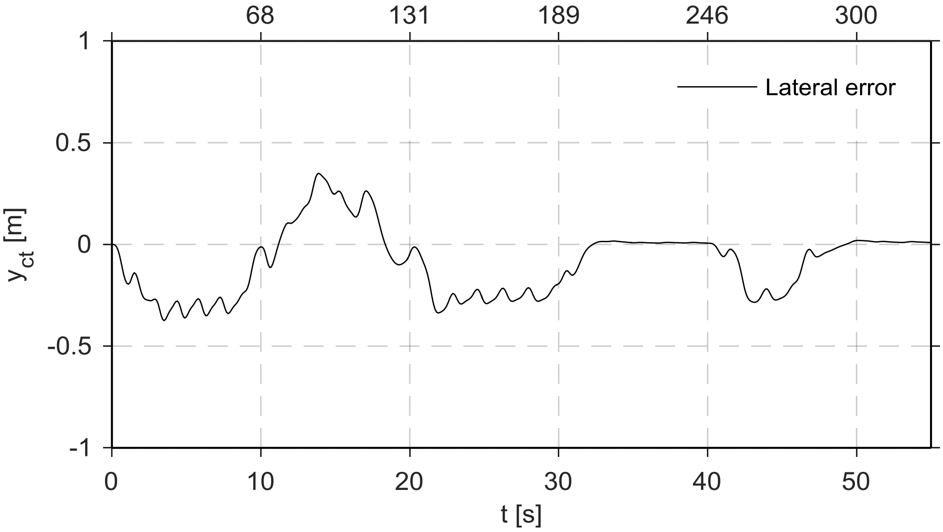


Figure 4.17 - Relative lateral distance between the vehicle and the reference trajectory

Figure 4.18 shows the angular error between the vehicle orientation and the reference trajectory tangent. The maximum angular error is 3.84 degrees, which is lower than the maximum error achieved in the case without preview. Still, the oscillations observed in Figure 4.18, as already observed in FiguresFigure 4.15, Figure 4.16 and Figure 4.17 can be explained either by an hyperactivity of the controller or by excessive slippage of the vehicle tyres, that the controller attempts to correct.

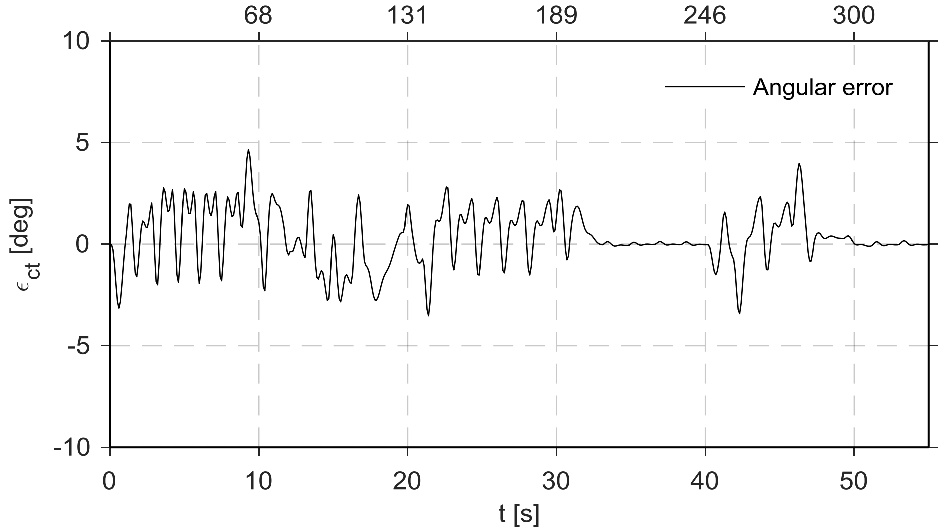


Figure 4.18 - Relative angular position between the vehicle heading and the reference trajectory

4.3.3. Demonstration case with speed control

In this demonstration case the controller is used with the speed control function, which ensures that the vehicle follows not only the trajectory but also a predefined speed profile. Considering the same track as in the previous demonstrations, a speed profile has to be computed and given as an input to the controller. The speed profile is constructed as a cubic spline interpolation, using the `spline` Matlab function, similarly to what happens with the trajectories, which means that a mesh of points is provided and interpolated afterwards. Likewise the speed cubic spline is also given as function of the curvature arc length parameter *s*.

For this demonstration case the vehicle initial velocity is 7.5 m/s and its initial position is placed at the origin, which is also coincident with the reference trajectory. Moreover, the speed profile is computed using the information provided in Table 4.2 in order to create the mesh of points. In this section 5.4 the construction of the speed profile for optimal time lap is discussed.

Table 4.2 - Information regarding the speed profile for the demonstration case

Point	Velocity [m/s]	*s* [m]
1	7.5	0
2	5.0	25
3	5.0	55
4	8.0	85
5	8.0	110
6	6.0	140

This demonstration case has a total simulation time of 22 seconds. As the objective is to analyze the controller speed following property, there is no need to perform the simulation on the entire track. Figure 4.19 shows the speed profile entered as input and together with the actual vehicle longitudinal velocity. It is evident that the controller correctly tracks the vehicle speed required, with a relative error when the speed profile changes velocity. Also a small delay in achieving the target velocity is observed throughout the simulation. Regardless, the results are satisfactory because the controller is able to follow a speed profile, even while cornering.

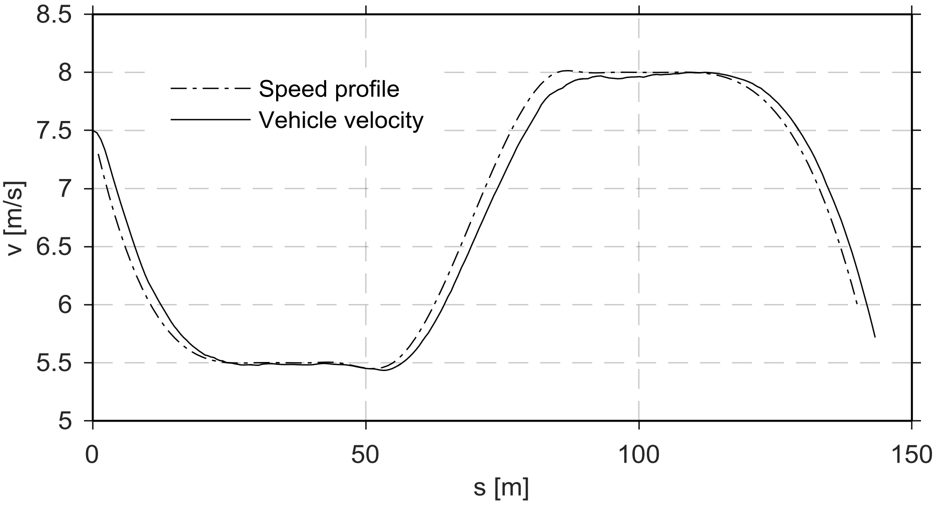


Figure 4.19 - Vehicle longitudinal velocity and predefined speed profile

Figure 4.20 shows the relative error between the vehicle velocity and the speed profile required, being the controller able to maintain the error close to zero while the speed is constant. When the speed changes the controller needs some time to return the error to zero again. The maximum value of the velocity error is 0.25 m/s.

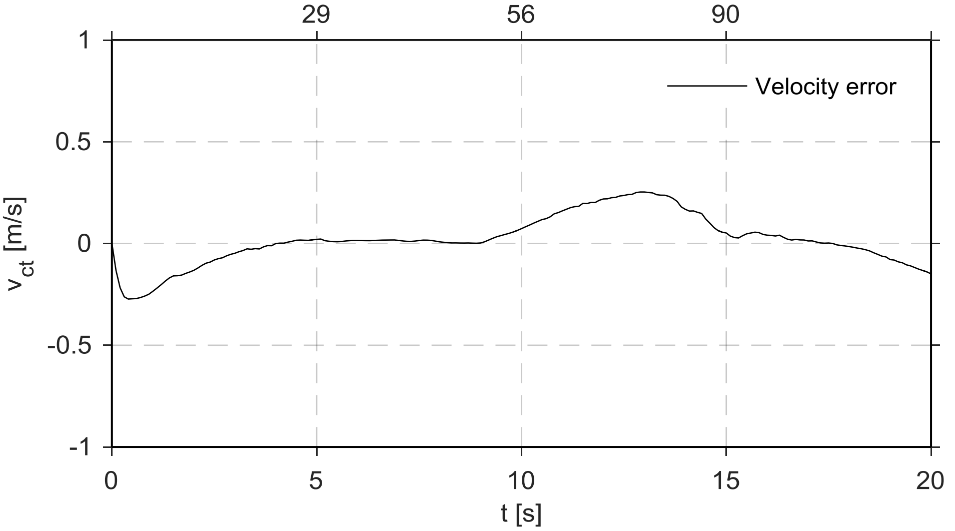


Figure 4.20 - Error of the vehicle longitudinal velocity with respect to the speed profile

The torque on the rear left wheel is also analyzed in Figure 4.21 which shows that the torque on the wheel is negative when the vehicle needs to reduce the velocity, which means braking, and has positive values during traction.

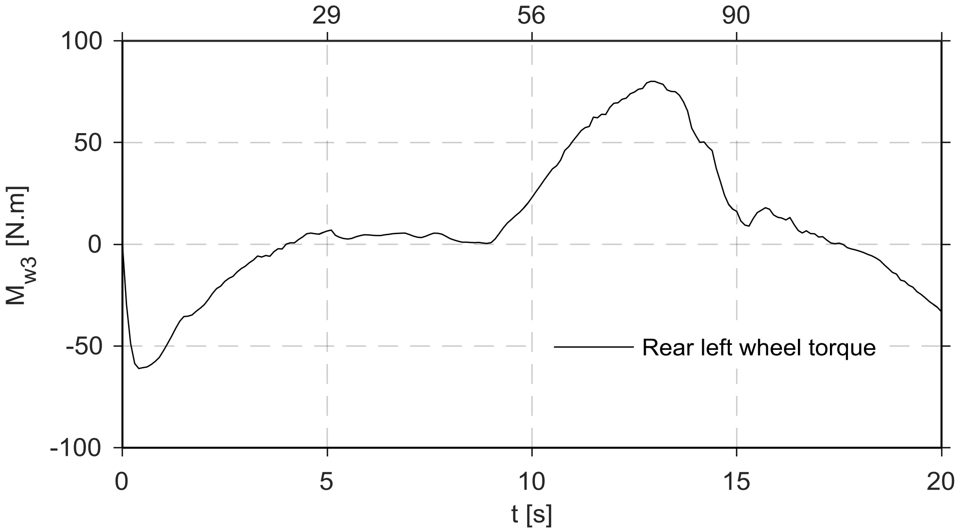


Figure 4.21 - Rear left wheel torque during the demonstration case

The front wheels only exhibit negative or null values for the torque, as shown in Figure 4.22, because the traction is only provided by the rear wheels being all the 4 wheels able to produce negative torques, for braking situations.

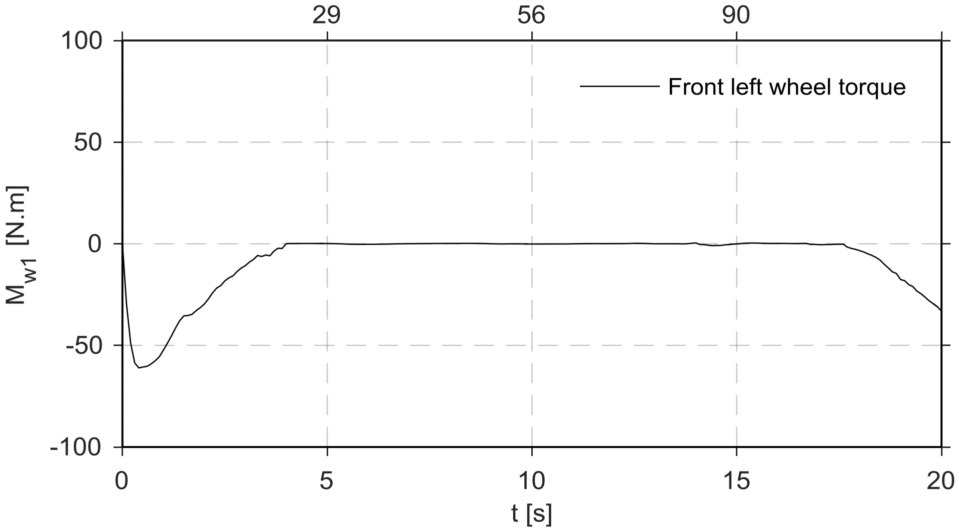


Figure 4.22 - Front left wheel torque during the demonstration case

5 The optimal lap time problem

In this chapter the optimal lap time problem is addressed. Many works have been developed in this field being the approach taken by Meier [2], which is also based in some previous works, the basis of the development presented here. Meier work addresses two different phases: the trajectory optimization and the speed profile optimization. Since the controller built in chapter 4 is able to ensure that the vehicle follows both trajectory and speed profiles, the methodology proposed by Meier [2] is adapted and implemented for the current work.

The optimal trajectory consists in finding the best compromise between the shortest trajectory and the least curvature trajectory, which allows the vehicle to have higher velocities while turning. Both shortest and least curvature are explained here as well as the speed profile optimization. Finally the best combination between trajectory and speed profile, which gives the minimal lap time, is found by an iterative process.

5.1. Shortest trajectory

The shortest trajectory is the shortest possible line that can be described within the track limits, i.e. between the right and left limits of the track. This seems to be a good manner of minimizing the lap time, since the travelling distance is minimized. However, it may not allow the vehicle to reach high speeds because it usually creates segments of the trajectory where the curvature is too high, which induces higher lateral accelerations on the car.

In order to describe the track geometrically, similarly to what was done to describe the reference trajectory for the control problem, the track is built here through an input file with the same type of information and then arranged in the form required for equation (4.49). Using that information, a mesh of n_{ps} points is built as function of the parameter s, not only for the track centerline but also with the inner and outer track limits. The track limits are considered to be at a fixed lateral distance from the centerline. With the track centerline definition and side limits the desired vehicle trajectory can be described as proposed in references [31] [32] [33]. The track is still discretized into several segments, using the mesh of points created, represented in Figure 5.1. The target vehicle trajectory is described as:

$$r(s_j) = r_{inner}(s_j) + \sigma(s_j) l_{tw}, \qquad j = 1, ..., n_{ps} \tag{5.1}$$

where *$r(s_j)$* gives the trajectory coordinates for the point *j*, *$r_{inner}(s_j)$* are the coordinates of the inner track limit for the same point and *$\sigma(s_j)$* is a factor that may assume values from 0 to 1. The track width *l_{tw}* multiplied by the *σ* factor gives the relative distance of the trajectory point from the inner limit. As before, the track width is considered constant for the complete track length.

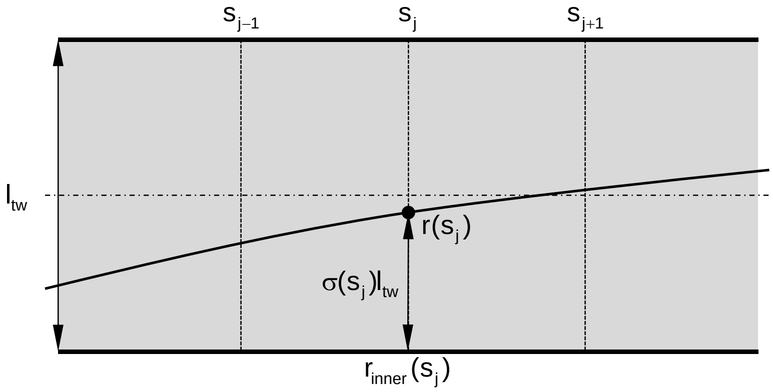


Figure 5.1 - Discretized track model representation

The shortest trajectory is found using an optimization process, where the factor *σ* is the optimization variable. The objective function of the problem to be minimized is the total track distance, subjected to some constraints. The total track distance for the discretized problem is just the line integral of the trajectory, which is approximated by the sum of the distances between consecutive points, from all points in the mesh and given by [2]:

$$\sum_{j=0}^{n_{ps}-2} \sqrt{(x_{j+1}-x_j)^2+(y_{j+1}-y_j)^2} \tag{5.2}$$

Using equation (5.2) the optimal problem is defined as:

$$\begin{aligned} \min \quad & \sum_{j=0}^{N-2}(x_{j+1}-x_j)^2+(y_{j+1}-y_j)^2 \\ s.t. \quad & 0 \le \sigma_j \le 1 \end{aligned} \tag{5.3}$$

Note that the squared Euclidean distance is used for the sake of computational simplicity, by omitting the square root [2].

For the computational implementation of the optimization, the auxiliary variable *ΔP* is computed to characterize the distance between two consecutive points, being its *x* and *y* components written as:

$$\Delta P_{x,j} = x_{j+1} - x_j = x_{inner,j+1} - x_{inner,j} + \sigma_{j+1} l_{tw} - \sigma_j l_{tw} \tag{5.4}$$

$$\Delta P_{y,j} = y_{j+1} - y_j = y_{inner,j+1} - y_{inner,j} + \sigma_{j+1} l_{tw} - \sigma_j l_{tw} \tag{5.5}$$

where equations (5.4) and (5.5) are deduced knowing that, based on equation (5.1), the coordinates of the trajectory x_j and y_j are given by:

$$x_j = x_{inner,j} + \sigma_j l_{tw} \tag{5.6}$$

$$y_j = y_{inner,j} + \sigma_j l_{tw} \tag{5.7}$$

Moreover, $\varDelta P_{x,j}$ is further developed and written as:

$$\Delta P_{x,j} = \Delta x_{j,0} + [\Delta x_{j+1}, -\Delta x_j] E_j \bar{\sigma} \tag{5.8}$$

where $\varDelta x_{j,0}$, $\mathbf{E}_j$ and $\bar{\boldsymbol{\sigma}}$ are given by:

$$\Delta x_{j,0} = x_{inner,j+1} - x_{inner,j} \tag{5.9}$$

$$\mathbf{E}_j = \begin{bmatrix} 0 & \dots & 0 & 1_j & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 1_{j+1} & 0 & \dots & 0 \end{bmatrix} \tag{5.10}$$

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_0 \\ \vdots \\ \sigma_{n_{ps}} - 1 \end{bmatrix} \tag{5.11}$$

The y component of $\varDelta P_j$ is computed in the same way using equations (5.8) to (5.11). The objective function L is now written as:

$$L = \sum_{j=0}^{n_{ps}-2} \Delta P_{x,j}^T \Delta P_{x,j} + \Delta P_{y,j}^T \Delta P_{y,j} \tag{5.12}$$

Providing the matrices $\mathbf{H}_j$ and $\mathbf{B}_j$:

$$\mathbf{H}_j = \begin{bmatrix} \Delta x_{j+1}^2 + \Delta y_{j+1}^2 & -\Delta x_{j+1}\Delta x_j - \Delta y_{j+1}\Delta y_j \\ -\Delta x_{j+1}\Delta x_j - \Delta y_{j+1}\Delta y_j & \Delta x_j^2 + \Delta y_j^2 \end{bmatrix} \tag{5.13}$$

$$\mathbf{B}_j = 2\left[\Delta x_{j+1}\Delta x_{j,0} + \Delta y_{j+1}\Delta y_{j,0} \quad -\Delta x_j \Delta x_{j,0} - \Delta y_j \Delta y_{j,0}\right] \tag{5.14}$$

the objective function given by equation (5.12) can be now written as:

$$L = \sum_{j=0}^{n_{ps}-2} \boldsymbol{\sigma}_j^T \mathbf{H}_j \boldsymbol{\sigma}_j + \sum_{j=0}^{n_{ps}-2} \mathbf{B}_j \boldsymbol{\sigma}_j + const. \tag{5.15}$$

which is equivalent to:

$$L = \boldsymbol{\sigma}_j^T \sum_{j=0}^{n_{ps}-2} \mathbf{E}_j^T \mathbf{H}_j \mathbf{E}_j \boldsymbol{\sigma} + \sum_{j=0}^{n_{ps}-2} \mathbf{B}_j \mathbf{E}_j \boldsymbol{\sigma} + const. \tag{5.16}$$

Matrices $\mathbf{H}_{st}$ and $\mathbf{B}_{st}$ are defined as:

$$\mathbf{H}_{st} = \sum_{j=0}^{n_{ps}-2} \mathbf{E}_j^T \mathbf{H}_j \mathbf{E}_j \tag{5.17}$$

$$\mathbf{B}_{st} = \sum_{j=0}^{n_{ps}-2} \mathbf{B}_j \mathbf{E}_j \tag{5.18}$$

and finally, the optimal problem is now written as:

$$\begin{aligned} &\min \quad \boldsymbol{\sigma}^T \mathbf{H}_{st} \boldsymbol{\sigma} + \mathbf{B}_{st} \boldsymbol{\sigma} + const. \\ &s.t. \quad 0 \le \sigma_j \le 1 \end{aligned} \tag{5.19}$$

In the work by Meier [2] also proposes an alternative method to find the shortest track which consists in extending the number of optimization variables by instead of only using *σ*, the addition *x* and *y* to the optimizations variables is proposed. This alternative approach is not developed here, for the sake of conciseness, although it has some promising features.

In terms of computational implementation, the problem is solved using Matlab being the minimization process handled by the Matlab function `fmincon`. This functions is applied for constrained non-linear multivariable problems which is suitable for the type of problem addressed here. The results of the optimal problem for the race track used throughout this work are presented in Figure 5.2. The geometry of the track used is the same that was described in Table 4.1 being the track side limits 5 meter from the centerline.

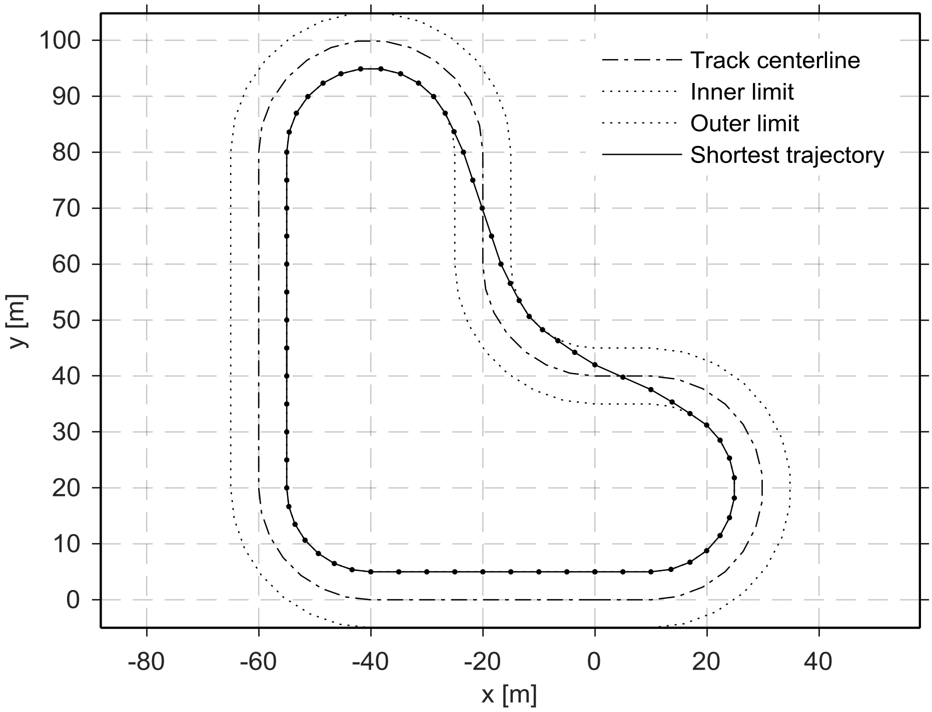


Figure 5.2 - Shortest trajectory representation for the demonstration track

Finally, it must be mentioned that a cubic spline interpolation is used with the shortest trajectory points being the control points of the curve. Using the Matlab `spline` function, the trajectory is fully described and ready to be used by the controller exactly in the same manner as it was used in chapter 4.

5.2. Least curvature trajectory

The least curvature trajectory is an alternative approach to the problem that allows the vehicle to reach higher velocities while cornering, which can be explained by the relation [2]:

$$a_{lat} = \frac{v^2}{R} = kv^2 \tag{5.20}$$

where a_{lat} is the vehicle lateral acceleration, v is its longitudinal velocity, R is the curvature radius and k is the curvature. Assuming that the vehicle has a maximum lateral acceleration, which is limited by the grip produced by the contact between the road and tyres, it becomes clear that when the curvature is lower, the velocity can reach higher values. Hence, with higher velocities lower lap times may be achieved. However, the total trajectory length is longer than the shortest trajectory one.

As for the shortest trajectory length, the variable σ is used as the optimization variable. The objective function to minimize in this approach is written as [2]:

$$\begin{aligned} &\min && \sum_{j=0}^{n_{ps}-2} k(s_j) \\ &s.t. && 0 \le \sigma_j \le 1 \end{aligned} \tag{5.21}$$

The curvature is given by the second order derivative of the trajectory, written as:

$$k(s) = \| r''(s) \| \tag{5.22}$$

which implies that the objective functions can be substituted by:

$$\begin{aligned} &\min && \sum_{j=0}^{n_{ps}-2} \| r''(s_j) \|^2 = \sum_{j=0}^{n_{ps}-2} [x''(s_j)]^2 + [y''(s_j)]^2 \\ &s.t. && 0 \le \sigma_j \le 1 \end{aligned} \tag{5.23}$$

In order to get the second order derivatives of the trajectory, it is suggested to use a spline interpolation [33]. The fact that cubic splines have a smooth shape and are continuous in their first and second order derivatives ensures that this type of interpolation does not lead to unnecessary numerical difficulties. The components z_x and z_y, of the second derivatives are obtained using the cubic spline interpolation as [2]:

$$z_{x,j} = x''(s_j) \qquad ; \qquad z_{y,j} = y''(s_j) \tag{5.24}$$

$$b_{x,j} = \frac{1}{h_j}(x_{j+1} - x_j) \qquad ; \qquad b_{y,j} = \frac{1}{h_j}(y_{j+1} - y_j) \tag{5.25}$$

Now, the distance between two sample points h_j is given by:

$$h_j = s_{j+1} - s_j = \Delta s \tag{5.26}$$

The cubic spline problem is now formulated:

$$\tilde{\mathbf{H}}_s \tilde{\mathbf{z}}_x = \tilde{\mathbf{b}}_x \qquad ; \qquad \tilde{\mathbf{H}}_s \tilde{\mathbf{z}}_y = \tilde{\mathbf{b}}_y \tag{5.27}$$

where $\tilde{\mathbf{H}}_s$, $\tilde{\mathbf{z}}_x$, $\tilde{\mathbf{z}}_y$, $\tilde{\mathbf{b}}_x$ and $\tilde{\mathbf{b}}_y$ are written as:

$$\tilde{\mathbf{H}}_s = \begin{bmatrix} 2(h_0+h_1) & h_1 & & & & \\ h_1 & 2(h_1+h_2) & h_2 & & & \\ & h_2 & 2(h_2+h_3) & h_3 & & \\ & & \ddots & \ddots & & \\ & & & h_{n_{ps}-4} & 2(h_{n_{ps}-4}+h_{n_{ps}-3}) & h_{n_{ps}-3} \\ & & & & h_{n_{ps}-3} & 2(h_{n_{ps}-3}+h_{n_{ps}-2}) \end{bmatrix} \tag{5.28}$$

$$\tilde{\mathbf{z}}_x = \begin{bmatrix} z_{x,1} \\ \vdots \\ z_{x,n_{ps}-2} \end{bmatrix} \quad ; \quad \tilde{\mathbf{z}}_y = \begin{bmatrix} z_{y,1} \\ \vdots \\ z_{y,n_{ps}-2} \end{bmatrix} \tag{5.29}$$

$$\tilde{\mathbf{b}}_x = 6\begin{bmatrix} b_{x,1} - b_{x,0} \\ \vdots \\ b_{x,n_{ps}-2} - b_{x,n_{ps}-3} \end{bmatrix} \quad ; \quad \tilde{\mathbf{b}}_y = 6\begin{bmatrix} b_{y,1} - b_{y,0} \\ \vdots \\ b_{y,n_{ps}-2} - b_{y,n_{ps}-3} \end{bmatrix} \tag{5.30}$$

The system of equations given by equation (5.27) is under-determined. However, if the first and last second derivatives o the cubic spline segments are zero, the system becomes a determined linear system, i.e.,

$$z_{x,0} = z_{y,0} = z_{x,n_{ps}-1} = z_{y,n_{ps}-1} = 0 \tag{5.31}$$

Note that **H**$_s$, **z**$_x$ and **b**$_x$ are defined by:

$$\mathbf{H}_s = \begin{bmatrix} 1 & 0_{1\times(n_{ps}-2)} & 0 \\ h_0 & & 0_{(n_{ps}-3)\times 1} \\ 0_{(n_{ps}-3)\times 1} & \tilde{\mathbf{H}}_s & h_{n_{ps}-2} \\ 0 & 0_{(1\times n_{ps}-2)} & 1 \end{bmatrix} \quad ; \quad \mathbf{z}_x = \begin{bmatrix} z_{x,0} \\ \tilde{\mathbf{z}}_x \\ z_{x,n_{ps}-1} \end{bmatrix} \quad ; \quad \mathbf{b}_x = \begin{bmatrix} 0 \\ \tilde{\mathbf{b}}_x \\ 0 \end{bmatrix} \tag{5.32}$$

The vectors **z**$_y$ and **b**$_y$ are computed similarly to equation (5.32). Based on equation (5.27) the second order derivative of the trajectory becomes:

$$\mathbf{z}_x = \mathbf{H}_s^{-1}\mathbf{b}_x \tag{5.33}$$

$$\mathbf{z}_y = \mathbf{H}_s^{-1}\mathbf{b}_y \tag{5.34}$$

where **b**$_x$ is now written as:

$$\mathbf{b}_x = \mathbf{B}_s x \tag{5.35}$$

and the matrix **B**$_s$ is built as:

$$\mathbf{B}_s = 6\begin{bmatrix} 0 & 0 & 0 & & & \\ \frac{1}{h_0} & -\left(\frac{1}{h_0}+\frac{1}{h_1}\right) & \frac{1}{h_1} & & & \\ & \frac{1}{h_1} & -\left(\frac{1}{h_1}+\frac{1}{h_2}\right) & \frac{1}{h_2} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \frac{1}{h_{n_{ps}-3}} & -\left(\frac{1}{h_{n_{ps}-3}}+\frac{1}{h_{n_{ps}-2}}\right) & \frac{1}{h_{n_{ps}-2}} \\ & & & 0 & 0 & 0 \end{bmatrix} \tag{5.36}$$

Finally, from equation (5.33) and (5.34) the system of equations that allows calculating the second order derivatives becomes:

$$\mathbf{z}_x = \mathbf{H}_s^{-1}\mathbf{B}_s\mathbf{x} \tag{5.37}$$

$$\mathbf{z}_y = \mathbf{H}_s^{-1}\mathbf{B}_s\mathbf{y} \tag{5.38}$$

With the second order derivatives, the optimization problem is reformulated for the parameter σ as the optimization variable. From equation (5.23) the objective function for the optimization problem is now written as:

$$\min \qquad \mathbf{z}_x^T\mathbf{z}_x + \mathbf{z}_y^T\mathbf{z}_y \tag{5.39}$$

By using equations (5.37) and (5.38) it is now possible to convert equation (5.39) into:

$$\min \qquad \mathbf{x}^T(\mathbf{H}_s^{-1}\mathbf{B}_s)^T(\mathbf{H}_s^{-1}\mathbf{B}_s)\mathbf{x} + \mathbf{y}^T(\mathbf{H}_s^{-1}\mathbf{B}_s)^T(\mathbf{H}_s^{-1}\mathbf{B}_s)\mathbf{y} \tag{5.40}$$

Considering equations (5.1) and (5.9), the optimization problem may be converted to a more convenient form as:

$$\begin{aligned} \min \qquad & \boldsymbol{\sigma}^T\mathbf{H}_{lc}\boldsymbol{\sigma} + \mathbf{B}_{lc}\boldsymbol{\sigma} + const. \\ s.t. \qquad & 0 \le \sigma_j \le 1 \end{aligned} \tag{5.41}$$

Where $\mathbf{H}_{lc}$, $\mathbf{B}_{lc}$ and the constant are given by:

$$\mathbf{H}_{lc} = diag(\Delta\mathbf{x})^T\mathbf{D}^T\mathbf{D}diag(\Delta\mathbf{x}) + diag(\Delta\mathbf{y})^T\mathbf{D}^T\mathbf{D}diag(\Delta\mathbf{y}) \tag{5.42}$$

$$\mathbf{B}_{lc} = 2(\mathbf{x}_{inner}^T\mathbf{D}^T\mathbf{D}diag(\Delta\mathbf{x}) + \mathbf{y}_{inner}^T\mathbf{D}^T\mathbf{D}diag(\Delta\mathbf{y})) \tag{5.43}$$

$$const. = \mathbf{x}_{inner}^T\mathbf{D}^T\mathbf{D}\mathbf{x}_{inner} + \mathbf{y}_{inner}^T\mathbf{D}^T\mathbf{D}\mathbf{y}_{inner} \tag{5.44}$$

Furthermore, **D** is given by:

$$\mathbf{D} = \mathbf{H}_s^{-1}\mathbf{B}_s \tag{5.45}$$

The work by Meier [2] also contains another approach for this where instead of having only the σ parameter as the optimization variable, the second order derivatives also enter in the problem as optimization variables. Similarly to what is done in the shortest trajectory optimization, Matlab `fmincon` function is also used for the least curvature trajectory optimization.

Figure 5.3 shows the least curvature trajectory, using the optimization method described by equation (5.41) and applied to the race track used throughout this work. The Matlab `spline` function is used in order to provide the trajectory for the controller.

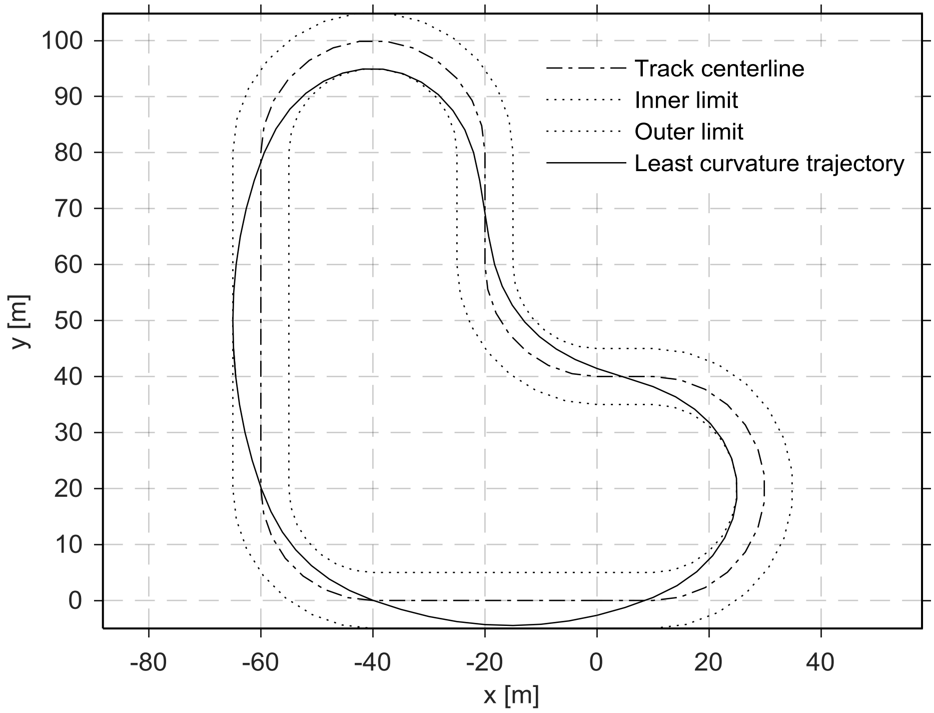


Figure 5.3 - Least curvature trajectory representation for the demonstration track

5.3. Combination of the shortest and least curvature trajectories

A combination of the two shortest and least curvature trajectories may provide the best trajectory for the optimal lap time. On one hand the shortest trajectory allows the vehicle to run a lower distance, which may minimize the time lap for lower velocities. However the velocities values reached while following this trajectory may be conservatively low and, in that sense, the trajectory is not time optimal. On the other hand, a trajectory with the least curvature within the track limits allows the vehicle to reach higher speed, but at the cost of having longer distances to run. As none of this two trajectories is optimal, a combination of the two may have good results and reduce the lap time.

Meier suggests to combine the two approaches in the following problem [2]:

$$L = (1-\tau)LC + \tau ST \tag{5.46}$$

where τ is the combination factor between the least curvature *LC* and shortest trajectory *ST* approaches. The equation (5.46) is explicitly written as:

$$L = \boldsymbol{\sigma}^T \mathbf{H}_L \boldsymbol{\sigma} + \mathbf{B}_L \boldsymbol{\sigma}^T \tag{5.47}$$

where $\mathbf{H}_L$ and $\mathbf{B}_L$ are given by:

$$\mathbf{H}_L = (1-\tau)\mathbf{H}_{lc} + \tau \mathbf{H}_{st} \tag{5.48}$$

$$\mathbf{B}_L = (1-\tau)\mathbf{B}_{lc} + \tau \mathbf{B}_{st} \tag{5.49}$$

and $\mathbf{H}_{lc}$, $\mathbf{H}_{st}$, $\mathbf{B}_{lc}$ and $\mathbf{B}_{st}$ are defined from equations (4.17), (4.18), (4.42) and (4.43). The optimization process for this approach is done using equation (5.47) as the objective function and having the parameter σ as the optimization variable. Matlab `fmincon` function is applied as in all previous optimal problems.

Once it is possible to get the optimal speed profile, it is also possible to get the best combination factor τ. By using an iterative process, where the combination factor varies from 0 to 1, and computing the optimal speed profile for each iteration, the best combination factor is calculated. Figure 5.4 shows an example of the optimal trajectory for the demonstration track, being in this case very similar to the least curvature track, for a given speed profile.

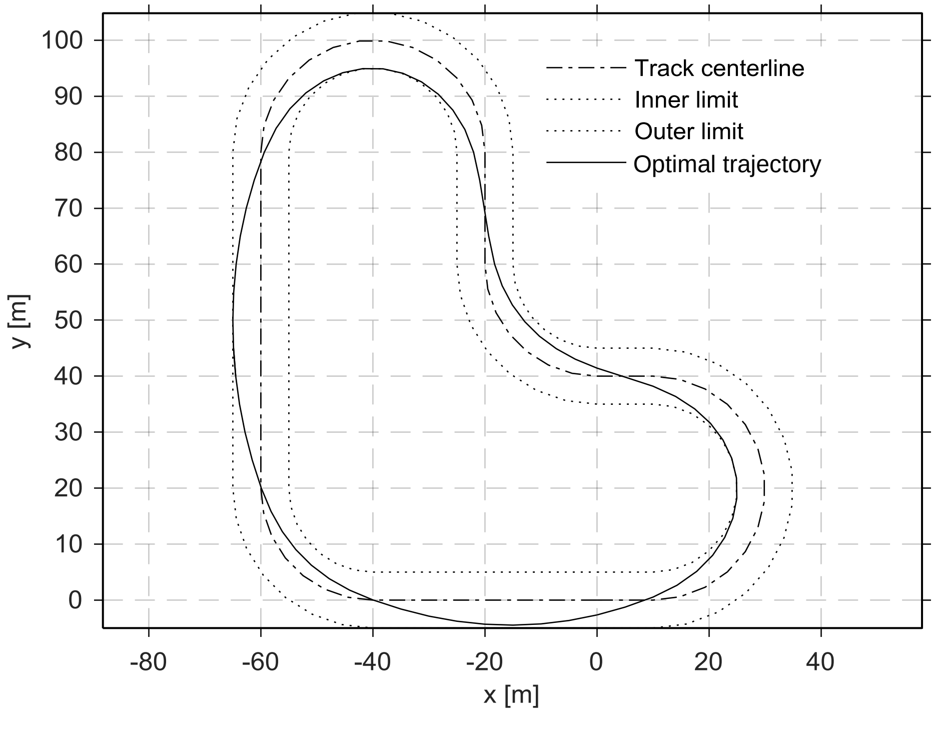


Figure 5.4 - Optimal trajectory for the demonstration track

The final combination factor value is 0.000025 and the lap time is 28.537 seconds. The maximum and minimum longitudinal accelerations used are 1.5 and -5 *m/s*^{2}, respectively. The maximum lateral acceleration allowed is 2.7 *m/s*^{2}.

5.4. Speed profile optimization

Having the optimal trajectory computed, it is necessary to get the maximum velocity that the vehicle can sustain at each point of the track, by optimizing the speed profile. The method implemented by Meier [2] tries to minimize the lap time, by relating the distance and the vehicle velocity at each track segment. While it is relatively simple approach it requires some simplifications. This method assumes that the maximum and minimum longitudinal and lateral accelerations of the vehicle are known and constant. As a consequence, if the values of the accelerations are not accurate enough, the results may be too conservative or too relaxed. It is necessary to keep in mind that the accelerations are limited by the ability of the tyres to produce lateral and longitudinal forces. Moreover, the torques that the engine and brakes are able to produce are also limited.

The discrete optimization problem for the optimal velocity profile is described by [2]:

$$\begin{aligned} \min \quad & \sum_{j=0}^{n_{ps}-2} \Delta t_j = \sum_{j=0}^{n_{ps}-2} \frac{\Delta s_j}{\frac{v_{j+1}+v_j}{2}} \\ s.t. \quad & a_{long\,\min,j} \le a_{long,j} \le a_{lon\,\max,j} \\ & 0 \le a_{lat,j} \le a_{lat\,\max,j} \\ & v_{\min,j} \le v_j \le v_{\max,j} \end{aligned} \tag{5.50}$$

where the absolute value of $a_{lat,j}$ is deduced from equation (5.20). The maximum and minimum values of the accelerations are predefined, while the longitudinal acceleration is written as:

$$a_{long,j} = \frac{\Delta v_j}{\Delta t_j} = \frac{\Delta v_j}{\Delta s_j}\frac{\Delta s_j}{\Delta t_j} = \frac{v_{j+1}-v_j}{\Delta s_j}\frac{v_{j+1}+v_j}{2} = \frac{v_{j+1}^2-v_j^2}{2\Delta s_j} \tag{5.51}$$

where v is the longitudinal velocity of the vehicle. Furthermore, it is now introduced the square of the velocity as a new variable [2]:

$$\tilde{v}_j = v_j^2 \tag{5.52}$$

By maximizing the square of the velocity, the lap time is minimized, which means that if the symmetric value of the square velocity is minimized, the lap time will be also minimum. The objective function becomes:

$$\min \qquad -\sum_{j=0}^{n_{sp}-2} \tilde{s}_j \tag{5.53}$$

With equation (5.53) as the objective function and the square of the velocity $\tilde{s}$ as the optimization variable, the problem is now fully structured as:

$$\begin{aligned} \min \quad & \mathbf{E}^T\tilde{\mathbf{s}} \\ s.t. \quad & 2\Delta s_j a_{long\min,j} \leq \tilde{s}_{j+1} - \tilde{s}_j \leq 2\Delta s_j a_{long\max,j} \\ & 0 \leq \tilde{s}_j \leq a_{lat\max,j} R \\ & s^2_{\min,j} \leq \tilde{s}_j \leq s^2_{\max,j} \end{aligned} \tag{5.54}$$

where $\mathbf{E}^T$ is given by:

$$\mathbf{E}^T = [-I_{n_{ps}\times 1}]$$

The minimization problem here is a linear problem. Therefore, differently from what happens in the trajectories optimization, in the speed profile problem the minimum of the objective function is found using Matlab `linprog` function, which solves linear programming problems. Figure 5.5 shows the optimal speed profile for the optimal trajectory provided from the previous demonstration case. The maximum and minimum longitudinal accelerations used are 1.5 and -5 *m/s*^{2}, respectively. The maximum lateral acceleration considered is 2.7 *m/s*^{2}.

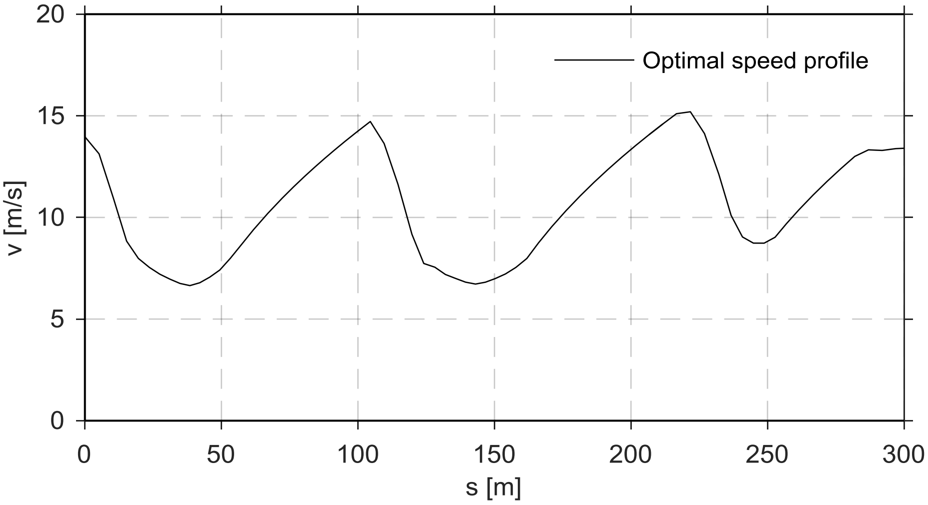


Figure 5.5 - Optimal speed profile for the optimal demonstration case trajectory

In Figure 5.6 the speed at each position in the track is depicted in the 3D graphical representation, being the optimal trajectory and the speed profile also represented.

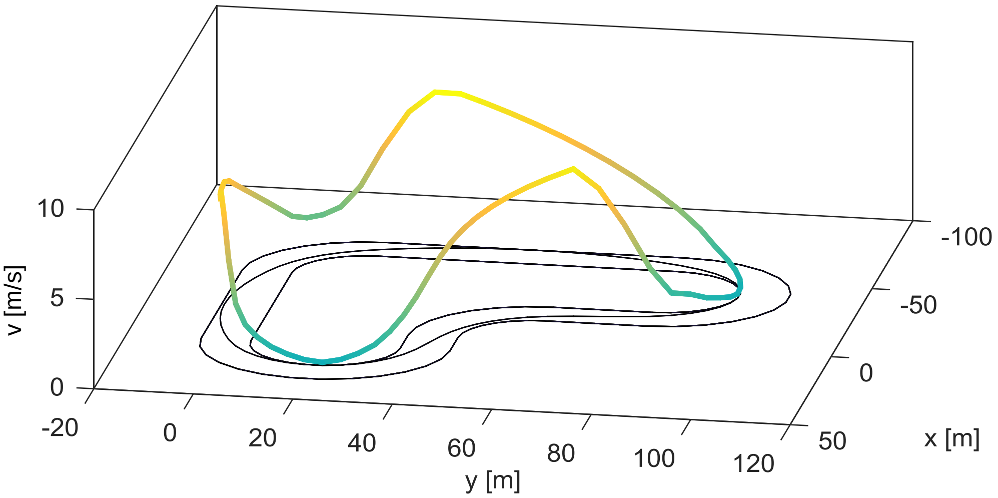


Figure 5.6 - Optimal trajectory and speed profile for the demonstration case

The speed profile is optimized for all of the trajectory cases in the demonstration track, being the values of the minimum and maximum accelerations the same for all of them and the same as used before. Table 5.1 shows the results, with respect to the lap times, of the different trajectories computed for the demonstration track.

Table 5.1 - Lap times for the different trajectories of the demonstration track

Trajectory	Lap Time [s]
Centerline	37.15
Shortest Track	31.33
Least Curvature Track	28.95
Optimal Track	28.54

5.5. Demonstration case and discussion

A demonstration case using the controller for the optimal trajectory and speed profile is presented. The controller with the previewing has some issues when running for the optimal lap time, because there are much more changes of curvature for the optimal trajectory and when the preview trajectories are computed, the vehicle behavior becomes very unstable and the simulation does not provide usable results. As discussed before, this problem can be solved by trying different preview distances and update distances, being this study an important step forward for the development of the preview methodology, which has some advantages, as seen before. The demonstration case for the optimal lap time is then performed without previewing, but the speed following feature is kept. The optimal speed profile and the optimal trajectory used are presented in figures Figure 5.4, Figure 5.5 and Figure 5.6. As the initial longitudinal velocity for the optimal speed profile is 14 *m/s*, the vehicle initial velocity is set for the same value.

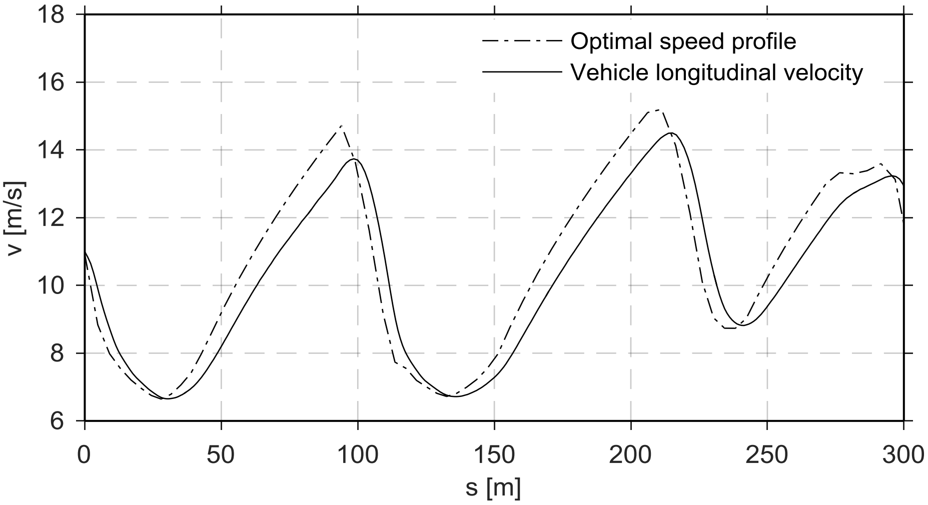


Figure 5.7 - Vehicle longitudinal velocity for the optimal lap time demonstration case

Figure 5.7 shows a comparison between the optimal speed profile and the actual vehicle longitudinal velocity. The controller behaves as expected, with similar results to what is described in previous sections. Figure 5.8 shows the vehicle trajectory during the optimal lap time demonstration case and also the reference trajectory, which is the optimal trajectory.

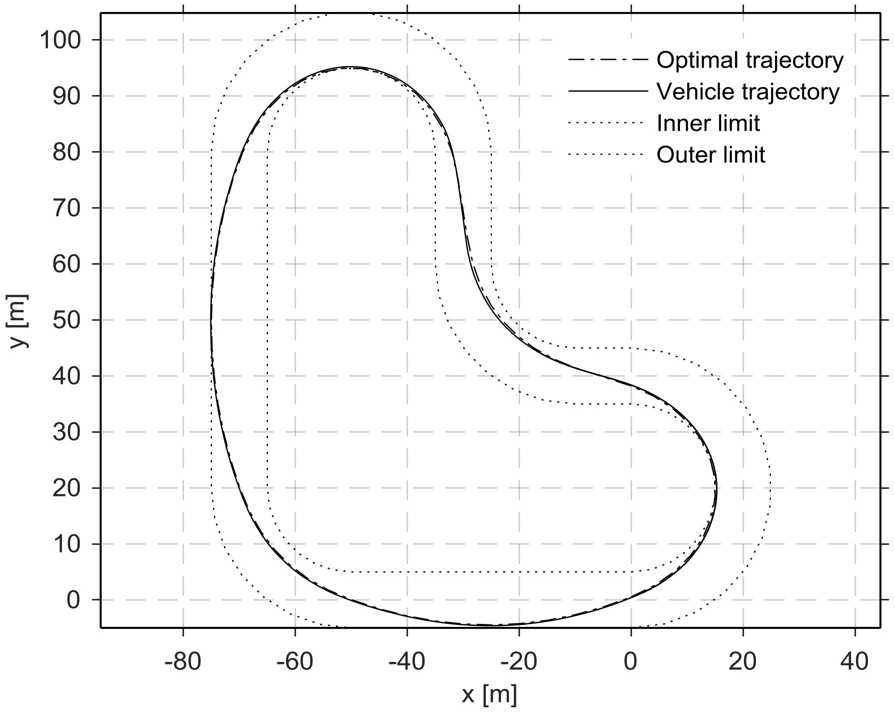


Figure 5.8 - Vehicle trajectory for the optimal lap time demonstration case

It takes 30,7 seconds for the vehicle to complete the entire lap, which is different from the expected 28,54 seconds of optimal lap time. The fact that the vehicle initial orientation does not exactly coincide with the track tangent orientation can explained this lap time error, as well as the maximum and minimum accelerations used to compute the optimal speed profile may be too optimistic and consequently the tyres are not able to produce the necessary forces in order to apply such accelerations on the vehicle center of mass. Nevertheless, the results achieved are well aligned with the objectives of this work, as all the tasks are accomplished with the desired degree of precision.

6 Spatial analysis

In this chapter the development of the 3D analysis is addressed. The approach used is similar to the approach used for the planar analysis, regarding the controller. The dynamic analysis is developed in 3D, using a spatial dynamic analysis program (DAP3D).

6.1. 2D and 3D dynamic analysis comparison

Advantages can be obtained by using the spatial analysis: it allows a much more detailed vehicle modelling, where the most important bodies and kinematic joints can be included in the model, instead of having a simplified planar vehicle model; the fact of having 3 dimensions assures the representations of several mechanical components, which influence the suspension behavior, that are not included in the planar model; it allows for the representation of the roll angle of the chassis as well as for the camber angle variation in the wheels. Moreover, mass transfers are computed in the dynamic analysis and have great impact in the variation of the normal load on the tyres, which affect all the tyre forces.

The tyre model used in the DAP3D program is the U/A Tyre Model [9] [10] [11], which is different from the one used in the planar dynamic analysis program, i.e., for the Pacejka's Magic Formula [8]. Furthermore, the tyre properties used also present some differences.

In what concerns to the road modelling, some advantages are taken from the use of the 3D analysis because it allows the track model to include a topology. Hence, real track geometries can be modelled and more accurate results can be obtained.

Regarding the control strategy, the same approach used in the planar analysis is applied here, which means that the controller acts the same way being the road planar or spatial. The lap time optimization the same method is also considered, being the road considered planar for the trajectory optimization and also for the speed profile optimization.

6.2. Spatial vehicle model

The multibody model of the Lancia Stratos for the spatial analysis is now presented and is based on the model developed by Ambrósio and Gonçalves [22], which is depicted in Figure 6.1.

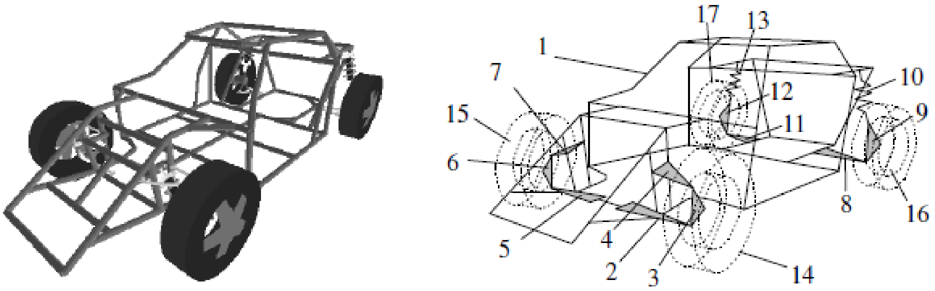


Figure 6.1 - Representation of the Lancia Stratos multibody model

Table 6.1 describes with detail all the bodies of the model with their masses, inertial characteristics and initial positions.

Table 6.1 - Mass, inertial characteristics and initial positions of the system components

Body	Description	Mass [Kg]	Inertia [Kg.m^2] ($J_{\xi\xi}$ / $J_{\eta\eta}$ / $J_{\zeta\zeta}$)	Center of mass [m] $(x/y/z)_0$
1	Chassis	723.3	98.9800/529.2000/587.9000	0.0000/0.0000/0.0890
2, 5	Front Lower A-Arm	2.0	0.0268/0.0001/0.0268	1.0100/± 0.3925/0.0000
3, 6	Front Knuckle	16.6	0.0621/0.0589/0.0231	1.0100/± 0.5875/0.1100
4, 7	Front Upper A-Arm	1.1	0.0077/0.0064/0.0140	1.0100/± 0.4435/0.2600
8, 11	Rear Lower A-Arm	1.5	0.0140/0.0057/0.0197	-1.0800/± 0.4630/0.0000
9, 12	Rear Knuckle	11.4	0.0313/0.0240/0.0257	-1.0800/± 0.6410/0.1070
10, 13	Rear Damper	0.001	0.0100/0.0010/0.0100	-1.0800/± 0.5000/0.6000
14, 15	Front Wheel	9.0	0.0000/1.9000/0.0000	1.0100/± 0.7150/0.1000
16, 17	Rear Wheel	9.5	0.0000/1.9000/0.0000	-1.0800/± 0.7300/0.1000

All the kinematic joints for the model are presented in Table 6.2.

Table 6.2 - Kinematic joints properties

Joint	Type	Body i	Body j	ξ_i^P ξ_i^Q	η_i^P η_i^Q	ζ_i^P ζ_i^Q	ξ_j^P ξ_j^Q	η_j^P η_j^Q	ζ_j^P ζ_i^Q
1	Spherical	4	3	0.0000 0.0000	0.1240 -0.0200	0.0000 0.1500	0.0000 0.0000	0.0000 0.0000	0.0000 0.0000
2	Spherical	7	6	0.0000 0.0000	-0.1240 0.0200	0.0000 0.1500	0.0000 0.0000	0.0000 0.0000	0.0000 0.0000
3	Spherical	2	3	0.0000 0.0000	0.1750 -0.0200	0.0000 -0.110	0.0000 0.0000	0.0000 0.0000	0.0000 0.0000
4	Spherical	5	6	0.0000 0.0000	-0.1750 0.0200	0.0000 -0.110	0.0000 0.0000	0.0000 0.0000	0.0000 0.0000
5	Spherical	8	9	0.0000 0.0000	0.1370 -0.0410	0.0000 -0.107	0.0000 0.0000	0.0000 0.0000	0.0000 0.0000
6	Spherical	11	12	0.0000 0.0000	-0.1370 0.0410	0.0000 -0.107	0.0000 0.0000	0.0000 0.0000	0.0000 0.0000
7	Revolute	1	2	1.0100 0.0000	0.2175 -0.1750	-0.089 0.0000	2.0100 1.0000	0.2175 -0.1750	-0.890 0.0000
8	Revolute	1	4	1.0100 0.0000	0.3175 -0.1260	0.1710 0.0000	2.0100 1.0000	0.3175 -0.1260	0.1710 0.0000
9	Revolute	3	14	0.0000 0.0000	0.1275 0.0000	-0.010 0.0000	0.0000 0.0000	1.1275 1.0000	-0.010 0.0000
10	Revolute	1	5	1.0100 0.0000	-0.2175 0.1750	-0.089 0.0000	2.0100 1.0000	-0.2175 0.1750	-0.890 0.0000
11	Revolute	1	7	1.0100 0.0000	-0.3175 0.1260	0.1710 0.0000	2.0100 1.0000	-0.3175 0.1260	0.1710 0.0000
12	Revolute	6	15	0.0000 0.0000	-0.1275 0.0000	-0.010 0.0000	0.0000 0.0000	-1.1275 -1.0000	-0.010 0.0000
13	Revolute	1	8	-1.080 0.0000	0.3000 -0.1630	-0.089 0.0000	-2.080 -1.000	0.3000 -0.1630	-0.089 0.0000
14	Revolute	1	10	-1.080 0.0000	0.5000 0.0000	0.5110 0.0000	-2.080 -1.000	0.5000 0.0000	0.5110 0.0000
15	Revolute	9	16	0.0000 0.0000	0.0890 0.0000	-0.007 0.0000	0.0000 0.0000	1.0890 1.0000	-0.007 0.0000
16	Revolute	1	11	-1.080 0.0000	-0.3000 0.1630	-0.089 0.0000	-2.080 -1.000	-0.3000 0.1630	-0.089 0.0000
17	Revolute	1	13	-1.080 0.0000	-0.5000 0.0000	0.5110 0.0000	-2.080 -1.000	-0.5000 0.0000	0.5110 0.0000
18	Revolute	12	17	0.0000 0.0000	-0.0890 0.0000	-0.007 0.0000	0.0000 0.0000	-1.0890 -1.0000	-0.007 0.0000
19	Translational	10	9	0.0000 0.0000	0.0000 -0.1410	0.0000 0.4930	0.0000 0.0000	0.1000 -0.0410	-0.350 0.1430
20	Translational	13	12	0.0000 0.0000	0.0000 0.1410	0.0000 0.4930	0.0000 0.0000	-0.1000 0.0410	-0.350 0.1430

Table 6.3 presents the data with respect to the springs and dampers used in the spatial model.

Table 6.3 - Suspension springs and dampers data

Force element	Connected bodies	Stiffness [N.m^{-1}]	Damping [N.s.m^{-1}]	Spring free length [m]
1	1, 2	57250	4000	0.4025
2	1, 5	57250	4000	0.4025
3	1, 9	44150	3000	0.5290
4	1, 12	44150	3000	0.5290

The tyre characteristics used for the Lancia Stratos multibody model are described in Table 6.4. Note that the data reported is suitable for the use of the U of A model. When using Pacejka Tire Model a different set of factors must be used.

Table 6.4 - Tyre characteristics.

Tire radius	0.2930	[m]
Torus radius	0.0512	[m]
Radial stiffness	200000	[N.m^{-1}]
Longitudinal stiffness	500000	[N.m^{-1}]
Lateral stiffness	150000	[N.m^{-1}]
Cornering stiffness	30000	[N.rad^{-1}]
Rolling friction coefficient	0.0100	[-]
Radial damping	0.0780	[N.s.m^{-1}]

6.3. 3D demonstration case

A demonstration case for the spatial analysis is presented, being the track used the same as for the planar analysis. The methodology applied for this demonstration case uses the standard controller, without preview and speed profile, as the computational effort needed for the spatial analysis is much greater than for the planar case. Furthermore, the simulation performed does not make the vehicle complete an entire lap, but only the first curve. However, it permits to conclude that the controller also have a good behavior for the spatial case. The vehicle initial velocity is 7.5 m/s and it starts at the origin, as shown in Figure 6.2, which depicts the vehicle trajectory along the track.

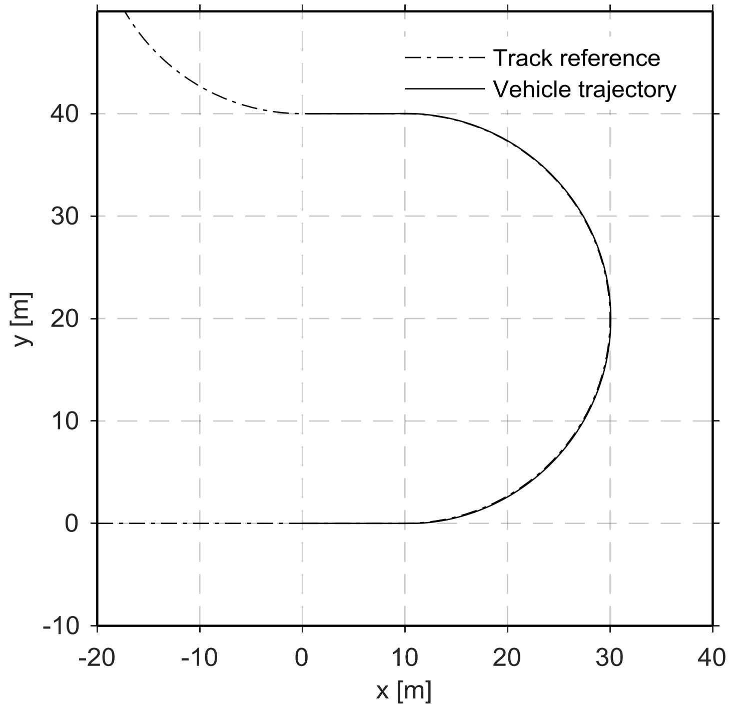


Figure 6.2 - Vehicle trajectory for the spatial demonstration case

Figure 6.3 shows a comparison between the lateral errors from the planar and the spatial case along the same track segment.

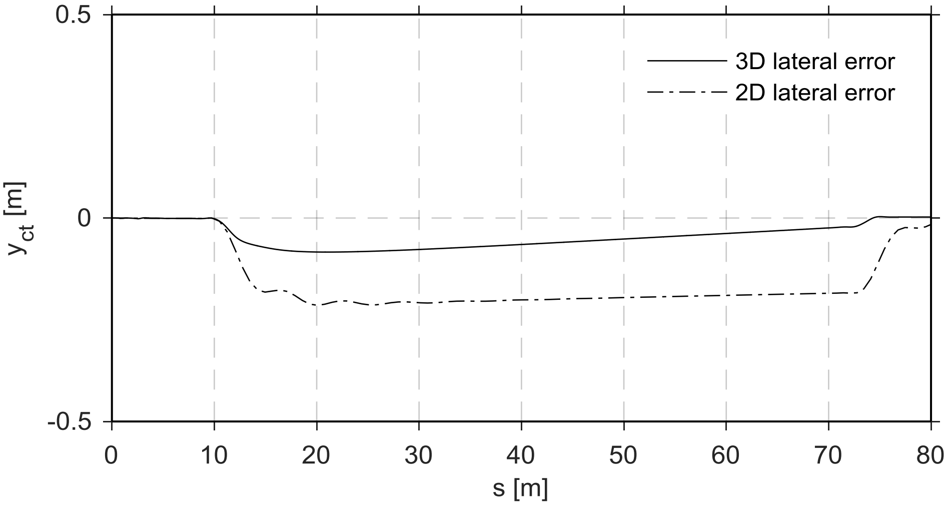


Figure 6.3 - Comparison between the relative lateral distances between the vehicle and the reference trajectory for the 2D and 3D case

As shown in Figure 6.3 the controller seems to have a similar behavior for both 2D and 3D cases, even if in the spatial case the maximum error is smaller. However, having results for such a small segment of the track and the fact that the tyre model used is not the same does not allow to have a definitive idea on the reason for the costly computational time for the spatial analysis.

7 Conclusions and recommendations

7.1. Conclusions

A 2D multibody dynamic analysis program was successfully developed with all the necessary features needed to simulate the behavior of a car in a race track, being the steering constraint and the tyre model the most important. The tyre parameters used from the literature are not the ideal ones to simulate the specific vehicle here presented, Lancia Stratos, mostly because this parameters belong to a tyre which is not built to support the normal loads that the Lancia Stratos produces. Nevertheless, the objective here was to implement the tyre model that enables the program to simulate the behavior of a vehicle in a road by computing the tyre-road contact forces, which was achieved by implementing lateral and longitudinal tyre forces and the self-aligning torque for combined slip correctly in a simplified planar Lancia Stratos model. The 2D dynamic analysis program developed is able to run either with input files, with steering and brake/traction information, or with a controller.

The second part of this work, with respect to the controller, is implemented with the update of a previously developed controller by adding new features to it, besides its trajectory following function. First of all, it is ensured that the controller has the ability to lead the vehicle to complete a complete round to any track. A preview distance is then added to the controller, which smooths the trajectory of the vehicle and better corresponds to the way that a human drives a car by continuously computing new trajectories from the actual position of the vehicle to a point in the track reference defined with a predefined preview distance ahead. The controller is updated with the ability to follow a prescribed speed profile by applying torques to the wheels of the car, which consequently leads the dynamic analysis program to produce tyre-ground contact forces. The Lancia Stratos is a rear-wheel driving vehicle, which means that only the rear wheels of the car are able to produce traction, being all of the 4 wheels available to brake the car. Hence, the controller is implemented to only produce torque in the rear wheels while accelerating and in the 4 wheels for braking situations.

The final part of this work consists in finding the optimal lap time for a race track, which is found by optimizing the trajectory first and by computing the optimal speed profile for the trajectory. The combination of this two different methodologies provides the optimal lap time. In order to find the optimal trajectory a conjunction of the shortest trajectory and the least curvature trajectory is computed. For the optimal speed profile, an optimization is performed using the velocity as the function to maximize by using lateral and longitudinal acceleration thresholds for the vehicle. Finally, the optimal trajectory and speed profile are provided to the controller which, together with the dynamic analysis program, simulates the behavior of the vehicle performing the optimal lap.

Finally the work intends to adapt all the developments to the spatial case, using a detailed Lancia Stratos multibody model, which was achieved in part. The fact that the 3D case needs much greater computational effort did not allow all the developments to be adapted from the planar analysis, as the only results obtained were for a small segment of the track with the standard controller. The comparisons between the planar and spatial case do not allow conclusions not only because tyre model used is different for the two cases but also due to the limited range of results analyzed. However, it became obvious that the controller apparently also work well for the spatial analysis.

7.2. Recommendations for future work

The objectives of this work were to develop a planar dynamic analysis program, implementing the tyre model which provides the tyre-road contact forces and to update a previously developed controller in order to ensure that the multibody vehicle of the Lancia Stratos is able to follow a prescribed trajectory and speed profile, which are computed by using an optimization process that minimizes the lap time for a race track and the results are well aligned with this objectives. However, there are some improvements that can be applied in order to get more trustable results when compared to experimental data from real vehicles.

The dynamics analysis program can be improved by implementing new tyre parameters more adequate to the vehicle model being studied. It is important to provide the correct parameters for the Pacejka Magic Formula, in order to compute correctly the lateral and longitudinal forces and the self-aligning moment, for combined slip. Moreover, the 2D dynamic analysis program does not take into account the car load transfers, which are very important to compute the tyre-road contact forces, as the normal load of each tyre changes with respect to the accelerations applied in the car center of mass.

The steering constraint implemented in the program can be reformulated by implementing the Ackerman condition, which is also important regarding the tyre forces, as it influences the side slip angles of the steered tyres.

The controller has some issues which can be solved. The weighting matrices coefficients can be improved, however there are no perfect values for them. The values have to be chosen by trying different combinations and checking which one gives the best results for each specific objective. More work has to be done concerning the preview and update distances. Nevertheless, it depends on each track and vehicle and several values have to be tested.

The optimal lap time problem can be improved specially in the optimization with respect to the speed profile. In the approach used here the speed profile is limited by the maximum and minimum lateral and longitudinal accelerations which are provided to the program by the user. This acceleration boundary values need to well represent the real limits of the vehicle, otherwise the optimal lap time is either too conservative or ambitious which, in the last case, may lead the multibody model of the vehicle, studied in the dynamic analysis program, to go out of the track limits and fail to follow the reference trajectory. Note that, in order to properly compute the longitudinal acceleration thresholds, the torques that the engine can produce, along with all the gearbox ratios (if it is applied for the vehicle studied) and the brake system characteristics have to be taken into account. Moreover, alternative methods for the optimizer have to be studied, as the one used takes too much time to compute the trajectories.

Regarding the spatial analysis, the most important aspect to be improved is the tyre model approach, which needs to be aligned with the planar case, using of the same model and tyre properties. Moreover, the trajectory and speed profile optimization have to be better adapted for the spatial case, as the track geometry may assume non-constant values in the vertical direction, i.e. the track can be modelled with ups and downs or with lateral inclination, being the optimal trajectory and speed profile obviously different from the planar case.

References

[1] P. Antos and J. A. C. Ambrósio, "A Control Strategy for Vehicle Trajectory Tracking Using Multibody Models," *Multibody System Dynamics 11,* pp. 365-394, 2004.

[2] K. Meier, "Sub-optimal path planning for 1:43 scale race cars," Zurich, 2010.

[3] P. E. Nikravesh, Planar Multibody Dynamics: Formulation, Programming and Applications, CRC Press - Taylor & Francis Group, 2008.

[4] A. A. Shabana, Dynamics of Multibody Systems, United Kingdom: Cambridge University Press, 1998.

[5] W. F. Milliken and D. L. Milliken, Race Car Vehicle Dynamics, United States of America: Society of Automotive Engineers, Inc, 1995.

[6] R. N. Jazar, Vehicle Dynamics: Theory and Applications, New York: Springer Science + Business Media, LLC, 2008.

[7] G. Genta and L. Morello, The Automotive Chassis - Volume 1: Components Design, Springer Science + Business Media B. V., 2009.

[8] H. B. Pacejka, Tyre and Vehicle Dynamics: 2nd Edition, Butterworth-Heinemann, 2006.

[9] G. Gim and P. E. Nikravesh, "An analytical model of pneumatic tyres for vehicle dynamic simulations. Part 3: Validation against experimental data," *International Journal of Vehicle Design,* vol. 12, pp. 217-228, 1991.

[10] G. Gim and P. E. Nikravesh, "An analytical model of pneumatic tyres for vehicle dynamic simulations. Part 1: Pure slips," *International Journal of Vehicle Design,* vol. 11, pp. 589-618, 1990.

[11] G. Gim and P. E. Nikravesh, "An analytical model of pneumatic tyres for vehicle dynamic simulations. Part 2: Comprehensive slips," *International Journal of Vehicle Design,* vol. 11, pp. 19-39, 1991.

[12] A. Rucco, G. Notarstefano and J. Hauser, "Computing minimum lap-time trajectories for a single-track car with load transfer," in *51st IEEE Conference on Decision and Control,* Maui, Hawaii, USA, 2012.

[13] D. Brayshaw and M. Harrison, "A quasi steady state approach to race car lap simulation in order to understand the effects of racing line and centre of gravity location," *Journal of Automobile Engineering,* vol. 219, pp. 725-739, 2005.

[14] D. P. Kelly, ""Lap time simulation with transient vehicle and tyre dynamics", Ph.D. dissertation," Cranfield University, 2008.

[15] J. P. M. Hendrikx, T. J. J. Meijlink and R. F. C. Kriens, "Application of optimal control theory to inverse simulation of car handling," *Vehicle System Dynamics,* vol. 26, pp. 449-462, 1996.

[16] D. Casanova, R. Sharp and P. Symonds, "Minimum time manoeuvring: The significance of yaw inertia," *Vehicle System Dynamics,* vol. 34, pp. 77-115, 2000.

[17] M. Thommyppillai, S. Evangelou and R. S. Sharp, "Advances in the development of a virtual car driver," *Multibody System Dynamics,* vol. 22, pp. 245-267, 2009.

[18] P. E. Nikravesh, Computer-Aided Analysis of Mechanical Systems, Englewood Cliffs, New Jersey: Prentice-Hall, 1988.

[19] J. Baumgarte, "Stabilization of Constraints and Integrals of Motion in Dynamical Systems," *Computer Methods in Apllied Mechanics and Engineering,,* vol. 1, pp. 1-16, 1972.

[20] W. Hirschberg, G. Rill and H. Weinfurter, "Tyre Model TMeasy," *Vehicle System Dynamics,* vol. 45, no. Issue S1, pp. 101-119, 2007.

[21] J. P. C. Gonçalves and J. A. C. Ambrósio, "Road Vehicle Modeling Requirements for Optimization of Ride and Handling," *Multibody System Dynamics,* vol. 13, pp. 2-32, 2005.

[22] J. A. C. Ambrósio and J. Gonçalves, "Complex Flexible Multibody Systems with Application to Vehicle Dynamics," *Multibody System Dynamics,* vol. 6, pp. 163-182, 2001.

[23] R. Sharp, "Motorcycle steering control by road preview," *ASMF Journal of Dynamic Systems,* vol. 129, no. Measurement and Control, pp. 373-381, 2007.

[24] A. Alleyne, "A Comparison of Alternative Intervention Strategies for Unintended Roadway Departure (URD) Control," *Vehicle System Dynamics,* vol. 27, pp. 157-186, 1997.

[25] J. Y. Wong, Theory of Ground Vehicles, New York, USA: John Wiley & Sons, Inc, 1993.

[26] H. Peng and M. Tomizuka, "Lateral Control of Front-Wheel-Steering Rubber-Tire Vehicles," PATH Report, UCB-ITS-PRR-90-5, University of California at Berkeley, USA, 1990.

[27] W. Langson and A. Alleyne, "Multivariable bilinear vehicle control using steering and individual wheel torques," *Journal of Dynamics Systems,* vol. 121, no. Measurement and Control, pp. 631-637, 1999.

[28] W. Arnold and A. Laub, "Generalized eigenproblem algorithms and software for algebraic Riccati equations," *Proceedings of IEEE,* vol. 72, 1984.

[29] K. Atkinson, "Modelling a road using spline interpolation," Department of Mathematics, University of Iowa, 2002.

[30] L. Shampine and M. Gordon, Computer Solution of Ordinary Differential Equations: The Initial Value Problem, San Francisco: W.H. Freeman, 1975.

[31] S. Colas, "Convex Optimization Approach To Time-Optimum Path Planning," 2009.

[32] Q. T. Dinh and M. Diehl, "An Application of Sequential Convex Programming to Time Optimal Trajectory Planning for a Car Motion," in *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference,* Shanghai, P.R. China, 2009.

[33] F. Braghin, S. Melzi, E. S. F. Cheli and C. F. Cheli, "Race Driver Model," Politecnico di Milano, Department of Mechanics, Milan, 2008.

[34] A. Alleyne, "A Comparison of Alternative Obstacle Avoidance Strategies for Vehicle Control," *Vehicle System Dynamics,* vol. 27, pp. 371-392, 1997.