ISSN 0280-5316
ISRN LUTFD2/TFRT--5797--SE

Vehicle Path Optimisation

Henrik Danielsson

Department of Automatic Control
Lund University
June 2007

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
June 2007

Document Number
ISRNLUTFD2/TFRT--5797--SE

Author(s)
Henrik Danielsson

Supervisor
Magnus Gäfvert and Johan Andreasson at Modelon
In Lund. Johan Åkesson Automatic Control in Lund.
Anders Rantzer Automatic Control in Lund (Examiner)

Sponsoring organization

Title and subtitle
Vehicle Path Optimisation (Banoptimering för fordon)

Abstract
Optimal paths are useful when testing the performances of vehicles. In this thesis, optimal paths for vehicle models created in Modelica have been obtained. The optimal paths have been obtained by applying the theory of optimal control. The optimal control problem has been formulated with the Optimica language and has been solved with AMPL and IPOPT. Two approaches for finding optimal paths have been applied. In the first approach the vehicle models are dependent of time, and the time it takes for the vehicle to traverse a specified track is minimized. Finding optimal paths when using this approach has been difficult. In the second approach the vehicle models have been transformed to be dependent of distance instead of time before finding the optimal path. In this approach the cost function is formulated in a different way, and the cost function is minimized over travelled distance. Optimal paths for an one-track vehicle model with linear tyre characteristics driving on specified tracks have been found with both approaches. Some optimal paths have also been found for an one-track vehicle model with nonlinear tyre characteristics.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
english

Number of pages
68

Recipient's notes

Security classification

Acknowledgements

The topic of this thesis has been chosen upon a request from a company called Modelon AB. I would like to thank Magnus Gäfvert and Johan Andreasson at Modelon for excellent supervision. I would also like to thank Johan Åkesson at the department of Automatic Control at Lund University for contributing with the The Optimica Compiler and for excellent supervision.

Contents

Chapter 1

Introduction

The study of vehicle dynamics has interested many people for decades. Plenty of studies with the objectives of improving the safety and the performances of vehicles have been published. Especially in race car sports the performances of the cars are important since the professional race car drivers operate at the limits of their stability envelope. In the studies concentrated on race car sports, optimal trajectories have been obtained with different optimisation techniques [2], [8], [9]. The purpose is usually to minimize the time it takes for a vehicle to drive through a segment of a track [2], [8], [9]. In some studies other variables than time are minimized or maximized. The velocity at the exit of a corner is maximized in [9]. In [8] the tyre forces are maximized. In [8] and [9] the optimisations are done over a short distance, while in [2] the optimal path is tried to be found for an entire lap of a race track but for different segments of the track at a time. The optimal path is useful when evaluating the performances of a race car. Race car models with different parameter sets can be simulated driving along the optimal path. Conclusions concerning what parameter set gives the best performances can then be made. Testing the vehicles in different simulation tests saves time and money compared to testing the vehicles in reality.

1.1 Objective

The main purpose of this thesis is to find an optimal path for a vehicle model driving on a specified track. By following the optimal path, the time it takes for the vehicle to traverse the track is minimized. In order to achieve the main goal, a method for finding the optimal path is needed to be introduced.

1.2 Method

The Modelica language and the Optimica language have been used trough out this project. Modelica is an object-oriented language for physical modeling [6]. The Optimica language is an extension of the Modelica language [11]. The optimal path will be found for vehicle models created in Modelica. In order to find the optimal path, the problem is formulated as an optimal control problem in Optimica. The optimal control problem is solved with AMPL and IPOPT [3].

1.3 Thesis Outline

The thesis begins with describing the modeling of the vehicles in Chapter 2. In Chapter 3 the optimal control problem is formulated and different approaches for solving the problem are discussed. In Chapter 4, results from different optimisation cases are presented. The thesis is finished with conclusions and a section discussing future work in Chapter 5.

1.4 Notations

a_x	Longitudinal acceleration of vehicle
a_y	Lateral acceleration of vehicle
B	Stiffness factor in the Magic Formula
c	General constraints
C	Shape factor in the Magic Formula
CG	Centre of gravity of vehicle
$C_f, (C_r)$	Cornering stiffness constant for front (rear) wheel
d	Distance between vehicle and centre of track
D	Peak value in the Magic Formula
E	Curvature factor in the Magic Formula
$f, (r)$	Distance from centre of gravity to front (rear) axis
$F_{x,f}, (F_{x,r})$	Longitudinal force acting on front (rear) wheel
$F_{xmax,f}, (F_{xmax,r})$	Maximum longitudinal force acting on front (rear) wheel
$F_{y,f}, (F_{y,r})$	Lateral force acting on front (rear) wheel
$F_{ymax,f}, (F_{ymax,r})$	Maximum lateral force acting on front (rear) wheel
F_z	Vertical load
h	Height of the centre of gravity
J	Cost function
J_z	Yaw Inertia
k_t	Curvature of the centre of the track
K_{us}	Under steer gradient
m	Mass of vehicle
$\overline{r}$	Direction vector of vehicle
R_e	Effective rolling radius
r_t	Radius of the centre of the track
s	Longitudinal slip
S_{cf}	Time to distance scaling factor
s_t	Distance from the start point of the track
t	Time
u	Vector of continuous control variables
u_L	Lower control bound
u_U	Upper control bound
v_x	Longitudinal velocity of vehicle
v_y	Lateral velocity of vehicle
v_{wx}	Velocity of wheel in the x-direction
v_{wy}	Velocity of wheel in the y-direction
w_t	Width of the track
x	General vector of continuous system state variables
x_c	The x-coordinate of the centre of the track

x_{cds}	The x-coordinate of the centre of the track at $s_t + d_s$
x_{pos}	Global x position of vehicle
x_{vel}	Global x velocity of vehicle
y_c	The y-coordinate of the centre of the track
y_{cds}	The y-coordinate of the centre of the track at $s + d_s$
y_{pos}	Global y position of vehicle
y_{vel}	Global y velocity of vehicle
α_f, α_r	Front (rear) lateral slip angle
δ	Steering angle of vehicle
δ_{ref}	Steering angle of driver model
μ	Friction coefficient between the tyre and road surface
ω_w	Angular velocity of wheel
ψ	Yaw angle of vehicle
ψ_t	The angle between the tangent of the centre of the track and the x-axis

Chapter 2

Vehicle Modeling

One of the larger parts of this thesis has been modeling of vehicles. The vehicle models have been used when solving the optimisation problem. The models have been created in a modeling language called Modelica. A short description of Modelica will begin this chapter. The chapter will then continue with describing different models of tyres and chassis. Since the tyres generate the forces that acts on the vehicle, the modeling of the tyres will be considered before the modeling of the chassis. Discussion concerning vehicle handling will finish the chapter.

2.1 Modelica

Modelica is an object-oriented modeling language, used for describing complex physical models. Models in Modelica are mathematically described by differential, algebraic and discrete equations. The Modelica language is free to use and is developed by a non-profit organisation called the Modelica Association [6].

2.2 Tyre Modeling

The forces acting on a vehicle when traction, braking or steering occur are generated by its tyres. Therefore the tyres are very important for several functions and properties of the vehicle. Tyre modeling is difficult and several different tyre models exist. Some of them are based on physical models and some of them are based on empirical models. In order for a tyre to generate force, slip has to occur. These forces are nonlinear in reality but can be modeled as linear functions of the slip angles. Under normal driving conditions it is sufficient to assume the forces to be linear. During more extreme driving conditions these forces will saturate and become nonlinear. There are three different types of slip: lateral, longitudinal and spin. Lateral slip will be considered first.

2.2.1 Lateral slip

Lateral slip occurs when the vehicle is cornering. The definition of the lateral slip angle, α is:

$$\tan \alpha = -v_{wy}/v_{wx} \tag{2.1}$$

where v_{wy} is the velocity of the wheel in the y-direction, while v_{wx} is the velocity of the wheel in the x-direction, see Figure 2.1.

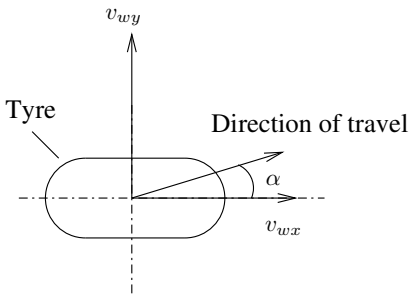

Figure 2.1: Definition of slip angle

For a linear tyre model the lateral forces acting on the vehicle are defined as:

$$F_{y,f} = -C_f \alpha_f \tag{2.2}$$

$$F_{y,r} = -C_r \alpha_r \tag{2.3}$$

where $F_{y,f}$ is the lateral force acting on the front wheel and $F_{y,r}$ is the lateral force acting on the rear wheel. C_f and C_r are cornering stiffness constants.

For a nonlinear tyre model, the lateral forces will saturate for large slip angles, shown in Figure 2.2.

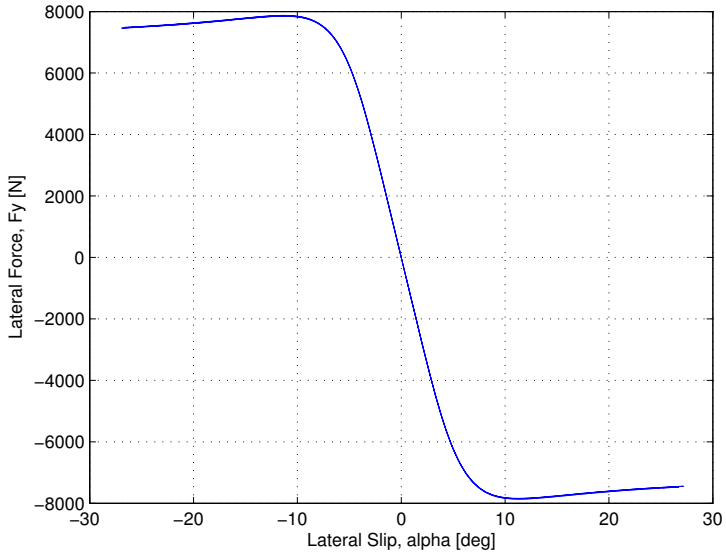

Figure 2.2: The relation between the lateral force, F_y and the lateral slip, α for a nonlinear tyre model.

Nonlinear tyre forces are often described with the Magic Formula [7]. This is an empirical method that has been developed from measuring tyre characteristics. For the lateral case the formula is defined as follows:

$$F_y = D\sin(C\arctan(B\alpha - E(B\alpha - \arctan(B\alpha)))) \tag{2.4}$$

where:

B is the stiffness factor
C is the shape factor
D is the peak value
E is the curvature factor

The factor B determines the slope at the origin of the curve. The coefficient C defines the extent of the sine function and therefore determines the shape of the curve. The coefficient D represents the peak value of the curve. The factor E controls the curvature at the peak and the horizontal position of the peak.

2.2.2 Longitudinal slip

Longitudinal slip occurs when the wheel is subjected to an external driving or braking moment, which will make the angular velocity of the wheel, ω_w different from when free-rolling. When the wheel is free-rolling, its angular velocity, ω_w is defined as the ratio between the wheel centre velocity in the x-direction, v_{wx} and the effective rolling radius, R_e;

$$\omega_w = v_{wx}/R_e \tag{2.5}$$

There are several definitions of longitudinal slip, s. In [10] the following definition is used:

$$s = \frac{R_e\omega_w - v_{wx}}{v_{wx}} \tag{2.6}$$

The following values for ω_w and s are obtained with definition 2.5 and 2.6:

Locked wheel $\omega_w = 0, \quad s = -1$
Free-rolling wheel $\omega_w = v_{wx}/R_e, \quad s = 0$
Spinning wheel $\omega_w = 2v_{wx}/R_e, \; s = 1$

The longitudinal force, F_x can be computed by the Magic Formula as well, but with the longitudinal slip, s as an input instead.

2.2.3 Combined slip

In some situations both lateral slip and longitudinal slip occur, for instance when cornering and braking at the same time. Figure 2.3 illustrates a friction ellipse, which is based on combined slip. The friction ellipse describes the dependency between lateral force and longitudinal force. The friction ellipse is defined as:

$$(\frac{F_x}{F_{xmax}})^2 + (\frac{F_y}{F_{ymax}})^2 = 1 \tag{2.7}$$

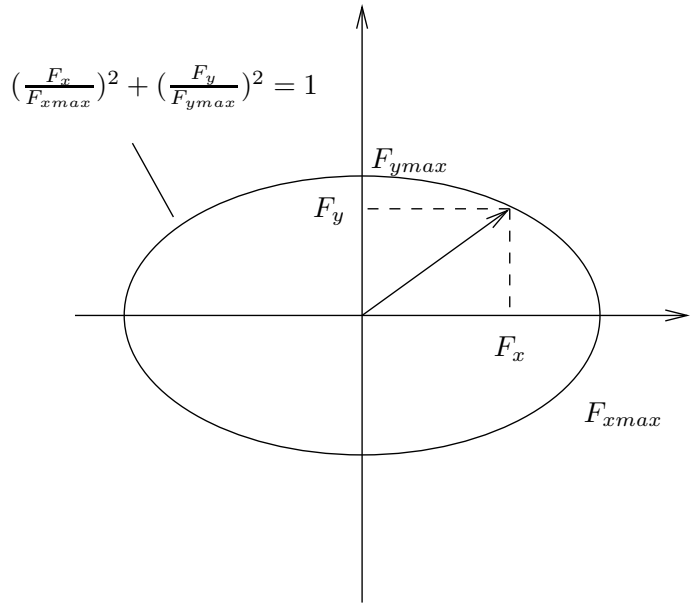

Figure 2.3: The friction ellipse

The idea is that a resultant force of F_x and F_y lies on the ellipse. F_x cannot exceed the maximum longitudinal force, F_{xmax} and F_y cannot exceed the maximum lateral force, F_{ymax}. F_{xmax} may be defined as:

$$F_{xmax} = \mu F_z \tag{2.8}$$

where μ is a coefficient of friction between the tyre and the road surface, and F_z is the vertical load.

2.3 Chassis Modeling

One of the simplest chassis models is the bicycle model. The bicycle model is a two dimensional vehicle model, see Figure 2.5. Basically it has only one wheel on each axis, which means that only planar motion in the earth fixed x-y plane is considered. To be able to derive the equations of motion it is necessary to introduce one additional coordinate system, the vehicle frame, which origin is located at the centre of gravity of the vehicle, as illustrated in Figure 2.4.

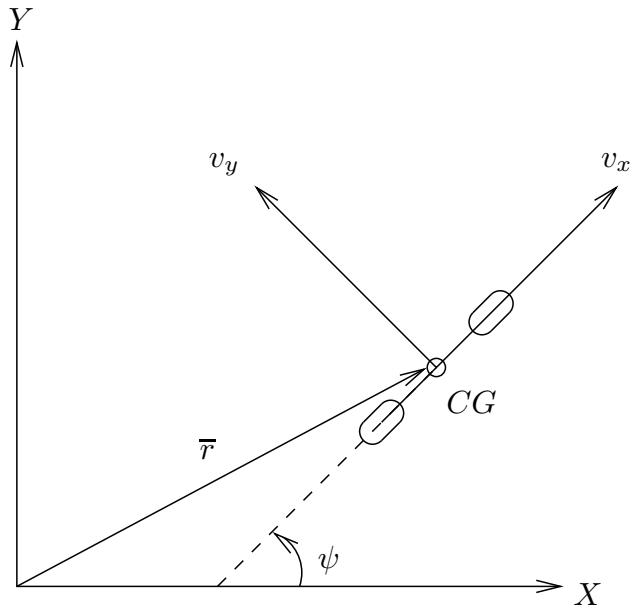

Figure 2.4: The earth fixed frame and the vehicle frame. The vehicle frame is rotated with ψ around the z-axis.

The yaw angle, ψ is the angle between the two coordinate systems. The equations of motion of the model originate from Newton's second law, $F = ma$. Lateral acceleration occurs while driving and steering, this is compensated for by forces acting on the vehicle. The acceleration of the vehicle can be obtained by derivation of the velocity vector of the vehicle, $\dot{\overline{r}}$. The velocity vector is denoted as:

$$\dot{\overline{r}} = \overline{v} = (v_x, v_y, 0) \tag{2.9}$$

and the yaw velocity of the vehicle, $\dot{\psi}$ as:

$$\overline{\omega} = (0, 0, \dot{\psi}) \tag{2.10}$$

The following derivation rule is applied when differentiating $\dot{\overline{r}}$:

$$\ddot{\overline{r}} = \frac{d}{dt}\dot{\overline{r}} = \dot{\overline{v}} + \overline{\omega} \times \overline{v} \tag{2.11}$$

$$\ddot{\overline{r}} = \begin{Bmatrix} \dot{v_x} \\ \dot{v_y} \\ 0 \end{Bmatrix} + \begin{Vmatrix} \hat{x} & \hat{y} & \hat{z} \\ 0 & 0 & \dot{\psi} \\ v_x & v_y & 0 \end{Vmatrix} = \begin{Bmatrix} \dot{v_x} - \dot{\psi} v_y \\ \dot{v_y} + \dot{\psi} v_x \\ 0 \end{Bmatrix} \tag{2.12}$$

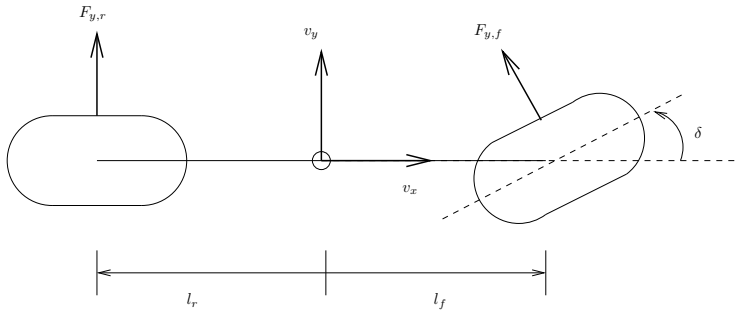

Figure 2.5: The bicycle model.

The equation of motion can now be written as:

$$\uparrow m(\dot{v_x} - \dot{\psi}v_y) = -F_{y,f}\sin\delta \tag{2.13}$$

$$\rightarrow m(\dot{v_y} + \dot{\psi}v_x) = F_{y,r} + F_{y,f}\cos\delta \tag{2.14}$$

$$J_z\ddot{\psi} = l_f F_{y,f}\cos\delta - l_r F_{y,f} \tag{2.15}$$

where J_z is the moment of inertia of the vehicle and $\ddot{\psi}$ is the angular acceleration of the vehicle around the z-axis. A more thoroughly derivation of the equations of motion for the bicycle model can be found in [5].

As shown in Section 2.2.1, the lateral forces, $F_{y,f}$ and $F_{y,r}$ are dependent of the lateral slip angles, α_f and α_r, while the slip angles are dependent of the velocities of the wheels. The expressions for the slip angles can be rewritten as functions of the velocities of the vehicle instead. This makes more sense when dealing with the lateral forces in the equations of motion. In order to accomplish this, the velocity vectors of the wheels need to be derived. The velocity vector of the front wheel, $\dot{\overline{r_f}}$ and the velocity vector of the rear wheel, $\dot{\overline{r_r}}$ can be obtained as follows:

$$\dot{\overline{r_f}} = \dot{\overline{r}} + \frac{d}{dt}\overline{l_f} + \overline{\omega}\times\overline{l_f} = \begin{Bmatrix} v_x \\ v_y \\ 0 \end{Bmatrix} + \begin{Vmatrix} \hat{x} & \hat{y} & \hat{z} \\ 0 & 0 & \dot{\psi} \\ l_f & 0 & 0 \end{Vmatrix} = \begin{Bmatrix} v_x \\ v_y + \dot{\psi}l_f \\ 0 \end{Bmatrix} \tag{2.16}$$

$$\dot{\overline{r_r}} = \dot{\overline{r}} + \frac{d}{dt}\overline{l_r} + \overline{\omega}\times\overline{l_r} = \begin{Bmatrix} v_x \\ v_y \\ 0 \end{Bmatrix} + \begin{Vmatrix} \hat{x} & \hat{y} & \hat{z} \\ 0 & 0 & \dot{\psi} \\ -l_r & 0 & 0 \end{Vmatrix} = \begin{Bmatrix} v_x \\ v_y - \dot{\psi}l_r \\ 0 \end{Bmatrix} \tag{2.17}$$

It is now possible to write the lateral slip angles for the front and the rear wheel as functions of the velocities of the vehicle:

$$\alpha_f = \arctan(\frac{v_y + \dot{\psi}l_f}{v_x}) - \delta \tag{2.18}$$

$$\alpha_r = \arctan(\frac{v_y - \dot{\psi}l_r}{v_x}) \tag{2.19}$$

2.4 Track Modeling

Two different tracks have been used during this project. These two tracks have been modeled in Modelica and the coordinates of the tracks are arbitrarily imposed. A trajectory has been defined for each of the two tracks, this trajectory represents the centre of the track, shown in Figure 2.6. The trajectory, $(x_c(s_t), y_c(s_t))$ is represented as a function of $s_t(t)$, while $s_t(t)$ is a function of time. This means $s_t(t)$ increases over time, while the trajectory increases over $s_t(t)$. x_c and y_c are the global x- and y-positions of the centre of the track.

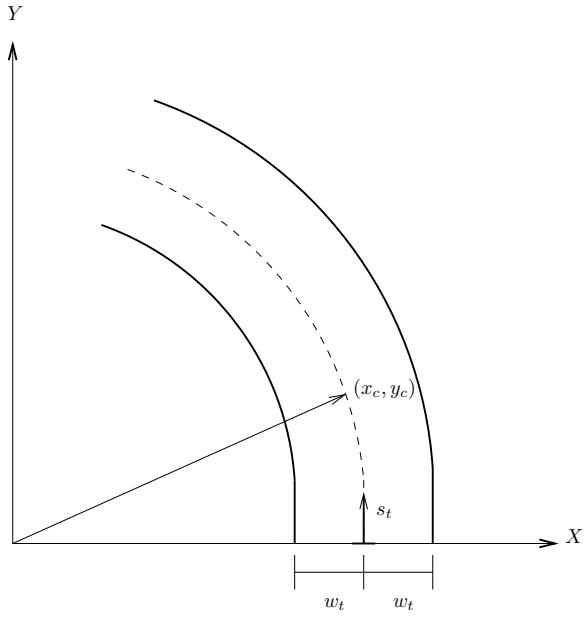

Figure 2.6: The track model.

The first track used was ellipse-shaped, see Figure 2.7. The coordinates was chosen in a way so the ellipse became twice as long as it is wide. The total length of the track is 450 meters. The centre of the ellipse-shaped track is defined as:

$$(x_c, y_c) = (45\cos(s_t), 90\sin(s_t)) \tag{2.20}$$

Further on a circular-shaped track with a sinus curvature added to it was introduced. As seen in Figure 2.8, this track has more corners than the previous one. This track is supposed to be more challenging for the vehicle model. The length of this track is about 1400 meters, which makes it more than three times further than the ellipse-shaped track. The coordinates of the centre of the track are defined as:

$$(x_c, y_c) = ((200+40\sin(4s_t-\pi/2))\cos s_t, (200+40\sin(4s_t-\pi/2))\sin s_t) \tag{2.21}$$

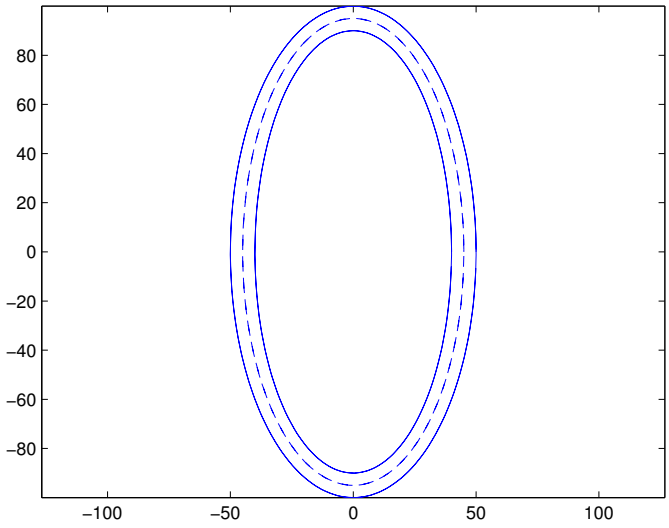

Figure 2.7: The ellipse-shaped track.

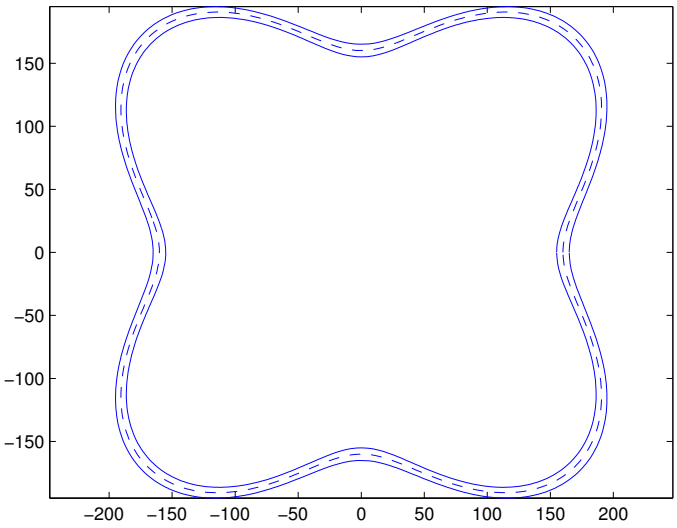

Figure 2.8: The flower-shaped track.

2.5 Time to Distance Transformation

In Section 2.3, the equations of motion of the vehicle model were derived with respect of time. In this section the equations will be derived with respect of travelled distance instead. In order to achieve this, a time to distance scaling factor, S_{cf} is needed to be introduced. In [2] the transformation from time to distance has been done with a scaling factor defined as:

$$S_{cf} = \frac{dt}{ds_t} = \frac{1 - \frac{d}{r_t}}{v_x \cos(\psi_v - \psi_t) - v_y \sin(\psi_v - \psi_t)} \tag{2.22}$$

Where d is the shortest distance between the vehicle and the centre of the track.

$$d = (y_{pos} - y_c)\cos(\psi_t) - (x_{pos} - x_c)\sin(\psi_t) \tag{2.23}$$

ψ_t is the angle between the tangent of the centre of the track and the x-axis. ψ_t can be defined as:

$$\psi_t = \arccos(\frac{(\dot{x_c}, \dot{y_c})(1,0)}{|(\dot{x_c}, \dot{y_c})||(1,0)|}) = \arccos(\frac{\dot{x_c}}{\sqrt{\dot{x_c}^2 + \dot{y_c}^2}}) \tag{2.24}$$

r_t is the radius of the centre of the track and can be written as $r_t = 1/k_t$, where k_t is the curvature of the centre of the track and defined as follows:

$$k_t = \frac{|(\dot{x_c}, \dot{y_c}) \times (\ddot{x_c}, \ddot{y_c})|}{|(\dot{x_c}, \dot{y_c})|^3} \tag{2.25}$$

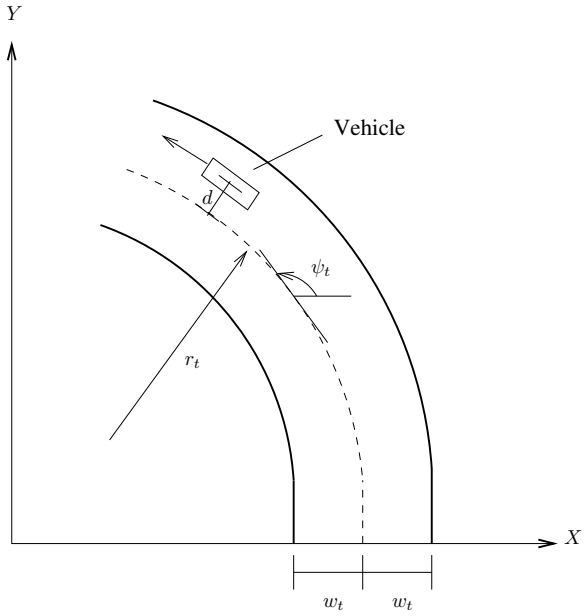

Figure 2.9: The track model.

Consider a model dependent of time with the following definition:

$$\dot{x} = f(x(t), u(t), t),\ x(0) = x_0,\ t \in [0, t_f]$$

When the model is transformed to be dependent of travelled distance, it will be defined as:

$$\frac{dx}{ds_t} = S_{cf} f(x(s_t), u(s_t), s_t),\ x(0) = x_0,\ s_t \in [0, s_{tf}]$$

2.6 The Driver Model

A quite simple driver model has been introduced with the purpose of obtaining a first initial guess for the optimisation problem. The driver model is supposed to make the vehicle follow a specified trajectory, in this case the centre of the track. The driver model consists of a type of steering control but doesn't have an acceleration control, the vehicle is supposed to have constant velocity. The steering control model spots a point on the trajectory a few meters ahead and simply tells the vehicle to steer towards this point. This can be compared to how a human driver would steer the vehicle, he looks at the road a few meters ahead and steers in order to keep the vehicle on the road. The position of the vehicle in the world frame is (x_{pos}, y_{pos}) and the point a few meters ahead of the vehicle is (x_{cds}, y_{cds}). The vector between these points is $(x_{cds} - x_{pos}, y_{cds} - y_{pos})$ and the direction of the vehicle is denoted as (x_{vel}, y_{vel}). The angle between these two vectors is equal to the steering angle, as illustrated in Figure 2.10.

$$\delta_{ref} = \arccos \frac{(x_{cds} - x_{pos})x_{vel} + (y_{cds} - y_{pos})y_{vel}}{\sqrt{(x_{cds} - x_{pos})^2 + (y_{cds} - y_{pos})^2} + \sqrt{x_{vel}^2 + y_{vel}^2}} \tag{2.26}$$

The vehicle model is then simulated with a constant velocity and with δ_{ref} as an input for the steering angle. The simulated result can then be used as an initial guess for the first optimisation run.

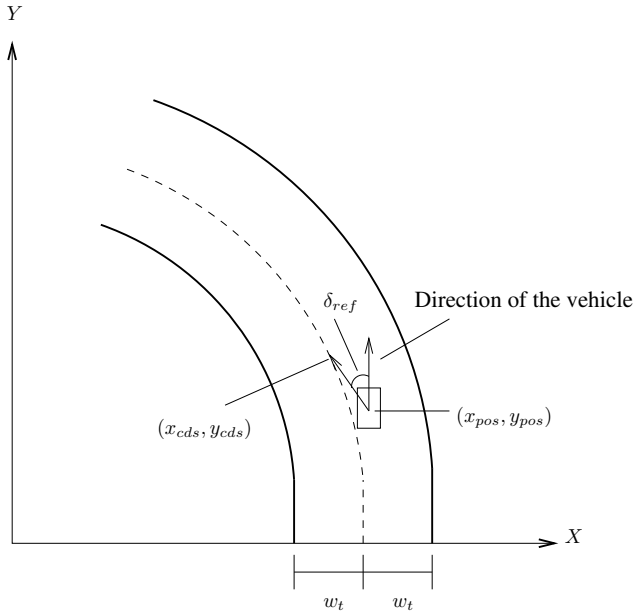

Figure 2.10: The driver model

2.7 Vehicle Handling

Vehicle handling describes the way vehicles perform transverse to their direction of motion, particularly during cornering. When dealing with planar vehicle models, the focus lies on yaw stability and response to steering input. An under steer gradient is often measured or computed when describing the steering characteristics of a vehicle. The under steer gradient, K_{us} is defined as:

$$K_{us} = \frac{\partial \delta}{\partial a_y} - \frac{\partial \delta_A}{\partial a_y} \tag{2.27}$$

where δ_A is called the Ackerman angle and is derived from a pure geometrical approach. An under steer vehicle has a positive under steer gradient ($K_{us} > 0$) and an over steer vehicle has a negative under steer gradient ($K_{us} < 0$). The steering characteristics of a vehicle can be tested, by driving the vehicle around a circle, while the vehicle constantly accelerates. Figure 2.11 illustrates the result from a test case where a bicycle model with linear tyre properties is simulated driving in a circle with a given radius. The lateral acceleration is plotted against the steering angle. The solid line represents an under steer bicycle model while the dashed line represents an over steer model. The under steer model was used later on in the optimisations. For an under steer vehicle the slip angles of the front wheels are larger than the slip angles of the rear wheels. This means the driver usually needs to compensate for this by steering harder. For an over steer vehicle it is the opposite, the driver should not steer as much and the rear wheels have larger slip angles than the front wheels.

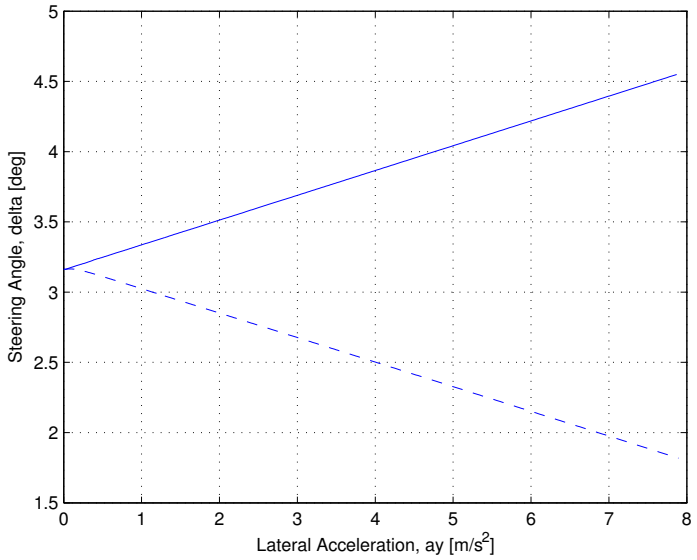

Figure 2.11: The relation between steering angle, δ and lateral acceleration, a_y. The solid line shows an under steer vehicle model while the dashed line shows an over steer vehicle model.

For a bicycle model with linear tyres, the steering characteristics depend on two factors, the cornering stiffness constants of the tyres and the place of the centre of gravity of the vehicle. The bicycle model is neutral steer if the cornering stiffness constants of the front and rear wheels are equal while the centre of gravity of the vehicle is placed in the middle.

$C_f < C_r$ -> under steer
$C_f = C_r$ -> neutral steer
$C_f > C_r$ -> over steer

$l_f < l_r$ -> under steer
$l_f = l_r$ -> neutral steer
$l_f > l_r$ -> over steer

Another way to illustrate the steering characteristics could be done by plotting the lateral forces $F_{y,f}$ and $F_{y,r}$ against the slip angles, α_f and α_r, see Figure 2.12.

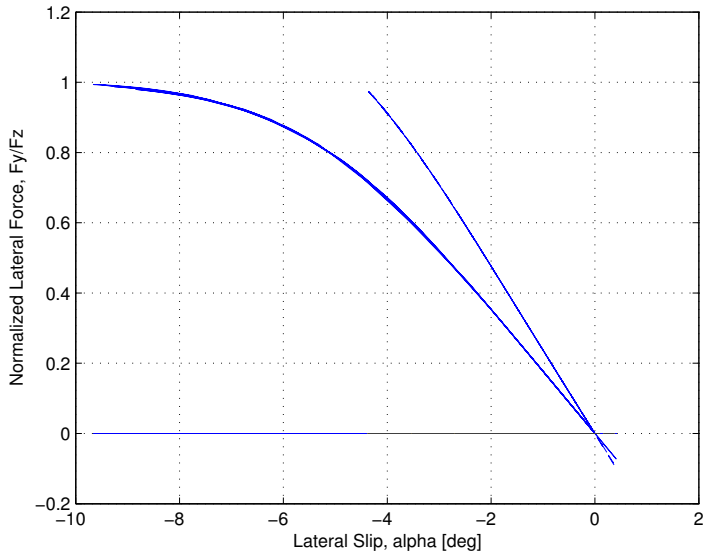

Figure 2.12: The relation between the normalized lateral forces, $F_{y,f}$ and $F_{y,r}$ and the lateral slip angles, α_f and α_r

The diagram shows the normalized lateral forces of the tyres plotted against the lateral slip angles of the tyres. The solid curve represents the front tyre while the dashed curve represents the rear tyre. The diagram shows a bicycle model with nonlinear tyre properties, which was used later on in the optimisations. This particular model is under steer during the entire simulation run, since the curve of the rear tyres is above the curve of the front tyres. If the curve of the rear tyres is below the curve of the front tyres at any point then the vehicle is over steer at that point.

Chapter 3

Optimisation

The following chapter will begin with a short description of an optimal control problem. AMPL and Optimica will be described briefly in the next section. Finally two different approaches for solving the optimal control problem will be introduced.

3.1 The Optimal Control Problem

The objective of this project has been to find an optimal path that will minimize the time for a vehicle model travelling along a specified track. The optimal path can be found by using the theory of optimal control. In optimal control the objective is to minimize or in some cases maximize a cost function. An optimal control problem is typically formulated as follows:

$$\min_{u,p} J = \min \int_0^{t_f} L((x(t), u(t), p)dt \tag{3.1}$$

$$\begin{aligned} \text{subject to: } & \dot{x} = f(x(t), u(t)) && x(0) = x_0 \\ & c(t) = c(x(t), u(t), t) \leq 0 \\ & u_L \leq u(t) \leq u_U \\ & \text{for all } t \in [t_0, t_f] \end{aligned}$$

where J is the cost function and

$$\dot{x} = f(x(t), u(t)),\ x(0) = x_0,\ t \in [t_0, t_f]$$

is a mathematical model of the system. In our case, this is the vehicle model.

$$c(x(t), u(t), t) \leq 0 \tag{3.2}$$

are the physical constraints of the optimal control problem. Physical constraints arise from limitations on control variables and limitations on the state variables of the model. For instance in the minimum time problem, there are constraints on the longitudinal and lateral acceleration of the vehicle and on the steering angle. Other constraints come from the fact that the vehicle must stay within the track boundaries.

$$u_L \leq u(t) \leq u_U \tag{3.3}$$

are the constant control bounds.

An optimal control, u^* and its corresponding optimal state trajectory, x^* are found when the cost function is minimized.

When dealing with a minimum time problem, the cost function takes the following form:

$$\min J = \min \int_0^{t_f} 1dt \tag{3.4}$$

There are several methods for solving an optimal control problem. These methods can be divided into two different main methods, direct and indirect methods. When applying direct methods, the optimal control problem is converted into a nonlinear programming problem and is solved directly using mathematical programming techniques. The continuous control history is replaced with a discrete approximation. This means that the control input can only be adjusted at a fixed number of positions along the trajectory, while the values in between the points are estimated by interpolation. The Optimica Compiler uses a simultaneous method, known as the direct collocation method. This method fully discretize the state and the control variables, which leads to large-scale NLP problems. Simultaneously methods are discussed in [1]. AMPL and an external solver, called IPOPT have been used when solving the nonlinear programming problem.

3.2 AMPL

AMPL is a high-level, mathematical programming language, used for describing and solving large scale optimisation problems. AMPL expresses the symbolic algebraic notation familiar to people in a way that can serve as direct input to a computer system [3]. AMPL does not solve the problem directly, instead it uses an external solver such as IPOPT. AMPL handles linear and nonlinear problems.

3.3 Optimica

Optimica is an extension of the Modelica language and the Optimica Compiler is an optimisation tool developed for Modelica. The Optimica language admits formulation of dynamic optimisation problems on the following form, according to [11]:

$$\min \int_0^{t_f} L((x(t), u(t), p)dt \tag{3.5}$$

$$\begin{aligned} \text{subject to: } & f(\dot{x}, x, u, p) = 0 \\ & c_i(x(t), u(t), p) \leq 0 \\ & c_e(x(t), u(t), p) = 0 \\ & c_f(x(t_f), u(t_f), p) = 0 \end{aligned}$$

The cost function and the constraints are written in an Optimica file. This Optimica file consists of three sections. In the first section the variable bounds are specified. The next section contains the cost function and the optimisation horizon. The last section contains all the constraints. The Optimica Compiler

compiles the Optimica file and the Modelica file, containing the model representation and generates a set of new files containing AMPL code. The problem is then solved using AMPL, which in turn invokes the numerical solver.

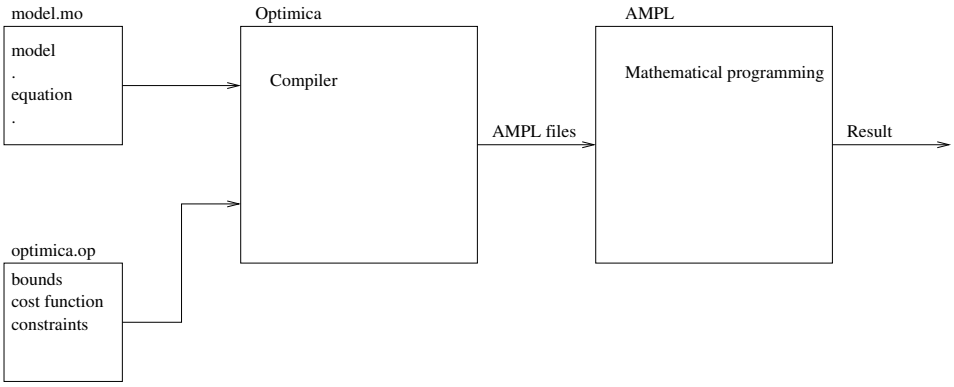

Figure 3.1: Flow chart for AMPL and Optimica.

3.4 Solving the Optimal Control Problem

The purpose has been to minimize the driving time for a vehicle model travelling along a specific track. Solving a minimum time problem like this one is very difficult. It is necessary to specify a good initial guess, in order to find an optimal solution. The initial guess needs to be close to the optimal solution. One approach for obtaining an initial guess is to come up with a driver model, which takes the vehicle around the track in a way that is close to the optimal way. A driver model that accomplish this will be quite complex [2]. Another approach for finding a good initial guess is to minimize another cost function for a fixed final time. For instance could the control inputs be minimized for a given time. The result obtained can then be used as an initial guess when solving the minimum time problem. This is a common method when solving minimum time problems and has been used in this project as well.

The driver model introduced in Section 2.6 is used for producing an initial guess for the optimisation case where the control inputs are minimized. This driver model is quite simple but good enough for taking the vehicle around the specified track. The main limitation of the driver model is the lack of an acceleration control, the driver model has constant velocity through out the track. The vehicle is simulated driving through the track with the specified driver model. The result obtained from the simulation will then be used as a first initial guess for the optimal control problem. This initial guess is too poor for solving the minimum time problem but is sufficient for minimizing the control inputs for a given time. This given time will be the same as the time it took for the driver model to drive around the track. The cost function is defined as:

$$\min \int_0^{t_f} u_1(t) + u_2(t) dt \tag{3.6}$$

where $u_1(t)$ and $u_2(t)$ are the control inputs, in most cases the longitudinal

acceleration, a_x and the steering angle, δ of the vehicle. t_f is the fixed final time. The optimal solution is put in a result file and this result will be used as an initial guess for the next optimisation. In the next optimisation the cost function will stay the same while the fixed final time, t_f is decreased. This procedure will keep going until t_f is sufficiently close to the optimal time. The optimal time is obviously not known at this point but one can have an hint of what it is. The cost function is then redefined in a way so the time is minimized. When minimizing the time, the result from the last optimisation of the control inputs is used as an initial guess. Hopefully this initial guess is good enough for finding an optimal solution for the minimum time case.

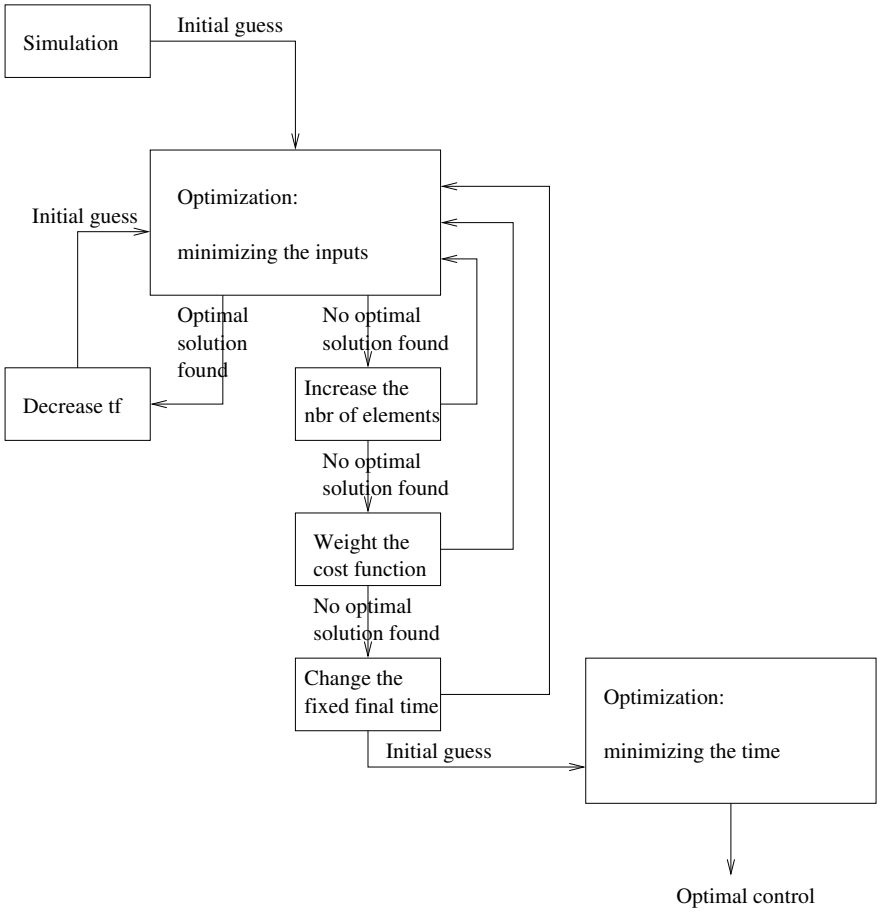

Figure 3.2: Flow chart for the optimisation procedure.

The procedure described above might not always behave as desired. If AMPL doesn't find an optimal solution during any of the optimisation runs, something needs to be changed. There are a few different possibilities for continuing when no optimal solution is found. The number of elements of the grid can be respecified, a larger number will improve the accuracy, which may lead to an optimal solution. If an optimal solution is still not found, the next step is to change the

cost function by penalizing one of the terms. For instance could the original cost function be changed into:

$$\min \int_0^{t_f} 10 * u_1(t) + u_2(t) dt \tag{3.7}$$

which means that the first control input is now more heavily penalized. If the search of an optimal solution fails once again, the fixed final time can be changed. Let's say that an optimisation case with the fixed final time, $t_f = 60$ is run and AMPL finds an optimal solution. The next step is to decrease t_f, for instance to $t_f = 56$ and then run a new optimisation with the same cost function. If an optimal solution isn't found, then it might be needed to run an optimisation for $t_f = 58$ first and then run the optimisation for $t_f = 56$, which might result in an optimal solution. The result from $t_f = 58$ is a better initial guess than the result from $t_f = 60$ when trying to solve the case for $t_f = 56$.

When all the above actions are carried through and an optimal solution is still not found then the problem cannot be solved because of the fixed final time, t_f is being unfeasible without violating the constraints. This also means that the last optimal solution found is close to the optimal solution for the minimum time case and will be a good initial guess for solving the minimum time problem.

3.5 Transforming the Optimal Control Problem

In the previous section, the optimal control problem was set up with the equations of the vehicle model dependent of time. In this section, the optimal control problem will be set up after the transformation from time to distance has been applied, which was described in 2.5. The equations of the vehicle model will be dependent on travelled distance. The purpose is still to find an optimal path, which will minimize the time. Since S_{cf} is defined as an increment of travelled distance, ds_t divided by an increment in time, dt, minimizing the integral of S_{cf} will have the same effect as minimizing time. The cost function will be formulated as follows:

$$\min t_f = \min \int_0^{s_{tf}} S_{cf} ds \tag{3.8}$$

The cost function is now minimized over a certain distance, s_{tf} instead of over a certain time. The reason why the transformation from time to travelled distance is applied, is because it should be easier for the optimisation tools to find an optimal solution. The variable that the cost function is minimized over, in this case the distance, s_{tf} is fixed as opposed to when minimizing over time. As mentioned before it is easier to find an optimal solution when having a fixed final variable.

A good initial guess is still vital when searching for an optimal solution. When the problem was formulated with respect to time, the inputs were minimized for a fixed final time. The fixed final time was decreased between every optimisation run until the fixed final time was close enough to the minimum time. This approach cannot be used when the problem is formulated with respect to travelled distance, because the cost function is not minimized over time. Instead another approach, which is based on increasing s_{tf}, is tried. The idea is to start with a low s_{tf}, which means S_{cf} is minimized for the first part of the track, in the next optimisation run s_{tf} is increased and S_{cf} is minimized

once again. s_{tf} is then increased until it equals the entire track. Let's consider a case where the purpose is to drive a lap around the ellipse-shaped track. In the first optimisation an optimal path is found for the first part of the track. In the next optimisation an optimal solution is found when the vehicle drives a bit further on the track, the s_{tf} is increased. When solving this problem the previous optimal solution is used as an initial guess. Then s_{tf} is increased again and a new optimal solution is found, this procedure will continue until the vehicle has driven around the entire track.

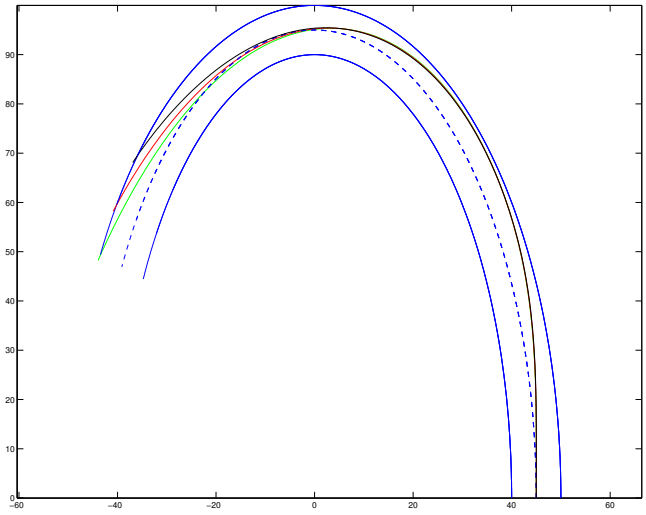

Figure 3.3: Optimal paths for different s_{tf}

Chapter 4

Optimal Path Results

Several optimisation results have been obtained, the following results will be presented in this section:

- Bicycle Model with Linear tyres
  - $\min t$
    - The Ellipse Track, 2 Laps
    - The Flower Track, 1 Lap
  - $\min S_{cf}$
    - The Ellipse Track, 2 Laps
    - The Flower Track, 1 Lap
- Bicycle Model with Nonlinear tyres
  - $\min t$
    - The Ellipse Track, 2 Laps
    - The Flower Track, 1 Lap
  - $\min S_{cf}$
    - The Ellipse Track, 2 Laps

4.1 Bicycle Model with Linear Tyres

For the first optimisation tests, a bicycle model with linear tyres was used. The parameters of the bicycle model are presented in Table 4.1. These parameters are arbitrarily imposed and not received from any real vehicle model, but the parameters are realistic. Variables with lower and upper bound constraints are presented in Table 4.2. Notice that the longitudinal velocity of the vehicle, v_x has a starting value. The entire vehicle model in Modelica can be found in Appendix A.

Parameter	
Yaw Inertia, J_z $[kgm^2]$	2800
Front Cornering Stiffness, C_f []	100000
Rear Cornering Stiffness, C_r []	150000
Distance from centre of gravity to front axle, l_f $[m]$	1.33
Distance from centre of gravity to rear axle, l_r $[m]$	1.43
Mass of vehicle, m $[kg]$	1550

Table 4.1: Parameters of the bicycle model.

Variable	*Lower Bound*	*Upper Bound*	*Start Value*
delta $[rad]$	-1	1	0
v_x $[m/s]$	0	100	10
a_x $[m/s^2]$	-10	10	0
d $[m]$	-5	5	0

Table 4.2: Variables of the bicycle model.

4.1.1 Minimizing Time

In the first optimisation case the time around an ellipse-shaped track was minimized and in the second case the time around a flower-shaped track was minimized. The cost function and the constraints for both of the two cases are formulated as follows:

Cost function:
$\min J = \min \int_0^{t_f} 1dt$

Constraints:
$a_x^2 + a_y^2 \leq 10^2$

The Ellipse Track, 2 Laps

The optimal path for the vehicle model when minimizing time is shown in Figure 4.1. The resulting control signals, a_x and δ are presented in Figure 4.2. The velocity of the vehicle and the lateral slip angles of the tyres are shown in Figure 4.3. In Figure 4.4, the lateral acceleration is plotted against the longitudinal acceleration, usually called g-g diagram. The purpose of the g-g diagram is to show how close the vehicle is to its limitations in acceleration during driving. The dashed line in Figure 4.4 illustrates the limitations in acceleration of the vehicle, noticeable is that the vehicle is extremely close to its limitations the entire run. In Figure 4.2 it can be seen that the steering angle, δ decreases very quickly at the very end of the run. Since it doesn't matter how the vehicle behave after the run is completed, the final value of the steering angle is unimportant. This also explains the straight line in the g-g diagram.

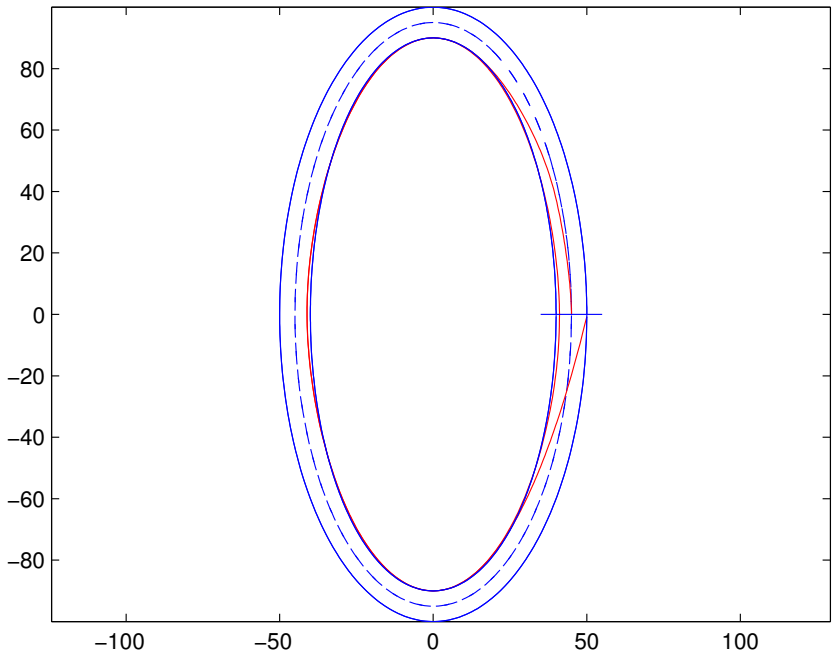

Figure 4.1: The optimal path obtained when minimizing the time, for a bicycle model with linear tyres driving 2 laps on the track.

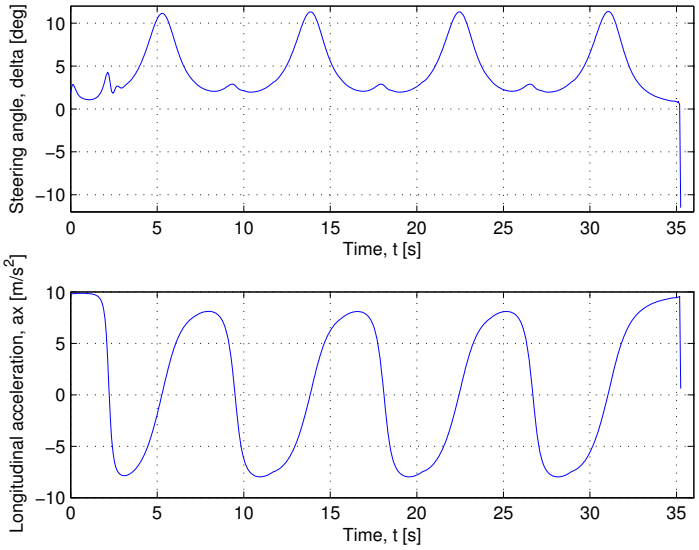

Figure 4.2: The control inputs for a bicycle model with linear tyres driving 2 laps on the ellipse-shaped track when minimizing the time. Longitudinal acceleration, a_x and steering angle, δ.

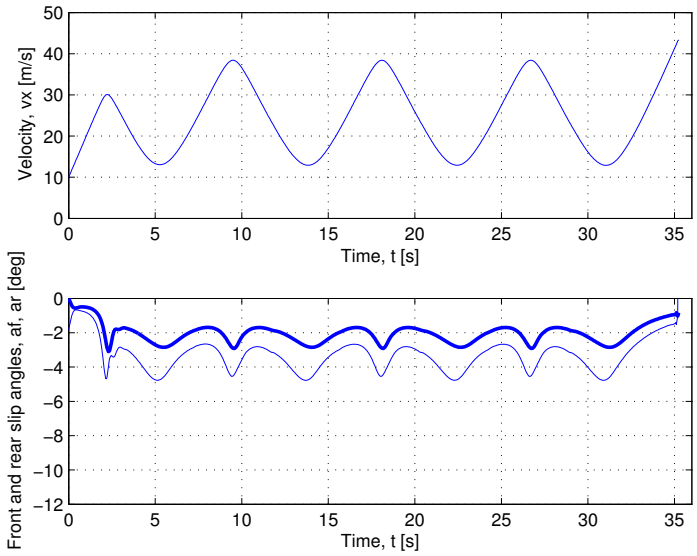

Figure 4.3: The velocity, v_x and the slip angles, α_f(thin) and α_r(thick) of the bicycle model with linear tyres driving 2 laps on the ellipse-shaped track when minimizing the time.

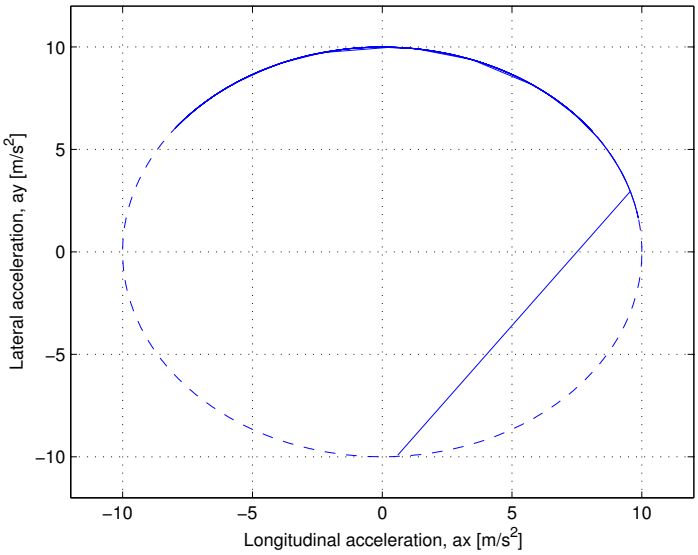

Figure 4.4: The relation between the lateral acceleration, a_y and the longitudinal acceleration, a_x of the bicycle model with linear tyres driving 2 laps on the ellipse-shaped track when the time is minimized. The dashed line shows the constraints on the accelerations.

The procedure described in Section 3.4 was applied in order to obtain the minimum time solution. The control inputs, a_x and δ were minimized for a decreasing fixed final time, t_f. After t_f was decreased a few times, the steering angle, δ started to oscillate. The result wasn't satisfying, since it was a poor initial guess for the minimum time case. Instead a_x and the derivative of δ were minimized, in addition the derivative of δ was penalized. In the received result the oscillations of the δ vanished and the result was good enough as an initial guess for the minimum time case.

The Flower Track, 1 Lap

The optimal path for the vehicle model driving one lap through the second track is shown in Figure 4.5. The acceleration, a_x and the steering angle, δ are presented in Figure 4.6. In Figure 4.7, the velocity of the vehicle and the lateral slip angles of the tyres are presented. Figure 4.8 illustrates a g-g diagram. Even for this track the vehicle is very close to its limitations trough out the run.

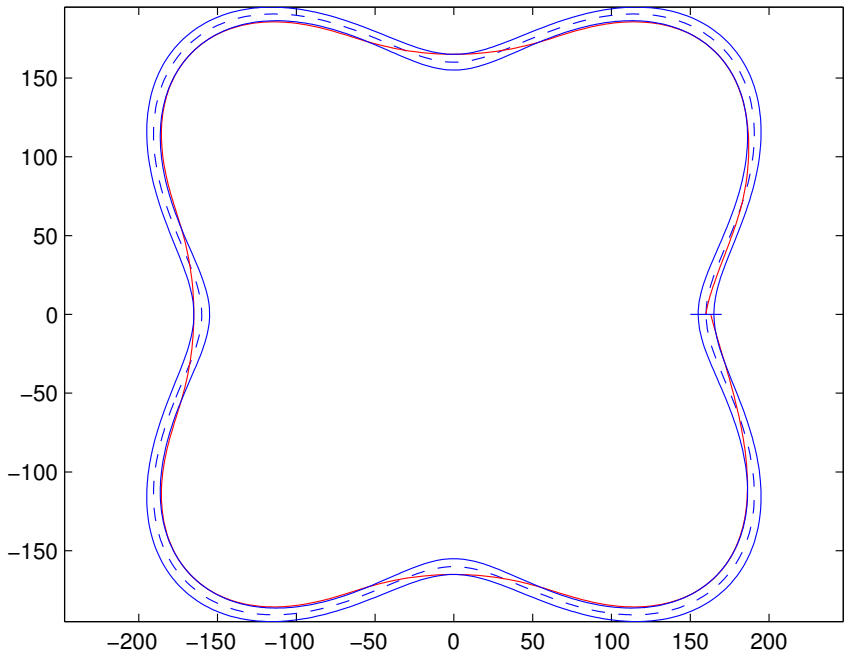

Figure 4.5: The optimal path obtained when minimizing the time, for a bicycle model with linear tyres driving 1 lap on the track.

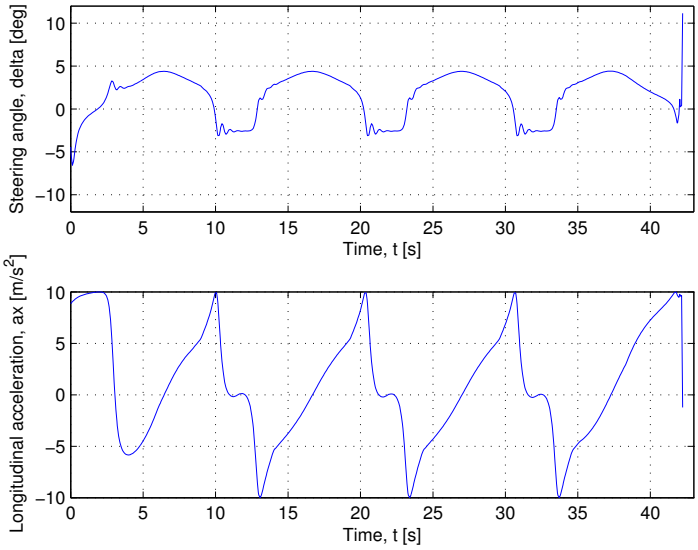

Figure 4.6: The control inputs for a bicycle model with linear tyres driving 1 lap on the flower-shaped track when minimizing the time. Longitudinal acceleration, a_x and steering angle, δ.

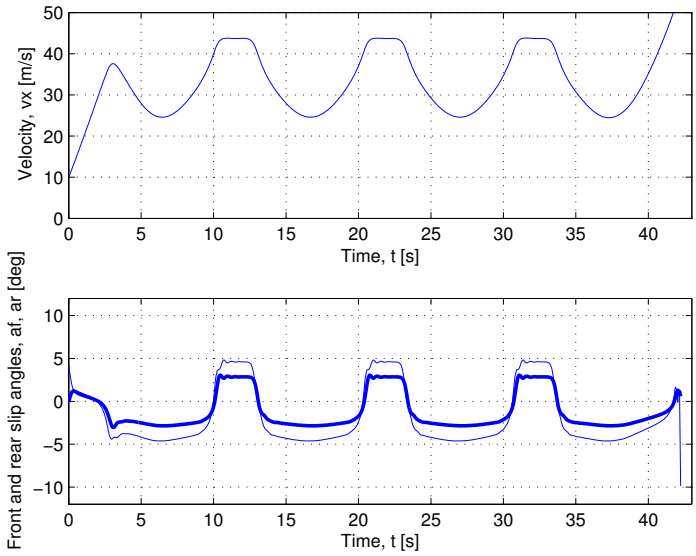

Figure 4.7: The velocity, v_x and the slip angles, α_f(thin) and α_r(thick) of the bicycle model with linear tyres driving 1 lap on the flower-shaped track when minimizing the time.

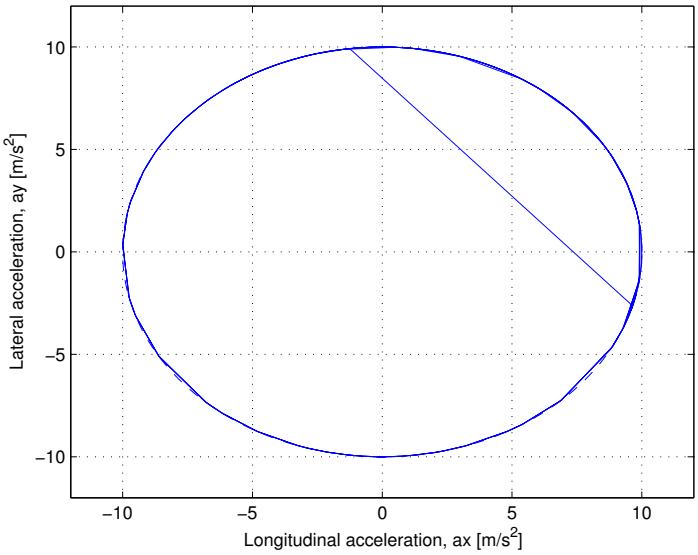

Figure 4.8: The relation between the lateral acceleration, a_y and the longitudinal acceleration, a_x of the bicycle model with linear tyres driving 1 lap on the flower-shaped track when the time is minimized. The dashed lines show the constraints on the accelerations.

Finding the optimal path was more difficult for this track, which is explained with the fact that this track has more corners. As for the previous case the derivative of δ and a_x were minimized for a fixed final time and the result was then used as an initial guess for the minimum time case.

4.1.2 Minimizing the Scaling Factor

In the two following optimisation cases, the equations of the vehicle model and the optimal control problem are dependent of distance. The cost function and the constraints are now defined as:

Cost function:
$\min J = \min \int_0^{s_{tf}} S_{cf} ds$

Constraints:
$a_x^2 + a_y^2 \leq 10^2$

The Ellipse Track, 2 Laps

The optimal path for the ellipse-shaped track is presented in Figure 4.9. Figure 4.10 shows the control inputs, a_x and δ, while Figure 4.11 shows the velocity of the vehicle, v_x and the slip angels of the tyres, α. The g-g diagram in Figure 4.12 illustrates that the vehicle once again is extremely close to its limita-

tions. The dashed curves in Figure 4.10 illustrate the result from Section 4.1.1, where the time was minimized. Since the objective is the same when minimizing the scaling factor as when minimizing time, the results may be compared. The results are quite similar as observed in Figure 4.10. In this case the steering angle, δ increases quickly at the end, but as mentioned earlier the final value of the steering angle doesn't affect the optimal path.

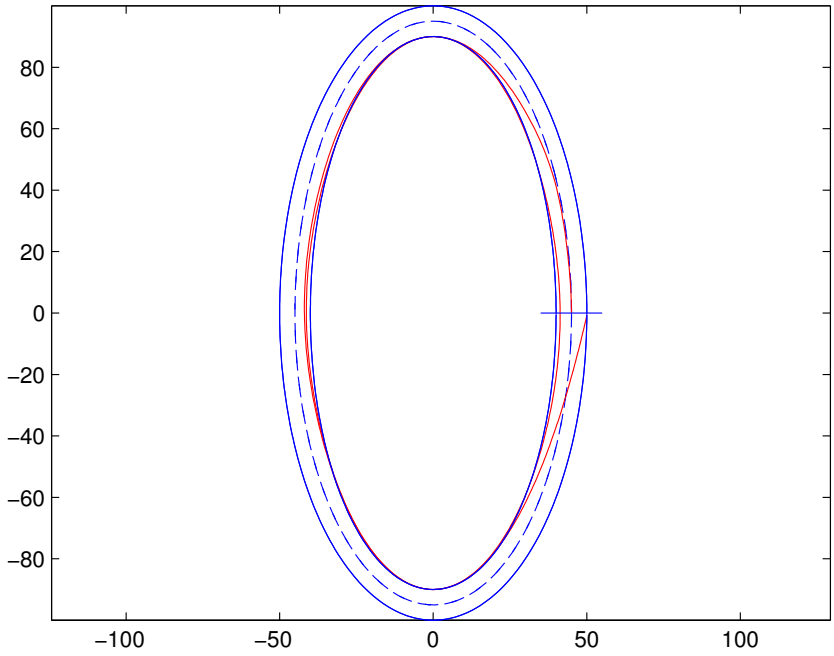

Figure 4.9: The optimal path obtained when minimizing the scaling factor, for a bicycle model with linear tyres driving 2 laps on the track.

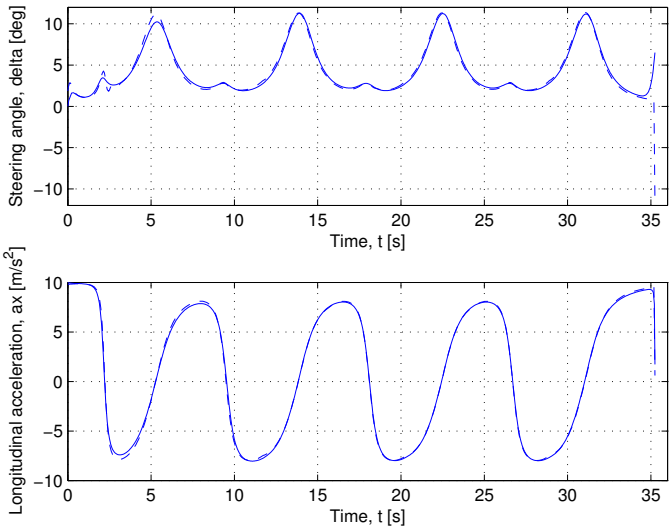

Figure 4.10: The control inputs for a bicycle model with linear tyres driving 2 laps on the ellipse-shaped track when minimizing the scaling factor. Longitudinal acceleration, a_x and steering angle, δ. The dashed lines show the control inputs from the case where the time was minimized.

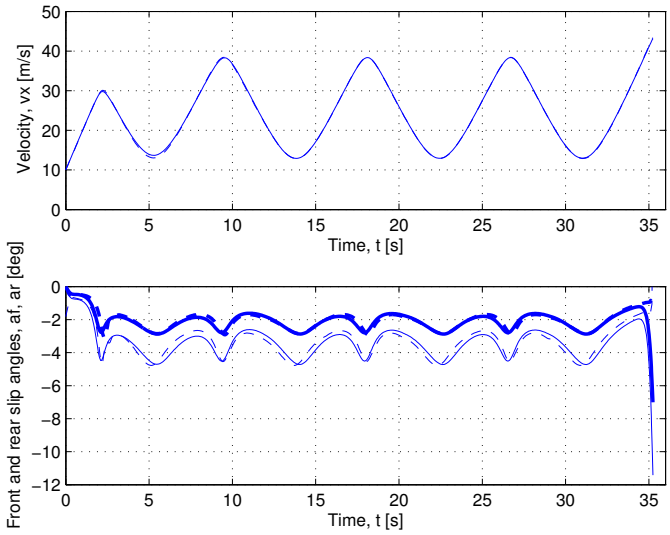

Figure 4.11: The velocity, v_x and the slip angles, α_f(thin) and α_r(thick) of the bicycle model with linear tyres driving 2 laps on the ellipse-shaped track when minimizing the scaling factor. The dashed lines show the control inputs from the case where the time was minimized.

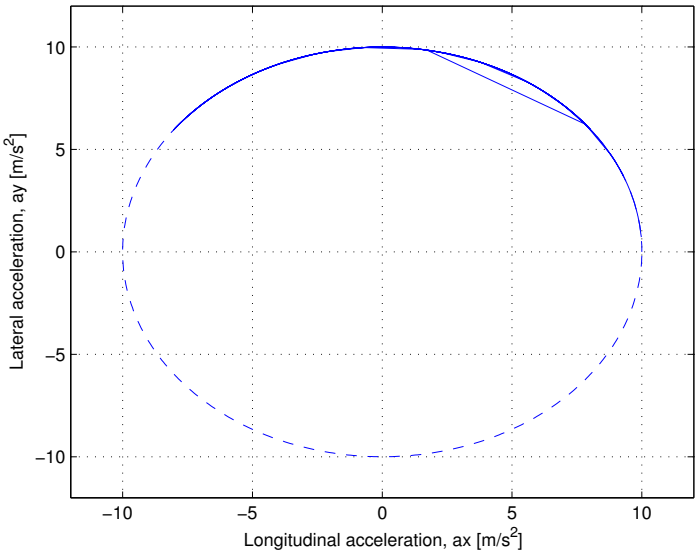

Figure 4.12: The relation between the lateral acceleration, a_y and the longitudinal acceleration, a_x of the bicycle model with linear tyres driving 2 laps on the ellipse-shaped track when the scaling factor is minimized. The dashed line shows the constraints on the accelerations.

The approach introduced in Section 3.5 was applied at first. In general this approach behaved well but at certain parts of the track, an optimal solution wasn't found and caused the method to fail. Particularly at the top and in the section after together with the bottom and the section after the bottom the search for an optimal solution was difficult. S_{cf} was minimized for the very first part of the track and an optimal solution was found. s_{tf} was then increased and an optimal solution was found again. This procedure worked out good until reaching the top of the ellipse, where an optimal solution were no longer found. The explanation was that the vehicle had a bad position and the velocity was too high for continuing on the track, thereby the failure of the next optimisation. This problem was solved by putting terminal constraints into the optimisation. Following terminal constraints were introduced:

- low terminal velocity of the vehicle
- the final position of the vehicle needs to be in the middle of the track
- the final direction of the vehicle needs to equal the final direction of the tangent of the track

The next problem arose in the section after the top of the ellipse. The steering angle began to oscillate, this problem was solved by adding the derivative of the steering angle to the cost function and penalizing it. This means both the scaling factor and the derivative of the steering angle were minimized. One might believe this will interfere with finding an optimal path, but that wasn't

the case. Instead it became easier for the optimisation tools to converge to the optimal solution. After introducing the above improvements an optimal path for the entire track was found. A script automating this process was used, see Appendix B.1. In the script a few different options may be chosen, one particular option called, "mu_strategy=adaptive" required less iterations in the optimisation runs. More about different IPOPT options can be found in [4].

The optimal path was also found when using the result from a simulation with the driver model as an initial guess. The simulated initial guess is preferable when it works because the procedure with all the other optimisations are not needed, but for more challenging tracks and more advanced vehicle models the simulated initial guess will probably not be good enough.

The Flower Track, 1 Lap

The optimal path for the second track when minimizing the scaling factor, S_{cf} is shown in Figure 4.13. The control inputs, the velocity of the vehicle and the slips angles are presented in Figures 4.14 and 4.15. The g-g diagram is presented in Figure 4.16. The steering angle and the slip angles are a bit smoother when minimizing the scaling factor compared to when minimizing time. In the case of minimizing the scaling factor, the derivative of the steering angle was added to the cost function. This means the derivative of the steering angle was minimized as well, which makes the steering angle smoother. As mentioned earlier, this doesn't affect the search of the optimal path. The vehicle is still driving on its limitations, as seen in Figure 4.16.

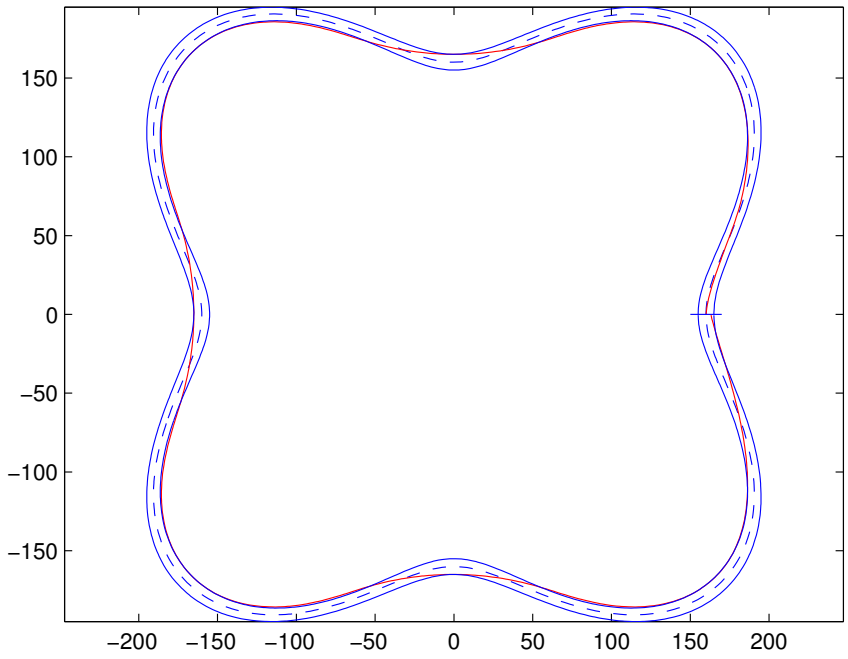

Figure 4.13: The optimal path obtained when minimizing the scaling factor, for a bicycle model with linear tyres driving 1 lap on the track.

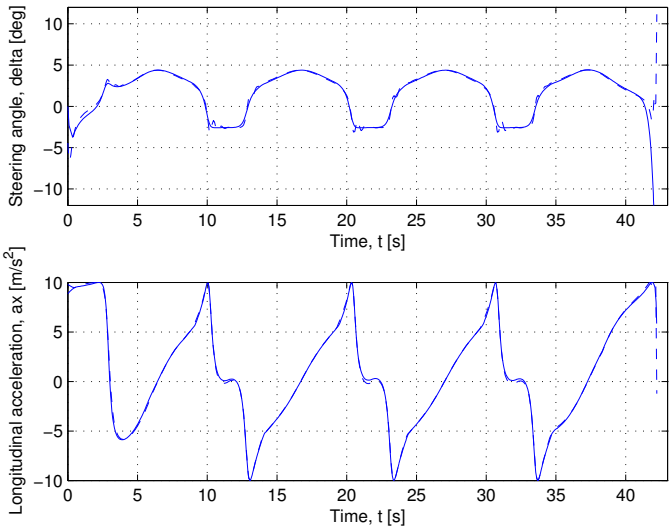

Figure 4.14: The control inputs for a bicycle model with linear tyres driving 1 lap on the flower-shaped track when minimizing the scaling factor. Longitudinal acceleration, a_x and steering angle, δ. The dashed lines show the control inputs from the case where the time was minimized.

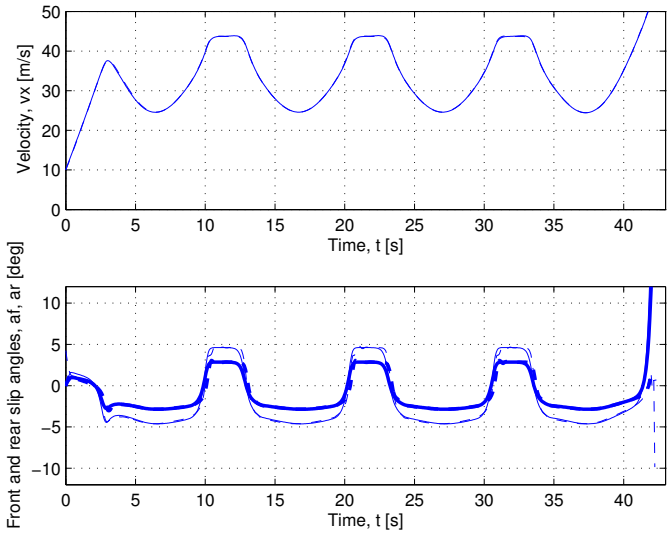

Figure 4.15: The velocity, v_x and the slip angles, α_f(thin) and α_r(thick) of the bicycle model with linear tyres driving 1 lap on the flower-shaped track when minimizing the scaling factor. The dashed lines show the control inputs from the case where the time was minimized.

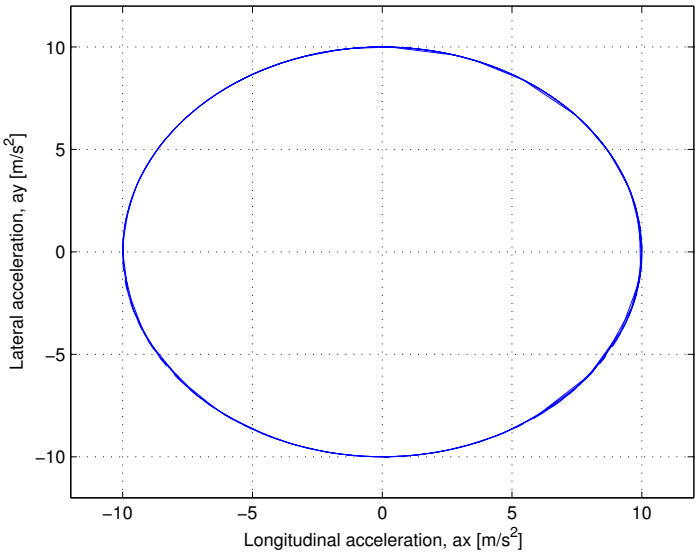

Figure 4.16: The relation between the lateral acceleration, a_y and the longitudinal acceleration, a_x of the bicycle model with linear tyres driving 1 lap on the flower-shaped track when the scaling factor is minimized. The dashed line show the constraints on the accelerations.

The approach described in Section 3.5 failed at first at certain points of this track as well, but with the improvements made in the previous case an optimal solution was obtained. The optimal path was also found when using the simulated initial guess, just like for the case with the ellipse-shaped track.

4.1.3 Comparison

The time it takes for the vehicle to traverse different numbers of laps on the two tracks are presented in Table 4.3. The table shows the minimum time received when minimizing the time and when minimizing the scaling factor.

Track	*Number of laps*	$\min t$	$\min S_c f$
Ellipse	1	$18.039s$	$18.042s$
Ellipse	2	$35.242s$	$35.243s$
Ellipse	3	$52.443s$	$52.443s$
Flower	1	$42.228s$	$42.220s$
Flower	2	$83.506s$	$83.504s$

Table 4.3: Comparison between minimizing time and minimizing the scaling factor for a bicycle model with linear tyres driving different number of laps on the two different tracks.

The execution times for the different optimisation cases are dependent on how good the initial guesses are. A better initial guess will obviously lead to a

shorter execution time. Generally the execution time is less when minimizing the scaling factor compared to when minimizing the time. Normally it takes a few minutes for each optimisation run when minimizing the time. When minimizing the scaling factor an optimisation run takes a few seconds but aggregately more optimisation runs are needed compared to when minimizing the time. The number of optimisation runs depend on the distance of the track and on how much s_{tf} is increased between the runs, when minimizing the scaling factor. In the case where the time is minimized the number of optimisation runs depend on how much t_f is decreased between the runs.

4.2 Bicycle Model with Nonlinear Tyres

In this section, optimal path results for a bicycle model with nonlinear tyres are presented. For the ellipse-shaped track, optimal paths were found when minimizing time and when minimizing the scaling factor. Unfortunately no optimal paths were found for the second track, instead the result from the case when minimizing the control inputs (the result that functioned as an initial guess when trying to minimize time) are presented. The parameters of the vehicle model are shown in Table 4.4. Just as for the previous vehicle model, these parameters are arbitrarily chosen and not received from any real vehicle. Variables with lower and upper bound constraints are presented in Table 4.5.

Parameter	
Yaw Inertia, J_z $[kgm^2]$	2800
Distance from centre of gravity to front axle, l_f $[m]$	1.33
Distance from centre of gravity to rear axle, l_r $[m]$	1.43
Mass of vehicle, m $[kg]$	1550
Distance from centre of gravity to the ground, h $[m]$	0.3
The shape factor of the front wheel, C_f	1.3
The shape factor of the rear wheel, C_r	1.3
The curvature factor of the front wheel, E_f	-2
The curvature factor of the rear wheel, E_r	-2.5

Table 4.4: Parameters of the bicycle model with nonlinear tyres.

Variable	*Lower Bound*	*Upper Bound*	*Start Value*
delta $[rad]$	-1	1	0
v_x $[m/s]$	0	100	15
a_x $[m/s^2]$	-10	10	0
d $[m]$	-5	5	0
α_f	-0.175	0.175	0
α_r	-0.175	0.175	0

Table 4.5: Variables of the bicycle model with nonlinear tyres.

4.2.1 Minimizing Time

The time was minimized for the vehicle model driving on the two tracks. The cost function for both of the two cases are formulated as follows:

Cost function:
$\min J = \min \int_0^{t_f} 1dt$

The Ellipse Track, 2 Laps

The result from the bicycle model with nonlinear tyres driving on the ellipse-shaped track are presented in Figures 4.17- 4.22. In Figures 4.21 and 4.22 the lateral tyre force and the maximum lateral tyre force are plotted against the longitudinal tyre force for the front and rear tyres respectively. The blue curve represents the lateral tyre force, while the red curve represents the maximum lateral tyre force. The lateral force is for most of the time close to the maximum lateral force for the front tyre except when braking heavily. For the rear tyre the lateral force is close to the maximum lateral force only when braking. The conclusion is, the vehicle tends to go from being under steer to over steer when braking and cornering heavily.

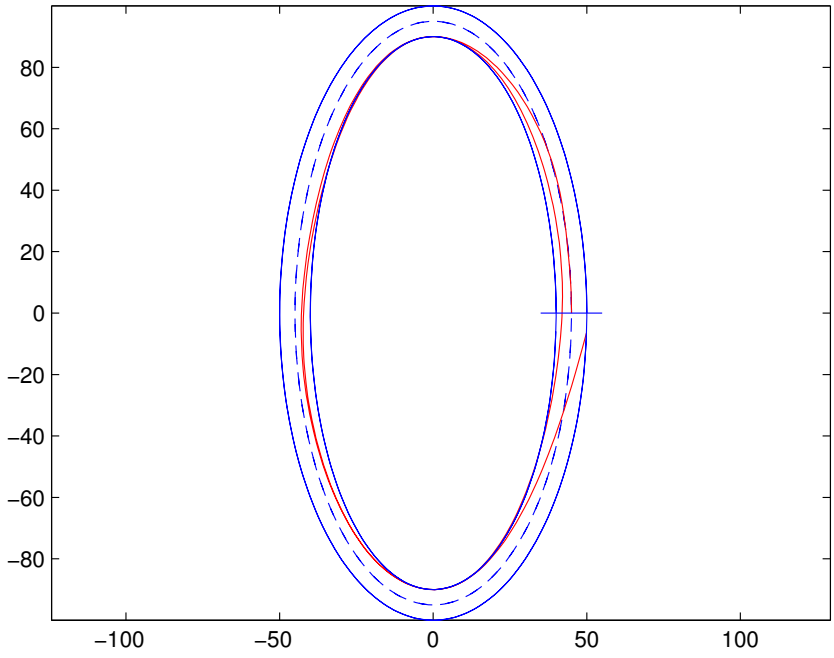

Figure 4.17: The optimal path obtained when minimizing the time, for a bicycle model with nonlinear tyres driving 2 laps on the track.

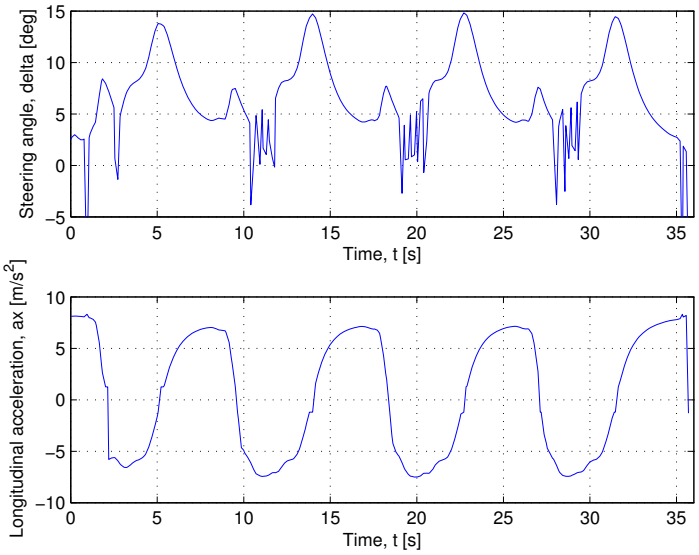

Figure 4.18: The control inputs for a bicycle model with nonlinear tyres driving 2 laps on the ellipse-shaped track when minimizing the time. Longitudinal acceleration, a_x and steering angle, δ.

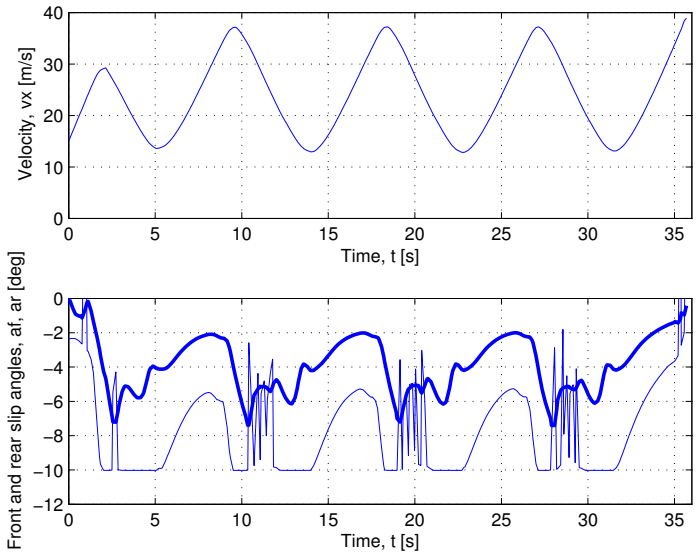

Figure 4.19: The velocity, v_x and the slip angles, α_f(thin) and α_r(thick) of the bicycle model with nonlinear tyres driving 2 laps on the ellipse-shaped track when minimizing the time.

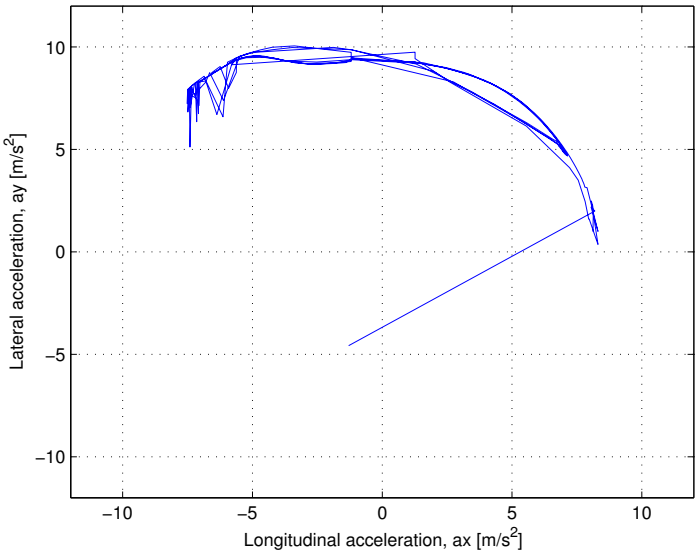

Figure 4.20: The relation between the lateral acceleration, a_y and the longitudinal acceleration, a_x of the bicycle model with nonlinear tyres driving 2 laps on the ellipse-shaped track when the time is minimized.

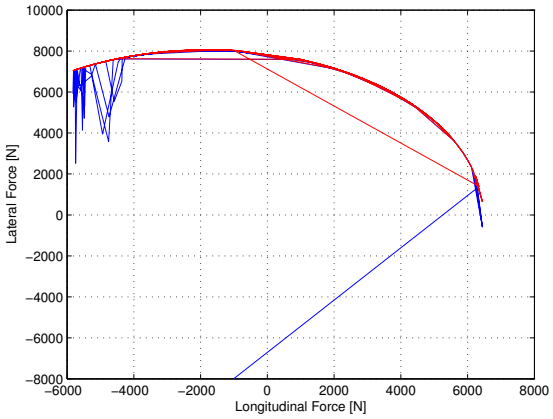

Figure 4.21: The lateral force(blue) and the maximum lateral force(red) plotted against the longitudinal force for the front tyre in the case where the time was minimized for a bicycle model with nonlinear tyres driving 2 laps on the ellipse-shaped track.

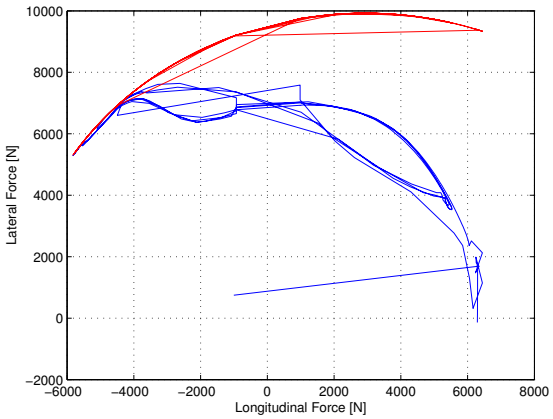

Figure 4.22: The lateral force(blue) and the maximum lateral force(red) plotted against the longitudinal force for the rear tyre in the case where the time was minimized for a bicycle model with nonlinear tyres driving 2 laps on the ellipse-shaped track.

A minimum time solution wasn't found at first but when upper and lower bounds on the slip angles of the tyres were introduced an optimal solution was obtained. This can be explained by observing Figure 2.3 in Section 2.2.3, the lateral force has a peak at a certain slip angle. If the slip angle increases more, then the lateral force will decrease. This peak makes it hard for the optimisation tools to converge to a solution. Therefore the upper and lower bounds on the slip angles were introduced preventing the slip angles from increasing pass the peaks.

Before the optimal solution was found, another problem concerning the equation 2.7, which defines the friction ellipse came up. The equation was at first written on the following form:

$$D = F_{ymax}\sqrt{(}1 - (\frac{F_x}{F_{xmax}})^2) \tag{4.1}$$

When the optimisation tools evaluate equation 4.1, the expression under the square root sign might become negative. This problem disappeared when the equation was rewritten as:

$$(\frac{F_x}{F_{xmax}})^2 + (\frac{D}{F_{ymax}})^2 = 1 \tag{4.2}$$

It is an advantage if the square root can be avoided in all equations.

The Flower Track, 1 Lap

No optimal solution was found when minimizing time for the vehicle model driving on the flower-shaped track. The initial guess used for the minimum time case might have been too poor. The initial guess used was the result from a optimisation case, where the acceleration, the derivative of the steering angle

and the derivative of the acceleration were minimized for a fixed final time. The cost function was formulated as:

$$\min J = \int_0^{t_f} a_x^2 + 1000\frac{d}{dt}\delta^2 + 10\frac{d}{dt}a_x^2 dt \tag{4.3}$$

The result from this case is presented in Figures 4.23- 4.28. A better initial guess couldn't be found without violating the constraints. By evaluating the figures showing the tyre forces, the conclusion that the vehicle is performing very close to its limitations can be made. This result should be a good initial guess for the minimum time case. Another reason for not finding an optimal solution when minimizing time could originate from the equations of the vehicle model, particularly the equations of the nonlinear tyre model. There might exist more than one equilibrium at certain points, if that's the case, it is hard for the optimisation tools to converge to an optimal solution.

If the optimal path results of the current vehicle model are compared to the optimal path results of the bicycle model with linear tyre properties, it can be seen that the results have the same tendencies. It is difficult to compare the different results more closely, since some of the constraints differ. For the model with linear tyres, the limitation of the lateral acceleration is set with an limitation envelope, depending on the longitudinal acceleration, as illustrated in every g-g diagram. For the vehicle model with nonlinear tyre properties the lateral force is limited by the friction ellipse, depending on the longitudinal force.

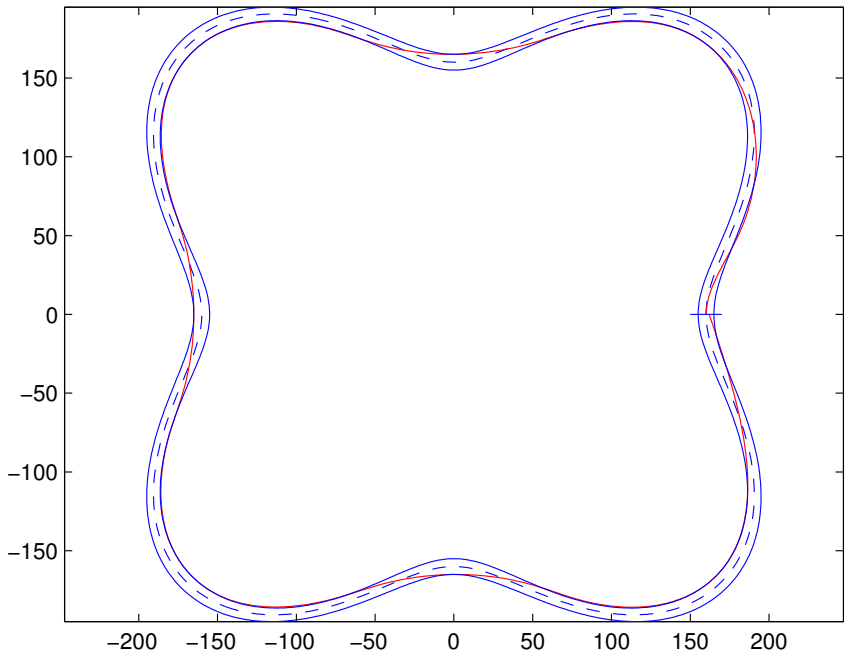

Figure 4.23: The optimal path obtained when minimizing cost function 4.3, for a bicycle model with nonlinear tyres driving 1 lap on the track.

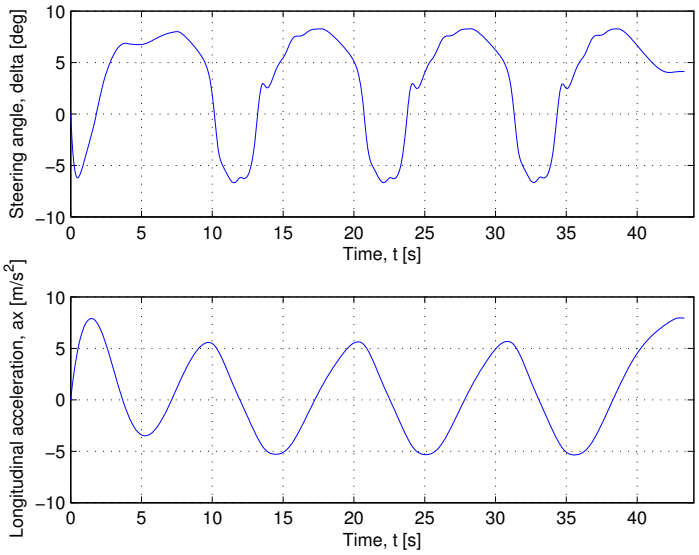

Figure 4.24: The control inputs for a bicycle model with nonlinear tyres driving 1 lap on the flower-shaped track when minimizing cost function 4.3. Longitudinal acceleration, a_x and steering angle, δ.

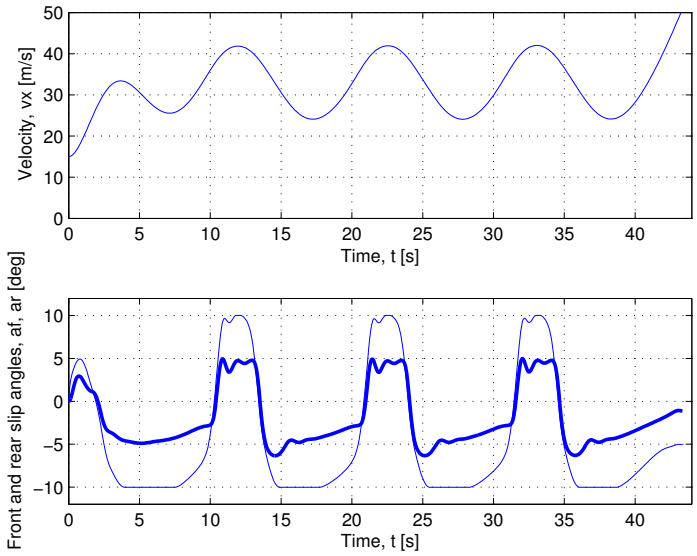

Figure 4.25: The velocity, v_x and the slip angles, α_f(thin) and α_r(thick) of the bicycle model with nonlinear tyres driving 1 lap on the flower-shaped track when minimizing cost function 4.3.

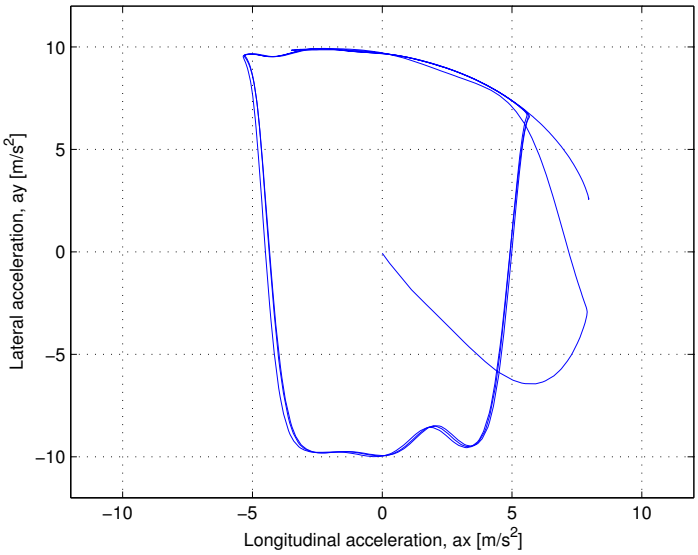

Figure 4.26: The relation between the lateral acceleration, a_y and the longitudinal acceleration, a_x of the bicycle model with nonlinear tyres driving 1 lap on the flower-shaped track when cost function 4.3 is minimized.

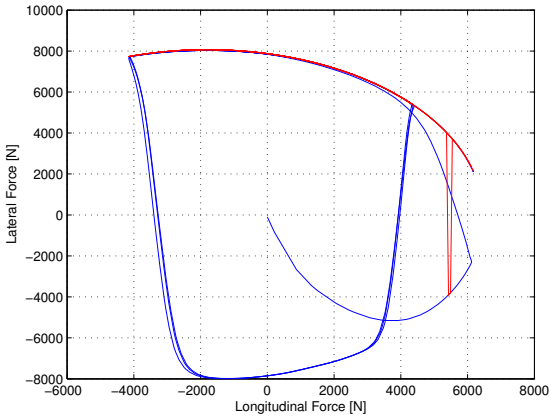

Figure 4.27: The lateral force(blue) and the maximum lateral force(red) plotted against the longitudinal force for the front tyre in the case where cost function 4.3 was minimized for a bicycle model with nonlinear tyres driving 1 lap on the flower-shaped track.

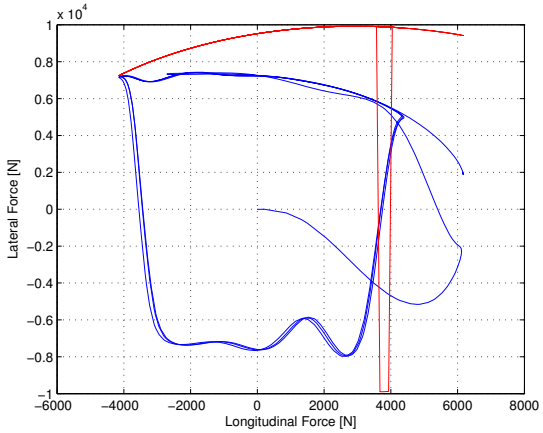

Figure 4.28: The lateral force(blue) and the maximum lateral force(red) plotted against the longitudinal force for the front tyre in the case where cost function 4.3 was minimized for a bicycle model with nonlinear tyres driving 1 lap on the flower-shaped track.

4.2.2 Minimizing the Scaling Factor

As mentioned before, an optimal path was only found for the ellipse-shaped track. The cost function was formulated as follows:

Cost function:
$\min J = \min \int_0^{s_{tf}} S_{cf} ds$

The Ellipse Track, 2 Laps

The optimal path is presented in Figure 4.29. The control inputs, a_x and δ are shown in Figure 4.30, the dashed lines are the resulting control inputs from Section 4.2.1, when minimizing time. The velocity of the vehicle, v_x and the slip angles of the tyres, α are presented in Figure 4.31.

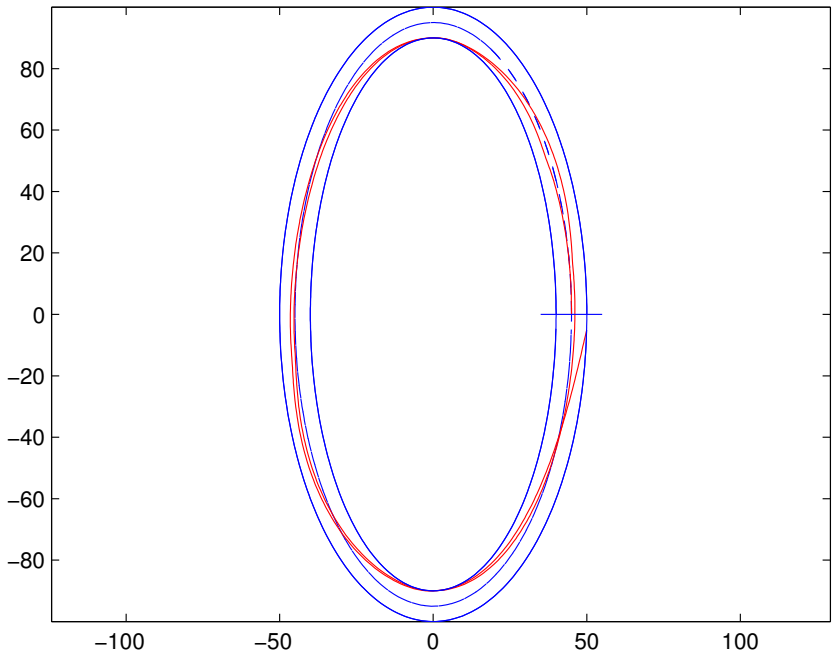

Figure 4.29: The optimal path obtained when minimizing the scaling factor, for a bicycle model with nonlinear tyres driving 2 laps on the track.

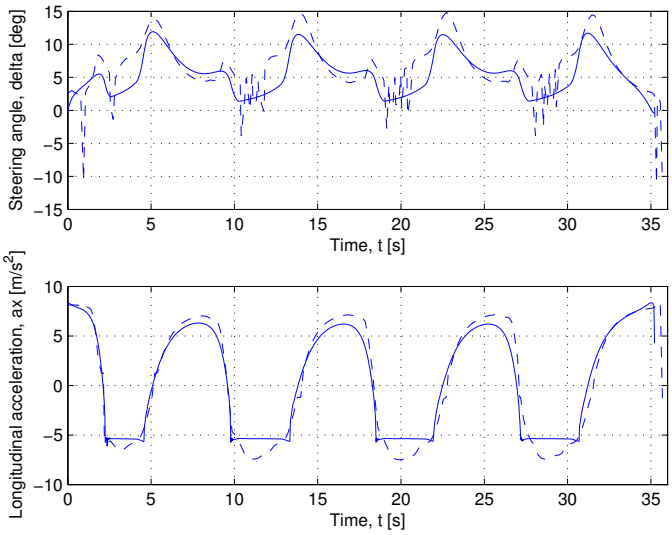

Figure 4.30: The control inputs for a bicycle model with nonlinear tyres driving 2 laps on the ellipse-shaped track when minimizing the scaling factor. Longitudinal acceleration, a_x and steering angle, δ. The dashed lines show the control inputs from the case where the time was minimized.

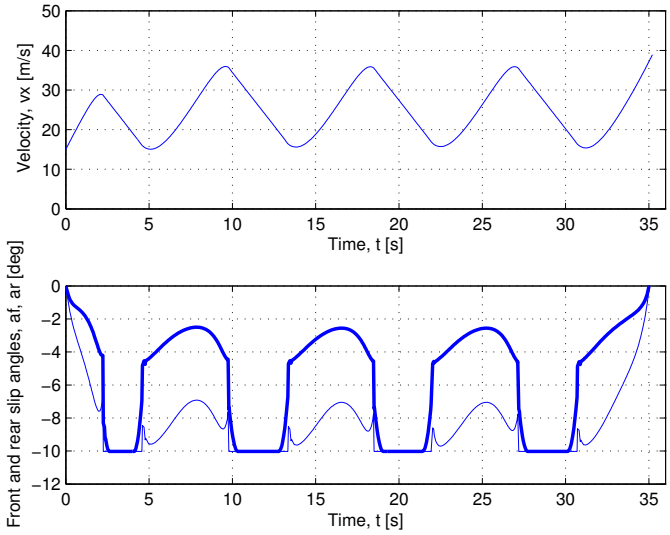

Figure 4.31: The velocity, v_x and the slip angles, α_f(thin) and α_r(thick) of the bicycle model with nonlinear tyres driving 2 laps on the ellipse-shaped track when minimizing the scaling factor.

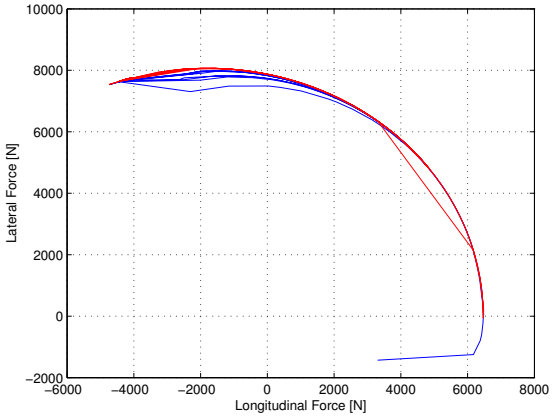

Figure 4.32: The lateral force(blue) and the maximum lateral force(red) plotted against the longitudinal force for the front tyre in the case where the scaling factor was minimized for a bicycle model with nonlinear tyres driving 2 laps on the ellipse-shaped track.

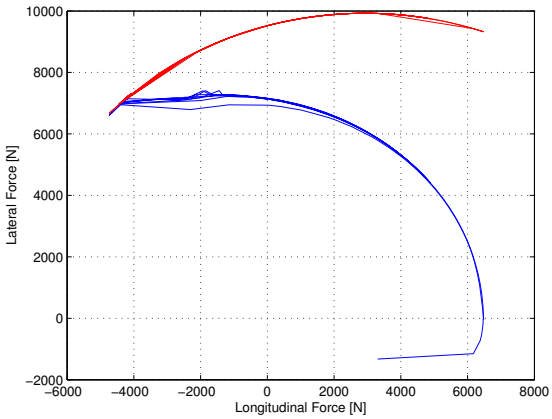

Figure 4.33: The lateral force(blue) and the maximum lateral force(red) plotted against the longitudinal force for the rear tyre in the case where the scaling factor was minimized for a bicycle model with nonlinear tyres driving 2 laps on the ellipse-shaped track.

The above result was obtained when using the result from a simulation run with the driver model as an initial guess. A simple acceleration control was added to the driver model before the simulation run. The approach described in Section 3.5 was applied at first but failed.

The control inputs are not very similar to the control inputs from the case when minimizing time. By observing Figure 4.30 it can be noticed that the final time it takes for the vehicle to drive around the track differ between the two

optimisation cases. The final time is less when minimizing the scaling factor, which might mean that the result obtained when minimizing the time is not really optimal. It may also be discussed if the result from the case where the scaling factor is minimized are really optimal, since some of the states behave strangely. The lateral acceleration, a_y doesn't behave as desired, see Figure 4.34. Unfortunately it is difficult to compare the results for the nonlinear model to the linear model, since the limitations differ. It is hard to find optimal paths, especially when the vehicle model becomes more advanced. Sometimes it is problematic for AMPL to converge to an optimal solution and there is usually a risk of finding a local minimum instead.

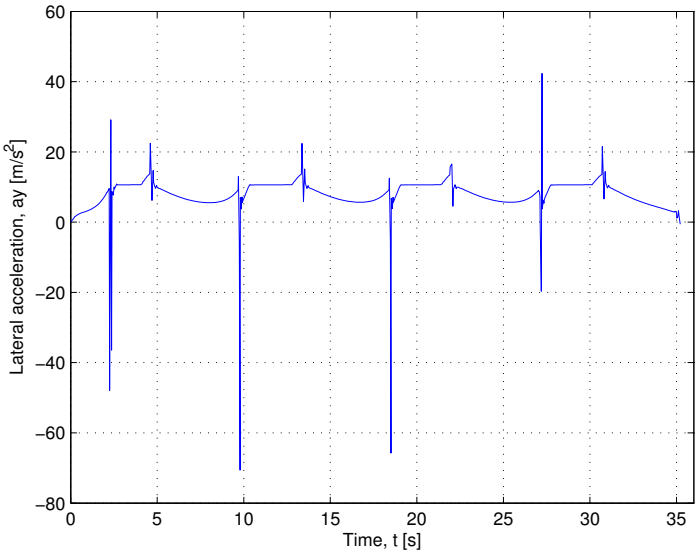

Figure 4.34: The lateral acceleration, a_y for a bicycle model with nonlinear tyres driving 2 laps on the ellipse-shaped track when minimizing the scaling factor.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this project it has been shown that it is possible to find optimal paths for vehicle models designed in Modelica. These optimal paths have been found by applying the theory of optimal control. The optimal control problem has been set up with the Optimica language and then been solved with AMPL.

A vehicle following the optimal path will traverse the track in minimum time, which means by minimizing the time, the optimal path could be found. An optimal control problem with the purpose of minimizing the time is a free final time problem. Solving free final time problems is difficult, usually a good initial guess is necessary in order to find a solution.

In this case the initial guess can be obtained by using a driver model. The driver model will be rather complex in order of producing a proper initial guess. The driver model needs to consist of a steering control and an acceleration control. Implementing a well-behaved acceleration control is particularly difficult. A different approach based on minimizing the control inputs for a fixed final time and reuse the result as an initial guess for the minimum time problem has been applied in this project. Optimal control problems with fixed final time are much easier to solve than problems with free final time, which has been experienced during this study. A disadvantage with the approach is the many optimisation runs that are needed before obtaining a proper initial guess, which might be time-consuming but a special script automates this process.

Transforming the vehicle model and the optimal control problem to be dependent of distance instead of time will facilitate the search for an optimal path. The main reason why, is that the optimal control problem becomes a fixed final time problem, still with the objective of finding a minimum time solution. It is therefore preferable to transform the vehicle model and the optimal control problem, before attempting to find an optimal path. One disadvantage is that the control inputs can no longer be minimized in order to obtain an initial guess. Instead another approach for solving the problem was introduced. The idea of the approach is to increase the distance the vehicle travels between every optimisation until the vehicle reaches the desired distance.

Satisfying optimal path results for a vehicle model with linear tyre properties have been obtained with both of the two approaches. The results have been

compared and the conclusion that the results are very similar has been made. Transforming the model and the problem to be dependent of distance is to prefer, since the initial guess is not as crucial. This means it will take less time to find a sufficiently good initial guess and the optimal path will be easier to determine.

During extreme driving conditions it is not adequate to approximate the tyre characteristics with linear properties. Unfortunately it is difficult to find optimal paths for vehicles with nonlinear tyre characteristics.

When it is possible to find optimal paths for complex vehicle models, resembling real race cars, conclusions considering the vehicle models can be made. These conclusion will play a significant role when designing real vehicles.

5.2 Future Work

There are several opportunities for continuation of this project. The next goal could be to find optimal paths for a vehicle model with nonlinear tyre properties. If accomplishing this isn't possible, vehicle models with linear tyres can still be used but some type of a constraint on the maximum lateral force should be introduced. Any of the following suggestions could be implemented:

1. $F_{ymax} = Constant$
2. $F_{ymax} = Constant - |F_x|$
3. $F_{ymax}^2 + F_x^2 = Constant^2$

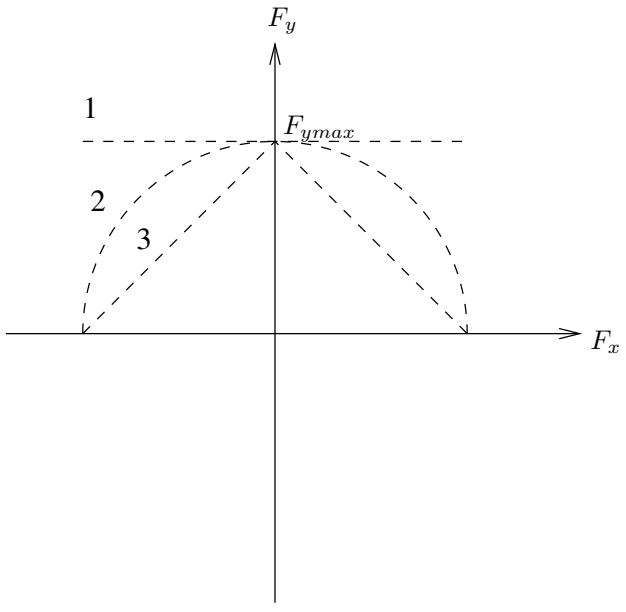

Figure 5.1: Three different types of limitations on the lateral force.

The first alternative is obviously the simplest. The third alternative, where

the constraint on the maximum lateral force is nonlinear is ideal and is similar to the friction ellipse used for the nonlinear tyres.

The chassis model should also be developed towards a more complex model. The next step is perhaps to introduce a two track model and then implement some type of suspension dynamics.

Other types of tracks can also be specified. The tracks could be defined in a different way, for instance with splines.

When searching for the optimal path other aspects but time could be considered. Different driving paths require different driving techniques, which in turn might influence the vehicle differently. For instance the wear of the tyres might be larger for one path, leading to worse performances of the vehicle, which in turn leads to a greater time. The wear of the tyres could in this case be a part of the cost function.

The approach for finding the optimal path when minimizing the scaling factor can be further developed. It would be interesting to see if the first part of the track can be decoupled when using the approach. It should at least be possible to decouple the first lap when finding an optimal path for the second lap.

Bibliography

[1] L. T. Biegler, A. M. Cervantes, A. Wächter *Advances in simultaneous strategies for dynamic process optimization* Carnegie Mellon University, 2001

[2] D. Casanova *On minimum time vehicle manoeuvring: The theoretical optimal lap* PhD Thesis, Cranfield University, 2000

[3] R. Fourer, D. M. Gay, B. W. Kernighan *AMPL: A Modeling Language for Mathematical Programming* Duxbury Press, 2002

[4] K. Kawajiri, C. D. Laird *Introduction to IPOPT: A tutorial for downloading, installing and using IPOPT.*

[5] W. F. Milliken, D. L. Milliken *Race Car Vehicle Dynamics* SAE International, 1995

[6] M. Otter, H. Elmqvist *Modelica - Language, Libraries, Tools, Workshop and EU-Project RealSim* German Aerospace Center and Dynasim AB, 2001

[7] H. B. Pacejka *Tyre and vehicle dynamics* Butterworth Heinmann, 2002

[8] P. Ramanata *Optimal Vehicle Path Generator Using optimisation Methods* Master's Thesis, Virginia Institute of Technology, 1998

[9] E. Velenis, P. Tsiotras *Minimum Time vs Maximum Exit Velocity Path optimisation During Cornering* Georgia Institute of Technology

[10] E. Wennerström, S. Nordmark, B. Thorvald *Fordonsdynamik* Fordonsdynamik, The Royal Institute of Technology, 2005

[11] J. Åkesson *The Optimica Compiler 0.3 User's Guide* Lund Institute of Technology, 2007

Appendix A

Vehicle Model

```
partial model BicycleModel_nonlinear

  extends Icons.Audit;

  parameter SI.Inertia Jz=2800  "Yaw inertia";
  parameter SI.Length lf=1.33  "Distance from centre of gravity to
front axle";
  parameter SI.Length lr=1.43  "Distance from centre of gravity to
rear axle";
  parameter SI.Mass m=1550  "Mass of vehicle";
  SI.Position xpos(start=45)  "Global x position";
  SI.Position ypos(start=0)  "Global y position";
  SI.Velocity xvel  "Global x velocity";
  SI.Velocity yvel  "Global y velocity";
  SI.Acceleration xacc  "Global x acceleration";
  SI.Acceleration yacc  "Global y acceleration";
  SI.Velocity vx(start=15,min=0.001) "Longitudinal velocity";
  SI.Velocity vy(start=0)  "Lateral velocity";
  SI.Acceleration dvy  "Derivative of lateral velocity";
  SI.Acceleration ax  "Longitudinal acceleration";
  SI.Acceleration ay  "Lateral acceleration";
  SI.Angle psi(start=Modelica.Constants.pi/2) "Yaw angle";
  SI.AngularVelocity z "Yaw velocity";
  SI.Angle delta "Steering angle";
  SI.Angle af "Front slip angle";
  SI.Angle ar "Rear slip angle";
  SI.Angle beta "Vehicle slip angle";
  SI.Force Fxf "Front wheel longitudinal force";
  SI.Force Fxr "Rear wheel longitudinal force";
  SI.Force Fyf "Front wheel lateral force";
  SI.Force Fyr "Rear wheel lateral force";

///NONLINEAR TYRE MODEL///

  SI.Force Fxmaxf(min=0.001) "Maximum longitudinal force, front tyre";
```

```
  SI.Force Fxmaxr(min=0.001) "Maximum longitudinal force, rear tyre";
  SI.Force Fymaxf(min=0.001) "Maximum lateral force, front tyre";
  SI.Force Fymaxr(min=0.001) "Maximum lateral force, rear tyre";
  SI.Force Fx "Longitudinal force of vehicle";
  Real Df "Peak value in the Magic Formula, front tyre";
  Real Dr "Peak value in the Magic Formula, rear tyre";
  Real Cf "Shape factor in the Magic Formula, front tyre";
  Real Cr "Shape factor in the Magic Formula, rear tyre";
  Real Bf "Stiffness factor in the Magic Formula, front tyre";
  Real Br "Stiffness factor in the Magic Formula, rear tyre";
  Real Ef "Curvature factor in the Magic Formula, front tyre";
  Real Er "Curvature factor in the Magic Formula, rear tyre";
  SI.Force Fzf "Vertical load, front wheel";
  SI.Force Fzr "Vertical load, rear wheel";
  SI.Height h=0.3 "Height of the centre of gravity";
  Real lamda=0.5;

/////////TRACK/////////

  SI.Position xc "The center of the track, x-axis";
  SI.Position yc "The center of the track, y-axis";
  SI.Velocity dxc "Derivative of xc";
  SI.Velocity dyc "Derivative of yc";
  SI.Acceleration ddxc "Derivative of dxc";
  SI.Acceleration ddyc "Derivative of dyc";
  Real st(start=0) "Distance from the start point of the track";
  SI.Distance d "Distance between the vehicle and the centre of track";
  SI.Position xmin;
  SI.Position ymin;
  SI.Position xmax;
  SI.Position ymax;
  Real ds;
  Real xcds "The x-coordinate of the centre of the track at st+ds";
  Real ycds "The y-coordinate of the centre of the track at st+ds";
  Real xcdsxpos "xcds-xpos";
  Real ycdsypos "ycds-ypos";
  Real delta_tmp;
  Real delta_ref "Steering angle of driver model";
  Real psit "The angle between the tangent of the centre of the track and
the x-axis";
  Real kt "Curvature of the centre of the track";
  Real rt "Radius of the centre of the track";
  Real Scf "Time to distance scaling factor";

equation
  assert(vx>0, "Longitudinal velocity (vx) is to low");

  /*Slip*/
  af=atan((vy+z*lf)/vx)-delta;
  ar=atan((vy-z*lr)/vx);
```

```
  beta=atan(vy/vx);

  /*Kinematics*/
  der(psi)=z;
  der(xpos)=xvel;
  der(ypos)=yvel;
  der(xvel)=xacc;
  der(yvel)=yacc;
  der(vx)=ax;
  der(vy)=dvy;

  /*Coordinate transform*/
  xvel = vx*cos(psi) - vy*sin(psi);
  yvel = vx*sin(psi) + vy*cos(psi);

  ay = dvy+z*vx;

  /*Equations of motion*/
//  der(vx)-z*vy=(-Fyf*sin(delta))/m;
  der(vy)+z*vx=(Fyr+Fyf*cos(delta))/m;
  der(z)=(lf*Fyf*cos(delta)-lr*Fyr)/Jz;

///NONLINEAR TIRE MODEL///

  Fyf=-Df*sin(Cf*atan(Bf*(af)-Ef*(Bf*(af)-atan(Bf*(af)))));
  Fyr=-Dr*sin(Cr*atan(Br*(ar)-Er*(Br*(ar)-atan(Br*(ar)))));

  1=(Df/Fymaxf)^2+(Fxf/Fxmaxf)^2;
  1=(Dr/Fymaxr)^2+(Fxr/Fxmaxr)^2;

  Fx=ax*m;
  Fxf=lamda*Fx;
  Fxr=(1-lamda)*Fx;
  Fxmaxf=Fymaxf;
  Fxmaxr=Fymaxr;
  Fzf=(m*Modelica.Constants.g_n*lr-m*ax*h)/(lf+lr);
  Fzr=(m*Modelica.Constants.g_n*lf+m*ax*h)/(lf+lr);

  Fymaxf=1.0*Fzf;  //myf*Fzf;
  Fymaxr=1.3*Fzr;  //myr*Fzr;
  Bf*Cf*Df=80000;
  Br*Cr*Dr=100000;
  Cf=1.3;
  Cr=1.3;
  Ef=-2;
  Er=-2.5;

///////ELLIPSE TRACK////////

  xc=45*cos(st);
```

```
  yc=95*sin(st);
  dxc=-45*sin(st);
  dyc=95*cos(st);
  ddxc=-45*cos(st);
  ddyc=-95*sin(st);
  xmin=40*cos(st);
  xmax=50*cos(st);
  ymin=90*sin(st);
  ymax=100*sin(st);
  xcds=45*cos(st+ds);
  ycds=95*sin(st+ds);

///////CIRCULAR TRACK///////

  // xc=(200+40*sin(4*(st)-Modelica.Constants.pi/2))*cos(st);
  // yc=(200+40*sin(4*(st)-Modelica.Constants.pi/2))*sin(st);
  // dxc=-(200)*sin(st)+40*4*cos(4*st-Modelica.Constants.pi/2)*cos(st)
-40*sin(4*st-Modelica.Constants.pi/2)*sin(st);
  // dyc=(200)*cos(st)+40*4*cos(4*st-Modelica.Constants.pi/2)*sin(st)
+40*sin(4*st-Modelica.Constants.pi/2)*cos(st);
  // ddxc=-200*cos(st)-680*sin(4*st-Modelica.Constants.pi/2)*cos(st)
-320*cos(4*st-Modelica.Constants.pi/2)*sin(st);
  // ddyc=-200*sin(st)+cos(4*st-Modelica.Constants.pi/2)*(640*sin(st)
+160*cos(st))+sin(4*st-Modelica.Constants.pi/2)*(160*cos(st)-40*sin(st));
  // xmin=(195+40*sin(4*(st)-Modelica.Constants.pi/2))*cos(st);
  // xmax=(205+40*sin(4*(st)-Modelica.Constants.pi/2))*cos(st);
  // ymin=(195+40*sin(4*(st)-Modelica.Constants.pi/2))*sin(st);
  // ymax=(205+40*sin(4*(st)-Modelica.Constants.pi/2))*sin(st);
  // xcds=(200+40*sin(4*((st+ds))-Modelica.Constants.pi/2))*cos((st+ds));
  // ycds=(200+40*sin(4*((st+ds))-Modelica.Constants.pi/2))*sin((st+ds));

  (xpos-xc)*dxc+(ypos-yc)*dyc=0;
  ds=0.07;
  xcdsxpos=xcds-xpos;
  ycdsypos=ycds-ypos;
  delta_tmp=acos((xcdsxpos*xvel+ycdsypos*yvel)/(sqrt(xcdsxpos^2+ycdsypos^2)
*sqrt(xvel^2+yvel^2)));
  delta_ref=if xcdsxpos>0 and xvel>0 and ycdsypos/xcdsxpos>yvel/xvel then
delta_tmp else
            if xcdsxpos<0 and xvel>0 and yvel/xvel>ycdsypos/xcdsxpos then
delta_tmp else
            if xcdsxpos<0 and xvel<0 and ycdsypos/xcdsxpos>yvel/xvel then
delta_tmp else
            if xcdsxpos>0 and xvel<0 and yvel/xvel>ycdsypos/xcdsxpos then
delta_tmp else
            if xvel>0 and xcdsxpos>0 and yvel/xvel>ycdsypos/xcdsxpos then
-delta_tmp else
            if xvel<0 and xcdsxpos>0 and ycdsypos/xcdsxpos>yvel/xvel then
-delta_tmp else
            if xvel<0 and xcdsxpos<0 and yvel/xvel>ycdsypos/xcdsxpos then
```

```
-delta_tmp else
            if xvel>0 and xcdsxpos<0 and ycdsypos/xcdsxpos>yvel/xvel then
-delta_tmp else
            0;
  psit=if ddxc<=0 then acos(dxc/sqrt(dxc^2+dyc^2)) else
       2*Modelica.Constants.pi-acos(dxc/sqrt(dxc^2+dyc^2));
  d=(ypos-yc)*cos(psit)-(xpos-xc)*sin(psit);
  kt=sqrt((dxc*ddxc)^2+(dyc*ddyc)^2)/(sqrt(dxc^2+dyc^2))^3;
  rt=if kt>0 then 1/kt else
     100000;
  Scf=(1-d*kt)/(vx*cos(psi-psit)-vy*sin(psi-psit));

end BicycleModel_nonlinear;
```

Appendix B

Optimisation Script

B.1 Minimizing Time

```
reset;
param i;
let i:=2;

#AMPL-files

model PlanarVehicles.Experiments.Optimization.mod;
data PlanarVehicles.Experiments.Optimization.dat;
model PlanarVehicles.Experiments.Optimization.InitialGuess.mod;
data PlanarVehicles.Experiments.Optimization.InitialGuess.dat;
model PlanarVehicles.Experiments.Optimization.SquareProblemCost.mod;
model PlanarVehicles.Experiments.Optimization.Constraint.mod;

option  solver "/work/jakesson/software_tools/Ipopt/Ipopt-3.2.0
/CoinIpopt/bin/ipopt";
option  ipopt_options "max_iter=10000";
solve;
include PlanarVehicles.Experiments.Optimization.GenLogFile.run;
model PlanarVehicles.Experiments.Optimization.Cost.mod;

param tmp;
let tmp:=TIME;
redeclare param TIME;
let TIME:=tmp;

#Loop finding optimal solutions when minimizing the optimal control
inputs and decreasing TIME(=tf).

repeat while i > 0.01 {

solve;
include PlanarVehicles.Experiments.Optimization.GenLogFile
```

```
.run;

#If an optimal solution is found then TIME(=tf) is decreased.

if solve_result_num == 0 then {
shell 'cp PlanarVehicles.Experiments.Optimization
_res.txt PlanarVehicles.Experiments.Optimization_res_tmp.txt';
let tmp := TIME;
let i := 1*i;
let TIME := TIME - i;
}else if solve_result_num != 0 then {
shell 'cp PlanarVehicles.Experiments.Optimization_res
_tmp.txt PlanarVehicles.Experiments.Optimization_res.txt';
let i := i/2;
let TIME := TIME + i;
let tmp := TIME;
}

display TIME;

};
```

B.2 Minimizing the Scaling Factor

```
reset;
param i;
let i:=1;

#AMPL-files

model PlanarVehicles_s.Experiments.Optimization.mod;
data PlanarVehicles_s.Experiments.Optimization.dat;
model PlanarVehicles_s.Experiments.Optimization.InitialGuess.mod;
data PlanarVehicles_s.Experiments.Optimization.InitialGuess.dat;
model PlanarVehicles_s.Experiments.Optimization.SquareProblemCost.mod;
model PlanarVehicles_s.Experiments.Optimization.Constraint.mod;

option  solver "/work/jakesson/software_tools/Ipopt/Ipopt-3.2.0
/CoinIpopt/bin/ipopt";
option  ipopt_options "max_iter=2000 mu_strategy=adaptive";
solve;
include PlanarVehicles_s.Experiments.Optimization.GenLogFile.run;
model PlanarVehicles_s.Experiments.Optimization.Cost.mod;

param tmp;
let tmp:=TIME;
redeclare param TIME;
let TIME:=tmp;
```

```
#Loop finding optimal solutions and increasing TIME(=sf), until
TIME=900.

repeat while TIME < 900 {

printf "grid(finalTime=fixedFinalTime(finalTime=%d)
,nbrElements=%d);", TIME, 0.5*TIME > temp_opt_u_2.op;
close temp_opt_u_2.op;
shell 'cat temp_opt_u_1.op temp_opt_u_2.op temp_opt_u_3.op
 > temp_opt_u.op';
shell 'optimicac temp_opt_u.op PlanarVehicles_s.mo
PlanarVehicles_s.Experiments.Optimization
PlanarVehicles_s.Experiments.Optimization_res.txt';
solve;
include PlanarVehicles_s.Experiments.Optimization
.GenLogFile.run;

#If an optimal solution is found then TIME(=sf) is
increased.

if solve_result_num == 0 then {
shell 'cp PlanarVehicles_s.Experiments.Optimization
_res.txt PlanarVehicles_s.Experiments.Optimization_res_tmp.txt';
let tmp := TIME;
let i := 1*i;
let TIME := TIME + i;
}else if solve_result_num != 0 then {
shell 'cp PlanarVehicles_s.Experiments.Optimization
_res_tmp.txt PlanarVehicles_s.Experiments.Optimization_res.txt';
let i := i/2;
let TIME := TIME - i;
let tmp := TIME;
}

display TIME;

};
```