Cooperative Control of Road Vehicles

K. Konuk
DCT 2009.129

Internship report

Coach: Ir. J. Ploeg (TNO)
Supervisor: Prof. dr. H. Nijmeijer

Eindhoven University of Technology
Department of Mechanical Engineering
Automotive Technology

Eindhoven, November, 2009
Abstract

Due to the increase in world’s mobility problem, the need for autonomous vehicles arises. By forming car platoons, a decrease in the traffic jams is desired to be obtained.

In this study, the cooperative control problem of a car platoon is considered. A tracking controller that can control the position and the orientation angle of the centre of a single vehicle, thereby tracking the desired trajectory, is designed and this controller is used to form the platoon by means of master-slave approach.

Two vehicle models are designed for the simulations, a simple bicycle model and a more realistic model including tire behavior. These models mimic the dynamics of a front driven front steered passenger car. In the controller design, a virtual control point is formed and input-output linearization by state feedback approach is used. With these models, the efficiency of the controller is investigated for both a single vehicle case and a car platoon case. The resulting plots and the conclusions derived from these plots are presented in the report. As a conclusion, the controller (especially used with the simple model) achieved to track its trajectory with negligible errors, which also enables the use of cooperative control strategy. The theories given in this report can be implemented to real world vehicles in order to have autonomous vehicle platoons after a decent number of real-time experiments.
# Table of Contents

Abstract ......................................................................................................................................................... 2

1. INTRODUCTION ......................................................................................................................................... 5
   1.1 Problem statement ............................................................................................................................. 5
   1.2 Outline................................................................................................................................................. 5

2. BICYCLE MODEL ........................................................................................................................................ 6
   2.1 The coordinate system ........................................................................................................................ 6
   2.2 The constraint equations .................................................................................................................... 7
   2.3 Posture kinematic model .................................................................................................................... 9
   2.4 Configuration kinematic model ........................................................................................................ 10
   2.5 Configuration dynamic model ........................................................................................................ 10
   2.6 Feedback equivalent configuration dynamic model and Posture dynamic model .................. 13
   2.7 Physical interpretation of the model ................................................................................................ 14
   2.8 Realistic bicycle model ...................................................................................................................... 15

3. CONTROLLER FOR THE BICYCLE MODEL ................................................................................................. 17
   3.1 The controller development for the bicycle model .......................................................................... 17
   3.2 Simulation results ............................................................................................................................. 20
      3.2.1 Position error compensation ..................................................................................................... 20
      3.2.2 Compensation of change in speed ............................................................................................. 22
      3.2.3 Response on a circle and 8-shaped track ................................................................................... 24
   3.3 Conclusions ....................................................................................................................................... 26

4. CONFIGURATION CONTROL FOR COOPERATIVE DRIVING ...................................................................... 27
   4.1 The configuration control theory ...................................................................................................... 27
   4.2 Simulation results ............................................................................................................................. 28
      4.2.1 Position error compensation ..................................................................................................... 28
      4.2.2 Compensation of change in speed ............................................................................................. 33
      4.2.3 Response on a circle and 8-shaped track ................................................................................... 34
   4.3 Conclusions ....................................................................................................................................... 37

_Toc248612505

_Toc248612505
CHAPTER 1

INTRODUCTION

The world’s mobility problem is getting larger and larger due to the increase in the population and the distances people travel. With the increase in the number of the vehicles and therefore, the increase in traffic; the safety concerns arise. In the light of these arguments, the cooperative driving is seen to be the recipe of these problems. With the introduction of the cooperative driving, the automated and communicating vehicles will form adaptive platoons, which will in turn increase the efficiency and the safety by forming smooth flows.

The modeling and control of a 1-track vehicle model and cooperative control of these models are the topics of this report. 1-track model (also referred as bicycle model) is used since it is enough to mimic all of the dynamics except the roll dynamics of the vehicle. A PD-controller is designed for the model in order to track the reference trajectory. The controller is designed by means of so-called feedback linearization. And afterwards, cooperative driving is obtained using master-slave approach.

1.1 Problem statement
The objectives that form the problem statement are;

1. Design a tracking controller for road vehicles.
2. Obtain cooperative driving by building a control theory using master-slave approach [1].

1.2 Outline
This report is presented as follows. In chapter 2, the kinetic and kinematic equations of the bicycle model are derived and the physical interpretation of found results is given. At the end of this chapter, the more complex and realistic bicycle model built for the simulations is described. In chapter 3, controllers for the individual vehicles are designed and the responses to some selected scenarios are given. In chapter 4, configuration control is discussed and the responses to the same scenarios are given. Finally, the conclusions are presented in chapter 5.
CHAPTER 2

BICYCLE MODEL

In this chapter, the general procedure for the determination of the bicycle model is described. In the study, the bicycle model (1-track model) is used since the modeling of a car is quite complicated and unnecessary for the purpose of the study. The bicycle model in this study is a front wheel driven, front wheel steered car which is the most-commonly used type of car.

The dynamics of the model will be determined using the velocity constraints. Since the front and rear wheels are each lumped into one wheel, the lateral dynamics are derived in a simplified way.

The basic solution steps described in [2] will be followed in this report.

2.1 The coordinate system

The first step in the derivation of the model is to define coordinate systems. The coordinate definitions are given in Figure 1. In the figure, there are two coordinates which are being absolute \((x,y)\) and local \((x_l,y_l)\).

![Figure 1: The Coordinate System](image)

To be able to switch from one coordinate system to another, a transformation matrix \(R(\theta)\) is introduced where \(\theta\) is the orientation angle of the car with respect to the global frame.

\[
R(\theta) = \begin{pmatrix}
\cos(\theta) & \sin(\theta) & 0 \\
-\sin(\theta) & \cos(\theta) & 0 \\
0 & 0 & 1
\end{pmatrix}
\] (2.1)

The transformation from global frame to local frame is done accordingly:

\[
\begin{pmatrix}
\dot{x}_l \\
\dot{y}_l \\
\dot{\theta}
\end{pmatrix} = R(\theta) \begin{pmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta}
\end{pmatrix}
\] (2.2)
2.2 The constraint equations
For the bicycle model shown in Figure 2, six generalized variables exist to describe the various motions of the model. These are the position variables as $x$, $y$ and $\theta$; steering angle $\beta$; and the rotation angles of the wheels $\varphi_1$ and $\varphi_2$. The sign convention of $\varphi_1$ and $\varphi_2$ is shown in Figure 3. These variables are shown in a matrix, which will be referred later:

$$q = \begin{pmatrix} x \\ y \\ \theta \\ \beta \\ \varphi_1 \\ \varphi_2 \end{pmatrix}$$

(2.3)
There are two holonomic constraints which describe the motion of the kinematic model. The first constraint is the slip constraint which is the restriction of the movement in lateral direction. The second constraint is again a slip constraint obtained by the assumption that the wheel rolls without slip. With this constraint relation between the forward velocity and the wheel rotation angle is derived.

The first slip constraint (lateral direction) equation for the model is:

\[
C_1 R(\theta) \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} \sin(\beta) & -\cos(\beta) & -L \cos(\beta) \\ 0 & 1 & -L \end{pmatrix} R(\theta) \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = 0
\]

where \( L \) is the half distance between the wheels (also the distance between the wheel and the local coordinate frame origin). In (2.4), the first row of \( C_1 \) describes the slip constraint in the front wheel, and the second row describes the one in rear wheel.

The second slip constraint (forward direction) equation for the model is:

\[
J_1 R(\theta) \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} + J_2 \begin{pmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{pmatrix} =
\]

\[
= \begin{pmatrix} \cos(\beta) & \sin(\beta) & L \sin(\beta) \\ 0 & 0 & 0 \end{pmatrix} R(\theta) \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} + \begin{pmatrix} r \\ 0 \\ r \end{pmatrix} \begin{pmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{pmatrix} = 0
\]

where \( r \) is the radius of the wheel. In (2.5), again the first row of \( J_1 \) and \( J_2 \) formulates the slip constraint in front wheel, and the second row formulates the one in rear wheel.

The degree of steerability and mobility should be defined before the derivation of the kinematic and dynamics models. The degree of steerability is the number of conventional steering wheels, in other words, number of the steering angles. The degree of mobility is the degree of freedom which can be obtained while keeping the steering angles constant.

The degree of steerability is given by the following equation:

\[
\delta_s = \text{rank}(C_1(\beta)) = \text{rank}(\begin{pmatrix} \cos(\beta) & \sin(\beta) & L \sin(\beta) \end{pmatrix})
\]

(2.6)

where \( C_1 \) is first matrix of the slip constraint (lateral) equation. Therefore, the degree of steerability of the model is 1 which is obvious since the car can be just steered from the front wheel.

The degree of mobility is given by the following equation:

\[
\delta_m = \dim(N[C_1]) = 3 - \text{rank}[C_1]
\]

(2.7)

where \( \dim(N[C_1]) \) is the dimension of the null space of \( C_1 \). Knowing the fact that the null space of \( C_1 \) includes the all possible movements and by applying the rank-nullity theorem, the above equation is derived. When calculated the mobility of the model is 1 as the only movement is going forward and backward in a line or in a circle if the front wheel is steered before.
As a result, the bicycle model is so-called (1,1) type mobile robot where the first term in brackets represents the degree of mobility and the second term represents the degree of steerability.

### 2.3 Posture kinematic model

The next step is to find the posture kinematic model which is just about the motion in the x-y plane. Therefore, wheels rotations are not included in this model which makes this model the simplest one. $C_1$ is the relevant matrix in finding the model. In order to find the model, the solution of $\dot{\xi}$ should be found. However, it is impossible to solve (2.4) for $\dot{\xi}$ which is the short notation of:

$$\dot{\xi} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix}$$  \hspace{1cm} (2.8)

In (2.4), $R(\theta)\dot{\xi}$ can be regarded as the null-space of $C_1$. For this reason, one should define a $\Sigma(\beta)$ matrix whose columns span the null-space of $C_1$ and which will give the following equation; $C_1 \Sigma(\beta) = 0$ and define an input vector $\eta_a$ ($\eta_a$ is scalar (=dimension of $\delta_m$) for this specific case). Using this definitions one can derive the following relation:

$$R(\theta)\dot{\xi} = \Sigma(\beta)\eta_a$$  \hspace{1cm} (2.9)

which gives:

$$\dot{\xi} = R^T(\theta)\Sigma(\beta)\eta_a$$  \hspace{1cm} (2.10)

Using the fact that $C_1 \Sigma(\beta) = 0$ the following solution can be obtained:

$$\Sigma(\beta) = \begin{pmatrix} 2L \cos(\beta) \\ L \sin(\beta) \\ \sin(\beta) \end{pmatrix}$$  \hspace{1cm} (2.11)

(2.10) almost describes the posture kinematic model; however, due to the steering angle, the system is not linear. In order to linearize the system, $\dot{\beta} = \zeta$ relation is added to the model. So, the final posture kinematic model is:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} R^T(\theta)\Sigma(\beta) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \eta_a \\ \zeta \end{pmatrix}$$  \hspace{1cm} (2.12)

knowing $R(\theta)$ and $\Sigma(\beta)$ from (2.1) and (2.11), respectively,

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} 0.5L \cos(\theta + \beta) + 0.5L \cos(-\theta + \beta) + L \cos(\theta + \beta) & 0 \\ 0.5L \sin(\theta + \beta) - 0.5L \sin(-\theta + \beta) + L \sin(\theta + \beta) & 0 \\ \sin(\beta) & 0 \end{pmatrix} \begin{pmatrix} \eta_a \\ \zeta \end{pmatrix}$$  \hspace{1cm} (2.13)

For this case $\eta_a$ represents the forward velocity in the direction of front wheel and $\zeta$ represents the steering velocity. (see also section 2.7)
2.4 Configuration kinematic model

In this step the configuration kinematic model is derived, which is just the addition of the wheel rotation variable equations to the posture kinematic model.

The configuration kinematic model is:

\[
\begin{pmatrix}
\dot{\xi} \\
\dot{\beta} \\
\dot{\phi}
\end{pmatrix} =
\begin{pmatrix}
R^T(\theta)\Sigma(\beta) & 0 & 0 \\
0 & I & 0 \\
-J_z^{-1}J_1(\Sigma(\beta)) & 0 & 0
\end{pmatrix}
\begin{pmatrix}
\eta_a \\
\zeta
\end{pmatrix}
\]

(2.14)

To make it compact, the equation can be expressed as:

\[
\dot{q} = S(q)\eta
\]

(2.15)

where the notations express the matrices in the order that is in the previous equation.

2.5 Configuration dynamic model

A dynamics equation is needed to express the already derived configuration parameters in a dynamical way. In order to do it, a standard mechanical differential equation is used.

\[
M(q)\ddot{q} = C(q, \dot{q}) + F + B(q)\tau
\]

(2.16)

where \(M(q)\) is the mass matrix, \(C(q, \dot{q})\) contains the centripetal, Coriolis terms and encompasses joint flexibility effects and gravitational effects, \(F\) is the total generalized force vector consisting of constraint forces, \(B(q)\) represents the generalized torque directions of the torques and \(\tau\) is the input torque vector.

While constructing the matrixes above, the assumption of absence of friction forces doing work against the system is made. Since the car is a front driven, front steered type, there are two input torques, the driving and the steering torque.

The matrices of the dynamical equation are:
Since $F$ is the force vector containing the constraint forces, it can be formulated by the constraint equation matrix $A$ and the so-called Lagrange multiplier vectors $\lambda$.

$$F = A\lambda$$  \hspace{1cm} (2.19)$$

where $A = \begin{pmatrix} C_1 R(\theta) & 0 & 0 \\ J_1 R(\theta) & 0 & J_2 \end{pmatrix}^T$ and $\lambda_1$ (2x1) is the friction force vector preventing the wheel from slipping laterally and $\lambda_2$ (2x1) is the friction force vector ensuring that the wheel turns without slip.

$$B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$  \hspace{1cm} (2.20)$$

$$\tau = \begin{pmatrix} T_s \\ \tau_d \end{pmatrix}$$  \hspace{1cm} (2.21)$$

where $m$ is the mass of the car, $I_\theta$ is the inertia of the car about the axis perpendicular to $x$ and $y$, $I_s$ is the inertia of the wheel about its axis of steering, $I_\varphi$ is the inertia of the wheel about the axis of rotation, $A\lambda$ is the representation of the lagrange forces ensuring the system obeys the constraints, $\tau_s$ is the steering torque and $\tau_d$ is the driving torque.

Now, the configuration dynamic model can be written by taking the derivative of the configuration dynamic model (2.15):

$$\ddot{\eta} = \dot{S}\eta + S\dot{\eta}$$  \hspace{1cm} (2.22)$$

From (2.4) and (2.5), it is known that $A^T\dot{q} = 0$. Using this fact and equalities $S^T F = 0$ is desired to be obtained as follows:

Using (2.15):

$$A^T\dot{q} = A^T S(q)\eta = 0$$  \hspace{1cm} (2.23)$$

Therefore:

$$A^T S(q) = 0$$  \hspace{1cm} (2.24)$$

since $\eta$ cannot be zero. (Except stand-still position which is trivial)

From (2.19), $A^T\dot{q} = 0$ implies:
\[ F^T \dot{q} = 0 \quad (2.25) \]

since \( \lambda \) cannot be equal to zero. So, (2.24) becomes:

\[ F^T S(q) = 0 \quad (2.26) \]

Finally taking the transpose of both terms leads to:

\[ S^T F = 0 \quad (2.27) \]

Since the values of the Lagrange forces are not known, it is desired to eliminate them by pre-multiplying (2.16) with \( S^T \) and knowing the fact that \( S^T F = 0 \):

\[
\begin{bmatrix} \dot{\eta} \\ \eta \end{bmatrix} = \left( (S^T MS)^{-1} S^T (-MS\eta + C + B\tau) \right)
\]

where

\[
\dot{S} = \begin{bmatrix}
2L \left( -\dot{\theta} \sin(\theta) \cos(\beta) - \dot{\beta} \sin(\beta) \cos(\theta) \right) - L(\dot{\theta} \cos(\theta) \sin(\beta) + \dot{\beta} \sin(\theta) \cos(\beta)) & 0 \\
n2L \left( \dot{\theta} \cos(\theta) \cos(\beta) - \dot{\beta} \sin(\beta) \sin(\theta) \right) + L(\dot{\beta} \sin(\theta) \sin(\beta) + \dot{\beta} \cos(\theta) \cos(\beta)) & 0 \\
\dot{\beta} \cos(\beta) & 0 \\
0 & 0 \\
0 & 0 \\
-\frac{2L}{r} \sin(\beta) \dot{\beta} & 0
\end{bmatrix}
\]

(2.29)

and makes the model:

\[
\begin{bmatrix} \dot{\xi} \\ \dot{\gamma} \\ \dot{\theta} \\ \dot{\beta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\eta}_a \\ \dot{\zeta} \end{bmatrix} = \begin{bmatrix}
\frac{2L \cos(\theta) \cos(\beta) - L \sin(\theta) \sin(\beta)}{r} \eta_a \\
\frac{2L \sin(\theta) \cos(\beta) + L \cos(\theta) \sin(\beta)}{r} \eta_a \\
\sin(\beta) \eta_a \\
0 \\
0 \\
-\frac{2L}{r} \eta_a \\
\frac{2L \cos(\beta)}{r} \eta_a \\
\frac{rrd}{mrr^2 + lq} \\
\frac{\tau_s}{l_s}
\end{bmatrix}
\]

(2.30)
2.6 Feedback equivalent configuration dynamic model and Posture dynamic model

Regarding the configuration dynamic model in the previous step, the final step is to find the feedback equivalent of the configuration dynamic model and reduce it to a more useful form, namely posture dynamic model. In order to achieve this, an arbitrary reference input, \( \nu = [\nu_1, \nu_2] \) is defined.

The torque input \( \tau \) can be found from the lower part of \( (2.28) \) as follows:

\[
\tau = (S^T B)^+ S^T M (S \nu + S \eta)
\]

where \((B)^+\) is the pseudo inverse of \(B\). However, pseudo inverse can be replaced by inverse for this specific case.

The torque input can easily be defined for the system as:

\[
\begin{pmatrix}
\tau_d \\
\tau_s
\end{pmatrix} = \frac{\nu_1 (mr^2 + I_g)}{r} \begin{pmatrix}
\nu_1 I \nu_1 \\
\nu_2 I_s
\end{pmatrix}
\]

(2.32)

The feedback equivalent configuration dynamic model is:

\[
\begin{pmatrix}
\dot{x} \\
\dot{\gamma} \\
\dot{\theta} \\
\dot{\beta} \\
\dot{\phi}_1 \\
\dot{\phi}_2 \\
\dot{\eta}_a \\
\dot{\zeta}
\end{pmatrix} = \begin{pmatrix}
(2L \cos(\theta) \cos(\beta) - L \sin(\theta) \sin(\beta)) \eta_a \\
(2L \sin(\theta) \cos(\beta) + L \cos(\theta) \sin(\beta)) \eta_a \\
\sin(\beta) \eta_a \\
\zeta \\
-\frac{2L}{r} \eta_a \\
-\frac{2L \cos(\beta)}{r} \eta_a \\
\nu_1 \\
\nu_2
\end{pmatrix}
\]

(2.33)

Finally, the posture dynamic model is:

\[
\begin{pmatrix}
\dot{x} \\
\dot{\gamma} \\
\dot{\theta} \\
\dot{\beta} \\
\dot{\eta}_a \\
\dot{\zeta}
\end{pmatrix} = \begin{pmatrix}
(2L \cos(\theta) \cos(\beta) - L \sin(\theta) \sin(\beta)) \eta_a \\
(2L \sin(\theta) \cos(\beta) + L \cos(\theta) \sin(\beta)) \eta_a \\
\sin(\beta) \eta_a \\
\zeta \\
\nu_1 \\
\nu_2
\end{pmatrix}
\]

(2.34)
2.7 Physical interpretation of the model

Up to now, all equations are found using mathematical formulae without giving any physical interpretation. Instead of using the above mathematical formulae, one can also obtain the same model by just using the mechanics. The physical meaning of the configuration dynamic model is elaborated in the following paragraphs.

Explaining the physical meaning of $\eta_a$ will be a good starting point. $\eta_a$ represents the forward velocity of the front wheel of the vehicle divided by $2L$. Using this fact, $\phi_1$, the rotational velocity of the front wheel, can be better understood:

$$\phi_1 = -\frac{V}{r} = -\frac{2L\eta_a}{r} = \frac{-2L}{r}\eta_a$$  \hspace{1cm} (2.35)

where $V$ is the forward velocity of the front wheel.

Knowing the fact that the rear wheel is the instantaneous centre of rotation, the equations of $\dot{\theta}, \dot{x}$ and $\dot{y}$ can be derived using the following physical relations.

In order to find $\dot{\theta}$, the principle of rotation around a fixed point (rear wheel) and the following relation is used. The perpendicular velocities are:

$$\dot{\theta}L = \frac{\sin(\beta)V}{2} = \frac{\sin(\beta)(2L\eta_a)}{2} = \sin(\beta)L\eta_a$$  \hspace{1cm} (2.36)

after cancelling $L$ from both sides, already derived equation is obtained with another way:

$$\dot{\theta} = \sin(\beta)\eta_a$$  \hspace{1cm} (2.37)

Similar to $\dot{\theta}$, derivations of $\dot{x}$ and $\dot{y}$ use the same principle, instantaneous point of zero velocity.

First the forward velocity is decomposed into its components (the corresponding angle is $\beta$), then each velocity vector is again decomposed into its components (the corresponding angle is $\theta$). Finally, they are added or subtracted according to their signs. The resulting equations are:

$$\dot{x} = V\cos(\beta)\cos(\theta) - 0.5V\sin(\beta)\sin(\theta) = (2L\cos(\theta)\cos(\beta) - L\sin(\theta)\sin(\beta))\eta_a$$  \hspace{1cm} (2.38)

$$\dot{y} = V\cos(\beta)\sin(\theta) + 0.5V\sin(\beta)\cos(\theta) = (2L\sin(\theta)\cos(\beta) + L\cos(\theta)\sin(\beta))\eta_a$$  \hspace{1cm} (2.39)

One should note that the division of the velocities by 2 is due to the translating the velocity at front wheel to the middle of the bicycle.

Furthermore, $\zeta$ represents the steering angle velocity and $\zeta$ represents the steering angle acceleration.
2.8 Realistic bicycle model

The tracking controller of this study is designed using the simple bicycle model since both use the same mathematical theory. However, another model is needed to evaluate this controller and see the effects of possible real-time complexities such as tire effects. Therefore, a more complex bicycle model is given in this chapter. Since this model is a more realistic model, the simulation results run with this model are given in the name of \textit{realistic model}. This model is built mainly with the components that are designed by [3].

The model that will be described in this chapter is an electric car with an electric drive line and electric steering system. Furthermore, it also has a tire model dealing with the slip angles and the relaxation length. The car body (1-track, 3-DOF) has the inputs which are the outputs of these components and calculates the final angles, positions, velocities, accelerations and the resulting forces. The car body also takes the air drag into consideration.

The electric drive line consists of a brushless DC drive, a gearbox and a wheel. The brushless DC drive is an electric motor which is equipped with a torque controller. A conventional gearbox is used in order to include the friction in the gears. A wheel is used to calculate the rotational wheel velocity knowing the driving, brake and rolling friction torques.

The electric steering system also consists of a brushless DC and a gearbox same as the drive line. Using this electric steering system and knowing the inertia about $z$-axis, the desired steering angle is found.

The modeling of the tire is rather complex due to the nonlinearities and the limited number of inputs. In order to model this nonlinear behavior, an empirical approach, the Magic Formula is used. The magic formula is nothing but a mathematical formula that correlates the slip values to the slip characteristics. To be more precise, the following characteristics are calculated using the base Magic formula:

\begin{equation}
F_x = f(\kappa), \quad F_y = g(\alpha), \quad M_z = h(\alpha)
\end{equation}

where $F_x$ denotes longitudinal slip, $F_y$ denotes lateral slip and $M_z$ denotes self-aligning moment. Additionally, $\kappa$ and $\alpha$ express longitudinal and lateral slip values, respectively. The behavior of the tire force as a function of the slip values is important and the relation between them are given in Figure 4 [4]. Finally the Magic Formula to be used in the functions is:

\begin{equation}
F = D \sin(C \arctan(1 - E)Bx + E \arctan(Bx)))
\end{equation}

where $D$ determines the peak value, $C$ determines the limit value when $x$ goes to infinity, $B$ & $C&D$ determine the slope near the origin and $B&C&E$ determine the location of the peak.
Despite not revealing the real case characteristics fully, the model is quite realistic to be used to express the maneuverability characteristics of a car in this kind of simulations.
CHAPTER 3

CONTROLLER FOR THE BICYCLE MODEL

In this chapter, the design process of the controller for a single vehicle is explained. Since the derived dynamic model is nonlinear, input-output linearization by state feedback approach is used at the beginning of the design. By means of that, the normal linear control techniques became possible to be used. After the design of the controller, some simulations in order to evaluate the performance of the controller are presented.

3.1 The controller development for the bicycle model

The aim of the study is to develop a controller for a car which can track a certain trajectory defined according to an arbitrary point of the car. This arbitrary point is assumed to be the center of the car at which the origin of the coordinate axes are defined, and the reference trajectory is in the form of \([x_{\text{ref}}, y_{\text{ref}}, \theta_{\text{ref}}]\) in this particular case. The largest subsystem that forms a controllable linear subsystem is in the dimension of \(2(\delta_x + \delta_m) = 4\). The dimension of the nonlinear subsystem is \(3 - \delta_m = 2\) [2]. The first step in the development of the controller for the bicycle model is to define the controller point which is also called as the virtual control point. The output linearizing vector \(z_1\) denotes this point and also in the dimension of \(\delta_x + \delta_m = 2\). Since the system is described by the posture variables \((q = [x, y, \theta])\) and the dimension of \(z_1\) is 2, there will be a point tracking problem. If a point on the bicycle such as \(x\) and \(y\) is chosen, then \(\theta\) cannot be made equal to the reference. Therefore, a point in front of the bicycle is selected which will make the variables equal to the reference signals in case of the presence of forward velocity. This control point is shown in Figure 5.

\[ y \]
\[ x \]

Figure 5: The virtual control point \(z_1\)
Before defining the control point, let’s write the dynamic model to be controlled using “input-output linearization by state feedback”.

\[
\dot{q}_1 = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = S_1(q_1)\eta = \begin{pmatrix} (2L\cos(\theta)\cos(\beta) - L\sin(\theta)\sin(\beta)) & 0 \\ (2L\sin(\theta)\cos(\beta) + L\cos(\theta)\sin(\beta)) & 0 \\ \sin(\beta) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \eta_a \\ \zeta \end{pmatrix}
\] (3.1)

and

\[
\dot{\eta} = \nu
\] (3.2)

Due to the nonlinearities involved in the system, linearization is required in order to use the normal control techniques. The first step is to define the control point, \(z_1\) which is:

\[
z_1 = h(q_1) = \begin{pmatrix} x + L\cos(\theta) + e_c\cos(\theta + \beta) \\ y + L\sin(\theta) + e_c\sin(\theta + \beta) \end{pmatrix}
\] (3.3)

where \(e_c\) is the distance between the front wheel and the virtual control point shown in Figure 5.

In the next step, \(z_2 = \dot{z}_1\) is defined which is equal to:

\[
z_2 = \dot{z}_1 = \frac{\partial h}{\partial q_1} q_1 + \frac{\partial h}{\partial q_1} S(q_1)\eta = H\eta = \begin{pmatrix} \dot{x} - L\sin(\theta)\dot{\theta} - e_c\sin(\theta + \beta)(\dot{\theta} + \dot{\beta}) \\ \dot{y} + L\cos(\theta)\dot{\theta} + e_c\cos(\theta + \beta)(\dot{\theta} + \dot{\beta}) \end{pmatrix}
\] (3.4)

In this case \(H\) becomes:

\[
H = \begin{pmatrix} (2L\cos(\theta + \beta) - 0.5e_c\cos(\theta) + 0.5e_c\cos(\theta + 2\beta) & -e_c\sin(\theta + \beta) \\ 2L\sin(\theta + \beta) - 0.5e_c\sin(\theta) + 0.5e_c\sin(\theta + 2\beta) & e_c\cos(\theta + \beta) \end{pmatrix}
\] (3.5)

Since the input \(\nu\) has not showed up yet, the next step is to differentiate \(z_2\):

\[
\dot{z}_2 = \ddot{z}_1 = H\dot{\eta} + H\eta = H\nu + b(q_1, \eta)
\] (3.6)

where \(b\) is:

\[
b(q_1, \eta) = \begin{pmatrix} \hat{H}_{11} & -e_c\cos(\theta + \beta)(\eta_a\sin(\beta) + \zeta) \\ \hat{H}_{21} & -e_c\sin(\theta + \beta)(\eta_a\sin(\beta) + \zeta) \end{pmatrix} \begin{pmatrix} \eta_a \\ \zeta \end{pmatrix}
\] (3.7)

where \(\hat{H}_{11}\) and \(\hat{H}_{21}\) are:

\[
\hat{H}_{11} = -2L\sin(\theta + \beta)(\eta_a\sin(\beta) + \zeta) + 0.5e_c\sin(\theta)\eta_a\sin(\beta) - 0.5e_c\sin(\theta + 2\beta)(\eta_a\sin(\beta) + 2\zeta)
\] (3.8)

\[
\hat{H}_{21} = 2L\cos(\theta + \beta)(\eta_a\sin(\beta) + \zeta) - 0.5e_c\cos(\theta)\eta_a\sin(\beta) + 0.5e_c\cos(\theta + 2\beta)(\eta_a\sin(\beta) + 2\zeta)
\] (3.9)

knowing the fact that:

\[
\eta_a = \frac{\dot{\theta}}{\sin(\beta)} \text{ and } \zeta = \dot{\beta}
\] (3.10)
Although $z_1$ and $z_2$ and their derivatives are enough to track the reference inputs, one more variable which will deal with the internal dynamics is needed. With the addition of this variable, full mapping between the controller parameters and the controlled parameters will be present. In the absence of this variable, the tracking will be achieved; however, the internal stability will not be succeeded for all cases. Therefore, $z_3$ is defined as:

$$z_3 = \dot{z}_2$$  \hspace{1cm} (3.11)

The following representations summarize the above explained procedure:

$$z_1 = h(q_1)$$  \hspace{1cm} (3.12)

$$z_2 = H(q_1)\eta$$  \hspace{1cm} (3.13)

$$z_3 = k(q_1)$$  \hspace{1cm} (3.14)

Furthermore, $\dot{z}_3$ can be defined as:

$$\dot{z}_3 = \frac{\partial k}{\partial q_1} q_1 = \frac{\partial k}{\partial q_1} S H^{-1} z_2 = Q(q_1) z_2$$  \hspace{1cm} (3.15)

It should be noted that the previous equation has a solution if and only if $H$ is a non-singular matrix. Therefore, $z_1$ should be chosen accordingly which makes $H$ invertible for all cases. Fortunately, the determinant of $H$ matrix is equal to $2e_z L$ which assures that $H$ is not a singular matrix for all cases.

In order to linearize the system the following equality is needed:

$$\dot{z}_2 = \dot{z}_1 = Hv + b(q_1, \eta) = w$$  \hspace{1cm} (3.16)

and by means, the following equation is derived:

$$v = H^{-1}(w - b)$$  \hspace{1cm} (3.17)

The solution for the input $w$ making the error in $z_1$ and $z_2$ zero is:

$$w = \dot{z}_{1,ref} - K_v (\dot{z}_1 - \dot{z}_{1,ref}) - K_p (z_1 - z_{1,ref})$$  \hspace{1cm} (3.18)

where $K_v$ and $K_p$ are the gains of the PD controller.

The final system for the controller part is:

$$\dot{z}_1 = z_2$$  \hspace{1cm} (3.19)

$$\dot{z}_2 = w$$  \hspace{1cm} (3.20)

$$\dot{z}_3 = Q(q_1) z_2$$  \hspace{1cm} (3.21)
3.2 Simulation results

In this section, simulation results carried out with both the simple and the realistic bicycle models will be presented. In the first part of this section, how good the controller compensate for position errors will be examined for simple disturbances and starting errors. Then, the effectiveness of the controller will be assessed for speed changes. Finally in the last part, the response of the controller in more complex trajectories will be shown.

The distance $e_c$ between the front wheel and the virtual control point is selected as a constant value of 0.35 m since the effect of $e_c$ is minor in these simulation and should be in the interval of 0.25 and 0.5 for velocities around 10 km/h as proposed in [2].

3.2.1 Position error compensation

In this section, two types of situations will be simulated where the position of the vehicle is examined.

The first situation is the response for starting errors, in other words initial conditions. The models will be started 1 meter away (in $y$-direction) from the starting point. The trajectory for these simulations is just a straight line where the car starts and ends at zero velocity and the middle part is travelled at constant speed. The constant speed is equal to 15 km/h and the maximum acceleration is $4 \text{ m/s}^2$. In Figure 6 and Figure 7, the resulting plots are shown. In both cases, the error, which is difference between the reference position and the car centre, becomes approximately zero around 15 meters in $x$-direction. The virtual control point is also given as well as the body centre point. One can observe that for the realistic model, oscillations are present in the virtual control point. This is basically resulted from the changes in the tire forces due to the nonlinear behavior of the tire force shown in chapter 2.8 in Figure 4 which makes the controlling more difficult.

![Figure 6: $y$-position vs $x$-position with a starting error (simple model)](image-url)
The second situation is the observation of a step response on a straight line which is the lane change action of a vehicle physically. The speed is kept constant after the startup and before the stop. This constant velocity is 10 km/h and the maximum acceleration seen in \( x \) direction is 4 m/s\(^2\). A step disturbance is applied at 3 meters of the straight line which has a magnitude of 0.25 meter. It can be observed from both Figure 8 and Figure 9 that the error becomes approximately zero around 16 meters. As in the first situation, the behavior of the realistic model is oscillatory and as a consequence it can get fully (i.e. error is less than 0.001 m) to the track 1 meter later.

Figure 7: \( y \)-position vs \( x \)-position with a starting error (realistic model)

Figure 8: \( y \)-position vs \( x \)-position for a step response (simple model)
3.2.2 Compensation of change in speed

The same reference trajectory as in the first simulation is also used in this set of simulations. However, in this case the speed change and how well the controller can track these changes are investigated. The speed curve of this trajectory is given in Figure 10: Reference. The speed is taken as the velocity in $x$-direction since the car is driven in a straight line and the error in speed is plotted as a time graph in Figure 11 and Figure 12. Although the error is quite small for the simple model, it gets larger for the realistic model as it includes the tire and also other driveline elements. The time required for the damping of the error is more or less the same for both situations.
Figure 11: Velocity error vs time (simple model)

Figure 12: Velocity error vs time (realistic model)
3.2.3 Response on a circle and 8-shaped track

In this set of simulations, more difficult tracks compared to previous ones are used to test the effectiveness of the controller. There are two types of trajectories which are in the shape of a circle and an eight. These trajectories are rather impossible to be tracked for the realistic bicycle model since they are designed for the four wheels driven four wheels steered mobile robots and the accelerations are quite high. The simulations of these tracks always start at the origin of the graphs and the cars start their motion heading to the right.

The first trajectory is the o-track where the maximum speed is 30 km/h and the maximum centripetal acceleration is 7 m/s$^2$. The radius of the track is approximately 10 meters which generates a challenging track for the designed realistic bicycle model including the tire behavior with these speed and acceleration values.

In Figure 13, tracking behavior of the controller in a circle track is given for the simple bicycle model. The controller can track the reference trajectory almost perfectly with an offset of 0.13 meters which diminishes when the virtual control point trajectory is the focus of concern.

In Figure 14, tracking behavior of the controller in a circle track is given for the realistic bicycle model. The controller cannot achieve to track the reference at the beginning. The reason is the fact that due to high accelerations, the controller tends to track the trajectory using shortcuts. After maintaining a constant speed, the controller can track the trajectory perfectly corresponding to the controller of simple model.
The second challenging trajectory is the eight-track where the maximum speed is 20 km/h and the maximum centripetal acceleration is $9 \text{ m/s}^2$. The minimum radius of curvature of this eight-shaped track is 3.5 m which makes the trajectory tougher than the o-shaped one.

The resulting plot for the simple bicycle model is given in Figure 15. The same offset is also seen in this track as were the case in the o-track.

The level of effectiveness of the controller for the realistic model can be seen in Figure 16. The trajectory becomes almost impossible to track when the tires are included. However, it can still maintain the shape of the trajectory. If the curvature is increased and the accelerations are decreased, the desired tracking can be achieved.
3.3 Conclusions
The controller described in this chapter, mostly achieved to track the reference trajectories. The simulations run by the simple model was satisfactory although some shifts are seen due to the controlled point. This result is expected since the virtual point control approach is used. The simulations run by the realistic model was also satisfactory, however oscillatory behavior could not be avoided. This result is also not surprising since this model includes the nonlinear tire model. In overall, the controller was successful to track the trajectories with minor errors.
CHAPTER 4

CONFIGURATION CONTROL FOR COOPERATIVE DRIVING

In this chapter, the control theory for cooperative driving of two or more cars is explained. The main goal of the controllers of each car will be to drive at certain trajectories; in turn the cooperative driving will be obtained which means that cars will follow each other at specified distances. After the design of this configuration control theory, the regarding simulations with two cars will be presented.

4.1 The configuration control theory

In the formation control, the master-slave approach, as referred in [1], will be used. Although, the virtual structure approach is better when the robustness with respect to keeping the formation in the presence of disturbances on the slave is the concern, the master-slave approach is used for its ease of implementation.

In this report, the master-slave approach will be simulated with two cars keeping a predetermined distance in both $x$ and $y$ directions. The car going at the front and tracking the reference trajectory will be called the master and the one tracking the trajectory of the master will be called the slave. These are indicated by the subscripts $m$ and $s$ will be added, respectively. The same models will be used in this chapter as used in chapter 3.

Therefore, the only difference apart from chapter 3 will be the reference trajectory of the slave which can be described as:

$$x_{ref,s} = x_m + l_x \text{ and } y_{ref,s} = y_m + l_y$$

(4.1)

where $l_x$ is the constant distance between the cars in $x$ direction and $l_y$ is the constant distance between the cars in $y$ direction.

For constant $l_x$ and $l_y$, it is guaranteed that:

$$\dot{x}_{ref,s} = \dot{x}_m, \dot{y}_{ref,s} = \dot{y}_m \text{ and } \ddot{x}_{ref,s} = \ddot{x}_m, \ddot{y}_{ref,s} = \ddot{y}_m$$

(4.2)

The only remaining variable is the orientation angle $\theta$ and its derivatives. It can also be proved that:

$$\theta_{ref,s} = \arctan\left(\frac{\dot{y}_{ref,s}}{\dot{x}_{ref,s}}\right) = \arctan\left(\frac{\dot{y}_m}{\dot{x}_m}\right) = \theta_m$$

(4.3)

which also means that the derivatives are equal.
4.2 Simulation results
In this section, the same simulations as they are given in chapter 3.1 are presented for two cars, one is following the other.

4.2.1 Position error compensation
The first simulation is again the compensation of the starting error. The master starts at a 1 meter vertical distance from the initial point of the trajectory and the slave tracks the new trajectory constructed by the master. In this simulation the distance between the center of the cars ($l_x$) is taken as -5 meters. In Figure 17, it is seen that the slave performs exactly the same trajectory as the master with 5 meter shift. This can be better understood by Figure 18. However, in Figure 19 the same oscillations also seen on the previous simulations for the realistic model exist. The effect of these oscillations is seen in Figure 20. Though, it can still track the reference.

![Figure 17: y-position vs x-position with a starting error at the master's position (simple model)]
Figure 18: Position difference (x -dir.) between the master and slave (simple model)

Figure 19: y-position vs x-position with a starting error at the master’s position (realistic model)
The second simulation is the lane change of two cars travelling collaterally. When the master car changes its line to the one at left, the slave car also does the same movement. The simulation for the simple model is presented in Figure 21, Figure 22 and Figure 23. The same behavior for the master and slave are observed with a 0.2 meters shift vertically for the slave. Figure 24, Figure 25 and Figure 26 shows the simulation result for the realistic model.
Figure 22: Position difference (x-dir.) between the master and slave (simple model)

Figure 23: Position difference (y-dir.) between the master and slave (simple model)
Figure 24: Lane change of two cars travelling collaterally (realistic model)

Figure 25: Position difference (x-dir.) between the master and slave (realistic model)
4.2.2 Compensation of change in speed

The compensation of the errors due to speed changes are desired to be observed in these simulations. A straight line trajectory is used, which starts from stand still, drive constantly and then again finishes at zero velocity. The errors are calculated by subtracting the velocity values from its own reference trajectory.

In Figure 27, the error in velocity due to the change of velocity is presented for the simple model. The error in the slave is larger than the master due to the accumulation of the errors. However, the errors of the both simulations are negligibly small.

![Graph of velocity error vs time (simple model)](image)

In Figure 28, the same procedure is given for the realistic model. However in this case, the error in the master is larger than the slave. This can be explained by the fact that the slave reference is a
more suitable trajectory that the controller can track, resulting in smaller errors. But it should also be noted that these errors are calculated according to own reference trajectories. In order to comprehend the real error for the slave, the errors shown in the graph should be added to each other, being also relevant for the simple model.

![Figure 28: Velocity error vs time (realistic model)](image)

4.2.3 Response on a circle and 8-shaped track
The remaining four simulations which are complex ones are given in this chapter. The same arguments stated in chapter 3.1.3 are also applicable for these simulations.

The cooperative control strategy for these four simulations has some small differences from the previously presented ones. Since it is meaningless for vehicles to drive in circles or eights which are apart from each other, \( l_x \) and \( l_y \) are not taken to be constant in order to achieve that the cars travelling side by side. In these cases \( l_x \) and \( l_y \) used in equation 4.1 are chosen as:

\[
\begin{align*}
    l_x &= -a \sin(\theta) \\
    l_y &= a \cos(\theta)
\end{align*}
\]

(4.4)

(4.5)

where \( a \) is the distance between cars in lateral direction.

Since \( l_x \) and \( l_y \) are not assigned as constant, equation 4.2 does not hold anymore for these specific cases. However, equation 4.3 is still valid since the cars are traveling at the same arc. The new equations which interchange with 4.2 are:

\[
\begin{align*}
    \dot{x}_{ref,s} &= \dot{x}_m - a \cos(\theta)\dot{\theta} \\
    \dot{y}_{ref,s} &= \dot{y}_m - a \sin(\theta)\dot{\theta} \\
    \ddot{x}_{ref,s} &= \ddot{x}_m + a \sin(\theta)\dot{\theta}^2 - a \cos(\theta)\ddot{\theta} \\
    \ddot{y}_{ref,s} &= \ddot{y}_m - a \cos(\theta)\dot{\theta}^2 - a \sin(\theta)\ddot{\theta}
\end{align*}
\]

(4.6)

(4.7)

(4.8)

(4.9)
The first trajectory is the o-track. The master drives in a circle with a radius of 10 meters and the slave derives in a circle with a radius of 8 meters. The results of these simulations are given in Figure 29 and Figure 30. The same discussions given for the non-cooperative case are also applicable for these figures. The controller of the slave is more successful than the master’s, since its trajectory is more suitable for the vehicle model.

The last simulations are done by using the eight-track. As in the previous one, cars travel side by side but in this case the arc radiuses for the slave change. The slave makes the cornering with a bigger radius (5.5 m) at the right arc but makes a smaller (1.5 m) at the left arc. The results are presented in Figure 31 and Figure 32. The slave of the simple model can track the trajectory perfectly; however, the slave of the realistic model is poor in tracking while cornering the small arc. The reason is not
surprising since the radius is too small for the vehicle model to corner. Another eight-track trajectory should be constructed for the application of this model.

Figure 31: Position graph of 8-track (simple model)

Figure 32: Position graph of 8-track (realistic model)
4.3 Conclusions
By using the controller explained in chapter 3, cooperative driving of two vehicles are obtained by introducing a configuration control theory. With the simulations, it is shown that the slave cars are capable of tracking their masters at a distance specified by the designer. Almost perfect tracking is obtained between the slave reference and the actual position of the slave center. Also the importance of the master’s actual position is emphasized as it becomes the reference trajectory of the slave.
CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

A position controller for a single bicycle model which reveals the basic properties of a front driven front steered car is designed in this report. Afterwards, a control strategy based on the designed controller for a single vehicle is developed for cooperative driving. Conclusions and recommendations based on the results of the simulations are presented in this chapter.

5.1 Conclusions

The first objective of this report is to develop a bicycle model which deals with the dynamics of a car except the roll motion. After applying various steps, the feedback equivalent configuration dynamic model is obtained to be used in the simulations. In addition to this model, a more complicated and realistic model which also includes the tire behavior and the properties of electric components is designed. A brief explanation of this model is also given.

The second objective of this report is to design a controller which will track the given trajectory. The concept of “virtual control point” is used for this purpose. Due to the nonlinearities involved, input-output linearization by state feedback principle is followed. For the linearized subsystem a feedback controller (PD) is designed. The controller when used with the simple mathematical model is robust with respect to the type of the trajectory and can track the desired trajectory with negligible errors. However, some shifts are observed for the complex trajectories since the controlled point is not the centre of the vehicle. The controller when used with the realistic model can still track the desired trajectory; however, oscillatory behavior cannot be eliminated because of the tire behavior, i.e. changing tire force for different slip angles. The errors are too large in the simulations of o-track and 8-track when it is performed with realistic model. However, this is not an unexpected result since in these tracks including high acceleration and deceleration with small radius arcs, the designed vehicle model cannot be kept on the track regardless of the controller. This is mainly due to the geometric limitations of the car model.

The last objective is to build a control theory so that the cooperative control of the vehicles with the predesigned controller is achieved. The master-slave approach [1] is followed to achieve this job. For the complex trajectories, i.e. ones including cornering, the approach is improved with some additions; defining the distances between the cars as a function of orientation angle. It is seen that the controllers when simulated with the simple model can track perfectly as for the single vehicle case. Oscillations are seen for the controllers when simulated with the realistic model. Moreover, in some cases even if the master does not have oscillations and can track pretty well, it is observed that the slave is not as successful as the master.

The final conclusion is that the controller (especially used with the simple model) is capable of tracking its trajectory with minor errors, which also enables the use of cooperative control strategy. The methods given in this report can be implemented to real world vehicles in order to have autonomous vehicles after a decent number of real-time experiments.
5.2 Recommendations
In this section, a number of recommendations are given to improve the designed controllers.

The trajectories used in this study are not designed specifically for the application of these models. New trajectories with larger radius of curvatures can be created.

More slaves can be used to see the effect of cumulative errors.

Instead of using the master-slave approach, a more robust approach, known as the Virtual Structure Approach [1], enabling the mutual coupling can be used. With the introduction of this approach, the disturbance effects on the slave can be better compensated.

The controllers should be verified with real-time experiments which were out of the scope of this study.
REFERENCES


