ISSN 0280-5316
ISRN LUTFD2/TFRT--5673--SE

Dynamic Tyre Models in Adaptive Slip Control

Lars Wallén

Department of Automatic Control
Lund Institute of Technology
March 2001

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

Document name
MASTER THESIS

Date of issue
March 2001

Document Number
TFRT LUTFD2/TFRT--5673--SE

Author(s)
Lars Wallén

Supervisor
Anders Rantzer, LTH
Jens Kalkkuhl, Ola Nockhammar, Daimler Chrysler

Sponsoring organization

Title and subtitle
Dynamic Tyre Models in Adaptive Slip Control
(Dynamiska modeller av däck för adaptiv slip-reglering)

Abstract

When designing an automatic brake-control system for a car it is important to have a mathematical model of the vehicle to base the design upon. Since careful modeling of the entyre car would encompass far too many variables, the most care is usually taken to the wheels and the tyres, as those parts contain most of the dynamics influential to the process.

During braking the tyres deform because of the forces between road and tyres. Because of this deformation the braking force from the brake discs is not immediately transmitted to road surface, resulting in a delay of the decelerating force. Tyre models mainly aims at describing the friction force between tyre and road surface, and most does this with a function where the friction force is mapped as a static function of the relative velocity between vehicle and tyre. Such a static model considers the tyre to be so sti_ that it does not deform at all during braking, and does not take deformation into account when calculating the friction force.

In this report it is investigated how influential the dynamics from the tyre deformation are. By using measurement data from a real car a model for the dynamics is made. This model is made so that it can easily be added to already existing models of tyre and car. Then by the use of this model a controller is designed which takes the tyre dynamics into account. It is implemented in simulink and compared to another controller that has been designed based on the non-dynamic tyre model.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
48

Recipient's notes

Security classification

The report may be ordered from the Department of Automatic Control or borrowed through:
University Library 2, Box 3, SE-221 00 Lund, Sweden
Fax +46 46 222 44 22 E-mail ub2@ub2.se

Contents

1 Introduction

ABS (acronym for Antilock Brake Systems) is a feedback control system for brake systems that aims at retaining the steerability of the vehicle as well as maximizing the braking force, thus decelerating the vehicle as much as possible while keeping the driver in control. The first designs where made as early as in 1947, and consisted of sensors that by detection of a wheel lock opened a valve to reduce brake pressure, a purely mechanical system.

Today the option to use electromechanical instead of hydraulic brake actuators introduces new possibilities to control the braking process. Use of such so called brake-by-wire systems are gaining widespread use, as they enable more accurate control of the braking forces and have significantly less time-delay than their hydromechanical counterparts. Also much more mathematically complicated controllers can be implemented when sufficient computing capacity is available, opening up possibilities of using complex estimators and control algorithms. Thus can the nonlinearities and uncertainties of the process be estimated and controlled with advanced systems, for instance by using nonlinear or adaptive techniques.

When developing these new control systems good models of the vehicle and especially the wheels are essential. Much work has already been made in this area, and several models of varying complexity exists and are in use. Many models used today in work with ABS-systems uses a stationary model for the tire. This means that at every instant the friction force is determined from the current measurements, without using knowledge about the past values. This is based upon a tire model where the tire has been modeled as so rigid that it doesn't deform at all during braking, an assumption that obviously is not true. Studies of tires have revealed that the tire does indeed deform when subjected to forces due to friction from the road during braking, and that this deformation has an impact on the development of forces between tire and road surface.

This work will focus on how a static tiremodel can be modified to take this deformation into account. In literature a multitude of different approaches and models are proposed, of whom some will be chosen to be examined. The gain in accuracy will then be weighted against the increase in complexity to see what extension that actually returns a better model for control purposes. Here other studies must be taken into account, as it is important that an extension works well together with models used in prior work. Then by using the new, dynamic model of the tire-behaviour, a controller is designed. To see if it can do an acceptable work in controlling the new tire model it is tested on simulated tire models in simulink. An already existing controller that is based on a static model is also implemented and used as a reference.

1.1 Structure of the report

The first half of the report is dedicated to the modeling of the car and the choosing of an appropriate model for the rest of the work; in sections 2 and 3 a mathematical model with emphasis on braking is designed, and different models of tire/road interaction are explained. In section 4 the models are examined to see which one that yields the most accurate estimate of the friction force when data from a real test-vehicle is used.

In the second half different approaches to control of the new dynamic model are examined; in section 5 mathematical design tools needed later are described briefly. Then different control designs are made in section 6 and finally simulated and compared to each other with simulink simulations of the vehicle in section 7. In the last section conclusions are drawn regarding the results and of what use they are.

2 Modeling of the car

To be able to design a functional controller for the braking process a mathematical model of the longitudinal car- and wheel dynamics has to be derived. This model can also later be used to simulate and test the controller.

If every component of the car would be modeled carefully, the order of the system and the number of parameters would be immense. Not only is such a model complex to construct, it is also computationally very demanding, and in a real-time system such as a car, computational capacity is limited. But it must not be made too simple, as it then would not be able to describe the process well enough to design an efficient controller. Thus the problem is to know what simplifications that can be done without affecting precision too much, and to design the model so that it is otherwise robust when measurement noise is added. It must also be robust enough to handle parameter variations, for instance tire wear, car loading or variations in road surface.

2.1 The complete car

Since the most important aspect of the model is to estimate the friction forces between road and tires, the interaction between the different parts of the car body are of minor importance, and thus not modeled. This means that the car is simplified to consist of a body, modeled as one single stiff piece, and four wheels that are modeled more carefully. The contact between wheels and body is considered stiff[1], i.e. the suspension is neglected.

During braking the brake actuators create a brake torque on each one of the wheels. This slows down the rotation of the wheels, which in turn creates a force between road and tire surface. For a wheel with a deviation angle α, as seen in figure 1, the friction force is divided into one vector along the traveling direction of the wheel (*longitudinal force* v_x) and one perpendicular to it (*latitudinal force* v_y).

2.2 Longitudinal model

If the modeled vehicle is traveling straight ahead with no steering input, the latitudinal forces and velocities are equal to zero. If care is taken when conducting experiments, so that the steering angle of the car is zero, this can be used to get a simpler model. A model where the car cannot turn is clearly non-realistic, but it is good enough for estimating longitudinal friction forces. This means that unless stated otherwise, the following is true for the rest of the report:

$$\begin{aligned} \alpha &= 0 \\ v_x &= v \\ a_x &= a \end{aligned}$$

With no latitudinal forces the car is 2-dimensional. The forces acting on the car are as shown in figure 2. This model can be used to deduce two things about the car:

[1] i.e. without dynamics

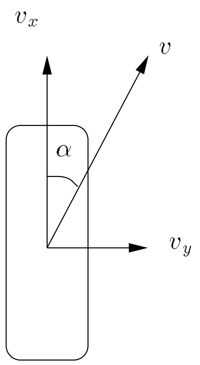

Figure 1: Single wheel, viewed from above. The vehicle travels along v, the wheel is turned an angle α.

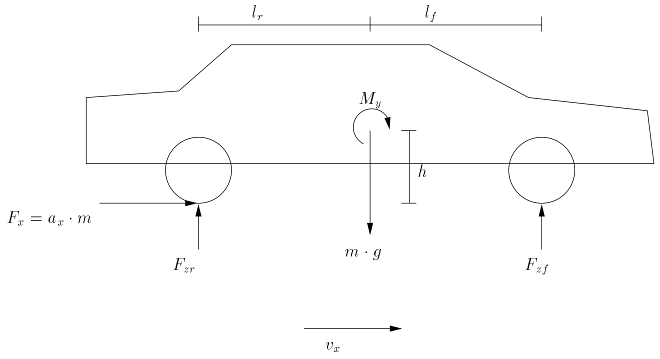

Figure 2: Forces acting on the longitudinal model of the car. Here F_x is the sum of forces acting between tires and road.

- The rate of acceleration as

$$a_x = \frac{F_x}{m} \tag{1}$$

- The variations in normal forces F_{zr} and F_{zf} as a function of the acceleration of the car.

In equation 1 the force F_x is the sum of accelerating force from the engine and braking force from the from the brakes. The sign is negative if braking force is greater than accelerating force. m is the total mass of the car.

The most simple model of the normal force is made by simply determining how much of the vehicles weight that is resting on each wheel (giving an expression independent of a_x). The normal force can be more accurately described by making and solving the equations of force and torque in two dimensions (x and z). The model then accounts for the change in F_z due to acceleration and braking. Note that suspension dynamics are ignored, with the motivation that

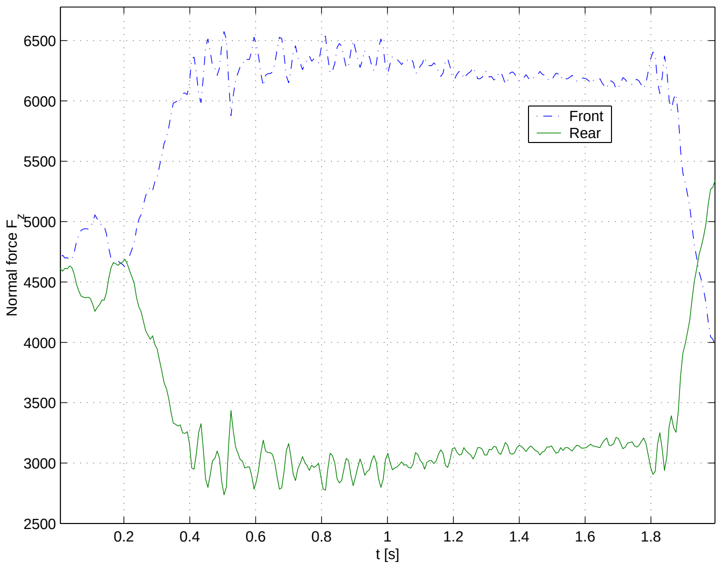

Figure 3: Normal forces when the model described by equations 4 and 5 is used. In this test the car begun braking at a speed of 50 kph, and was stopped at $t = 1.8s$.

the time-constants of the accelerations are considerably larger than those of the dynamics of the springs (see [8]). Equations are:

$$\sum F_z = 0 \Longrightarrow F_{zf} + F_{zr} - \frac{m_{car}}{2} \cdot g = 0 \tag{2}$$

$$\sum M_y = 0 \Longrightarrow F_{zf} \cdot l_f - F_{zr} \cdot l_r - \frac{m_{car}}{2} \cdot a_x \cdot h = 0 \tag{3}$$

Where forces and torques are as depicted in figure 2. The system has two equations and two unknowns (F_{zf} (front) and F_{zr} (rear)) and can thus be solved. Since only one side of the car is considered, the mass used is $\frac{m_{car}}{2}$. Solution is:

$$F_{zf} = \frac{m_{car}}{2} \cdot \frac{gl_r - a_x l_h}{l_r + l_f} \tag{4}$$

$$F_{zr} = \frac{m_{car}}{2} \cdot \frac{gl_f + a_x l_h}{l_r + l_f} \tag{5}$$

Figure 3 shows a plot of a typical behaviour of the normal forces F_z during a test where this model has been applied to data of acceleration by a car during decisive braking. The change in normal forces are considerable, and should not be ignored when designing the control. This model gives a slight exaggeration of the change in F_z, as part of the deceleration is a result of air drag and doesn't induce any torque.

2.3 Quarter-car

The next step is to model the wheels. Every wheel is modeled separately attached to a mass representing the part of the car 'resting' on that wheel. As the quarter car moves forward the force ${F_x}^2$ is generated between tire and road. This force creates a torque that in turn creates an angular acceleration $\dot{\omega}$. When the brakes are used they generate a braking torque T_b, causing an angular deceleration to stop the wheel's rotation. With the coordinates as shown in figure 4 this results in the equations of motion for the two-dimensional quarter car as:

$$\begin{aligned} m\dot{v} &= -F_x && (6) \\ J\dot{\omega} &= rF_x - T_b \cdot \mathrm{sign}(\omega) && (7) \end{aligned}$$

where the variables are:

m	mass of quarter car
v	speed of car
F_x	force acting between road and tire
J	inertia of wheel
ω	wheel angular speed
r	radius of wheel
T_b	braking torque

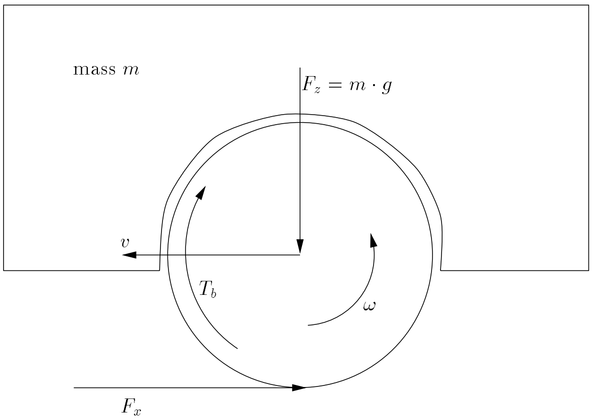

Figure 4: The quarter car

The behaviour of all the variables except the force F_x is linear and easy to describe with traditional mechanics. F_x , however, is highly nonlinear and difficult to model.

2.4 Unmodeled properties

The model has not taken every known property of a car into account. Some of these simplifications have been made because they would increase complexity far more than they would increase accuracy, some have been made because there has been no time to thoroughly investigate their influence on the model. One of those properties that are suspected to influence the process significantly is the suspension.

[2] Force F_x is similar, but not identical to a friction force. More on this in chapter 3

2.4.1 Suspension

When considering suspension dynamics the quarter-car model is no longer so simple. Considering vertical dynamics only the model including suspension consists of a wheel[3] attached to the quarter-car with springs and dampers in between, and a spring-and-damper equivalent between the tire and the road surface, depicting the tire's radial deformation (the tire's radius varies due to variation in F_z). These will together act as a second-order dynamic system on the vertical force F_z. Suspension modeling will grow even more complicated when longitudinal and inter-wheel suspension are considered.

[3] no longer without mass

3 Modeling tire–road interaction force

The modeling of the tire friction force F_x plays an important role in determining the vehicle behaviour. It has shown to be notorious difficult to model, and several models with different properties are in use. The term friction may be misleading, as the force not only is dependent on friction alone, but also on other physical processes such as deformation. Nonetheless, the term friction is used, and a friction-like definition is used to name the effective force quotient

$$\mu = \frac{F_x}{F_z} \tag{8}$$

In all models, the normal force F_z is considered known (estimated or measured, see section 2.2), and the friction force can be calculated when μ has been modeled. In most models, μ is a function of the following:

- Normal force F_z
- Road and tire materials and their interaction
- Speed v of vehicle and rim velocity ωr of the wheel

Where the normal force is of minor importance.

In the most basic models the tire is modeled as if it didn't deform at all during braking. In those models, called *static*, the effective force quotient μ is causal, i.e. dependent of present values of F_z, v and ω only.

When the wheel is modeled more carefully, the deformation can no longer be neglected. As the tire is made of (soft) rubber, the force acting on the tire from the road surface deforms the tire, which then in a sense acts a spring. Physically this adds an extra state of tire deformation to the process, and μ can no longer be determined from present measured values only. By modeling or estimating the state of tire deformation the actual friction force F_x can be calculated from the values of F_z, v and ω ([9]).

3.1 Wheel slip

The basic assumption in most friction models is that the quotient μ primarily is a function of the so-called *slip* λ. The slip is the normalized difference in velocity between the vehicle and the perimeter of the tire carcass. When $v > \omega r$ it is defined as:

$$\lambda_x = \frac{v_x - \omega r}{v} \tag{9}$$

and for $v < \omega r$ it is:

$$\lambda_x = \frac{v_x - \omega r}{\omega r} \tag{10}$$

This ensures that the value of the slip is between -1 and 1. A slip value of zero means that the the wheel rolls freely and has full contact with the road, a value of $\lambda = 1$ means a complete lock with the wheel sliding on the road, and a value

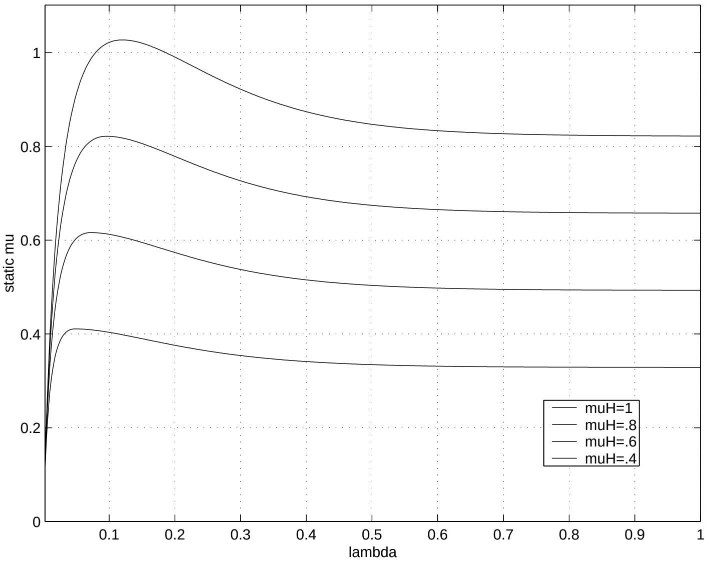

Figure 5: Plotted relationships between slip and friction quotient μ for different values of road friction parameter μ_H. Functions used in this case comes from a model using simple functions suitable for real-time control.

$\lambda = -1$ means that the wheel spins though the vehicle is stopped.

By conducting empirical experiments, where v and ω are constant while the measurement is done, the relationship between μ and λ can be plotted. All road surface and tire charasteristics are lumped together into a variable μ_H, that approximately takes the value of the maximal reachable μ. For low values of λ, where the slip mainly comes from deformation of the tire, μ increases approximately linearly. The rate of increase is mostly dependent of tire profile and the properties of the tire sides. At a value around $\lambda = 0.1$, μ reaches a peak at about μ_H, and after that converges towards an end value of μ_G. In this latter part the slip is mainly due to sliding.

The relationship between λ and μ is different for every combination of tire and road surface, an example is seen in figure 5, more can be found in [4, 7]. Note that the use of slip in any controller puts a lower bound for the velocities it can be used for, as either v or ωr are used in the denominator.

3.2 Generic static models

The most basic models based on the wheel slip use mathematical formulas to describe the relationship between μ and λ. These are chosen to match the results from empirical experiments, and have properties selected to fit their intended use. For instance must models that are to be used in real-time environments be compact and computationally non-demanding, where models meant for use in computer simulations should be more precise.

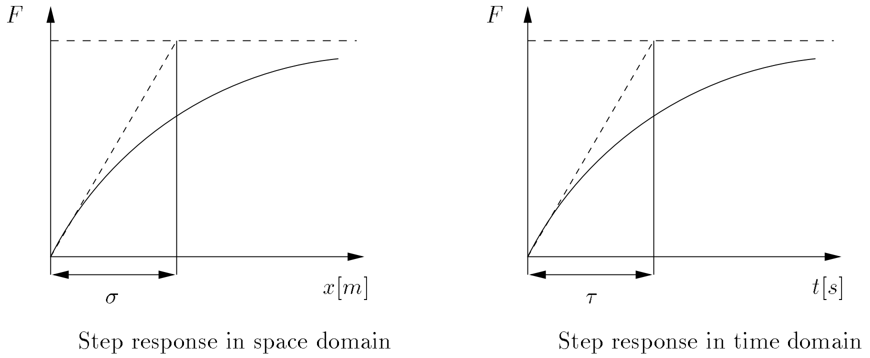

Figure 6: Behaviour of the force dynamics

If no care is taken to dynamic effects, that is if for every measurement of μ and λ the system is allowed to reach a static state with constant[4] v and ω, the friction can be modeled as a static map of μ:

$$\bar{F_x} = \mu(\lambda, \mu_H, F_z) \tag{11}$$

The bar over $\bar{F_x}$ is used to signify a static F_x.

3.2.1 Dynamic properties

As mentioned earlier, the static models above will lose their accuracy as the slip starts to change with time. This is due to that the pneumatic rubber tire is not rigid, and that the tire deformation doesn't instantly follow the variations in slip. When a brake-torque is applied by the brake actuators the rim of the tire will not instantly be affected (and thus the friction force will not instantly increase) meaning that the rim will start to deflect relative the wheel axis. This means that the force F_x will be a function of time as it is dependent of the deflection. The distance the wheel must travel to build up the deflection necessary to transmit two thirds of the force is called *relaxation length*[5] and is a tire charasteristic. The relaxation length in itself is not dependent on velocity, but as the vehicle travels faster it will require less time to cover this length, which means that the time constant in time-domain is dependent on v. A simple model often used (see for example [5] p. 79) is the first-order differential equation (12), with plotted behaviour in figure 6

$$\begin{aligned} \tau\dot{F_x} &= \bar{F_x} - F_x && (12)\\ \tau &= \frac{\sigma}{v_x} && (13)\\ \sigma &= \text{relaxationlength} \end{aligned}$$

These dynamics can then be added to the static tire/road interaction model to produce a dynamic model. In a state-space system it will add an extra state

[4] This can of course not be done on a real vehichle, as non-zero slip means non-zero friction and thus unavoidable deceleration.

[5] In German *Einlauflänge*

for the actual force between tire and road surface, increasing the order of the system by one.
When an even more sophisticated model is needed this approach will prove inadequate, as it is good only for lower frequencies. At higher frequencies (and velocities) the inertia of the tire become more important (see [6]). This will in the interest of simplicity not be used.

3.3 Other models

The above model is a simple way to model friction that only requires some kind of friction-slip relationship to work. It can be made very fast by the choice of function $\mu(\lambda)$. However, several other ways to model F_x exist.

3.3.1 Pacejka magic model

One of the most widely used models is the model due to Pacejka, sometimes called the 'magic formula'. It's functions describing the F_x relationship for longitudinal slip are

$$F_x = c_1 \sin(c_2(\arctan(c_3\lambda - c_4(c_3\lambda - arctan(c_3\lambda)))) \tag{14}$$

where the constants c_i are chosen to match experimental data, and have no special physical meaning. For real-time car-control it is somewhat too complicated, as it has several parameters that must be estimated in order to calculate road/tire interaction variation.
It can be extended to model F_x dynamically by modeling the deformation of the tire, introducing a dynamic slip state. Detailed description of the model is found in [9].

3.3.2 LuGre model

An other model describing the tire/road interaction is the LuGre-model. Compared to the other models, it has a different approach as it is based on a physical model, and not only on empirical experiments. It is based on a model where the contact area is modeled as a large number of separate contact points, each acting as a spring. Friction is then modeled as the average deflection force of those elastic springs, and the contact points will start to slip when deflection is large enough. The model is mathematically described by:

$$\dot{z} = v_r - \sigma_0 \frac{|v_r|}{g(v_r)} z \tag{15}$$

$$F_x = -(\sigma_0 z + \sigma_1 \dot{z} + \sigma_2 v_r) F_z \tag{16}$$

$$g(v) = \mu_C + (\mu_S - \mu_C) e^{-|v_r/v_s|^{1/2}} \tag{17}$$

where some freedom exists in the choice of $g(v)$

Variables used are:

σ_i	Tire rubber properties
v_r	relative speed, $= (r\omega - v) = -\lambda v$
v_s	Stribeck relative velocity
μ_C, μ_S	Normalized Coulomb and static friction values

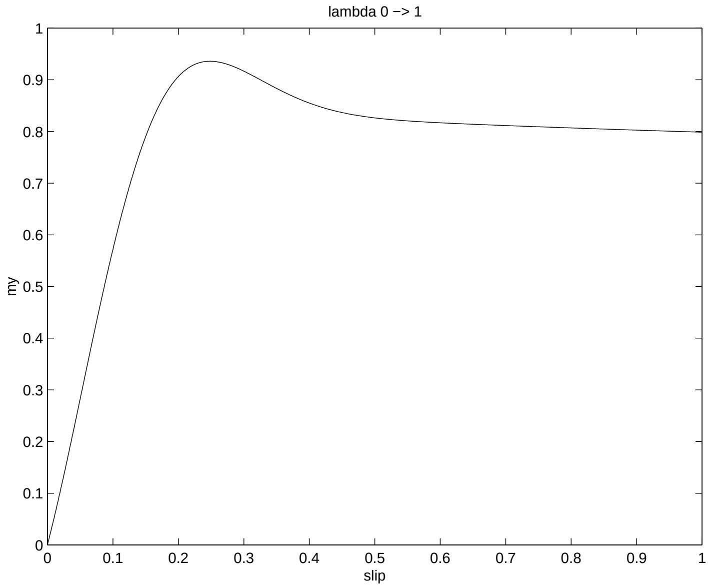

Figure 7: Curve $\mu(\lambda)$ for the Lugre friction model. Shape of curve is very similar to the corresponding curves for static empirical models. However, no static state exists, making curve dependent of $\dot{\lambda}$

The model models friction dynamically, but due to it's rather non-intuitive state z that models an average of virtual spring deflection it is difficult to relate the model to the other models. However, when plotting slip force coefficient μ versus slip λ a curve very similar to those of other models are obtained (see figure 7), indicating similar behaviour if implemented. The LuGre slip curve is however dependent on the rate of which λ increases, indicating existence of dynamics that not present in the other models.

3.3.3 More advanced models

For cases where the frequency of the disturbance on the vehicle is high, for instance when it rolls over short wavelength unevennesses, the models above are not sufficient. For frequencies of up to 80 Hz the free tire belt is almost rigid, and the above models can be used. But for higher frequencies the tire belt starts to deform, and a flexible-ring model should be used. These models are however computationally very demanding, and are thus not suitable for a real-time control system.

3.4 Slip-curve uncertainty

To be able to model the friction with the models described in section 3.2 or 3.3.1 the $\mu(\lambda)$–function must be known. The maximum value of the curve, the static friction constant μ_H, can be evaluated by some kind of estimator and

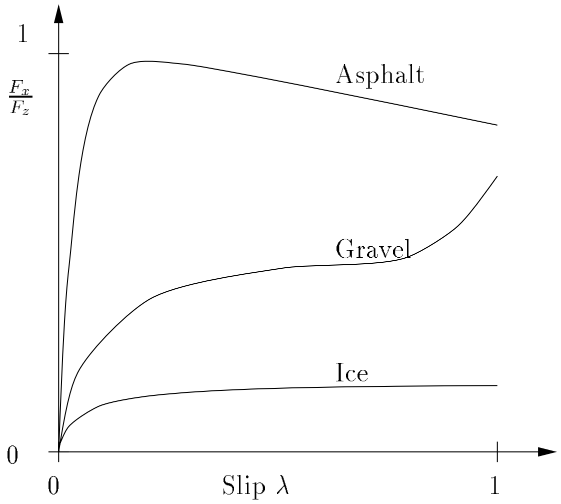

Figure 8: Example of how the shape of the $\mu(\lambda)$–function can vary depending on the road surface.

poses no unsolvable problem. But as shown in figure 8 the entire shape of the curve is also dependent on the road surface. The charasteristic with a distinct maximum in friction force is much less distinct when the surface is icy or winter tires are used, and for deformable surface materials such as gravel or snow the maximum is reached at a slip $\lambda = 1$ instead of a λ around 0.1.

And even if the surface and tire properties would be known exactly, one curve would still not do to model precisely, as there is also a dependence on velocity (see for instance [4] p. 19-21). This does not only affect the magnitude of μ_H, but also the entire shape of the curve, see figure 9.

These uncertainities poses problems when it comes to designing controllers for the system. Attempts have been made to use estimators to identify the surface properties on-line, but the approach to be used in this report is to design the controller with enough robustness to handle the $\mu(\lambda)$ function uncertainties.

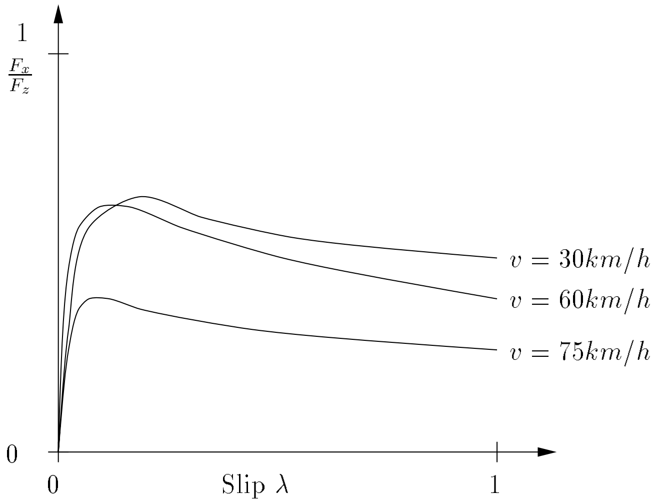

Figure 9: A certain combination of tire and road surface has not only one, but a complete family of curves, each one for a different velocity. Example of curves on wet concrete

4 Evaluation of the models

To test how well the models derived in the last sections describe reality, and what model that is most useful for car–control purposes, tests are made with a real test vehicle. These tests will also be used to find an approximate value for the tire relaxation length constant σ.

Instead of conducting new tests data is used from tests performed earlier. These have been made to evaluate the performance of different brake-controllers, and thus have all data relevant to tire– and carmodeling logged and available.

4.1 Test properties

As the tests have already taken place there is no influence over the design of the experiments. However, there are several different tests to choose from. In those that are chosen to be used for testing the model the vehicle only performs braking along a straight line, with wheel angle $\alpha = 0$ and lateral acceleration at a minimum. This is to avoid side forces and velocities, so that the longitudinal model described in section 2 applies. Additionally, only data from the front wheels are used. This is because the vehicle is driven on the rear wheels, and thus the moments of inertia J of those wheels are dependent on the engine. There is also the risk of an accelerating torque.

4.1.1 Properties of the measured data

All the logged data are sampled with a two-sample time–delay. Since this applies to all variables, it does not affect the system as there is no feed–back for these model evaluation tests. Also, some of the variables needed cannot be measured directly as the number of sensors is limited. To amend this, a Kalman filter that supplies estimates of the needed variables is used. It supplies those in real-time, and works independently from other control and estimation programs. In those tests where the controller have been turned off the estimates from the Kalman filter have not been logged, and instead the measured data is re-run through simulink where the Kalmanfilter is implemented as a C-function.

In the tests the friction force F_x is not available for direct measuring. To get a value equation 7 is used to calculate a reference value F_{meas} ('measured' value) as:

$$F_{meas} = \frac{J\dot{\omega} + T_b}{r} \tag{18}$$

Where T_b and ω are measured variables and r and J are parameters of the tire known beforehand.
The slip λ is calculated by using the definition in section 3.1:

$$\lambda = \frac{v - \omega r}{v}$$

This calculation is numerically sensitive as it contains a subtraction of two numbers with similar magnitude. Since the measured signals of ω is rather noisy, it leads to a relatively high level of uncertainty.

4.1.2 Filtering

As mentioned earlier the calculation of the slip is very sensitive to noise. Thus it may be tempting to use filtering to reduce the noise levels. However, the dynamics investigated are in some cases very fast, and filtering may remove or change those dynamics that are investigated.

For those cases where the noise level is too high a two-sided filtering is done before the data is imported to the simulink model. Such a filtering does not delay the signals, but care should be taken as it still might introduce foreign dynamics to the process.

4.2 Creating the $\mu(\lambda)$–curve

The generic static tire model described in section 3.2 says that for a specific combination of road surface and tire there exists a function $\mu(\lambda)$ describing the relationship between friction and slip. Measured data from tests is used to try to construct this function. With the normal force F_z calculated according to equation (4), the tire/road force quotient μ is:

$$\mu = \frac{F_{meas}}{F_z} \tag{19}$$

It is plotted on the y–axis against λ on the x–axis.

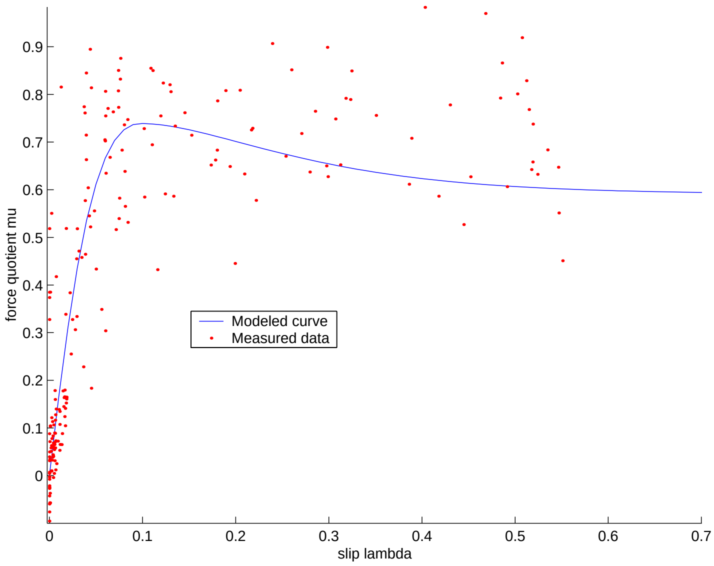

Figure 10: Plot of μ versus λ. Large noise levels may make filtering tempting, but as dynamics are fast that cannot de done without introducing error.

The result as seen in figure 10 is by no way the clear curve that modeling with a single mathematical function supposes. Measurement noise is not the only reason to the uncertainties, as the dependence on velocity described in section 3.4 makes it impossible to attain a clear curve, as the speed (naturally) cannot be kept constant during braking. To get the perfect curves (for example as seen in [9]), special equipment must be used where the data for the curve can be measured with a constant velocity v.

However, what can be seen that μ initially rises rapidly for small increases in λ, and reaches a maximum for a λ between 0.1 and 0.2. This will prove to be very useful when a controller for maximizing braking force is designed.
The plotting of the $\mu(\lambda)$–curve is made with the static value of force F_x. The dynamic addition described in section 3.2.1 can not be tested because of the noise; With the dynamic model the measured F_{meas} would be the dynamic value, and to plot the curve the static value $\bar{F}_x$ would be needed. To solve equation 12 would yield

$$\bar{F}_x = \tau\dot{F}_{meas} + F_{meas} \tag{20}$$

where the derivation would add far too much uncertainty.

4.3 Real value and model comparison

Another way to test a model is to see how well it can predict the friction force F_x when only receiving the measured data of ω and a_x along with an estimated value of speed from the Kalman filter. This is done with a simulink model, where data from the test is imported and processed.

4.3.1 Static model

A simulink model is made where the force F_x is calculated in two ways. The first is the force calculated directly from the measurements, F_{meas}, the second is an estimation of the force calculated by using the static model:

$$F_{stat} = \bar{F}_x(\lambda(t), \mu_H, F_z) \tag{21}$$

where the function $\bar{F}_x$ exists as a simulink 'black box' that takes the arguments of the function as insignals. μ_H is considered known beforehand (The static model does by itself not include parameter estimation). The function used is plotted in figure 5.

As can be seen in plots 11 and 12 the static model can give a rather good estimation of the friction force F_x. However, on two points are the results of the model poor:

- The estimated force F_{stat} respond to variations in force earlier than the real value F_{meas}

- The ratio between noise and measured slip is high. With the static model the friction force is nearly linearly dependent on the slip for low values of λ, which means that also the force estimation will be noisy.

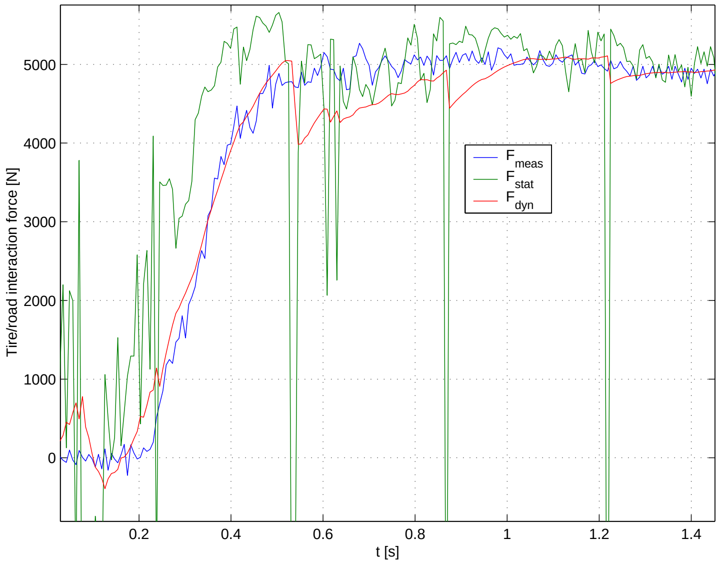

Figure 11: Tire forces during hard braking, where braking is at a maximum until the vehicle is stopped. Test performed on road with high friction, no filtering have been applied. F_{meas} is the measured braking force, F_{stat} is the braking force as calculated by the static model, and F_{dyn} is the modeled force when the dynamic model is used for the calculations.

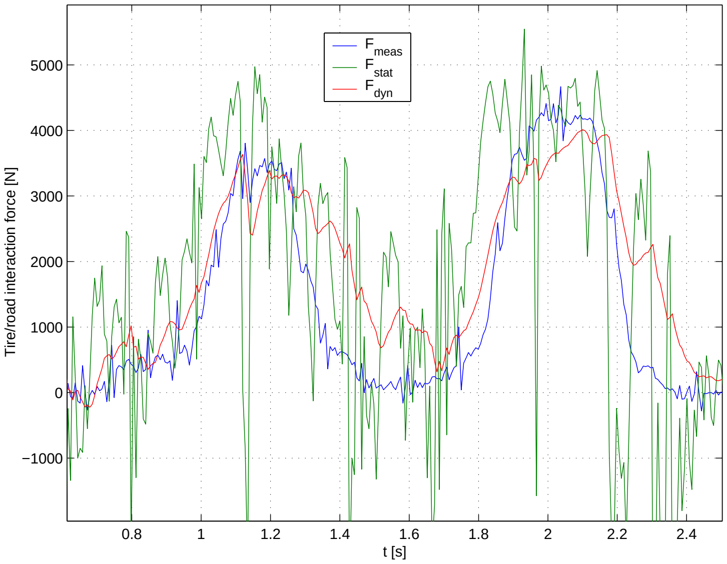

Figure 12: Brake signal in the form of pulses show that estimation is poor when braking force subsides.

4.3.2 Dynamic model

To test if the dynamic model can be used to improve the results the theory from section 3.2.1 is used. To do this the first–order dynamics is represented by a first–order filter acting on the static estimate F_{stat}. The timeconstant used is $\frac{v}{\sigma}$, where the value of v used comes from the Kalman filter. The acceleration of the car is low enough for the filter to be implemented as if it were time-invariant.

As can be seen in figures 11 and 12, adding the dynamic model solves both problems that existed with the static model. The filter delays the estimated value enough to make a good match of the measured value, and at the same time reduces the amount of noise in the signal, without introducing any dynamics more than those relating to the tire deformation.

According to the proposed dynamic model the tire deformation is modeled by first–order dynamics. To test whether this is true, and if not then of what order the process really is, the results are evaluated mathematically. This is done by applying ARX-models of different order, where F_{stat} is used as input and F_{meas} as output. These evaluations also serve the purpose of finding a value for the relaxation length for the tire used in the experiments. As the first–order dynamics also dependend upon velocity, tests where braking is done at several different velocities are used in the computation. (Those tests are plotted in figure 13)
The results show that the dynamics are best described by a first–order system, and the relaxation length for the tires used in the experiments were found to be $\underline{0.68}(m)$.

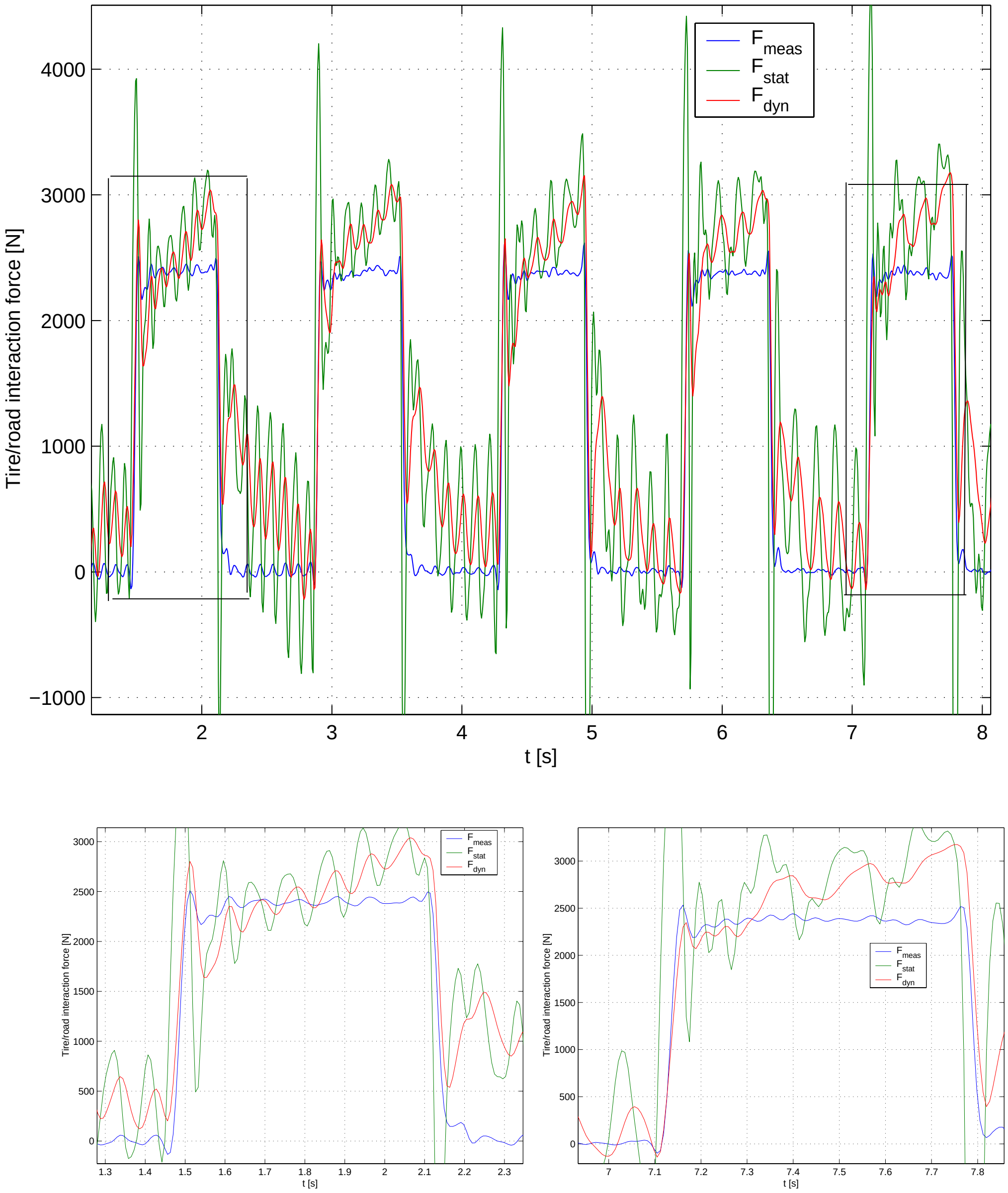

Figure 13: Test to show velocity dependence of relaxation length. The two lower plots are blow-ups of the marked areas on the upper plot. Wished braking force a square signal, first braking initiated at $v = 90$kph, last at $v = 15$kph. Dynamic model based on section 3.2.1 show good correspondence both for high and low velocities. The short-wavelength oscillations are probably due to suspension dynamics

4.3.3 LuGre model

Also the LuGre model is implemented and tested to see whether it can predict the friction force accurately. The procedure is the same as for the static and dynamic models; a LuGre model is implemented in simulink and fed with data of ω and v from the tests. The result is compared with a force F_{meas} calculated as earlier.

As seen in figure 14 the estimation is of acceptable accuracy for an increase in braking force, but when the braking force decreases the estimation does not follow the real force. Noise level is kept low without the use of extra filtering.

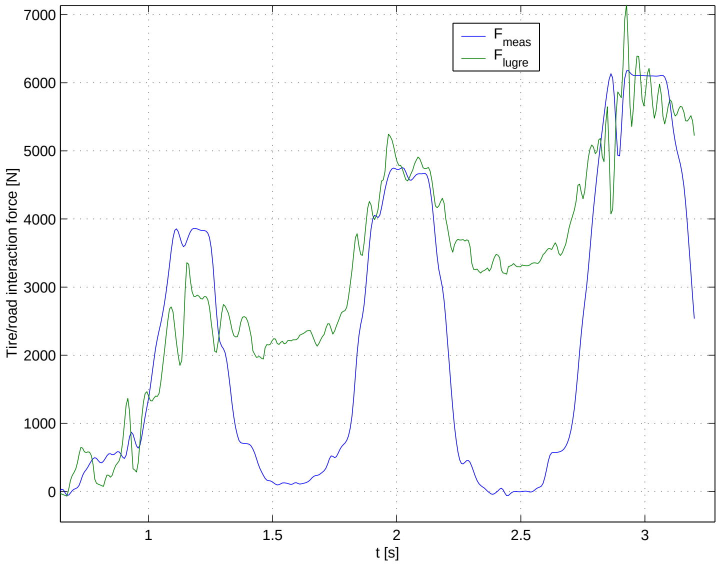

Figure 14: Lugre model applied on same data as used in figure 12. Model estimation is acceptable for the higher values of braking force, but is unable to estimate decrease.

4.4 Conclusions

Attempting to re-create the $\mu(\lambda)$–function from a single test-run is very hard to do, as the curve is dependent on velocity. And as the tests are ones of braking, the speed naturally decreases during the test. However, what can be seen is that the friction force reaches a maximum at a slip between 0.1 and 0.2.

The attempts to model the tire/road interaction force gave better results. It can be seen quite clearly that the correspondence from the static model is improved when the dynamics from the tire deformation are added. Both the dynamic and the LuGre models show poor results when the braking force is reduced, indicating some sort of dynamics that are not covered by the model. For the dynamic model the results are however acceptable, and when all factors considered the estimation is overall far better than that of the static version.

Those tests have been made to investigate which model that is best suited for modeling the friction. A static model where the friction force F_x is modeled as a function of slip is chosen together with the dynamic part, where the choice is based on:

- Less noise than the static version
- In all better fit than the static version
- Better fit than Lugre model, especially for case of decreasing brake force
- Using a model based on the static model allows for prior research in the area to be used

5 Introduction to Lyapunov backstepping

To find a controller that both can provide an estimate of μ_H and can handle the nonlinearities from the tire models, methods based on Lyapunov theory will be implemented. These methods will be described together with a short description of the mathematical theory that they are based upon to ease the understanding of the methods. For complete Lyapunov theory together with proofs and theorems see [1, 2].

5.1 Lyapunov stability

What Lyapunov theory can do is essentially to prove stability and/or convergence for a solution in an area D to the dynamic system:

$$\begin{aligned} \dot{x} &= f(x) \qquad (22) \\ f(0) &= 0 \end{aligned}$$

This is accomplished by finding a *Lyapunov function* $V(x)$ which is a function with the following properties:

- $V(x) > 0$ for $x \subseteq D$, $V(0) = 0$
- $\dot{V}(x) \leq 0$ for $x \subseteq D$

The theory then says that $x = 0$ is a stable solution. More so, it says that if $\dot{V}(x) < 0$ in D then for all starting values of $x \subseteq D$ the solution will in time approach 0. The idea is very simple, and by determining the magnitude of $\dot{V}(x)$ the convergence rate can be estimated. A problem is that there is no systematic way to find the Lyapunov function, although there are several widely used methods for specific cases. Usually choosing quadratic terms are a good start, as it automatically means that $V(x) > 0$.

Example To make an example a relatively simple system is chosen:

$$\dot{x} = x^3 - 3x$$

By plotting it can easily be shown that the system is stable for starting values $|x| < 1$. To prove it with Lyapunov theory a candidate function has to be chosen. Quadratic terms are usually a good start, the choice and the following calculations can for instance look like:

$$\begin{aligned} V(x) &= \frac{1}{2}x^2 > 0 \\ \dot{V}(x) &= x \cdot \dot{x} = \\ & x^4 - 3x^2 = x^2(x^2 - 3) \end{aligned}$$

$x^2(x^2 - 3) < 0$ when $|x| < \sqrt{3}$, thus stability and convergence to 0 is proven in this area.

An easier way of explaining the idea of Lyapunov stability can be made with an example from physics: Any system whose total energy never increases can never be unstable. If it's energy dissipates with time it will finally reach a stable state of zero energy, and thus be asymptotically stable.

5.2 Lyapunov-based controller

For systems containing unknown parameters the Lyapunov theory can be applied to construct rather elegant controllers that can estimate the unknown variables. Such controllers can also be used to regulate a certain class of nonlinear systems as long as the nonlinearity is known.
Basically the idea is to find a Lyapunov function for the system. This will prove the system to be stable, and if it's derivative is negative definite it will also prove asymptotic stability. By choosing the function so that it contains squares of the errors of the system, the errors will be proven stable. Then $\dot{V}$ governs the rate with which the errors converge to zero.

Example A controller with an estimator for a first-order system with an unknown parameter θ and a nonlinearity $\varphi(x)$ will be made. Parameter θ is considered to be constant.

$$\dot{x} \;=\; u + \theta\varphi(x)$$

In this example the state x is driven to zero by the controller, with a variable change $e = x - x_{ref}$ the calculations can as well be used for control to any x_{ref}. Constant θ gives:

$$\begin{aligned} \tilde{\theta} &= \theta - \hat{\theta} \\ \dot{\tilde{\theta}} &= \dot{\theta} - \dot{\hat{\theta}} = -\dot{\hat{\theta}} \end{aligned}$$

where $\hat{\theta}$ is estimate and $\tilde{\theta}$ is error. Direct approach by choosing Lyapunov function candidate as squares of variable and estimation error results in following calculations:

$$\begin{aligned} V &= \frac{1}{2}x^2 + \frac{1}{2\gamma}\tilde{\theta}^2 \\ \dot{V} &= x \cdot \dot{x} + \frac{1}{\gamma}\tilde{\theta}\dot{\tilde{\theta}} = x(u + \theta\varphi(x)) - \frac{1}{\gamma}\tilde{\theta}\dot{\hat{\theta}} \end{aligned}$$

by choosing control law for u as $u = -cx - \hat{\theta}\varphi(x)$, $\dot{V}$ will be:

$$\dot{V} \;=\; -cx^2 + \tilde{\theta}(x\varphi(x) - \frac{1}{\gamma}\dot{\hat{\theta}})$$

The second term is eliminated by choosing $\dot{\hat{\theta}} = \gamma x\varphi(x)$, which is the update law for estimate $\hat{\theta}$. This makes $\dot{V}$ negative semidefinite, which proves that the state x and the estimation error $\hat{\theta}$ are bounded. By choosing values for c and γ properties of the controller can be chosen.

5.3 Backstepping

Finding Lyapunov functions may be easy for simple systems of low order, but complexity increases rapidly with the order of the dynamic system. For systems where not all states can be measured, and where perhaps more than one nonlinearity is present, finding Lyapunov functions without guidelines can be almost impossible. One such systematic strategy called *backstepping* is thoroughly described in [3].

Backstepping is a recursive design method that designs a controller in several basic steps. Compared to the more standard concept of 'feedback linearization' it has the advantage that it can avoid cancelling of nonlinearities that are helpful for stabilization. It can also be used when there are unknown constants by using parameter estimation, introducing a dynamic part to the controller resulting in a nonlinear adaptive controller.
The key idea and also the starting point is to make a controller (by a Lyapunov function) for the first state by introducing a new state for the 'desired value' of higher states, in a way making a controller for a first-order system. This new state is called 'virtual control' and it's desired value $\alpha(x_1)$ a stabilizing function. $\alpha(x_1)$ is chosen so that the first state is virtually stable, but the only real effect from the first step is to deduce a $\alpha(x_1)$ that can be used in the next step. The difference between the higher states and $\alpha(x_1)$ is the corresponding error variable z_1. This step is then repeated until the last state, and the error variables z_i from the different steps are used to construct a Lyapunov function for the system.

Example

$$\begin{aligned}\dot{x}_1 &= x_2 + \varphi(x_1)\\ \dot{x}_2 &= u\end{aligned}$$

u is insignal to process and φ is a known and differentiable function. In the first step the most basic clf is tried, and state x_2 is replaced by it's virtual estimate and error:

$$\begin{aligned}V_1 &= \frac{1}{2}x_1^2\\ \dot{V}_1 &= x_1\dot{x}_1 = x_1(x_2+\varphi(x_1)) = x_1(\alpha(x_1)+z+\varphi(x_1))\end{aligned}$$

Stabilizing function α is chosen to be as close to desired state as possible. Only error z left.

$$\begin{aligned}\alpha(x_1) &= -c_1x_1 - \varphi(x_1) \Rightarrow\\ \dot{V}_1 &= -c_1x_1^2 + x_1z\end{aligned}$$

A new function is tried. The old one is taken and augmented with a quadratic term for error z:

$$\begin{aligned}V_2 &= \frac{1}{2}x_1^2 + \frac{1}{2}z^2\\ \dot{V}_2 &= \dot{V}_1 + z\dot{z} = -c_1x_1^2 + z(x_1 + u + c_1\dot{x}_1 + \frac{\partial\varphi(x_1)}{\partial x_1}\dot{x}_1)\end{aligned}$$

By choosing an appropriate control signal u the second term can be made negative definite, making V_2 a Lyapunov function.

$$\begin{aligned}u &= -x_1 - c_1\dot{x}_1 - \frac{\partial\varphi(x_1)}{\partial x_1}\dot{x}_1 - c_2z \Rightarrow \qquad (23)\\ \dot{V}_2 &= -c_1x_1^2 - c_2z^2\end{aligned}$$

For example on an adaptive backstepping design see for instance [1] p.250 or [3] p.92.

6 Control design for the dynamic system

Prior controllers for the ABS-system have been based on the static, less complicated model of the tire. One that has been implemented and that has shown good results is a controller based upon Lyapunov theory decribed in section 5.2. This controller will be explained so that it later can be used as a comparison to other designs. Then, by using the more complex and accurate dynamic model a new controller will be designed. Due to the systems complexity, the Lyapunov backstepping techniques from section 5.3 will have to be used.

6.1 Slip control

An effective approach to design brake–control systems is to control the slip λ to a certain value $\lambda = \lambda_0$. This is based on the assumption that the function for the brake force coefficient μ (equation 11), and thereby F_x, reaches a maximum for a certain value of longitudinal slip. It is also so that in this range of slip the lateral force is high enough to retain the steerability of the car. For most combinations of road and tire this maximum is somewhere in the range between 0.1 and 0.2. Examples of this can be seen in figure 5 and 10, even though the latter is noisy.

6.2 System dynamics

To get the system equations relating to the slip dynamics equation 9 is derivated. Then by using the equations of motion (equations 6 and 7) to get rid of ω, the system is written as:

$$\dot{\lambda} = \frac{1}{v}\left[\left(\frac{1-\lambda}{m} + \frac{r^2}{J}\right) F_x + \frac{r \cdot T_b}{J}\right] \tag{24}$$

$$\dot{v} = -\frac{F_x}{m} \tag{25}$$

Only the speed of λ–dynamics are dependent of the v–state, the qualitative properties are not affected. With the presumption that the speed dynamics are relatively slow, v can be omitted from the control, with only it's magnitude being used in the system as if it were static.

The system can be reduced further as for normal vehicles

$$\frac{1-\lambda}{m} << \frac{r^2}{J} \tag{26}$$

This only leads to a more compact model. Dynamics are basically the same.

6.2.1 Tire deformation dynamics

The dynamic model described in section 3.2.1 adds an extra state for friction force F_x to the system. With this extra state and the proposed simplification

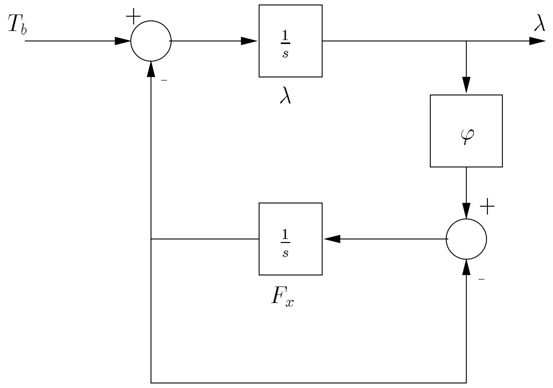

Figure 15: Block diagram describing state-relations. Constants have been omitted. φ signifies the nonlinearity.

26 the system to control is:

$$\begin{aligned}
\dot{\lambda} &= \frac{-r^2}{J\cdot v}\cdot F_x + \frac{r}{J\cdot v}\cdot T_b && (27)\\
\dot{F_x} &= \frac{v}{\sigma}\cdot \bar{F_x}(\lambda,\mu_H,F_z) - \frac{v}{\sigma}\cdot F_x && (28)\\
\dot{v} &= -\frac{F_x}{m} && (29)
\end{aligned}$$

With a block diagram depicting the relationship between states, inputs and output in figure 15.

The system that results from adding the state for tire longitudinal force is difficult to control. For time-invariant, linear systems there exists several methods, but this system has several properties that makes special methods necessary:

- Nonlinear function $F_x = \mu(\lambda, \mu_H, F_z)$ makes system nonlinear
- Road surface parameter μ_H is not only unknown, it can also change during braking
- Dynamics that have been removed to make system easier to model still exists. These will still act on the process, increasing need for robustness. Some examples are:
  - Suspension
  - Normal force F_z is dependent on $\dot{v}_x$
  - Brake actuator cannot transmit braking force instantly
  - Measurement noise
- Input data is limited, as F_x cannot be measured.

Control can be designed either on this system, trying to recognize all characteristics, or on a simpler system.

6.3 Reference controller

The controller that will be used as a reference is based on equations 24 and 25 only, i.e. a model using the static tire model. It is an adaptive controller that is based upon the Lyapunov function:

$$V = \frac{1}{2}v^2 \cdot (\lambda_0 - \lambda)^2 + \frac{1}{2}\frac{\theta}{\gamma}(\frac{\mu(\lambda_0)}{\theta} - \eta)^2 \tag{30}$$

where η estimates $\frac{\mu(\lambda)}{\theta}$(indirectly this estimates the unknown friction parameter μ_H) with the adaptation gain γ.
Function V can be made a Lyapunov function with the update law:

$$\dot{\eta} = -\gamma v \frac{r}{J}\sigma_0(\lambda - \lambda_0) \tag{31}$$

for the estimate η and the control law:

$$u = \sigma_0 \cdot [\eta - k_p(\lambda - \lambda_0)] \tag{32}$$

for control signal u, σ_0 is a constant dependent on the set point value λ_0. The parameters γ and k_p are functions dependent on speed and normal force.

The controller is thoroughly explained and analyzed in [7].

6.4 Lyapunov observer backstepping controller

A controller structure that can control the non-simplified system must be able to cope with:

- Nonlinearity
- Unknown constant requiring adaptation
- Non-measurable state F_x

Even if the brake actuators are fast, they still influence the process. Thus the controller should also be designed to handle

- Brake actuator dynamics

Especially the nonlinearity makes this difficult. The method chosen is a variant to the Lyapunov backstepping controller described in section 5, and as the system contains a non-measurable state the method called *observer backstepping* is used.

In this method is first an exponential observer constructed that supplies estimates of all the states. Then backstepping is used to deduce a controller for a new system, where the unmeasured states have been replaced by their estimates from the observer. All the time the observer-errors are treated as noise and is taken care of by nonlinear damping.
To use the method the control signal must be present in the same equation as

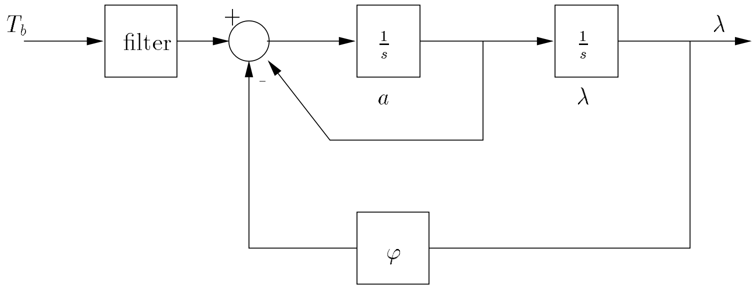

Figure 16: Block diagram describing state-relations after the variable change. The new system is suitable to control with backstepping techniques, as both the nonlinearity and the control affects the same state.

the nonlinearity, and the nonlinearity must be dependent of measurable states only. Therefore a variable change is made with:

$$a = \frac{-r^2}{J \cdot v} \cdot F_x + \frac{r}{J \cdot v} \cdot T_b \Rightarrow \tag{33}$$

$$F_x = \frac{T_b}{r} - \frac{J \cdot v}{r^2} \cdot a \tag{34}$$

$$\dot{F_x} = \frac{\dot{T_b}}{r} - \frac{J \cdot v}{r^2} \cdot \dot{a} \tag{35}$$

This system is depicted in figure 16. It must be further modified to handle the brake actuator dynamics. These are:

$$T_b = \frac{c_b}{s + c_b} \cdot u \tag{36}$$

where u is the control signal, and T_b is the applied braking force. T_b is also has upper and lower boundaries, but these will not be accounted for in the design. Solving for $\dot{T_b}$ yields

$$\dot{T_b} = c_b \cdot (u - T_b) \tag{37}$$

As the brake force can be measured it can be used in the control, and is considered a known quantity in the design, allowing equation 35 to be substituted for:

$$\dot{F_x} = \frac{c_b(u - T_b)}{r} - \frac{J \cdot v}{r^2} \cdot \dot{a} \tag{38}$$

resulting in the new system, where the dynamics of v have been ignored: (i.e. the acceleration have been considered to be zero)

$$\dot{\lambda} = a \tag{39}$$

$$\dot{a} = -\frac{r^2}{J \cdot \sigma} \bar{F}_x(\lambda, \mu_H, F_z) - \frac{v}{\sigma} \cdot a + T_b\left(\frac{r}{J\sigma} - \frac{c_b r}{Jv}\right) + \frac{c_b r}{Jv} \cdot \hat{u} \tag{40}$$

$\hat{u}$ is the control signal that is sent to the brake actuators. In the controller, an internal variable is used for the design:

$$u = T_b(\frac{r}{J\sigma} - \frac{c_b r}{Jv}) + \frac{c_b r}{Jv} \cdot \hat{u} \tag{41}$$

The system thus yields a control signal u, which is then passed through a function of T_b. The out-signal is called $\hat{u}$ and is:

$$\hat{u} = \frac{Jv}{c_b r} \cdot u - T_b(\frac{v}{c_b \sigma} - 1) \tag{42}$$

The last modification that has to be made is to move the unknown parameter μ_H out of the nonlinearity to stand as a constant before the function. This is a further modification to the system, but it doesn't affect the dynamics significantly as μ_H already behaves approximately as a constant, as seen in figure 5. Additionally, parameters with the nonlinearity are included in it:

$$\varphi(\lambda) = \frac{-r^2}{J \cdot \sigma} \cdot \bar{F}_x(\lambda, 1, F_z) \Rightarrow \tag{43}$$

$$\frac{-r^2}{J \cdot \sigma} \cdot \bar{F}_x(\lambda, \mu_H, F_z) = \mu_H \cdot \varphi(\lambda) \tag{44}$$

6.4.1 Design of the controller

Now the system is finally on a form suitable for backstepping. First an estimator for the system is designed. Two filters are used to reconstruct all states, one part for the μ_H-dependent part and one for the other.

$$\begin{aligned} \dot{\xi}_{01} &= k_1(\lambda - \xi_{01}) + \xi_{02} \\ \dot{\xi}_{02} &= k_2(\lambda - \xi_{01}) - \frac{v}{\sigma}\xi_{02} + u \\ \\ \dot{\xi}_{11} &= -k_1\xi_{11} + \xi_{12} \\ \dot{\xi}_{12} &= -k_2\xi_{11} - \frac{v}{\sigma}\xi_{12} + \varphi(\lambda) \end{aligned} \tag{45}$$

The states can then be reconstructed as:

$$\lambda = \xi_{01} + \mu_H \xi_{11} + \varepsilon_1 \tag{46}$$

$$a = \xi_{02} + \mu_H \xi_{12} + \varepsilon_2 \tag{47}$$

To ensure that the errors go to zero asymptotically the constants k_1 and k_2 must be chosen so that the eigenvalues of matrix A_0 are less than zero (according to theory for Kalman filters) A_0 is:

$$A_0 = \begin{bmatrix} -k_1 & 1 \\ -k_2 & -\frac{v}{\sigma} \end{bmatrix} \tag{48}$$

First error variable is the difference between input and real value for slip:

$$z_1 = \lambda - \lambda_0 \tag{49}$$

$$\dot{z}_1 = \dot{\lambda} - \dot{\lambda}_0 = a - \dot{\lambda}_0 \tag{50}$$

With 'normal' backstepping, a would now be used for the virtual control. As it is not measured it is substituted with it's estimate and inserted in equation 50:

$$\dot{z}_1 = \xi_{02} + \mu_H \xi_{12} + \varepsilon_2 - \dot{\lambda}_0 \tag{51}$$

One of the variables must be used to construct the virtual control. Since ξ_{02} contains the control signal u it is used:

$$\begin{aligned} z_2 &= \xi_{02} - \alpha_1 - \dot{\lambda}_0 && (52)\\ \alpha_1 &= -c_1 z_1 - d_1 z_1 - \hat{\theta}_1 \xi_{12} && (53) \end{aligned}$$

Where $\hat{\theta}_1$ is an estimate of the unknown parameter μ_H.
With these definitions $\dot{z}_1$ is:

$$\dot{z}_1 = -c_1 z_1 + z_2 - d_1 z_1 + (\mu_H - \hat{\theta}_1)\xi_{12} + \varepsilon_2 \tag{54}$$

First Lyapunov function candidate is chosen as a quadratic function of errors. If this would prove to be a Lyapunov function, it would mean that the error z_1 would go to zero and that the estimate for μ_H converges to it's real value.

$$\begin{aligned} V_1 &= \frac{1}{2} z_1^2 + \frac{1}{2\gamma_1}(\mu_H - \hat{\theta}_1)^2 + \frac{1}{d_1}\varepsilon^T P_0 \varepsilon && (55)\\ \dot{V}_1 &= z_1 \dot{z}_1 - \frac{1}{\gamma_1}(\mu_H - \hat{\theta}_1)\dot{\hat{\theta}}_1 + \frac{1}{d_1}\frac{d}{dt}(\varepsilon^T P_0 \varepsilon) && (56) \end{aligned}$$

Here the constant γ_1 is the adaptation gain for the first estimate of μ_H. Matrix P_0 has the properties

$$\begin{aligned} P_0 = P_0^T &> 0 && (57)\\ P_0 A_0 + A_0^T P_0 &= -I \end{aligned}$$

Note that P_0 doesn't have to be known, it must only exist. Further calculations (Described in [3]) lead to

$$\dot{V}_1 = \ldots \leq z_1 z_2 - c_1 z_1^2 + (\mu_H - \hat{\theta}_1)(z_1 \xi_{12} - \frac{1}{\gamma_1}\dot{\hat{\theta}}_1) - \frac{3}{4d_1}\varepsilon^T \varepsilon \tag{58}$$

With the update law $\dot{\hat{\theta}}_1 = \gamma_1 z_1 \xi_{12}$ the term with the uncertainty of μ_H is eliminated.

Function V_1 can not be proven to be a Lyapunov function since z_2 is not known. To continue design, z_2 is derivated:

$$\begin{aligned} \dot{z}_2 &= \dot{\xi}_{02} - \dot{\alpha}_1 - \ddot{\lambda}_0 \\ &= u + k_2(\lambda - \xi_{01}) - \frac{v}{\sigma}\xi_{02} - \frac{\partial \alpha_1}{\partial \lambda}(\xi_{02} + \theta \xi_{12} + \varepsilon_2) \\ &\quad - \frac{\partial \alpha_1}{\partial \xi_{12}}(-k_2 \xi_{11} - \frac{v}{\sigma}\xi_{12} + \varphi(\lambda)) - \frac{\partial \alpha_1}{\partial \hat{\theta}_1} \cdot \gamma_1 \xi_{12} z_1 \\ &\quad - \frac{\partial \alpha_1}{\partial \lambda_0}\dot{\lambda}_0 - \ddot{\lambda}_0 && (59) \end{aligned}$$

To deal with the unknown μ_H a new estimate $\hat{\theta}_2$ has to be employed, and to deal with the estimation error ε_2 nonlinear damping is used. Otherwise the control signal u is chosen to cancel those parts of $\dot{z}_2$ that doesn't increase stability:

$$\begin{aligned} u &= -c_2 z_2 - z_1 - d_2(\frac{\partial \alpha_1}{\partial \lambda})^2 z_2 - k_2(\lambda - \xi_{01}) + \frac{v}{\sigma}\xi_{02} + \\ &\quad + \frac{\partial \alpha_1}{\partial \lambda}(\xi_{02} + \hat{\theta}_2 \xi_{12}) + \frac{\partial \alpha_1}{\partial \xi_{12}}(-k_2 \xi_{11} - \frac{v}{\sigma}\xi_{12} + \varphi(\lambda)) \\ &\quad + \frac{\partial \alpha_1}{\partial \hat{\theta}_1} \cdot \gamma_1 \xi_{12} z_1 + \frac{\partial \alpha_1}{\partial \lambda_0}\dot{\lambda}_0 + \ddot{\lambda}_0 \qquad (60) \end{aligned}$$

Insert control law for u in equation for $\dot{z}_2$,

$$\dot{z}_2 = -c_2 z_2 - z_1 - d_2(\frac{\partial \alpha_1}{\partial \lambda})^2 z_2 - \frac{\partial \alpha_1}{\partial \lambda}\varepsilon_2 - \frac{\partial \alpha_1}{\partial \lambda}\xi_{12}(\mu_H - \hat{\theta}_2) \qquad (61)$$

The equation for $\dot{z}_2$ can now be used to make a new Lyapunov function candidate. Augment V_1 with quadratic terms of errors z_2 and $(\mu_H - \hat{\theta}_2)$:

$$\begin{aligned} V_2 &= V_1 + \frac{1}{2}z_2^2 + \frac{1}{2\gamma_2}(\mu_H - \hat{\theta}_2)^2 + \frac{1}{d_2}\varepsilon^T P_0 \varepsilon \qquad (62) \\ \dot{V}_2 &= \dot{V}_1 + z_2 \dot{z}_2 - \frac{1}{\gamma_2}\dot{\hat{\theta}}_2(\mu_H - \hat{\theta}_2) - \frac{1}{d_2}\varepsilon^T \varepsilon \\ &\leq -c_1 z_1^2 - c_2 z_2^2 - (\frac{3}{4d_1} + \frac{3}{4d_2})\varepsilon^T \varepsilon - (\mu_H - \hat{\theta}_2)(\frac{\partial \alpha_1}{\partial \lambda}\xi_{12} z_2 + \frac{1}{\gamma}\dot{\hat{\theta}}_2) (63) \end{aligned}$$

Only one non-quadratic term, and that can be eliminated with an update law for $\dot{\hat{\theta}}_2$. This makes V_2 a Lyapunov function, thus tracking error z_1 goes to zero. So in addition to the update laws for estimator ξ there are the following:

$$\begin{aligned} \dot{\hat{\theta}}_1 &= \gamma_1 \xi_{12} z_1 \qquad (64) \\ \dot{\hat{\theta}}_2 &= \gamma_2(c_1 + d_1)\xi_{12} z_2 \qquad (65) \\ u &= -c_2 z_2 - z_1 - d_2(c_1 + d_1)^2 z_2 - k_2(\lambda - \xi_{01}) + \frac{v}{\sigma}\xi_{02} \\ &\quad -(c_1 + d_1)(\xi_{02} + \hat{\theta}_2 \xi_{12}) - \hat{\theta}_1(-k_2 \xi_{11} - \frac{v}{\sigma}\xi_{12} + \varphi(\lambda)) \\ &\quad -\gamma_1 \xi_{12}^2 z_1 + (c_1 + d_1)\dot{\lambda}_0 + \ddot{\lambda}_0 \qquad (66) \end{aligned}$$

These are the control laws for the system with the state a. To make these control laws applicable for the real system equation 42 is used to get the real outsignal $\hat{u}$.

6.4.2 Other design issues

Too keep the design relatively simple, several properties of the model have not been considered in the design. They do however still affect the process, and therefore some additions have been made after the mathematical design of the Lyapunov controller. These modifications are not proven to maintain stability, and must thus later be tested so that they do not render the feedback system unstable. As the reason to why they are added is to increase stability and reduce nonlinearities, they should reduce rather than increase the instabilities.

- As the brake-actuators cannot transmit forces above a certain limit, and no negative forces, there are risks of integrator windup. This is avoided by an anti-windup function that stops the updating of the μ_H-estimates as soon as the output from the controller is out of bounds.

- The normal-force dynamics are slow enough for F_z to be considered static in the design, but in the implementation the measured value for a_x is used with equation 4 is used to calculate the value to be used.

- v is used in the design as known static parameter, but in fact a measured value is used. As v is used in the nonlinear estimator, it leads to troubles regarding proving the convergence for the estimator.

- Even though the brake-force T_b can be measured by a sensor, the value is estimated in the controller. This is to avoid the sampling time-delay, and thus be able to increase the performance of the control.

6.4.3 Analysis of the controller

The backstepping controller derived in last section is theoretically well suited to control the dynamic system described by equations 27 to 29. But when the controller is implemented for real, noise and unmodeled dynamics will add to the process and change the specifications.

For a condensed view of the error variables they are written as a matrix equation:

$$\begin{aligned}
\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} &= \begin{bmatrix} -c_1 - d_1 & 1 \\ -1 & -c_2 - d_2\left(\frac{\partial \alpha_1}{\partial \lambda}\right)^2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} \xi_{12} & 0 \\ 0 & -\frac{\partial \alpha_1}{\partial \lambda}\xi_{12} \end{bmatrix} \begin{bmatrix} \tilde{\theta}_1 \\ \tilde{\theta}_2 \end{bmatrix} + \begin{bmatrix} 1 \\ -\frac{\partial \alpha_1}{\partial \lambda} \end{bmatrix} \varepsilon_2 \\
\dot{\varepsilon} &= A_0 \varepsilon \qquad (67) \\
\begin{bmatrix} \dot{\tilde{\theta}}_1 \\ \dot{\tilde{\theta}}_2 \end{bmatrix} &= -\begin{bmatrix} \gamma_1 \xi_{12} & 0 \\ 0 & -\gamma_2 \frac{\partial \alpha_1}{\partial \lambda}\xi_{12} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \begin{bmatrix} \tilde{\theta}_1 \\ \tilde{\theta}_2 \end{bmatrix} \equiv \begin{bmatrix} \mu_H - \hat{\theta}_1 \\ \mu_H - \hat{\theta}_2 \end{bmatrix}
\end{aligned}$$

This way it is shown in what way the design parameters affect the system.

The design requires that the estimation errors ε_1 and ε_2 goes to zero asymptotically. To ensure this k_1 and k_2 must be chosen so that the eigenvalues for matrix A_0 are negative. This poses a problem as A_0, and thus its eigenvalues s, are a function of the non-constant velocity:

$$\begin{aligned}
\det(sI - A_0) &= 0 \Rightarrow \qquad (68) \\
s &= -\frac{k_1 + \frac{v}{\sigma}}{2} \pm \sqrt{\frac{(k_1 - \frac{v}{\sigma})^2}{4} - k_2} \qquad (69)
\end{aligned}$$

Study of the function reveals that the eigenvalues cannot be non-negative as long as k_1 and k_2 are higher than zero. Care has to be taken when choosing k_1 and k_2 to avoid a too poorly damped system.

7 Evaluation of controller performance

To test if the designed controller is able to perform satisfactory brakecontrol simulations are carried out in simulink. By using the mathematical models derived in section 2 and 3 several plants representing the car are implemented. These plants are of different complexity, where gradually more and more of the vehicle dynamics are taken into account, and by the last model all the properties that are mentioned in the modeling sections are taken into account. The models represent only a single wheel by the use of the equations of the quarter-car, and where the normal force is used it is calculated as a function of acceleration as described by the longitudinal model.

Thus is can first be controlled that the controller functions well for the reduced model of the car that it is designed for. Then the model is made more and more similar to the real test-vehicle to test how robust the controller is. This is also useful as a preparation to implementation,where the parameters must be known. By using simulations the amount of testing can be reduced.

7.1 Design of the tests

The simulations are carried out with both controllers from section 6. They are connected to the plants as in figure 17, where the controller blocks contain all functionality of the controllers, including estimators and filters.

The aim of control is to regulate the slip so that the lateral force can be kept high while the braking force is at a maximum. For some guidelines to what is considered to be good control a set of specifications from [7] is used. The ABS controller should be able to comply with:

- no wheel lock is allowed for $v > 4\frac{m}{s}$
- wheel lock for less than 0.2 seconds is allowed for $0.8\frac{m}{s} < v < 4\frac{m}{s}$
- no demands for $v < 0.8\frac{m}{s}$

Of course should also the time for stopping the vehicle be at a minimum.
Unless stated otherwise, the tests are performed with the following parameters (when applicable)

- Initial speed $v = 10\frac{m}{s}$
- Initial value for μ_H estimators ($\hat{\theta}_1$ and $\hat{\theta}_2$)is 0.7
- Real value for μ_H is 0.9
- Reference signal :
  $$\begin{array}{ll} \lambda_0 = 0 & 0 < t < 0.5 \\ \lambda_0 = 9 \cdot (t - 0.5) & 0.5 < t < 0.516 \\ \lambda_0 = 0.14 & 0.516 < t \end{array}$$

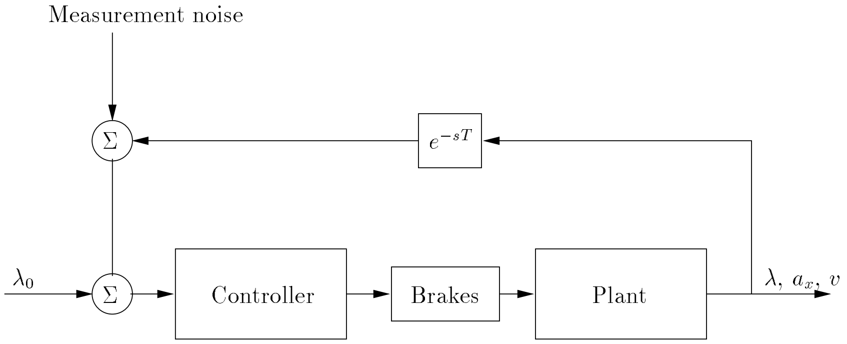

Figure 17: The schematics used when testing the controllers. The time delay and measurement noise are equal to zero unless stated otherwise.

- The physical parameters of the modelled car are:

Wheel moment of inertia	J	1	kgm^2
Wheel radius	r	0.31	m
Tire relaxation length	σ	0.68	m
Mass of quarter car	m	450	kg

  These values are approximately equal to those of a real vehicle. The value for the tire relaxation length is the one found in section 4.

- The controller parameter values varies between the experiments.

- The simulations simulate a front tire. This is only relevant for the calculation of F_z and for the value of J.

7.2 Basic dynamic system

In the first test the plant is based on exactly the equations used when designing the Lyapunov observer backstepping controller, i.e. on equations 36, 27 and 28, where the nonlinearity is simplified as in equation 43. This means that the model uses static values of v and F_z. This test is made to verify that the controllers are able to control the system at all.

The control is quick for both controllers, even though both display slight oscillations. The unnatural conditions (neither noise nor sampling delays) make it possible to use high gain and thus get quick response times. Plots of the results are found in figure 18.

7.3 Dynamic speed and normal force, brake-force limitations

Next test is made on a more realistic plant where the model is implemented fully, though without the uncertainties introduced by measurement, i.e sam-

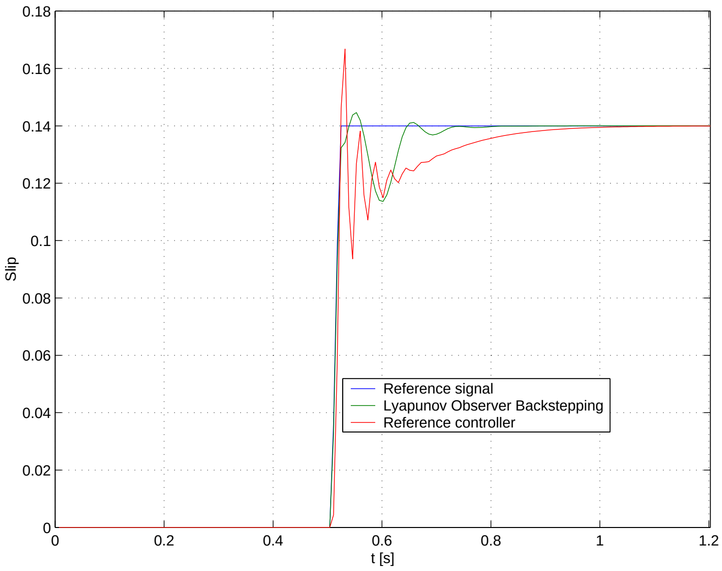

Figure 18: Simulation for both controllers using the most basic system. Neither noise nor timedelay, no dynamic v or F_z.

pling time-delay and measurement noise. This means that F_z and v are not stationary anymore, and that a state for v is added (as in equation 29). The brake actuators are also implemented fully, meaning that the effective brakeforce is limited to the interval $0 < T_b < 3500$ [Nm], and that the signal to the actuator cannot be negative. The anti-windup function from section **??** is added to prevent integrator windup.

Adding the dynamics for v and F_z does little impact on the results. However, the actuator limitations does change the results for the Lyapunov observer backstepping controller slightly, and the performance is compared versus the results from last test in figure 19.

The effects of the rewriting of the nonlinearity in equation 43 can be seen in the convergence values for the μ_H estimators; for the simple plant the convergence is exact, and it should be as the simple plant uses exactly the same equations as the design is based upon. The dynamic plant uses an approximation of the nonlinearity, and this obviously leads to a small static error as mentioned in the design section.

7.4 Rapid changes in θ and λ_0

With the rest of the dynamics added to the model in the last section the controllers showed that they were robust enough to handle a more realistic process. Using the same model, the following two simulations are made:

- Step changes of the friction parameter μ_H to simulate rapidly changing road conditions. This also tests the convergence rates of the friction esti-

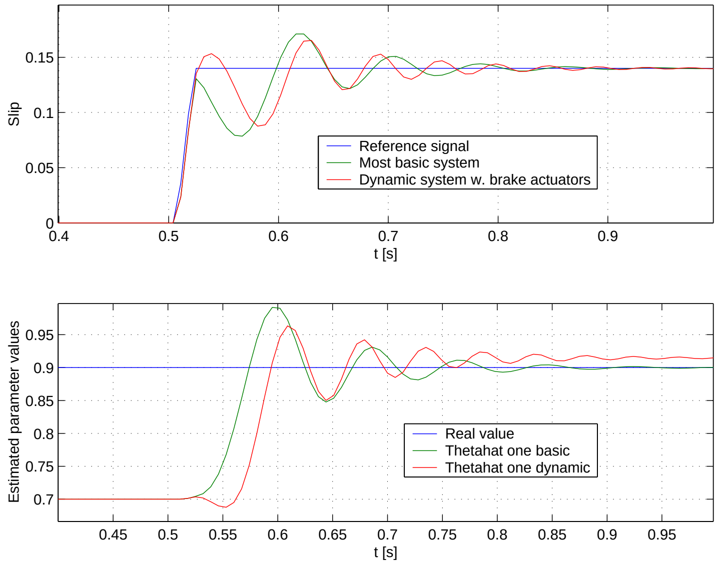

Figure 19: Comparison of the Lyapunov observer backstepping controller for simulations using both the simple (described in section 3.2) and the dynamic plant (described in section 3.2.1). The control itself is good, but there is a static error for the estimation of μ_H on the dynamic plant. Reason is that the $\mu(\lambda)$ function used by the dynamic model sightly differs from the one used by the controller.

mators $\hat{\theta}_i$.

- Rapid changes in reference signal λ_0. To retain flexibility, the controller should be able to control the slip to different values of λ_0, where the changes can be made during the control.

Both simulations are made with both controllers so that a comparison can be made. The control itself is as earlier quick, though slightly oscillatory. When the friction parameter decreases rapidly the slip makes a jump upwards, and thereafter settles. This settling time is the time it takes for the estimators to find their correct values. The adaptation speed is decided by the design parameter γ_i, though there is an upper limit as a too high value renders the system unstable.

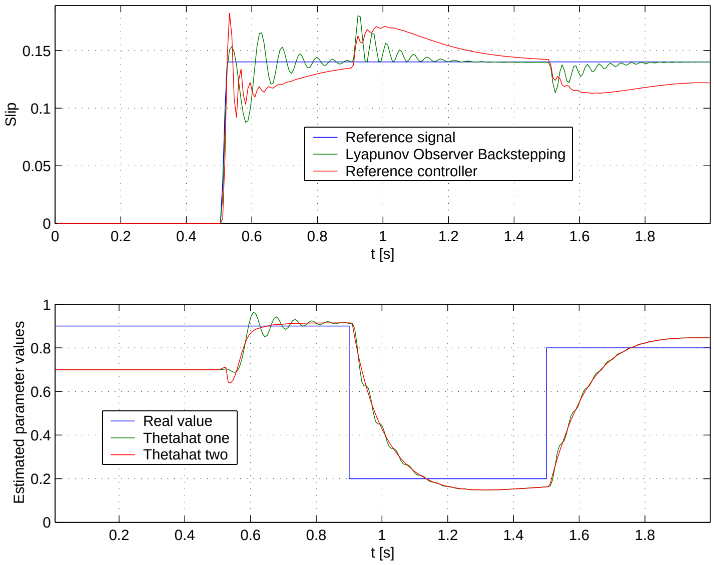

Figure 20: Response to steps in the friction parameter μ_H used by the plant. The plot of estimated parameter values is for the Lyapunov observer backstepping controller.

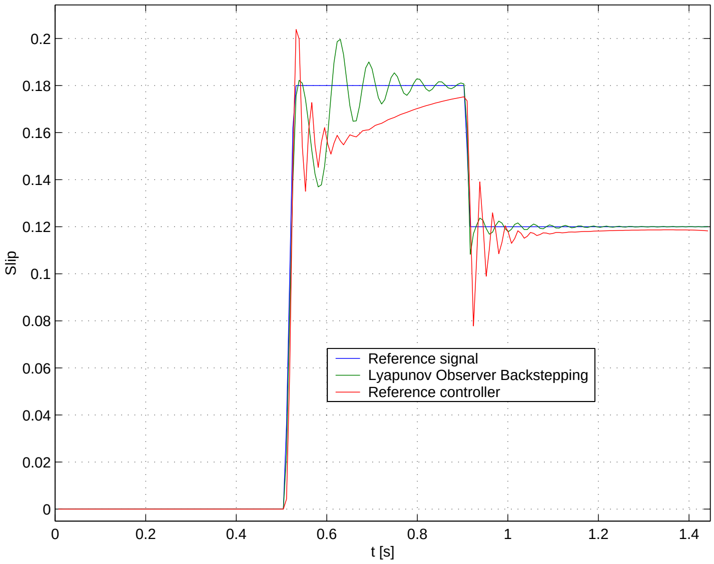

Figure 21: Response to steps in λ_0. That the controller could handle steps was proven earlier (by the step $0 \rightarrow 0.14$) and this test shows that it can also control the slip at other values for λ_0.

7.5 Time-delays and measurement noise

The last simulations are performed with a system as closely representing the real (quarter)car as possible. It is the same system as were used in the last two sections, but first with the sampling time-delay of 14 milliseconds, then with both time-delay and measurement noise added to the λ-signal.

The results of the two tests can be seen in figures 22 and 23. The Lyapunov observer backstepping controller suffers from a relatively long rise time after the initial step increase in λ_0. This is because of the estimators for μ_H , which due to the timedelay must be set to a rather slow adaptation rate. Once the estimators have found their correct values, the control is fast. The noise adds even more uncertainty, but the only really adverse effects it does bring is the poor control for low values of v, where the dynamic effects from the relaxation length are most influential (Due to the time constant for the tire dynamics being proportional to $\frac{1}{v}$).

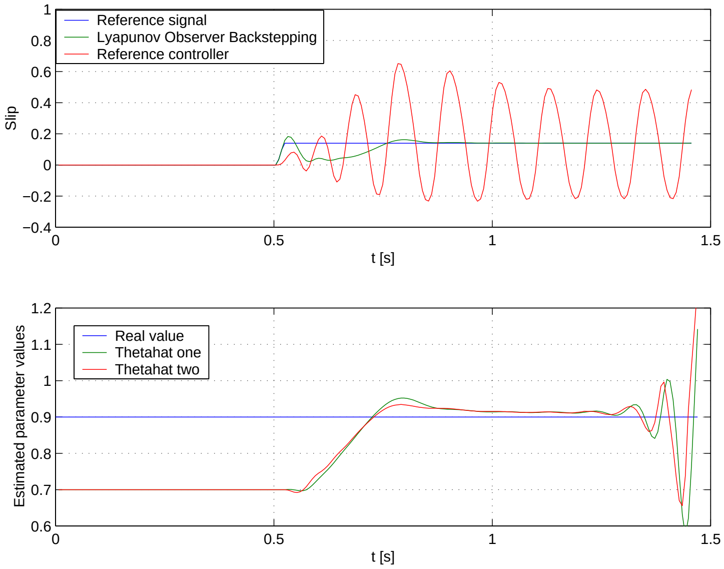

Figure 22: Simulations with time-delay of 14 ms added. This greatly reduces performance of both controllers, and forces adaptation rate to be set lower, thus the increased convergence time for $\hat{\theta}_i$.

For the reference controller the system becomes too complex with the time-delay. It does succeed in stopping the car without locking the wheels, but the slip is far too high and oscillatory for the control to be considered successful. The oscillations are the result of trying to regulate the plant around an unstable point (the value of λ chosen is on the unstable part of the $\mu(\lambda)$-curve) together with a time–delay.

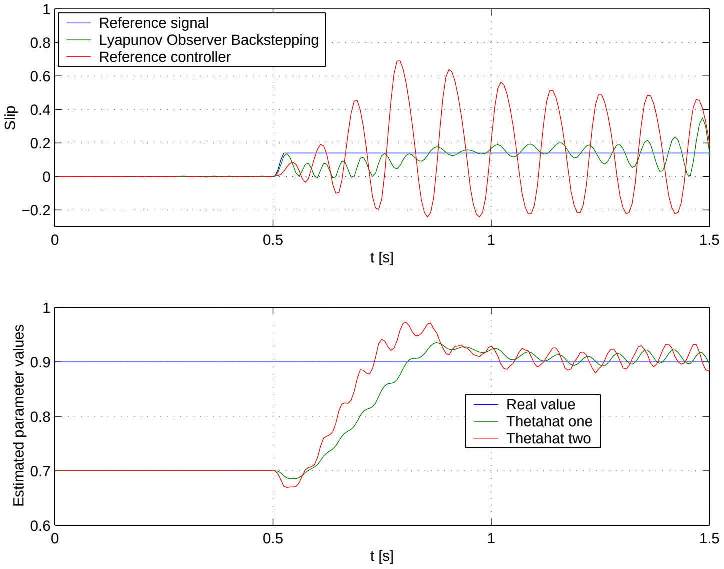

Figure 23: Same test-setup as in figure 22, but with noise added to the λ measurement.

7.6 Test evaluation

Some of the results are worth extra commenting:

- The minor discrepancies seen when comparing the model used for design and the model with brake actuators and dynamic v and F_z shows that the simpler model is enough to base designs on. This is due to the dynamics of v and F_z being so much slower than other variables that they can be considered static for the design.

- The poor performance of the reference controller implicates that something is wrong with the model. Since it doesn't oscillate when implemented for real, it seems that the real system is more damped than the model, either is the wheel dynamics not so influential or is the brake actuator dynamics of less importance.

- Another problem is the uncertainties associated with using the $\mu(\lambda)$ function in the design. That the function varies greatly is shown in section 3.4, and as the function is used in the controller this introduces an uncertainty that must be handled through robustness. That the controller works even though it uses an different function shows that the adaptation of $\hat{\theta}_i$ is enough to compensate for this uncertainty.

It should also be mentioned that these tests favor the Lyapunov observer backstepping controller, since the equations used for the tests are the same that were used for the design. The controller should be evaluated in a real vehicle to get a non-biased evaluation of it's performance.

8 Conclusions

The static model that had priorly been used was obviously unable to provide a good estimation of the actual friction force. That didn't mean that it cannot be used for control design, but it meant that it missed out on some property of the process.

A search for suitable models to describe the tire longitudinal dynamics revealed that most models are based on the same approach. By modeling the deformation of the tire as a first-order dynamic system, they get an extra state that describes the present state of the dynamics. This approach functions well as it is relatively simple, it automatically reduces measurement noise, it represents a physical phenomenon and it is easy to add to already existing static models. By using this extended model on data from simulations it was shown that it could model the friction force quite precisely. Though when the extended system was used to make a model in simulink, the controller that had earlier proven its use in real tests failed to control the model. This indicates that something is wrong with the model.

An other approach is the LuGre model, but this was discarded as it neither could model the friction on the test-vehicle nor is it possible to add easily into the already existing models.

The main reason to find a model that could better model the friction was to use it for design. Thus a controller is designed which takes the extra dynamic state as well as brake actuator dynamics into account. In tests this controller shows results that are better than those for the existing controller. The time it uses to converge is mainly due to the estimation for μ_H being relatively slow.

Even though the controller performed well in simulations it is still not really proven until it has been implemented and tested in a real vehicle. Not only might there exist dynamics that are not covered by the model, it might also prove that the controller requires too much computational capacity due to its complexity.

8.1 Suggestions on further work

The most obvious continuation of the work is to implement the controller and test it on a real vehicle. This work must begin with the transformation of the controller to use a discrete representation of time. Since Lyapunov theory can only be used to prove stability for continous systems, the transformation must be done with care so that the stability of the design is not destroyed.

After that the controller can be implemented, which should be quite straightforward once the discretization is done, though through the sheer size of the controller it might prove difficult to make the implementation quick enough for the car's computer system too handle.

The relatively poor convergence rate for the μ_H–estimators and thus the control could be amended by using a Multi-Model Observer, an observer that in parallel evaluates the correspondence for several values of the parameter to be estimated, and in case of an abrupt change is able to switch to the estimate most close to the new real value, thus improving convergence rate significantly.

The extended model could be used to base other control design on. A controller based on Sontag's formula should be quite straightforward to implement by using the Lyapunov functions used by the observer backstepping controller. Other designs could be made by for instance linearising the nonlinearity.

As for modeling a few properties were mentioned that were known to influence the process but weren't included for various reasons. These could be investigated, and if shown that they make the model more accurate they should be included in the model. Examples of such properties include suspension and higher-frequency dynamics for the tire model.

References

[1] Åström, Karl J. & Wittenmark, Björn *Adaptive Control, 2nd ed.* Addison-Wesley 1995

[2] Khalil, Hassan K. *Nonlinear Systems* MacMillan 1992

[3] Krstić, Miroslav & Kanellakopoulos, Ioannis & Kokotović, Petar *Nonlinear and Adaptive Control Design* John Wiley & Sons 1995

[4] Mitschke, Manfred *Dynamik der Kraftfahrzeuge, Band A:Antrieb und Bremsung* Springer-Verlag 1995

[5] Kortüm, Willi & Lugner, Peter *Systemdynamik und Regelung von Fahrzeugen* Springer-Verlag 1994

[6] Zegelaar, Peter W.A. *The Dynamic Tire Response to Brake Torque Variations and Road Unevennesses* Delft U.T. 1998

[7] Kalkkuhl, J & Giesa, D *A nonlinear adaptive tire slip controller* Daimler-Chrysler AG 1999

[8] Gustafsson, Fredrik *Slip-Based Tire-Road Friction Estimation* Department of Electrical Engineering, Linköping University, Sweden 1996

[9] van der Jagt, P. & Pacejka, H: *Influence of tyre and suspension dynamics on the braking performance of an anti-lock system on uneven roads* Delft U.T. 1989