Introduction ABS

Introduction ABS

C.J.S. Nederkoorn
University of Techology Eindhoven
Reportnumber: 93-096
July 9, 1993

Summary

In this report a short introduction in ABS is given. After some simplifications and special conditions we build a simple ¼-car model. With this model for the car and the tyre model of Pacejka we make simulations. In these simulations we are able to view the effects of different controllers on the input M_{br} , and the main variable: the relative slip λ.

1 Introduction

The purpose of this short essay, is to get an idea of the functoning of Anti Lock Brake Systems (ABS) in personal cars. These systems are the result of the developments in the car industries and the growing demands on safety. As early as during the thirties' ideas regarding ABS arose, which however did not become reliable and payable until the developments in the hydraulics and micro electronics in the seventies' took place.

To design an ABS, one needs a good knowledge of the effects of braking. The manner we brake strongly influences the maneuverability. The most important parameter in this process is the slip of the tyre in relation to the road that should be maintained at a certain degree to control the car.

2 Purposes and problems

The first problem we come across with ABS is to determine the desired behaviour of the system. Furthermore this system should never be inferior to a conventional system.

The most important features are summarized below:

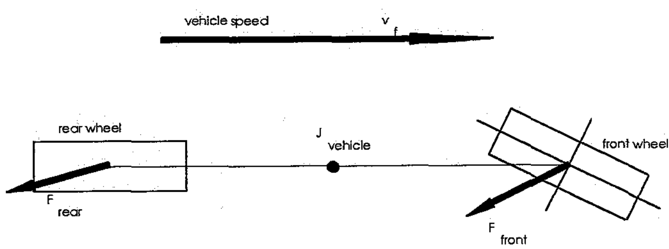
- Maneuverability is preferred to a short braking distance.
- One should be able to correct swerves, caused by different brake forces at the tyres, by steering.
- ABS must function from very high speeds down to low speeds where safety is not relevant any more.
- ABS should adapt fast to the variations of traction (i.e., force and torque the tyres can practice on the car).
- Especially at rough surfaces, during aquaplaning, at ice or snow, the system should function well.
- Excitation of other parts of the car must be avoided.
- When the system fails, the system should switch over to the conventional system and generate a warning to the driver.
- The system should be cost-effective, light and compact.
- The system should be connectable to existing hydraulic energy supplies (e.g., for servo-assisted brakes and steering).

The complications that occur are:

- Adaptation times that arise during the variations of traction can lead to unexpected phenomena.
- The interaction process between road and tyre is unstable for large slip coefficients.
- The input signals are physical limited, the number of measurements has to be restricted on cost-effective viewpoint.

- Interaction between tyre and road is strongly non-linear and is influenced by longitudinal and side slip, weight on the tyre, moment of inertia of the tyre, steering edge, etc.

Before we can continue with modeling of the brake process, we first need to know what the purpose of ABS is.
The main goal of ABS is to supply a control law, which controls the torque on the tyres in such way that the car stops as quickly as possible, maintaining maneuverability.
Maneuverability implies the possibility to change direction, between limits around the driving direction. Besides the force needed for braking the car, there has to be a torque that turns the car round its center of gravity. Looking at a ½-car model (bicycle model) one can see that maneuverability is possible when the tyres can supply longitudinal forces as well as side forces.
Our purpose is to get a first insight into the relevant parameters. Also we want to be aware of the simplifying conditions and assumptions we've made to obtain this vision.



3 Modeling

3.1 Tyre model

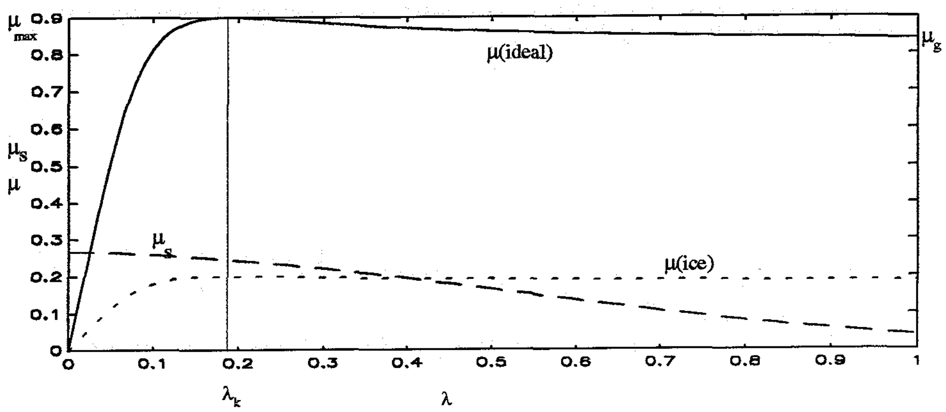
The interaction of forces between tyre and road is mainly a function of the relative slip λ. In our situation of braking λ is defined as:

(f 1) $$\lambda = \frac{v_f - v_r}{v_f}$$

with: v_f: *speed of the car with regard to the road*
v_r: *speed of the wheel with regard to the road*

Note that λ must have a value between 0 and 1.

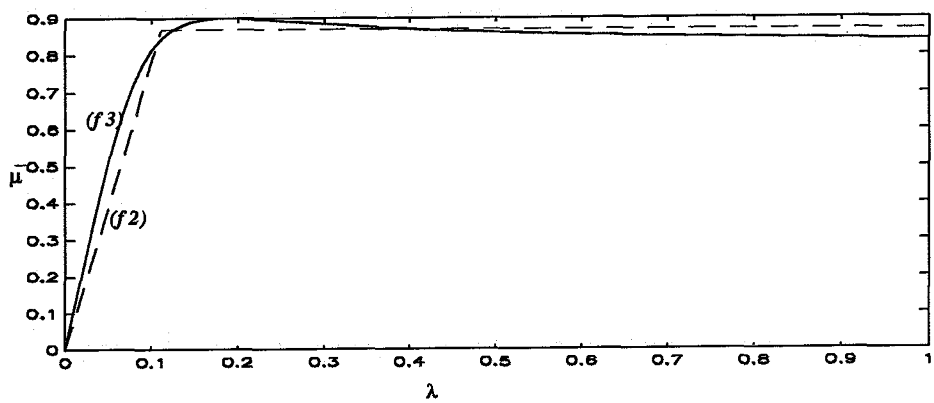
We can split the forces of the tyres in longitudinal and side forces. The longitudinal friction coefficient, from here called the brake coefficient μ, and the side friction coefficient μ_s as functions from λ, look as follows:



(p 1)

De highest value of $\mu=\mu_{max}$, occur at $\lambda=\lambda_k$. As we can see, μ_s has an acceptable value for this value of λ (around the 90 % of its maximimum). When λ increases, μ_s decreases towards zero and the side forces delivered by the tyres are insufficient to supply a torque that is big enough to control the maneuverability of the car.
The value of λ_k differs from situation to situation, but dependent on the sort of tyre, surface of the road, steering edge, its value will be between 0.08 and 0.30. The value of μ_{max} can vary among ±0.9 for ideal circumstances and 0.2 for an icy road ***(p 2)***.
When we want to approach μ as a function of λ we can formulate:

(f 2)
$$\bar{\mu}(\lambda) = \begin{cases} k * \lambda & voor\ \lambda \le \lambda_k \\ \mu_{max} & voor\ \lambda \ge \lambda_k \end{cases} \quad with: \begin{array}{l} k = \mu_{max} / \lambda_k \\ 0 \le \lambda \le 1 \end{array}$$



(p 2)

Besides the fact that the outline of this formula doesn't look like its real outline, there are some applications when we wish to use eventually the derivative of $\bar{\mu}(\lambda)$ for $\lambda=\lambda_k$. More elegant and numerical convenient is to use the tyre model of Pacejka. This is the so called magic formula (it means just a formula) where one can choose parameters (A, B, C, D) so $\bar{\mu}(\lambda)$ resembles ***(p 1)***.

$$(f3)\quad \begin{aligned} \nu &= (1-E) * \lambda + E / B * \arctan(B * \lambda) \quad \text{with } \nu\text{: help variable} \\ \bar{\mu} &= D * \sin(C * \arctan(B * \nu)) \end{aligned}$$

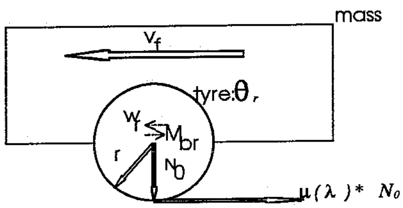
As one can see in *(f 3)*, D is equal to the maximum value of $\bar{\mu}(\lambda)=\bar{\mu}_{max}$.
From here we make no diference between μ and the approach of μ, $\bar{\mu}$.

To get familiar with the relevant parameters that describe the brake process, we build a numerical model. As we've seen before, maneuverability is guaranteed when λ lies round λ_k. If we want to study strong braking, we can neglect two dimensional effects by looking only at λ, so for our purpose a one dimensional model will be sufficient. We assume the following:

- When λ temporarily exceeds λ_k, we assume that the car brakes in a straight line.
- We don't take maneuverability into account.
- We look only at one wheel, we use a one dimensional model.
- The brake coefficient may vary but place and value are known by the system. The steering edge does not influence the brake coefficient. In practice there must be an algorithm to calculate $\mu(\lambda)$ from measurements.
- The side friction coefficient is acceptable at λ_k for any situation.
- The starting vehicle speed is 5 [m/s].
- Vehicle speed below 0.2 [m/s] is not of importance for safety.
- We don't take the hydraulics into account. Instead, we restrict the derivative of M_{br} to 20.000 [Nm/s], equivalent with the performance of a common hydraulic system.

3.2 Mechanical model

(p 3)



With the preceding assumptions and simplifications a ¼-car model will do for us. This model consists of a mass and a massless wheel with moment of inertia θ_t. On the wheel operate the input M_{br} controlled by the ABS and the torque caused by friction on the road ***(p 3)***. Formulating the equations of motion yields:

(f 4) $mass * \dot{v}_f = -\mu(\lambda) * N_0$

(f 5) $\theta_r * \dot{\omega} = M_{wr} - M_{br}$

and:

(f 6) $\lambda = 1 - \omega_r\, r / v_f$

(f 7) $mass * g = N_0$

(f 8) $M_{wr} = \mu(\lambda) * r * N_0$

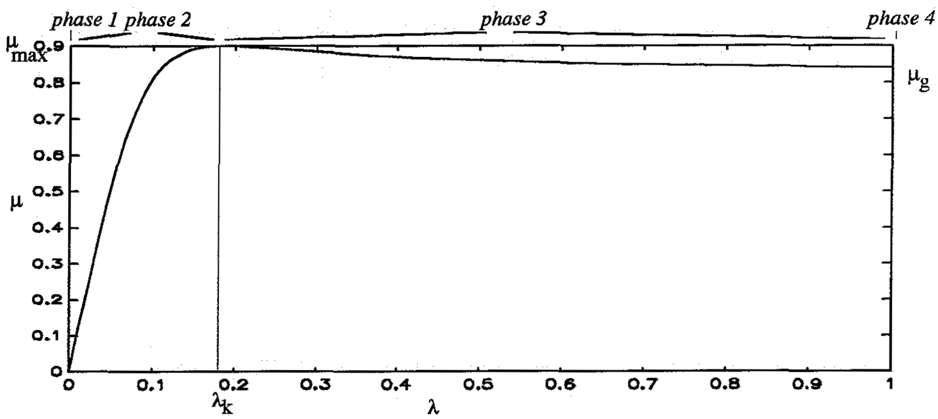
with:

λ	*as defined in **(f 1)***
N_0	*weight of the mass*
r	*radius of the tyre*
g	*gravity acceleration*

We have chosen the parameters in the magic formula *(f 3)* B=.104, C=1.27, E=-1.61.
D is chosen between 0.2 and 0.9 dependent on the simulation (for D represents the maximum brake coefficient).

The slip condition of a wheel is divided in 4 phases ***(p 4)***:

- 1^{st} *phase*: phase without braking. λ and μ equal zero, there is no brake action.
- 2^{nd} *phase*: stable phase. λ lies in between zero and λ_k, μ lies in between zero and μ_{max}.
- 3^{rd} *phase*: instable phase. λ lies between λ_k and 1, μ between μ_{max} and μ_g. In this phase a wheel will lock very soon, dependent on his moment of inertia.
- 4^{th} *phase*: λ equals 1 and μ equals μ_g. A wheel stays in the 4^{th} *phase* until the input M_{br} decreases under μ_g*r*N_0.



(p 4)

The equations of motion in *phase 4* differ from *phase 2* and *phase 3*. In this phase the wheels are locked so $\dot{\omega}$ and ω equal zero and λ equals 1. This yields:

(f 4a) $$\dot{v}_f = \frac{-\mu_g * N_0}{mass}$$

(f 5a) $$0 = M_{wr} - M_{br}$$

Last formula implicates that M_{br} must equal M_{wr} and thus an input controlling a higher value has no practical sense.

3.3 Feedback linearisation

With the formulas *(f 4)* and *(f 5)* we can make the following non-linear equations:

(f9) $\dot{\omega}_r = \dfrac{M_r(\lambda) - U}{\theta_r}$ *(f10)* $y = \lambda$

(f11) $\dot{v}_f = \dfrac{-\mu(\lambda)\, N_0}{mass}$ *(f12)* $\underset{\sim}{x} = \begin{vmatrix} \omega_r \\ v_f \end{vmatrix}$

With: $U = M_{br}(t) = input$

$\underset{\sim}{x} = state\ vector$

$\mu(\lambda) = \mu(\underset{\sim}{x})$

So we obtain: *(f13)* $\dot{\underset{\sim}{x}} = \underset{-}{f}(\underset{\sim}{x}, \mu)$

(f14) $y = \underline{g}(\underset{\sim}{x})$

If we want λ to be λ_k we get:

(f15) $y_d = \lambda_k$ so: *(f16)* $\dot{y}_d = \dot{\lambda}_k = 0$

(f10) $y = \lambda$ *(f17)* $\dot{y} = \dfrac{d}{dt}\left|\dfrac{v_f - \omega_r r}{v_f}\right| = \dfrac{d}{dt}\left|\dfrac{x_1 - x_2 r}{x_1}\right| = r\dfrac{\dot{x}_1 x_2}{x_1^2} - r\dfrac{\dot{x}_2}{x_1}$

with *(f 13)* and linearising around an (unknown) operating point we obtain:

(f18) $\dot{y} = \alpha(\underset{\sim}{x}) + \beta(\underset{\sim}{x}) * U$

With α and β unknown functions.

When we choose the input U as follows:

$$(f\,19)\quad U = \frac{\alpha(\tilde{x})}{\beta(\tilde{x})}\dot{y}_d - \frac{1}{\beta(\tilde{x})}\{U_s\}$$

where the 1st component (zero in our case) takes care of the following of the trajectory. The 2nd component with U_s free to choose should take care of the convergence of y to y_d. From *(f 19)* we get:

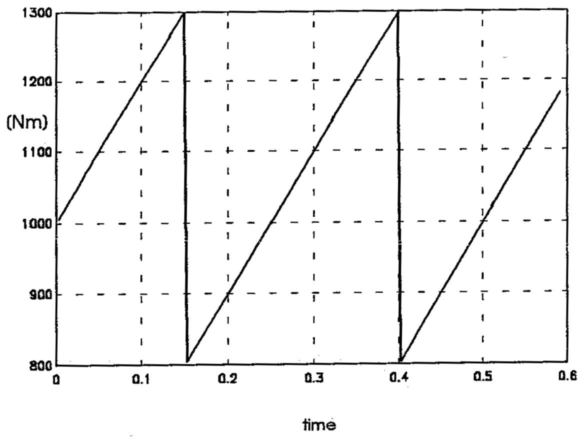
$$(f\,20)\quad U_s = \dot{y}_d - \dot{y} = \dot{e} \qquad \text{and:} \qquad (f\,21)\quad e = y_d - y$$

If:

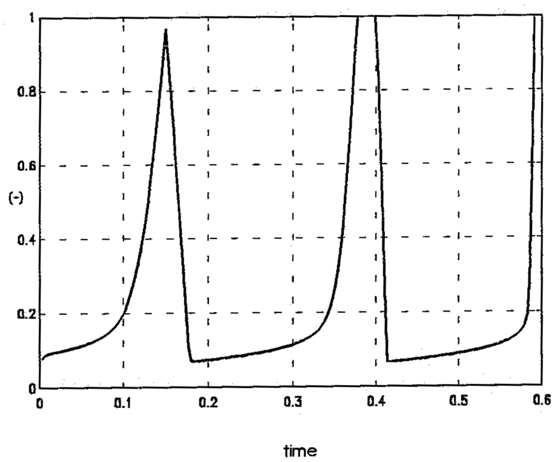
$e > 0$	then	$\dot{e} \leq 0$
$e < 0$	then	$\dot{e} \geq 0$

So we can choose U_s:

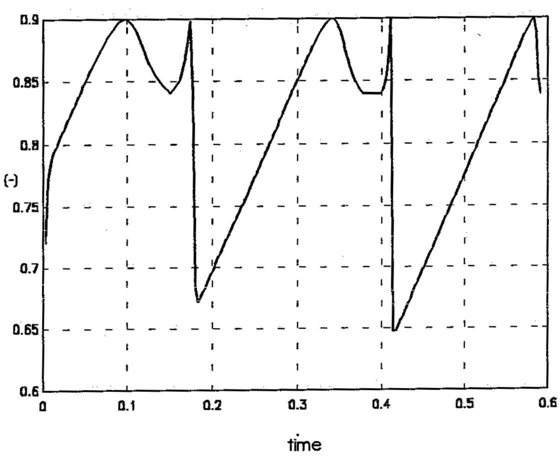
$$(f\,23)\qquad \boxed{U_s = -P\,e}$$



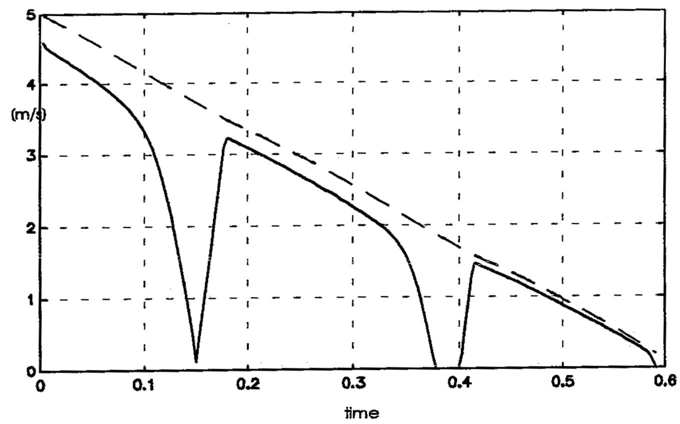
(g 1) $M_{br}(t)$



(g 2) λ *(t)*



(g 3) $\mu(t)$



(g 4) Wheel speed $v_r(t)$ — *and*
vehicle speed $v_f(t)$ - -

Simulation with a saw-toothed input M_{br}. *During the first peak* λ *just doesn't get the value 1. The second and third peak show that* λ *=1, so the wheels are locked. During the first decrease of* M_{br} μ *increases. Then the first decrease takes place (the wheel is in the 3rd phase). The following increase is caused by the jump of* M_{br} *back to 800[Nm]. The wheel is then accelerated by the speed of the mass.*

4 Simulations

Now we will look for three controllers of the input M_{br} and analyse the behaviour of the wheel.

4.1 Saw-toothed torque

The program structure for this input looks as follows:

```
Torque
Mbr(t=0)  = 1000[Nm]
Mbr   =   Mbr + 2000*Δt            Δ t is the integration time
if Mbr greater then 1300[Nm]
            Mbr 800[Nm]
return
```

When we control the input in this way, we obtain a clear view of the different phases of slip the wheel is going through. After the moment the input is activated, the wheel comes into the 2^{nd} *phase*. λ and μ increase both and the mass speed v_f will diminish proportional to μ. After a little while the wheel comes into *phase 3* and λ increases very fast. Looking at ***(f 5)*** we see that ω_r will decrease fast, for:

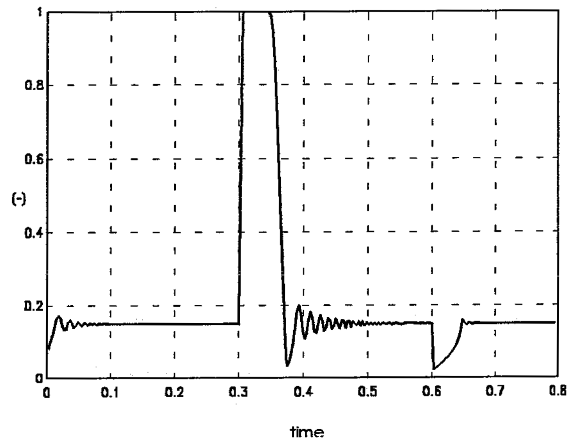
$$(f\,5b) \quad \dot{\omega} = \frac{M_{wr} - M_{br}}{\theta_r}$$

The difference between $M_{wr}(\lambda)$ and $M_{br}(t)$ is extra great because of the decreasing of $M_{br}(\lambda)$. Dependent on the radial speed of the wheel, the wheel comes into *phase 4*. During the first "tooth" of the input M_{br} this doesn't happen. When the input jumps back to 800[Nm], the speed of the wheel increases fast, so the wheel comes into *phase 2*. Here we can find an

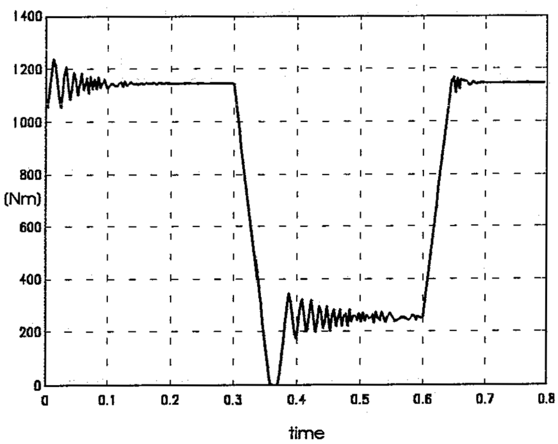
equilibrium between λ and input M_{br} . This process will repeat during the second "tooth", but the speed of the wheel is now lower. The wheel comes now into *phase 4*. The input now is higher than what's possible in reality, because of the locking wheels (a negative wheel speed is not possible in this case).
After the torque jumps back again, the wheel speed increases and the speed of the mass and wheel will finally go to zero.

Remarks:

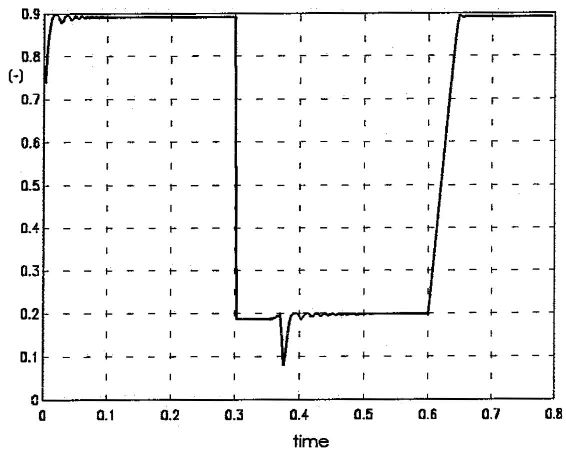
- In *phase 4* a bigger torque than possible is send. The program takes this in consideration and calculates with the biggest possible torque.
- During the locking of the wheels, the decline of the vehicle remains almost constant. The change of the input M_{br} is absorbed mostly by change in λ (dependent on the moment of inertia and ω_r). The value of λ(t) shows that we can say little about maneuverability.
- This way of braking (a sort of pumped braking) gives an almost maximum of decrease, but no guarantee for maneuverability.



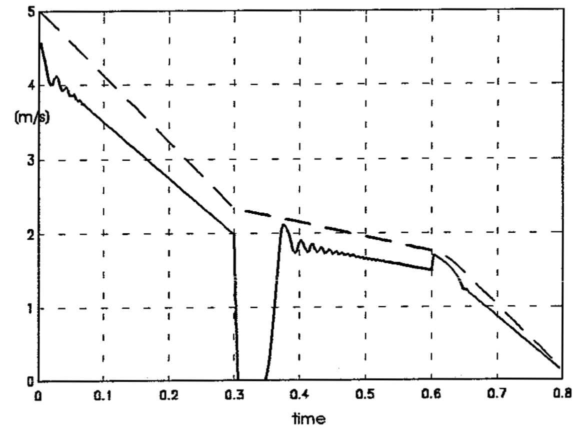
(g 5) λ *(t)*



(g 6) $M_{br}(t)$

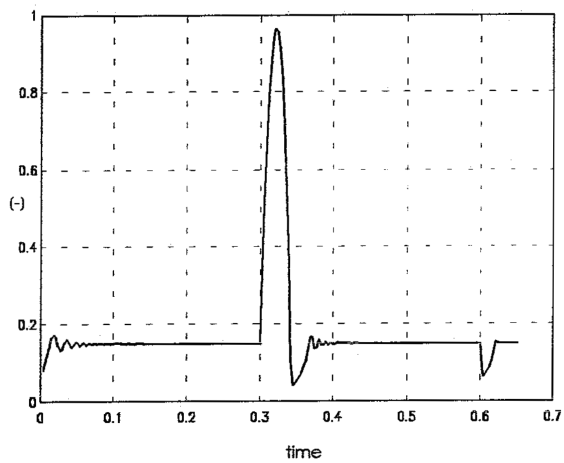


(g 7) $\mu(t)$



(g 8) $v_f(t)$ *- - and* $v_r(t)$—

Results of simulations with a sign control. The peaks in λ *(t) are caused by the jumps of* μ *(t) from 0.9 to 0.2 and back (actually D in the magic formula (f 3) jumps when t=0.3 and t=0.6).*
Through the character of the control there arises a vibration around the equilibrium ($\lambda \approx \lambda_k$*), that could influence the comfort in the vehicle. The performance of the system depends on the quickness that* M_{br} *can change. The quality of the controller depends on the time* $\lambda \neq \lambda_k$.



(g 9) Again λ *(t). This time* μ_{max} *jumps from 0.9 to 0.6 and back.* λ *stays below 1.*

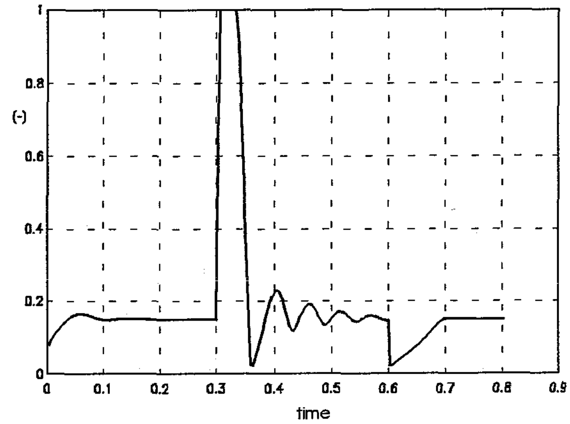
4.2 Control λ to a desired value

4.2.1 Sign control

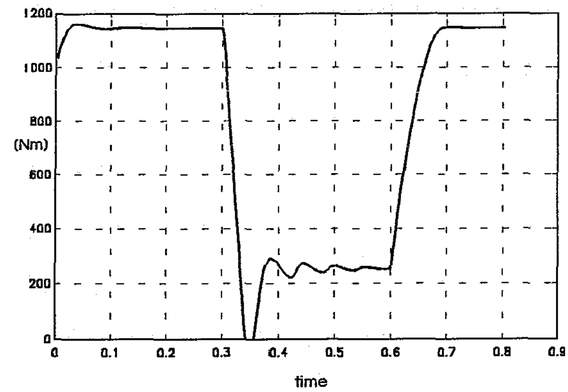
The structure of the first controller we take a look at, looks as follows:

```
Torque
    if(λ<0.15)     ΔMbr= 20.000[Nm/s]*Δt      (Maximum change torque)
    if(λ>0.15)     ΔMbr= -20.000[Nm/s]*Δt
    if(Mbr<0)      Mbr = 0                     (Negative torque not possible)
return
```

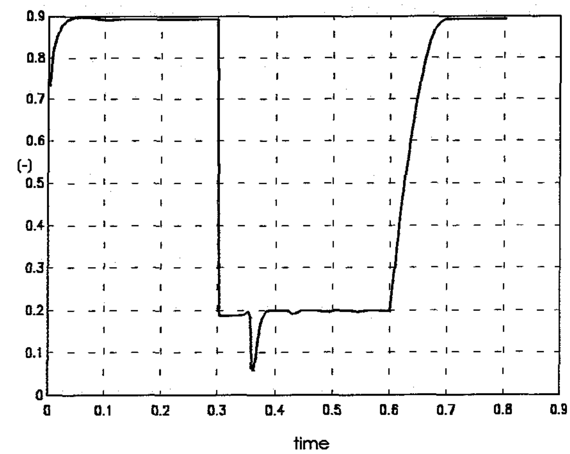
We assume that $\mu(\lambda)$ is known and doesn't change. This means that we can control λ to an adjusted value, for example 0.15. Dependent on the sign of the expression $(0.15-\lambda)$ the torque increases (sign positive) or decreases (sign negative), both with maximum change. Good functioning during variable brake coefficients is demanded, so we vary the brake coefficient between 0.9 and 0.2.



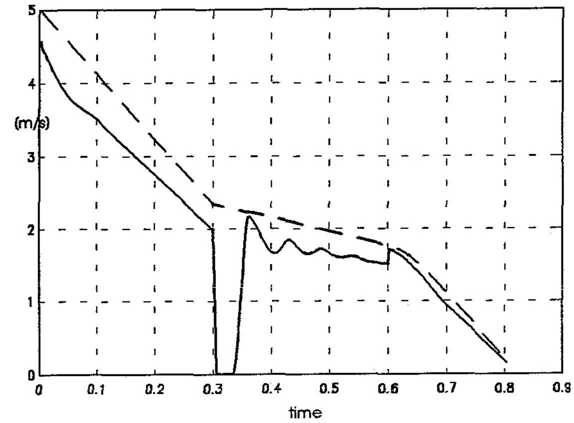
(g 10) $\lambda(t)$



(g 11) $M_{br}(t)$

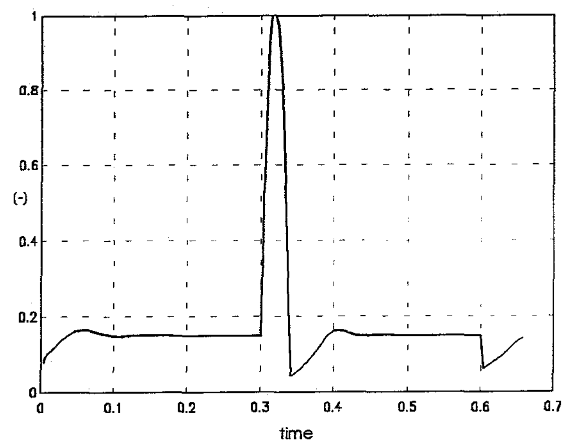


(g 12) $\mu\ (t)$



(g 13) $v_f(t)$ - - and $v_r(t)$—

Results of the simulations with a proportional control. At this control μ_{max} also jumps from 0.9 to 0.2 and back.
All of the plots show an even passage regarding to the sign contol. The efforts of the control around $\lambda = \lambda_k$ is less so $\lambda = \lambda_k$ will not be achieved fast.
Underneath $\lambda\ (t)$ is plotted again with μ_{max} between 0.9 and 0.6.



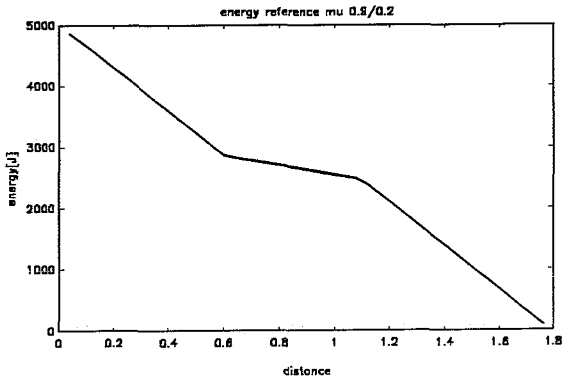
(g 14) $\lambda\ (t)$

4.2.2 Proportional control

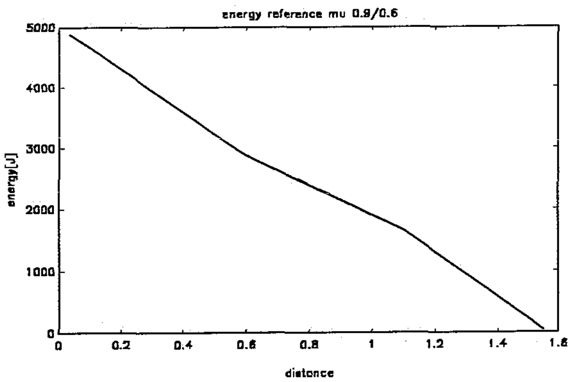
The structure of the second controller we take a look at, looks as follows.

```
Torque
if (λ<0.15)    ΔMbr  =  133.333[Nm/s] * (0.15-λ) * Δt
if (λ>0.15)    ΔMbr  =  -35.000[Nm/s] * (λ-0.15) * Δt
if (Mbr< 0)    Mbr = 0                (Negative torque not possible)
return
```

Dependent on the value of $(0.15\text{-}\lambda_g)$ we change the torque, limiting $\dot{M}_r$ to 20.000 [Nm/s] (this explains the choice of the values in the controller). As λ is closer to the desired value λ_k, the torque changes less.

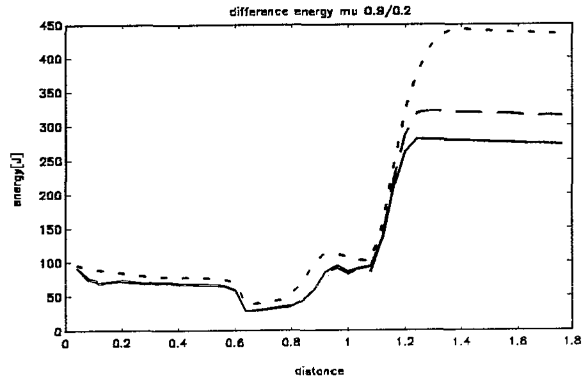


(g 15) *Energy(x_f) (μ_{max} 0.9 or 0.2)*

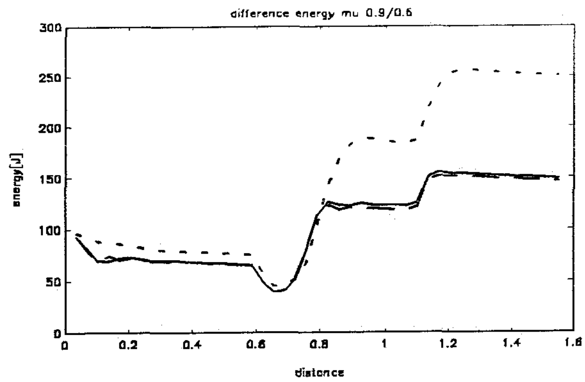


(g 16) *Energy(x_f) (μ_{max} 0.9 or 0.6)*

*Reference for the energy as function of the distance x_f during the simulations. Figure **(g 15)** shows the simulation where μ_{max} jumps between 0.9 and 0.2. Figure **(g 16)** shows the simulation where μ_{max} jumped between 0.9 and 0.6.*
In contrast with the previous simulations when μ_{max} is a function of time, the jumping of μ_{max} is related to the done distance of the vehicle for a fair comparison.
Energy 0[J] implicates a vehicle speed of 0[m/s]. The energy must decrease continuously for the system is pure dissipative.



(g 17) *Δ-Energy(x_f) (μ_{max} 0.9 or 0.2)*



(g 18) *Δ-Energy(x_f)(μ_{max} 0.9 or 0.6)*

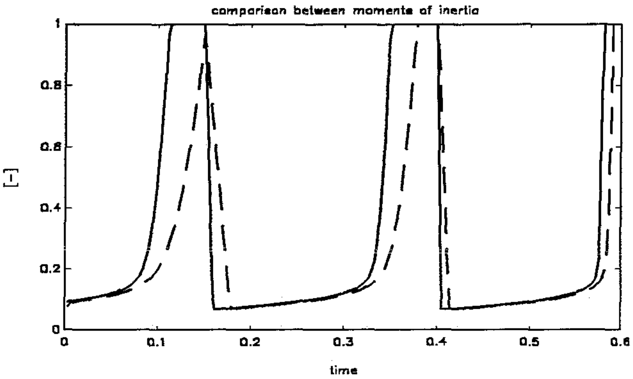
Differences between the energy of the simulations and references.
Sign controller: — —
Proportional controller: - -
Sign/proportional: —
The lower the difference, the better the controller. When λ is around λ_k, the energy of the system with sign controller decreases faster because of the lower efforts of the proportional controller.

4.3 Comparison of energy

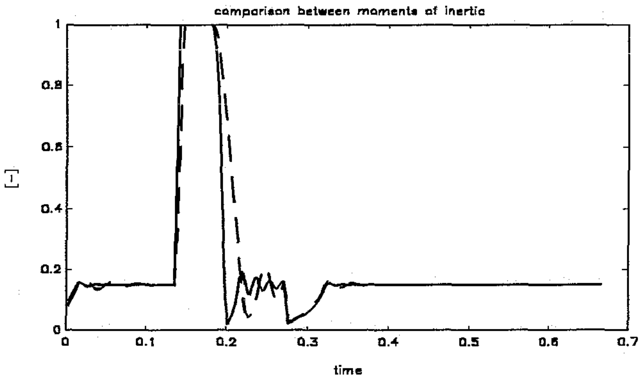
When we want to make a first comparison between the controllers, we can use an energy criterion. A simulation where we hold μ on μ_{max} is useful as a reference trajectory. The energy as a function of time is plotted for μ between 0.9 and 0.2 and μ between 0.9 and 0.6. For the differences between the energies are small, we've printed the differences separately.
A logical conclusion is to try a sign controller that is proportional around λ_k (between 0.1 and 0.2). The energy of this controller is also compared with the other controllers. When μ_{max} switches between 0.9 and 0.2, the overall performance is better. When μ_{max} switches between 0.9 and 0.6 however, the sign controller performs better with respect to brake distance.

The structure of the sign/proportional controller looks as follows:

```
Torque
if (λ<0.10)          ΔMbr =  20000[Nm/s] * Δt
if (λ>0.20)          ΔMbr = -20000[Nm/s] * Δt
if(λ<0.20&&λ>0.10)   ΔMbr = 400000*(0.15-λ)*Δt
if (Mbr< 0) Mbr = 0                 (Negative torque not possible)
return
```



***(g 19)** Moment of inertia 0.25 (—)*
and 0.75 (- -) with saw-toothed torque



(g 20) *Moment of inertia 0.25 (—)*
and 0.75 (- -) with a proportional
controller

4.4 Effect of θ_r

The difference between M_{br} and M_{wr} results in an angular acceleration formulated by ***(f 5)***. When the wheel is in *phase 3* and the torque increase is not large enough, the resulting torque can grow because of the shape of $\mu(\lambda)$. From this point of view we can see the wheel (and its moment of inertia) as a kind of buffer. So a large torque of inertia is good for a low $\dot{\omega}$ and consequently for a low λ. The disadvantage of a large moment of inertia is the chance on a shaky $\lambda(t)$. It results in shaking of $\lambda(t)$ and the input $M_{br}(t)$. The differences caused by the moment of inertia are printed on the opposite page.

5 Simulation program

Underneath is written the structure of the body of the simulation program in ANSI-C. The routines for the controller `Torque` are described in the previous sections. When the program switches from equations of motion I ***(f 4)*** and ***(f 5)*** to equations of motion II ***(f 4a)*** and ***(f 5a)*** λ must be 1. Therefore a new Δt must be calculated. After the integrationstep, λ will then be 1.

bold :	program modules
italic:	program commands
normal:	program module name or variables
italic:	comments

```
Main program
    Startup                          Preparations, opening files
    Get parameters                   Import of parameters
    While t < tend and v > 0.2       simulation conditions
        get D(xf,t)                  calculating friction as a function of the
                                     done distance or time
        onestep                      calculate one step
        archive to file              write results to a file
    end
exit
```

```
Onestep
   If phase 2/3   then  model_magic
   If phase 4     then  model_lock
      calculate torque
      calculate mu
return

Torque
   (User defined)            Described in the previous chapters
return

Mu
   Magic formula (λ,D(t))
return

Model_magic
   calculate equations of motion I   (f4) and (f5)
   integrate with Euler
   calculate λ
   if λ ≥ 1
      calculate new Δt          new Δ t so next λ will be 1
      recalculate equations of motion I
      set phase 4
return
```

```
Model_lock
   calculate equations of motion II         locking wheels, (f 4a)
                                            and (f 5a)
   integrate with Euler
   calculate λ
   if λ < 1
      calculate new Δt                      new Δ t so next λ will be 1
      recalculate equations of motion II
      set phase 2/3
return
```

Bibliography

[1] Bosch Techn. Berichte 7 (1980) 2:
Antiblockiersystem (ABS) für personenkraftwagen

[2] Vehicle System Dynamics Volume 20, Number 3-4:
Shear Force Development ....: page 159