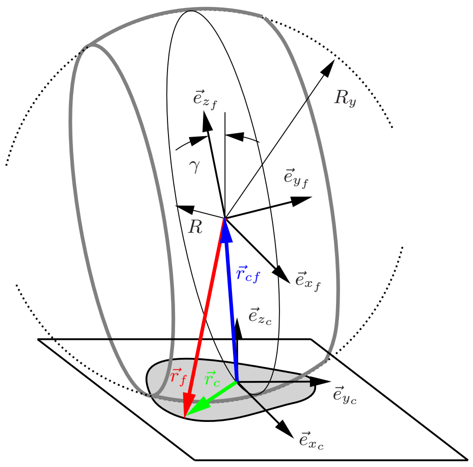
RMOD-K Formula Documentation

Prof. Dr.-Ing. Ch. Oertel

TH Brandenburg, FB Technik - Mechatronik,
Magdeburger Str. 50, PSF 2132,
14737 Brandenburg an der Havel, Germany,
oertel@fh-brandenburg.de

last change: April 24, 2021



Abstract

RMOD-K Formula is an open source collection of two simple tire models and related stuff like optimisation tools and simulation interfaces. It was intended to build a package reaching from parameter optimisation based on measurements of steady state force and moment to simulation models of steady state tire behaviour in vehicle dynamics. This may be used for instance in some field of education or any other field, where steady state tire behaviour is of interest. The basic idea was to provide an algebraic formula with physical meaningfully parameters and a discrete version of tangential contact formulation to get some insights in tangential contact mechanics. The analytic formula allows for instance to investigate the stability of small vehicle models by analytical linearisation even in combined slip situation. This may lead to stability maps with respect to certain tire parameters expressing friction or stiffness properties. The extended discrete version takes into account more detailed information about normal stress distribution, contact area shape and friction properties and is able to deal with camber.
This documentation explains the basic theory behind the models and how to create custom version within the framework of optimisation and simulation interfaces. The structure of the documentation is divided into two parts, in which the first deals with the model equations and assumptions made while the second part explains the software tools and gives some application examples. The author is grateful for all further developments and information about the application of the package. Everybody is invited to use the software, to contribute to the development and to give related comments.

Contents

List of Figures

List of Tables

1 Introduction

Simple tire models are developed for several reasons. The first one is to use such model in vehicle handling simulations, where the steady state force and moments are of great importance [3, 4, 5]. An other may be the aim to have a model which is able to reproduce steady state force and moments in realtime and can be used in state observers within control systems. A third example is the idea to do some analytical work like linearisation at larger slip values, where a closed formula may by helpful to study stability regions of vehicle motion.
If the simple tire model belongs to a family of tire models, it may be used in terms of parameter identification and optimization. More complex tire models up to Finite Element models use a subset of parameters in tangential contact dynamics, which are related to those of the simple tire model. If these parameters are obtained by optimization, they can be used as estimates or initial values in the parameter optimization procedure of complex tire models [2, 8]. This can save a lot of effort, because the function evaluation of the simple tire model is very fast.
Finally, this documentation and the software framework is intended as free download to give every interested person the possibility to use it, to modify the code and add new features or give some comments on software development. If you find the package useful, please cite this technical report in any resulting publications or reports and give some feed back to the author. The BibTex reference is as follows:

```
@TECHREPORT{RMOD-K-Formula,
author = {Ch. Oertel},
title = {RMOD-K Formula Documentation},
institution = {Brandenburg University of Applied Sciences, Faculty of Engineering},
month = {October},
year = {2011},
location = {www.RMOD-K.com}}
```

2 Basic equations

Some assumptions are made in this section to derive a first version of a steady state tire force approximation or formula with a small number of parameters with physical meaning.

2.1 Tangential contact

The assumptions

- of a rectangular contact area shape,
- of longitudinal and lateral slip without turn slip and
- of a normal stress distribution depending quadratically only on the longitudinal direction of the contact area

lead to very compact formulas for force and moments at steady state rolling conditions. Figure 1 shows the contact area dimensions.

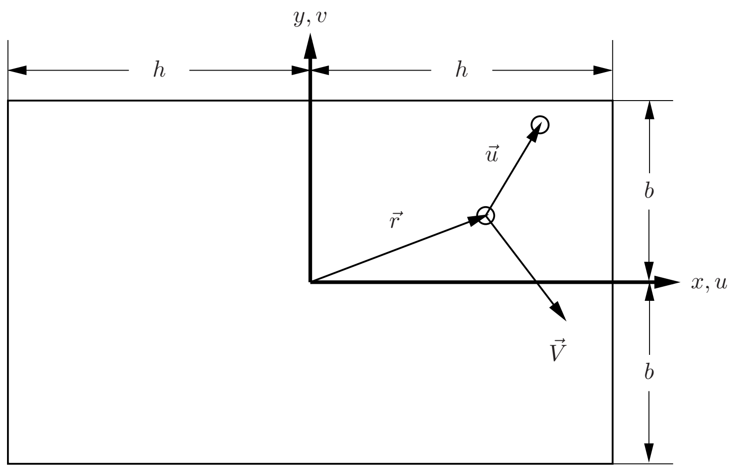


Figure 1: Contact area dimensions and coordinate systems

The differential equations

$$-\frac{\partial u(x,y,t)}{\partial x}\,V_T(x,y,t)+\frac{\partial u(x,y,t)}{\partial t} = V_{s_x}(x,y,t) \tag{1a}$$

$$-\frac{\partial v(x,y,t)}{\partial x}\,V_T(x,y,t)+\frac{\partial v(x,y,t)}{\partial t} = V_{s_y}(x,y,t) \tag{1b}$$

can be rewritten in the steady state case

$$-\frac{\partial u(x)}{\partial x}\,V_T = V_{s_x} \tag{2a}$$

$$-\frac{\partial v(x)}{\partial x}\,V_T = V_{s_y}. \tag{2b}$$

Therein V_T is the transport velocity, which is a measure of the material tire tread velocity. If the wheel is locked, the transport velocity vanishes. In all other cases, in (2a) and (2b) the slip can be introduced by dividing with $V_T = R_{dyn}\,\Omega$. The dynamic rolling radius R_{dyn} is a tire parameter and Ω the angular velocity of the rotation about the rim joint.

$$\frac{\partial u(x)}{\partial x} = -\frac{V_{s_x}}{V_T} = -s_x \tag{3a}$$

$$\frac{\partial v(x)}{\partial x} = -\frac{V_{s_y}}{V_T} = -s_y. \tag{3b}$$

Integration over x of (3a) and (3b) together with the boundary condition $u(x=h)=0$ and $v(x=h)=0$ leads to

$$u(x) = \left(1-\frac{x}{h}\right) h\, s_x \tag{4a}$$

$$v(x) = \left(1-\frac{x}{h}\right) h\, s_y, \tag{4b}$$

simply a linear distribution of displacement with respect to the longitudinal coordinate.

2.2 Normal contact

Figure 2 shows a simple way to get an approximative expression for the contact area length $2\,h$.

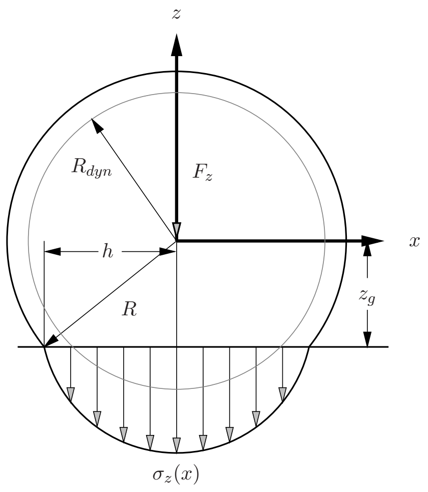


Figure 2: Normal contact and normal stress distribution

Wheel load F_z and radial stiffness c_R as well as the inflated but unloaded tire radius R are used in

$$h = \sqrt{R^2 - \left(R - \frac{F_z}{c_R}\right)^2}. \tag{5}$$

Using the assumption concerning σ_z

$$\sigma_z = \sigma_0 \left(1 - \left(\frac{x}{h}\right)^2\right) \tag{6}$$

the wheel load is

$$F_z = \int\limits_{A_C} \sigma_z\, dA = \int\limits_{-b}^{b}\int\limits_{-h}^{h} \sigma_0 \left(1-\left(\frac{x}{h}\right)^2\right) dx\, dy = 2\,b\,\sigma_0 \int\limits_{-h}^{h} \left(1-\left(\frac{x}{h}\right)^2\right) dx = 2\,b\,\sigma_0\, \frac{4\,h}{3}. \tag{7}$$

(5) and (7) together with the approximation $\sigma_0 = \frac{3\,p_i}{2}$ results in

$$F_z - 4\,b\,p_i \sqrt{R^2 - \left(R - \frac{F_z}{c_R}\right)^2} = 0 \tag{8}$$

and finally[1]

$$b = \frac{F_z}{4\,p_i \sqrt{R^2 - \left(R - \frac{F_z}{c_R}\right)^2}}. \tag{9}$$

1 Alternatively, b could be a model parameter, then σ_0 can be calculated from equation 7.

2.3 Parameter estimation - slip stiffness

Under the assumption of total sticking, the longitudinal and lateral force (see (4a) and (4b)) are obtained by

$$F_{x_{ts}} = \int_{A_C} \tau_x \, dA = \int_{-b}^{b} \int_{-h}^{h} c_x \, u(x) \, dx \, dy = 2\,h\,b\,c_x\,s_x \int_{-h}^{h} \left(1 - \frac{x}{h}\right) dx = 2\,h\,b\,c_x\,s_x\,2\,h = 4\,h^2\,b\,c_x\,s_x = c_{s_x}\,s_x \quad (10a)$$

$$F_{y_{ts}} = \int_{A_C} \tau_y \, dA = \int_{-b}^{b} \int_{-h}^{h} c_y \, v(x) \, dx \, dy = 2\,h\,b\,c_y\,s_y \int_{-h}^{h} \left(1 - \frac{x}{h}\right) dx = 2\,h\,b\,c_y\,s_y\,2\,h = 4\,h^2\,b\,c_y\,s_y = c_{s_y}\,s_y. \quad (10b)$$

Substituting (5) and (9) in this result, an estimate to get $c_{x,y}$ from the gradient around zero slip $F_{x,y}/s_{x,y}$ is

$$c_{x,y} = \frac{c_{s_{x,y}}\,p_i}{F_z \sqrt{R^2 - \left(R - \frac{F_z}{c_R}\right)^2}} \quad (11)$$

which can be further improved by adding the dependency $c_R = c_R(p_i)$.

2.4 Sticking and non sticking area

The linear equations (10a) and (10b) are only valid in a small region around zero slip. At larger slip, the tangential stresses are limited by friction. The border between sticking and non sticking area (often called sliding area) is given by

$$\mu_0\,\sigma_z(\bar{x}) = \mu_0\,\sigma_0 \left(1 - \left(\frac{\bar{x}}{h}\right)^2\right) = \sqrt{\tau_x^2(\bar{x}) + \tau_y^2(\bar{x})} = h \left(1 - \frac{\bar{x}}{h}\right) \sqrt{c_x^2\,s_x^2 + c_y^2\,s_y^2} \quad (12)$$

wherein μ_0 is the first friction related parameter. The solution of (12)

$$\bar{x} = \frac{h^2}{\mu_0 \sigma_0} \sqrt{c_x^2\,s_x^2 + c_y^2\,s_y^2} - h \quad (13)$$

is limited by h. To get the combined slip case included in the formula, the slip direction

$$\xi = \frac{s_x}{\sqrt{s_x^2 + s_y^2}}, \qquad \eta = \frac{s_y}{\sqrt{s_x^2 + s_y^2}} \quad (14)$$

is introduced. This leads to extended versions of the tangential force and moment calculation including sticking and non sticking parts

$$F_x = 2\,b\,h\,c_x\,s_x \int_{\bar{x}}^{h} \left(1 - \frac{x}{h}\right) dx + 2\,b\,\xi\,\mu_x\,\sigma_0 \int_{-h}^{\bar{x}} \left(1 - \left(\frac{x}{h}\right)^2\right) dx \quad (15a)$$

$$F_y = 2\,b\,h\,c_y\,s_y \int_{\bar{x}}^{h} \left(1 - \frac{x}{h}\right) dx + 2\,b\,\eta\,\mu_y\,\sigma_0 \int_{-h}^{\bar{x}} \left(1 - \left(\frac{x}{h}\right)^2\right) dx \quad (15b)$$

$$M_z = 2\,b\,h\,c_y\,s_y \int_{\bar{x}}^{h} x \left(1 - \frac{x}{h}\right) dx + 2\,b\,\eta\,\mu_y\,\sigma_0 \int_{-h}^{\bar{x}} x \left(1 - \left(\frac{x}{h}\right)^2\right) dx \quad (15c)$$

and integrated

$$F_x = b\,c_x\,(s_x + s_{x0}) \frac{(\bar{x} - h)^2}{2} + b\,\mu_x\,\sigma_0\,\xi \left(\frac{2\,h}{3} - \frac{\bar{x}^3 - 3\,h^2\,\bar{x}}{3\,h^2}\right) \quad (16a)$$

$$F_y = b\,c_y\,(s_y + s_{y0}) \frac{(\bar{x} - h)^2}{2} + b\,\mu_y\,\sigma_0\,\eta \left(\frac{2\,h}{3} - \frac{\bar{x}^3 - 3\,h^2\,\bar{x}}{3\,h^2}\right) \quad (16b)$$

$$M_z = b\,c_y\,(s_y + s_{y0}) \frac{(\bar{x} - h)^2\,(2\,\bar{x} + h)}{6} - b\,\mu_y\,\sigma_0\,\eta \left(\frac{h^2}{4} + \frac{\bar{x}^4 - 2\,h^2\,\bar{x}^2}{4\,h^2}\right). \quad (16c)$$

The friction values μ_x and μ_y are in the basic form of the formula independent from σ_z

$$\mu_x = \mu_0 + (\mu_{1_x} - \mu_0)\,(1 - e^{-\left|(\sqrt{s_x^2+s_y^2}/c_{\mu_x})\right|}) \tag{17a}$$
$$\mu_y = \mu_0 + (\mu_{1_y} - \mu_0)\,(1 - e^{-\left|(\sqrt{s_x^2+s_y^2}/c_{\mu_y})\right|}). \tag{17b}$$

Another two parameters s_{x_0} and s_{y_0} are used to match nonzero forces at zero slip, which result from ply-steer effects. With (13,14) and (17a,17b) in (16a,16b,16c), three coupled closed formulas with physical meaningful parameters describe the steady state tire behaviour, which can be used in realtime models or in stability investigations.

2.5 Linearisation

Linearisation of for instance (16a) has to be done with respect to $\bar{x}$. If $\bar{x}$ has reached the limit h, (16a) simplifies to

$$F_x = b\,\mu_x\,\sigma_0\,\xi\left(\frac{2\,h}{3} - \frac{h^3 - 3\,h^3}{3\,h^2}\right) = b\,\mu_x\,\sigma_0\,\xi\,\frac{4\,h}{3} \tag{18}$$

and in the case of $\xi = 1, \eta = 0, s_x > 0$ the linearisation is

$$F_x(s_{x_0} + \Delta s_x) = F_{s_0} + \frac{4\,h\,b\,\sigma_0}{3}\left(\frac{\partial \mu_x}{\partial s_x}\right)\bigg|_{s_{x_0}} \Delta s_x = F_{s_0} + \frac{4\,h\,b\,\sigma_0}{3}\left(\frac{\mu_{1_x} - \mu_0}{c_{\mu_x}}\right) e^{-\frac{s_{x_0}}{c_{\mu_x}}}\,\Delta s_x. \tag{19}$$

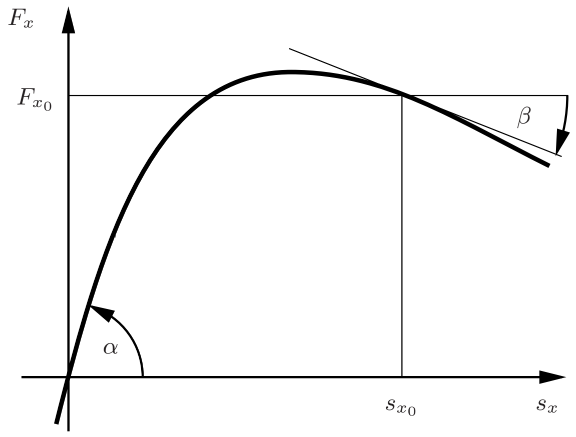


Figure 3: Linearisation of F_x

If $\mu_{1_x} < \mu_0$ holds, the gradient of F_x in s_{x_0} is negative and may cause a loss of stability. Because the slip s_x depends on the degrees of freedom $\underline{q}$ of a vehicle model (longitudinal vehicle velocity and rim angular velocity are involved), the damping matrix of such a system will be influenced by the tire parameters. Then, by the dependency

$$\frac{\partial F_x}{\partial \underline{q}} = \left(\frac{\partial F_x}{\partial s_x}\right)\left(\frac{\partial s_x}{\partial \underline{q}}\right) \tag{20}$$

the linearisation has to be carried out to get the additional damping matrix elements. It should be pointed out here, that additional elements to the stiffness matrix of a mechanical system have to a determined under different assumptions.

2.6 Results

To be able to deal which asymmetric tire forces, the friction parameters $\mu_{1_{x,y}}$ and $c_{\mu_{x,y}}$ are extended to a set for positive and a set for negative slip values. In this first version, 13 parameters have to be determined, this is done by optimization. An optimization method without constraints has been used. R, c_R, p_i and μ_0 have been not included in the set of design variables since they are to be understood as known fixed values. One possible extension is a dependency of c_x and c_y on the wheel load F_z which may be also applied to the friction parameters. An example is

$$c_x = c_{x_0} + c_{x_1}\, F_z, \quad c_y = c_{y_0} + c_{y_1}\, F_z \tag{21}$$

adding two new parameters to the set of design variables. Further improvement can be achieved by load depended friction properties. Load depended friction can be implemented as local or global effect. In the second case there are just two additional parameters, for instance μ_{F_z} and $\bar{F}_z$ in

$$\mu_{0_{F_z}} = \mu_0 \left(1 + \mu_{F_z}\left(\bar{F}_z - F_z\right)\right) \tag{22}$$
$$\mu_{1_{x_{F_z}}} = \mu_{1_x} \left(1 + \mu_{F_z}\left(\bar{F}_z - F_z\right)\right) \tag{23}$$
$$\mu_{1_{y_{F_z}}} = \mu_{1_x} \left(1 + \mu_{F_z}\left(\bar{F}_z - F_z\right)\right) \tag{24}$$
$$c_{\mu_{x_{F_z}}} = c_{\mu_x} \left(1 + \mu_{F_z}\left(\bar{F}_z - F_z\right)\right) \tag{25}$$
$$c_{\mu_{y_{F_z}}} = c_{\mu_y} \left(1 + \mu_{F_z}\left(\bar{F}_z - F_z\right)\right) \tag{26}$$

The target function is defined by

$$Z = \left(\sum_{i=1}^{N_C} \frac{100}{N_C\, N_{P_i}} \left(\sum_{j=1}^{N_{P_i}} \frac{\sqrt{\left(F_{Mea_{ij}} - F_{Sim_{ij}}\right)^2}}{F_{z_{ij}}}\right)\right) \tag{27}$$

wherein N_C denotes the number of reference curves and N_{P_i} the number of reference points per curve. $F_{z_{ij}}$ is used to *normalize* the difference between simulation and reference in order to make the target function dimensionless and independent on the wheel load. Weight factors for individual curves and points are not yet implemented.

The results of the optimisation are shown in figure 4 and 5. In this figure, a tolerance of $\pm 5\%$ is layed around the measurements. The approximation fits the measurements very well. The measurements are based on a 205/65-R16 high performance tire.

Taking into account another load depended effect – the variation of zero slip forces with wheel load – leads also to better values of the target function. This is simply done by

$$s_x = s_{x_I} + s_{x_0} + s_{off}(\frac{F_z}{\bar{F}_z}), \quad s_y = s_{y_I} + s_{y_0} + s_{off}(\frac{F_z}{\bar{F}_z}) \tag{28}$$

Both effects are included in the results of figure 4 and 5. This version of the formula contains 16 design variables and 5 constants.

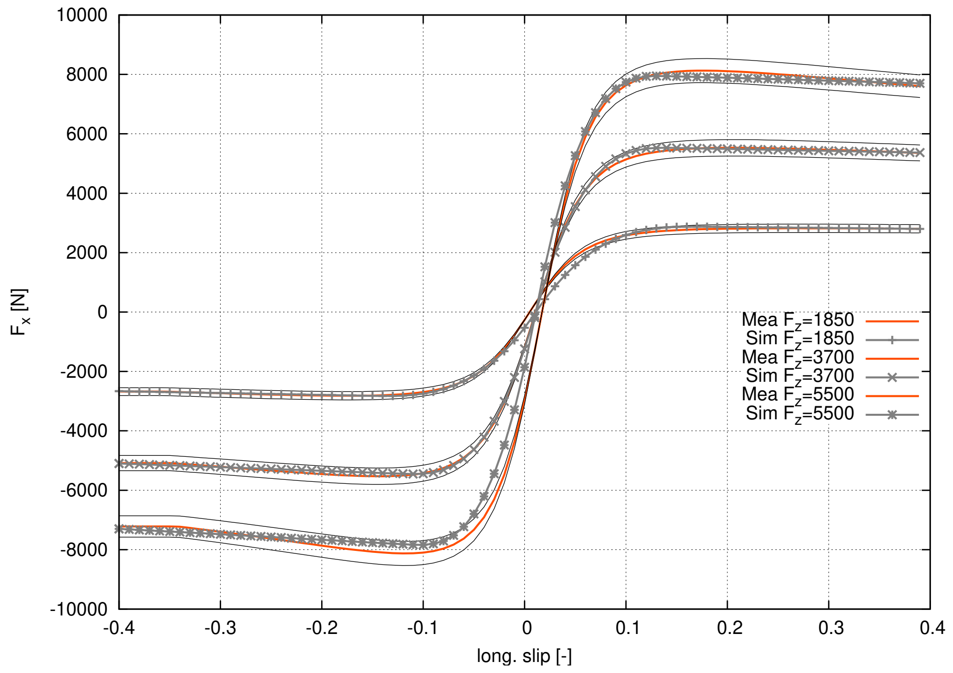


Figure 4: Optimization result F_x

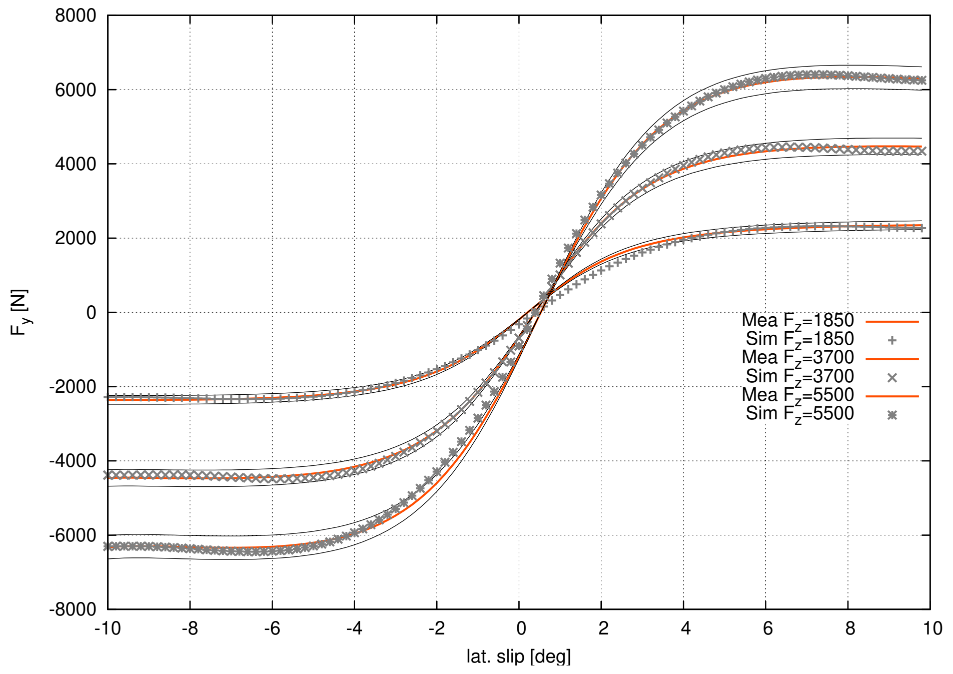


Figure 5: Optimization result F_y

3 Extensions

The turn slip is the third slip input resulting from transient steering or steady state camber. The velocity field including turn slip is

$$V_{sx} = V_{sx_0} - y\,\omega_z \tag{29a}$$
$$V_{sy} = V_{sy_0} + x\,\omega_z \tag{29b}$$

and in consequence, the border between sticking and non sticking part of the contact area is no longer a function of x, but depends on y also. Dealing with camber has another consequence, the contact area shape will no longer be nearly rectangular.

3.1 Segmentation

The contact area is cut into a number N of segments, each with individual length h_j and width Δb located at y_j.

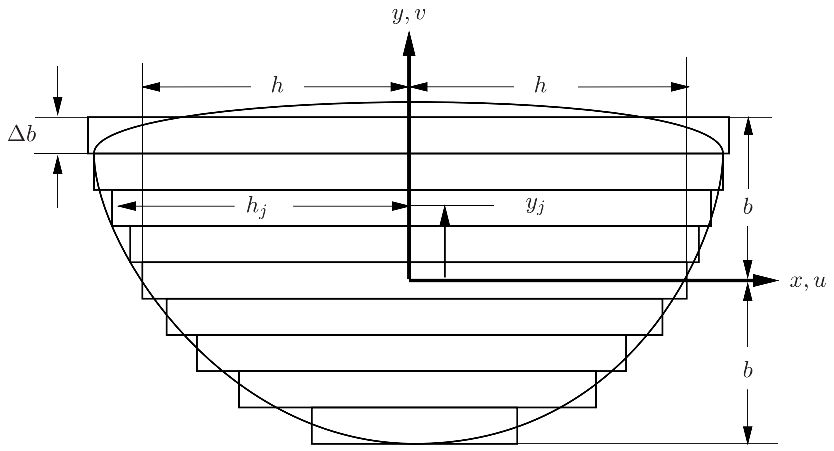


Figure 6: Contact area dimensions and coordinate system

The segment length and width are

$$h_j = h + k_\gamma\,\gamma\,y_j, \quad \Delta b = \frac{2\,b}{N}. \tag{30}$$

The normal stress is also a function of y_j. Each segment has a individual σ_0, which is taken from

$$\sigma_{0_j} = \sigma_0 + k_{\sigma_1}\,\gamma\,y_j + k_{\sigma_2}\,y_j^2. \tag{31}$$

The resulting normal force

$$\hat{F}_z = \sum_{j=1}^{N}\left(\Delta b\,\sigma_{0_j}\int_{-h_j}^{h_j}\left(1-\left(\frac{x}{h_j}\right)^2\right)dx\right) = \sum_{j=1}^{N}\left(\frac{4\,\Delta b}{3}\,\sigma_{0_j}\,\left(h + k_\gamma\,\gamma\,y_j\right)\right) \tag{32}$$

will not equal the given wheel load. Therefore, σ_0 has to be corrected by $F_z/\hat{F}_z$. Due to the modified slip velocity, the tangential displacement in longitudinal direction is

$$u(x, y_j) = h_j\,\left(s_{x_0} - y_j\,s_t\right)\left(1 - \frac{x}{h_j}\right) \tag{33}$$

introducing the turn slip $s_t = \omega_z/(R_{dyn}\Omega)$. In lateral direction, a quadratic component arises from turn slip

$$v(x) = h_j\, s_{y_0} \left(1 - \frac{x}{h_j}\right) + \frac{h_j^2\, s_t}{2} \left(1 - \left(\frac{x}{h_j}\right)^2\right). \tag{34}$$

The boundary of sticking in the case of the segmented contact area follows from

$$\mu_0\, \sigma_{0_j} \left(1 - \left(\frac{\bar{x}_j}{h_j}\right)^2\right) = \sqrt{c_x^2\, u(\bar{x}_j, y_j)^2 + c_y^2\, v(\bar{x}_j)^2} \tag{35}$$

and – using some abbreviation as $\kappa = x/h$ and $a_0 = \mu_0\, \sigma_{0_j}$ – can by rewritten as

$$f(\kappa) = a_0^2\, \left(1-\kappa^2\right)^2 - a_1^2\, (1-\kappa)^2 - \left(a_2\, (1-\kappa) + a_3\, \left(1-\kappa^2\right)\right)^2 = 0. \tag{36}$$

The four roots are

$$\kappa_{1,2} = 1, \quad \kappa_{3,4} = -1 + \frac{a_2\, a_3 \pm \sqrt{a_0^2\, (a_1^2 + a_2^2) - a_1^2\, a_3^2}}{a_0^2 - a_3^2}. \tag{37}$$

In the case of zero longitudinal slip ($a_1 == 0$), (37) simplifies to

$$\kappa_{1,2} = 1, \quad \kappa_{3,4} = -1 + \frac{a_2\, a_3 \pm \sqrt{a_0^2\, a_2^2}}{a_0^2 - a_3^2}. \tag{38}$$

Different cases are as follows

$$\begin{array}{lll} \text{a)} & |\kappa_3| > 1\ \&\ |\kappa_4| > 1 & \kappa_1 = max(\kappa_4, \kappa_3),\ \kappa_2 = min(\kappa_4, \kappa_3) \\ \text{b)} & |\kappa_3| > 1\ \&\ |\kappa_4| < 1 & \kappa = \kappa_4 \\ \text{c)} & |\kappa_3| < 1\ \&\ |\kappa_4| > 1 & \kappa = \kappa_3 \\ \text{d)} & |\kappa_3| > 1\ \&\ |\kappa_4| > 1 & \kappa = h_j \end{array} \tag{39}$$

where case a) addresses the fact, that more then one boundary exists. For instance, the first part of the contact patch is non sticking, a second one is sticking while the third again is non sticking. This can happen at a larger positive camber angle and a small negative yaw angle as illustrated in Figure 7.

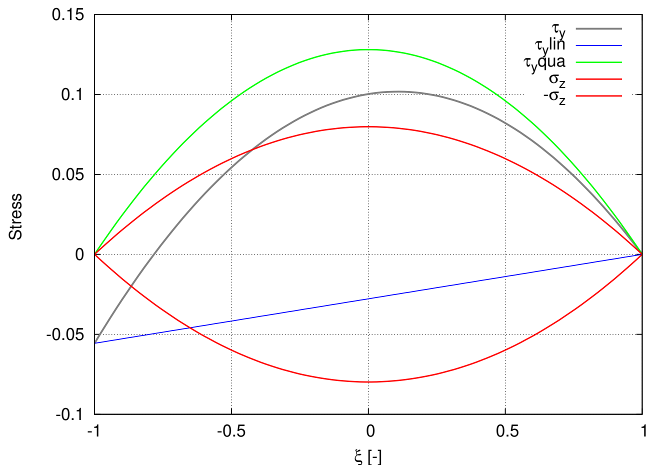


Figure 7: Case a) with two boundaries of sticking area

The direction of slip now becomes a local property

$$\xi(x, y_j) = \frac{s_{x_0} - y_j\, s_t}{\sqrt{(s_{x_0} - y_j\, s_t)^2 + (s_{y_0} + x\, s_t)^2}}, \quad \eta(x, y_j) = \frac{s_{y_0} + x\, s_t}{\sqrt{(s_{x_0} - y_j\, s_t)^2 + (s_{y_0} + x\, s_t)^2}}. \tag{40}$$

In consequence, the integration

$$F_{x_{sl}} = \int_{-h}^{\bar{x}} \xi(x, y_j) \left(1 - \left(\frac{\bar{x}}{h}\right)^2\right) dx \tag{41}$$

has no unique solution. To avoid this, the direction of sliding is approximately taken by

$$\xi(x, y_j) = \frac{s_{x0} - y_j\, s_t}{\sqrt{(s_{x0} - y_j\, s_t)^2 + s_{y0}^2}}, \quad \eta(x, y_j) = \frac{s_{y0}}{\sqrt{(s_{x0} - y_j\, s_t)^2 + s_{y0}^2}} \tag{42}$$

and the tangential contact force are

$$F_x = \sum_{j=1}^{N} \Delta b \left(\int_{\bar{x}_j}^{h_j} \tau_x(x, y_j)\, dx + \xi\, \mu_x\, \sigma_{0_j} \int_{-h_j}^{\bar{x}_j} \left(1 - \left(\frac{\bar{x}}{h_j}\right)^2\right) dx \right) \tag{43a}$$

$$F_y = \sum_{j=1}^{N} \Delta b \left(\int_{\bar{x}_j}^{h_j} \tau_y(x)\, dx + \eta\, \mu_y\, \sigma_{0_j} \int_{-h_j}^{\bar{x}_j} \left(1 - \left(\frac{\bar{x}}{h_j}\right)^2\right) dx \right). \tag{43b}$$

In case a), two sliding areas lead to

$$F_x = \sum_{j=1}^{N} \Delta b \left(\xi\, \mu_x\, \sigma_{0_j} \int_{\bar{x}_{j1}}^{h_j} \left(1 - \left(\frac{\bar{x}}{h_j}\right)^2\right) dx + \int_{\bar{x}_{j2}}^{\bar{x}_{j1}} \tau_x(x, y_j)\, dx + \xi\, \mu_x\, \sigma_{0_j} \int_{-h_j}^{\bar{x}_{j2}} \left(1 - \left(\frac{\bar{x}}{h_j}\right)^2\right) dx \right) \tag{44a}$$

$$F_y = \sum_{j=1}^{N} \Delta b \left(\eta\, \mu_y\, \sigma_{0_j} \int_{\bar{x}_{j1}}^{h_j} \left(1 - \left(\frac{\bar{x}}{h_j}\right)^2\right) dx + \int_{\bar{x}_{j2}}^{\bar{x}_{j1}} \tau_y(x)\, dx + \eta\, \mu_y\, \sigma_{0_j} \int_{-h_j}^{\bar{x}_{j2}} \left(1 - \left(\frac{\bar{x}}{h_j}\right)^2\right) dx \right). \tag{44b}$$

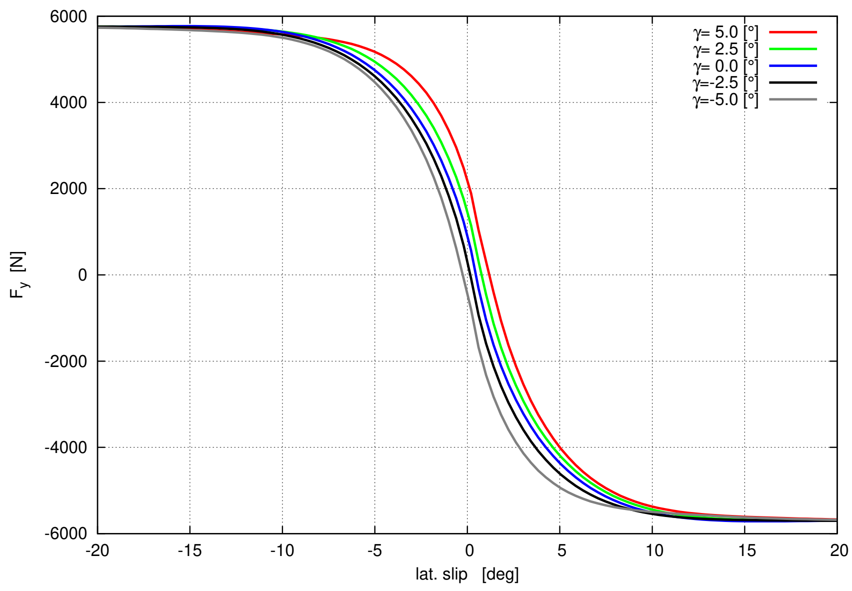


Figure 8: Lateral Force at different camber angles

The results in figure 6 demonstrate the model's reaction to camber inputs, but the basic idea of the formula-approach with closed equations does not work with camber input mainly because of the non unique solutions of (41).

3.2 Discretisation

A second version of the model uses a regular grid with $\Delta h = 2\,h/n_i$ and $\Delta b = 2\,b/n_j$ as *potential* contact area. A normal contact module and a friction module as well as the kinematics of tangential contact form the discrete version of the model. The sticking condition can be tested at every grid point and arbitrary normal stress distributions and contact area shapes can be taken into account. Starting with the simple case similar to the analytical formula, contact area shape remains rectangular and normal stress distribution is a quadratic function of x. Consequently, the results are also very similar when taking the optimized parameters set directly from the optimization with the analytical formula.
A first higher order function to approximate the normal force (instead of stress) distribution is given by

$$F_{z_{ij}}(x_i, y_j) = \Delta b\,\Delta h\,\sigma_{z_{ij}} = F_{0_{ij}} \left(1 - \left(\frac{x_i}{h}\right)^4\right)\left(1 + \left(\frac{y_i}{b}\right)^2\right). \tag{45}$$

This is more closely to a real normal stress distribution known from measurement or FEM analysis like the distribution in the right part of figure 9.

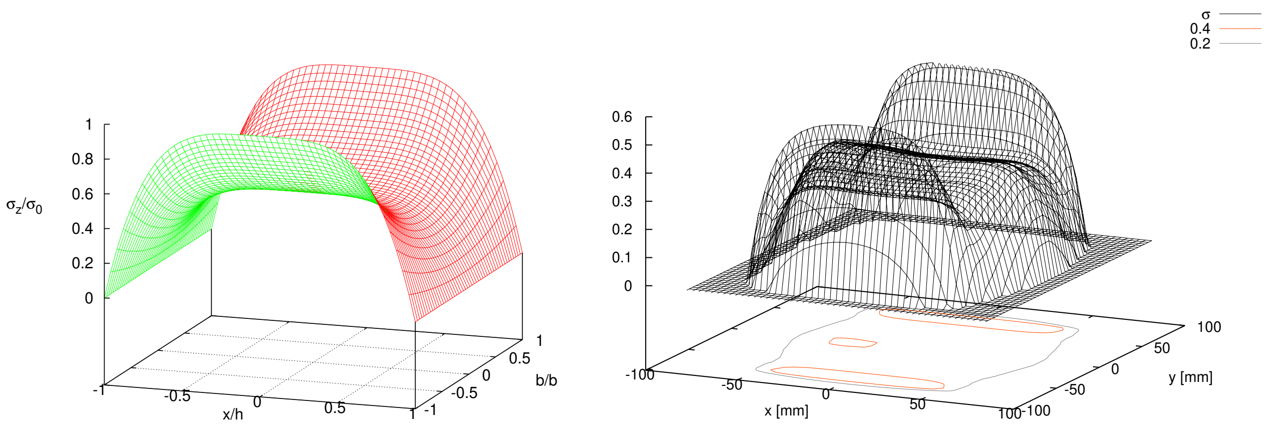


Figure 9: $\sigma_z(x, y)$ approximation and FE results

$F_{0_{ij}}$ has to satisfy

$$F_z = \sum_{i=1}^{n_i}\sum_{j=1}^{n_j} F_{z_{ij}}. \tag{46}$$

Together with the stiffness $\bar{c}_{x,y} = c_{x,y}\,\Delta A$ and $\Delta A = 4\,b\,h\,/(n_i\,n_j)$ with h and b taken from (5) and (9) as estimate

$$F_{x_{st}}(x_i, y_j) = \bar{c}_x\,u(x_i, y_j) = \bar{c}_x\,h\,(s_{x_0} - y_j\,s_t)\left(1 - \frac{x_i}{h}\right) \tag{47}$$

$$F_{y_{st}}(x_i) = \bar{c}_y\,v(x_i) = \bar{c}_y\,h\,s_{y_0}\left(1 - \frac{x_i}{h}\right) + \frac{\bar{c}_y\,h^2\,s_t}{2}\left(1 - \left(\frac{x_i}{h}\right)^2\right) \tag{48}$$

and (45), the grid point sticking condition is

$$\mu_0\,F_{z_{ij}} > \sqrt{F^2_{x_{st}}(x_i, y_j) + F^2_{y_{st}}(x_i)}. \tag{49}$$

If the conditions is not satisfied, the local force direction in the non sticking area is needed to formulate the non sticking tangential point force. This can be done in direction of the slip velocity, in the direction of the sticking force or in the direction of the displacements

$$F_{x_{sl}}(x_i, y_j) = \frac{\mu_x\,u(x_i, y_j)\,F_{z_{ij}}}{\sqrt{u(x_i, y_j)^2 + v(x_i)^2}} \tag{50}$$

$$F_{y_{sl}}(x_i, y_j) = \frac{\mu_y\,v(x_i)\,F_{z_{ij}}}{\sqrt{u(x_i, y_j)^2 + v(x_i)^2}}. \tag{51}$$

To deal with camber, a contact area shape and a related normal force distribution approximation is needed. One possibility is to take that by interpolation from a number of simulations with the FE model. If such data is not available, other sources most be provided.

3.3 Normal contact

A tire like shape is generated by rotating a curved line about the y_f axis and cutting the resulting figure at $\pm b$ as shown in figure 10 with gray lines. Two additional values describe the shape, the lateral radius R_y and the the lateral direction curvature exponent in

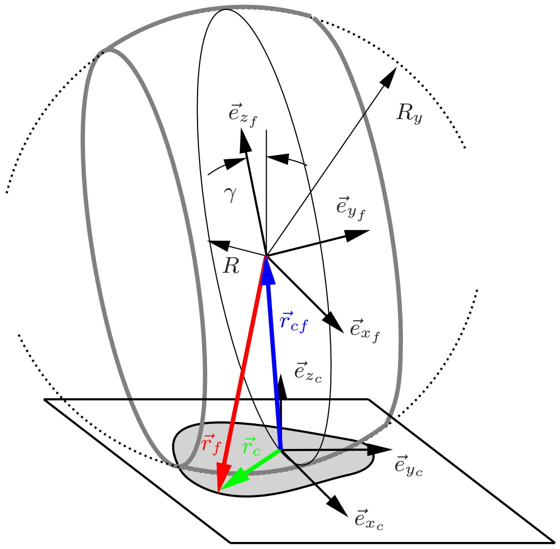


Figure 10: Shapemodel

$$\left(\frac{x_f}{R}\right)^2 + \left(\frac{z_f}{R}\right)^2 + \left(\frac{y_f}{R_y}\right)^{n_y} - 1 \;=\; 0. \tag{52}$$

Different tire shapes from normal passenger car tires to super single truck tires are covered by adjusting the two values R_y and n_y. The vector bases in figure 10 belong to the figure $< \vec{e}_{x_f}, \vec{e}_{y_f}, \vec{e}_{z_f} >$ and to the ground area $< \vec{e}_{x_c}, \vec{e}_{y_c}, \vec{e}_{z_c} >$. The figure fixed base is rotated about $\vec{e}_{x_c}$ with the camber angle γ. The intersection between the figure and a flat ground is an approximation for the contact patch area shape. Since the intersection is described by a function $G(x_c, y_c)$, the sign of the function is used to determine if a point in the *potential* contact area belongs to the *true* contact area or not. Points outside the *true* contact area are assigned $F_{z_{ij}} = 0$ and therefore the contact area shape is implicit given by $F_{z_{ij}}(x_i, y_j) > 0$. Solving the shape equation for z_f, only a solution $z_f < 0$ is of interest and it is

$$z_f = -\sqrt{R^2 - x_f^2 - R^2\left(\frac{y_f}{R_y}\right)^{n_y}}. \tag{53}$$

Arbitrary points on figure and ground are described by

$$\vec{r}_f \;=\; x_f\,\vec{e}_{x_f} + y_f\,\vec{e}_{y_f} + \left(-\sqrt{R^2 - x_f^2 - R^2\left(\frac{y_f}{R_y}\right)^{n_y}}\right)\vec{e}_{z_f} \tag{54}$$

$$\vec{r}_c \;=\; x_c\,\vec{e}_{x_c} + y_c\,\vec{e}_{y_c} \tag{55}$$

and the intersection follows from

$$(\vec{r}_f + \vec{r}_{fc} - \vec{r}_c)\cdot\vec{e}_{z_c} = 0 \tag{56}$$

wherein $\vec{r}_{fc} \cong (R - F_z/c_R)\,\vec{e}_{z_c}$ (small camber angles). The relation between the vector bases is

$$\vec{e}_{x_f} = \vec{e}_{x_c}, \quad \vec{e}_{y_f} = \cos\gamma\,\vec{e}_{y_c} + \sin\gamma\,\vec{e}_{z_c}, \quad \vec{e}_{z_f} = -\sin\gamma\,\vec{e}_{y_c} + \cos\gamma\,\vec{e}_{z_c} \tag{57}$$

leading to

$$\begin{aligned}\vec{r}_f = x_f\,\vec{e}_{x_c} \;&+\; \left(y_f\,\cos\gamma + \sin\gamma\left(\sqrt{R^2 - x_f^2 - R^2\left(\frac{y_f}{R_y}\right)^{n_y}}\right)\right)\vec{e}_{y_c}\\ &+\; \left(y_f\,\sin\gamma - \cos\gamma\left(\sqrt{R^2 - x_f^2 - R^2\left(\frac{y_f}{R_y}\right)^{n_y}}\right)\right)\vec{e}_{z_c}\end{aligned} \tag{58}$$

and after substituting in (56) the border function is

$$G(x_f, y_f) = y_f \, \sin\gamma - \cos\gamma \left(\sqrt{R^2 - x_f^2 - R^2 \left(\frac{y_f}{R_y}\right)^{n_y}} \right) + (R - F_z/c_R) \,. \tag{59}$$

A point on the border line of the contact area is

$$\begin{array}{rcllll} \vec{r}_b & = & x_{c_b} \, \vec{e}_{x_c} & + y_{c_b} \, \vec{e}_{y_c} & & \\ & = & x_{c_b} \, \vec{e}_{x_f} & + y_{c_b} \, \cos\gamma \, \vec{e}_{y_f} & & + y_{c_b} \, \sin\gamma \, \vec{e}_{z_f} \\ & = & x_{f_b} \, \vec{e}_{x_f} & + y_{f_b} \, \vec{e}_{y_f} & + & (z_{f_b} - R + F_z/c_R) \, \vec{e}_{z_f} \end{array} \tag{60}$$

which finally yields the contact indicator function

$$G(x_i, y_j) = \cos\gamma \left(y_j \, \sin\gamma - \sqrt{R^2 \left(1 - \left(\frac{y_j \, \cos\gamma}{R_y}\right)^{n_y}\right) - x_i^2} \right) + (R - F_z/c_R) \,, \tag{61}$$

points with $G(x_i, y_j) < 0$ are inside the contact area.

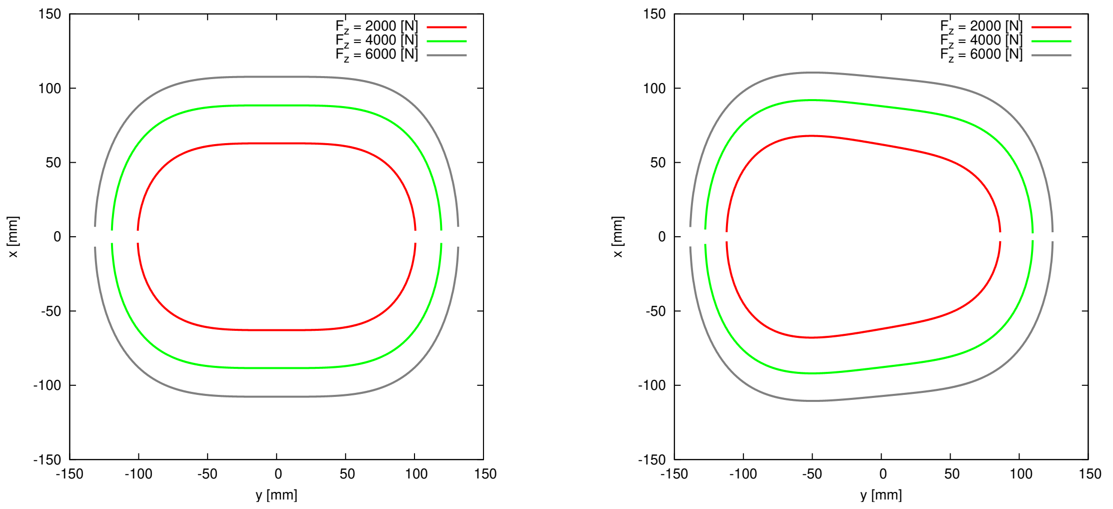


Figure 11: Contact area with $\gamma = 0$ and $\gamma = 3°$

The border between inside and outside in Figure 11 depends on wheel load and camber angle. Setting $G(x_i, y_j) = 0$ and solving for x_i results in a formula for the length h_i of a gridline with $y_j = const.$ and $\bar{z} = R - F_z/c_R$

$$h_j = \frac{\sqrt{-R_y^{n_y} \, (y_j \, \cos\gamma \, \sin\gamma \, (y_j \, \cos\gamma \, \sin\gamma + 2\,\bar{z}) - R^2 \, \cos^2\gamma + \bar{z}^2) - y_j^{n_y} \, R^2 \, (\cos\gamma)^{n_y+2}}}{\sqrt{R_y^{n_y}} \, \cos\gamma} \tag{62}$$

or in the case of $\gamma = 0$

$$h_j = \frac{\sqrt{R_y^{n_y} \, (R^2 - \bar{z}^2) - y_j^{n_y} \, R^2}}{\sqrt{R_y^{n_y}}}, \tag{63}$$

a result similar to that of (5) when $y_j = 0$. The maximum value of all $h_j(y_j)$ denotes the length of the *potential* contact area, but there is no such equation for the maximum width in the case of arbitrary n_y. Instead of an analytical solution, b is obtained by numerical solution of $f(y_j) = G(x_i = 0, y_j) = 0$. The *potential* contact area is then given by a rectangular area while the *true* contact area is described by $G(x_i, y_j) < 0$. The flexibility of this approach is demonstrated by figure 12 and 13, where the influence of the parameters R_y, n_y and R, c_R at constant wheel load and constant camber angle is shown. If footprints of a tire at several wheel loads and

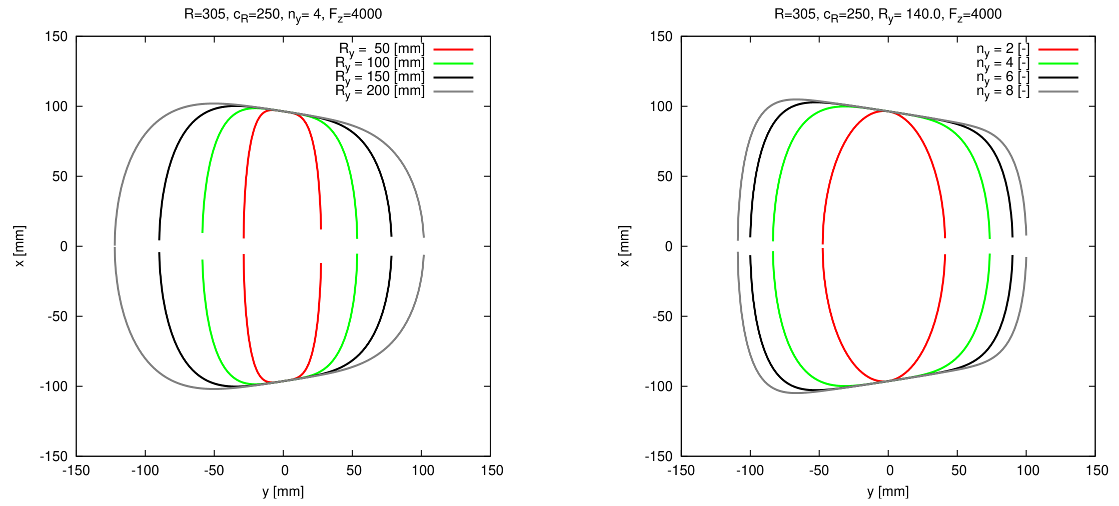


Figure 12: Contact area, Variation of R_y and n_y and $\gamma = 3°$

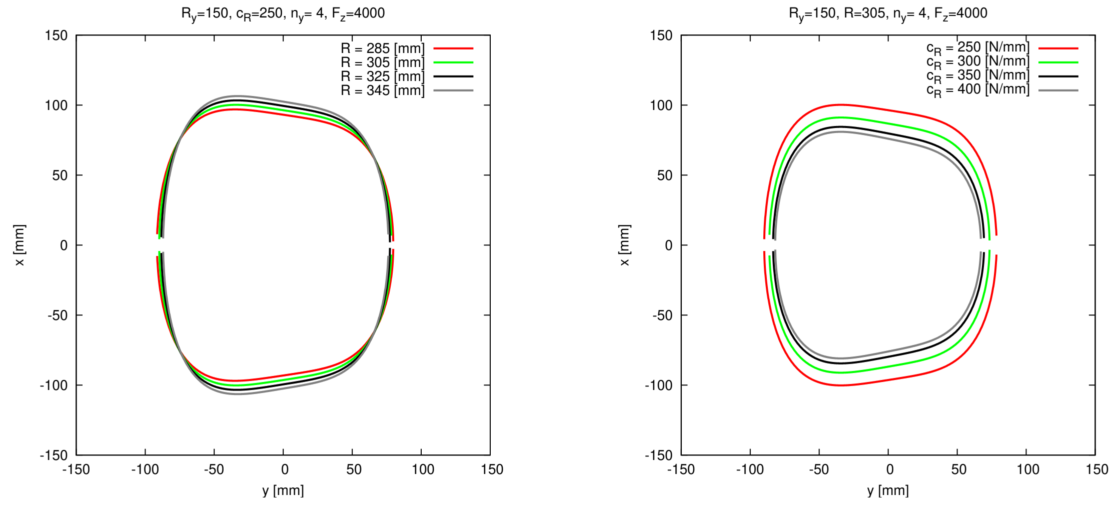


Figure 13: Contact area, Variation of R and c_R and $\gamma = 3°$

camber conditions are known from measurement or from FE results, parameter R_y and n_y can be found by comparison between the measurements and model results.
The normal contact model has to be completed by a normal force distribution as function of wheel load and camber. The approximation (45) is replaced by

$$F_{z_{ij}}(x_i, y_j) = F_{0_{ij}} \left(1 - \left(\frac{x_i}{h}\right)^4\right) \left(1 + a_{F_{z_1}} \left(\frac{y_j}{b_i}\right)^2 - (a_{F_{z_1}} + 1) \left(\frac{y_j}{b_i}\right)^6\right) \left(1 - a_{F_{z_2}} \sin(\gamma) \left(\frac{y_i}{b_i}\right)\right) \tag{64}$$

to cover the influence of camber. b_i is the *true* contact area width at the line with constant x_i. This approximation with $a_{F_{z_1}} = 2$ and $a_{F_{z_2}} = 5$ is able to reproduce FE results (Figure 14) quite good and is also adjustable via $a_{F_{z_1}}$ and $a_{F_{z_2}}$ to given stress or normal force distributions.
The results of the optimization in figures 15 to 17 show a good agreement between measurement and simulation. The self aligning torque M_z was not included in the target function, but the results are reasonable also. Parameters like R_y and n_y were held fixed in the optimization, because this parameters result from the comparison between footprints and the model's contact area shape. Only stiffness parameters, offset parameters and friction parameters are regarded as design variables.

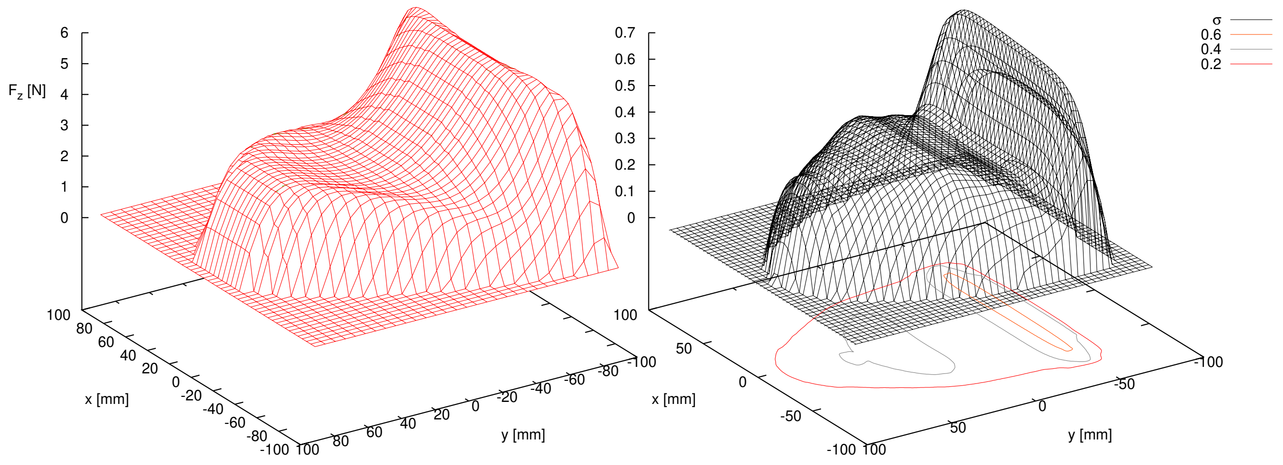


Figure 14: $F_{z_{ij}}$ approximation and $\sigma_z(x, y)$ FE results under $\gamma = 3°$

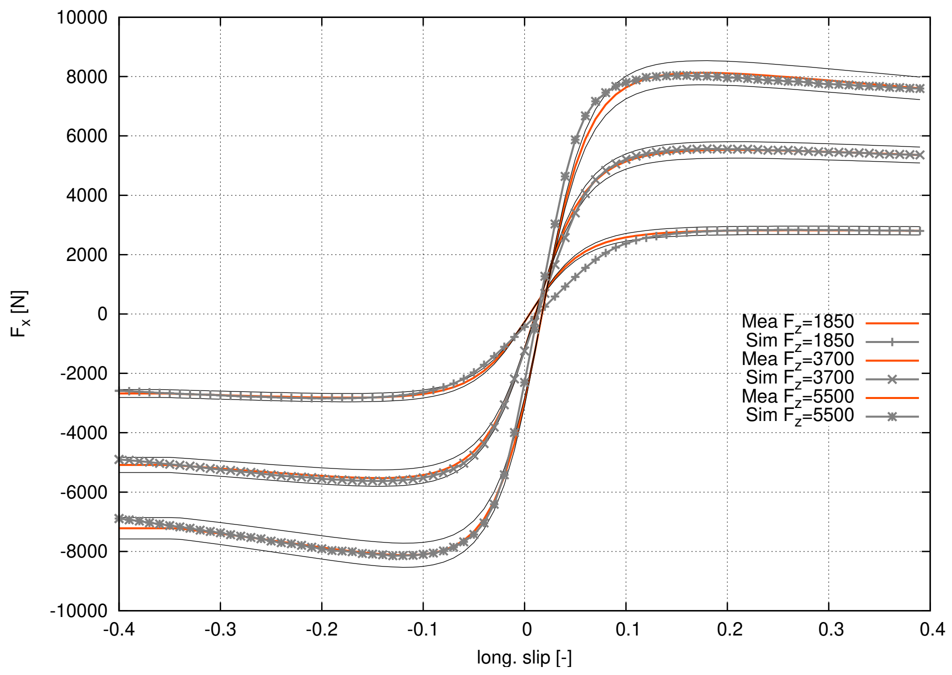


Figure 15: Optimization result F_x, discrete model

3.4 Variation of friction

In the above formulation, the friction functions are separated from each other and given by an exponential function. A more physical related formulation is based on one single function and some additional parameters to cover the influence of tread design. This is done by a piecewise linear function $\mu(V)$, a normal stress depend factor $\mu(\sigma_z)$ and a global load dependency $\mu(F_z)$:

$$\mu(V, \sigma_z) = \mu(v)\,\mu(\sigma_z)\mu(F_z) = \mu\left(\sqrt{V_{\mu_{xy}}^2\, V_x^2 + V_y^2}\right)\frac{1}{2}\left(a_0 + e^{-a_1\,\sigma_z}\right)\left(1 + \mu_{F_z}\left(\bar{F}_z - F_z\right)\right). \tag{65}$$

In the sliding area, the forces are taken in the direction of the sticking forces

$$F_{x_{sl}}(x_i, y_j) \quad = \quad \mu\, F_{z_{ij}}\frac{\mu_{xy}\, F_x}{\sqrt{F_x(x_i, y_j)^2 + F_y(x_i)^2}} \tag{66}$$

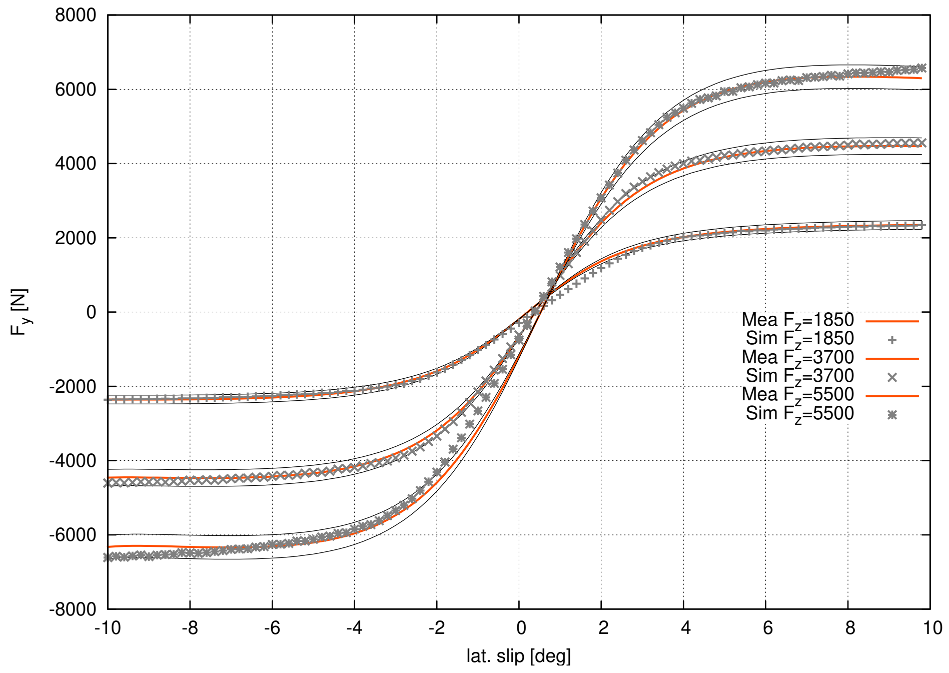


Figure 16: Optimization result F_y, discrete model

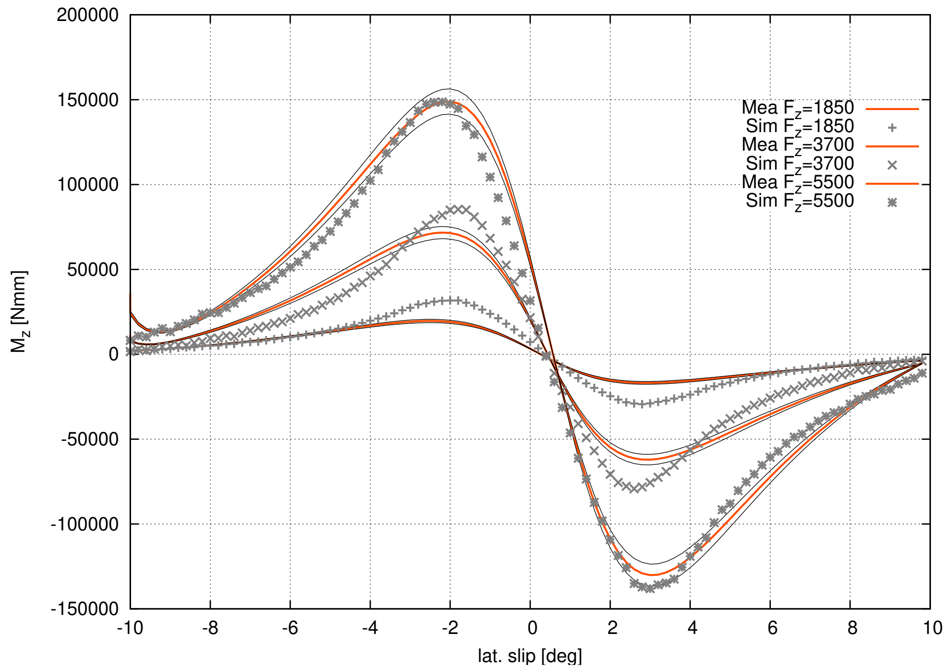


Figure 17: Optimization result M_z, discrete model

$$F_{y_{sl}}(x_i, y_j) \quad = \quad \mu \, F_{z_{ij}} \frac{F_y}{\sqrt{F_x(x_i, y_j)^2 + F_y(x_i)^2}}. \tag{67}$$

No parameters depending on the sign of slip velocity are included in this approach. The initial values concerning the friction related design parameters in the optimization – the samples of $\mu(V)$ – and the final values are given in table 1.

V_s	V_{slide}	$V_{slide}\,10^1$	$V_{slide}\,10^2$	$V_{slide}\,10^3$	$V_{slide}\,10^4$	$V_{slide}\,10^5$
μ_S	1.00	1.00	1.00	1.00	1.00	1.00
μ_F	1.43	1.33	1.27	1.17	1.01	0.88

Table 1: Values of friction

The final values are decreasing with increasing sliding velocity. If only F_x and F_y are taken into account when building the target function in the optimization process, the result may be not unique. Different distribution of local $F_{x_{ij}}$ and $F_{y_{ij}}$ may lead to identical global results F_x and F_y. Therefore, the usage of an optimization method with bounded design variables is strongly recommended. From figures 18 to 20 an improvement is visible compared with the above results.

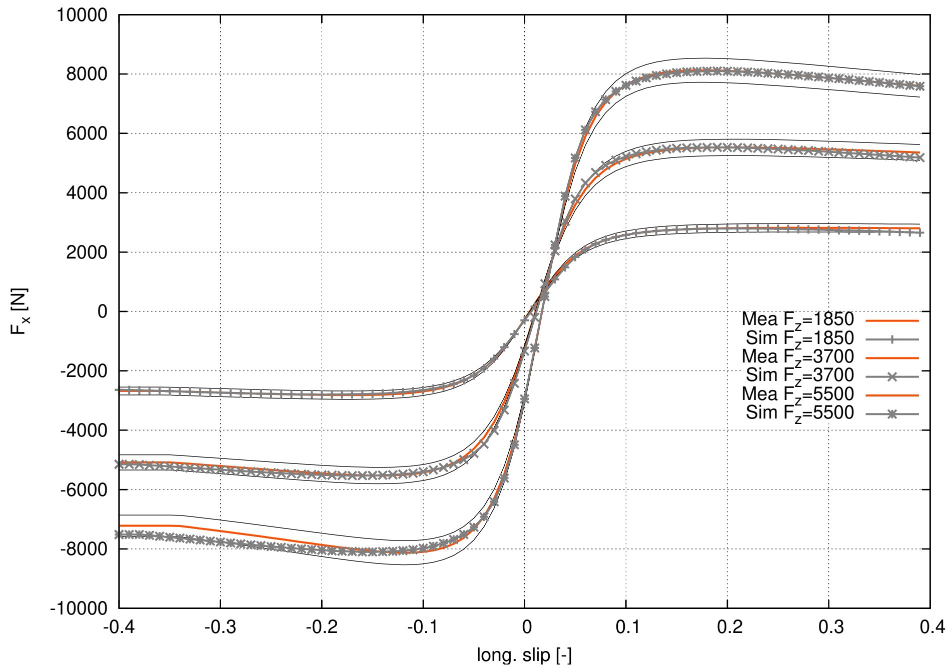


Figure 18: Optimization result F_x, discrete model and data table friction

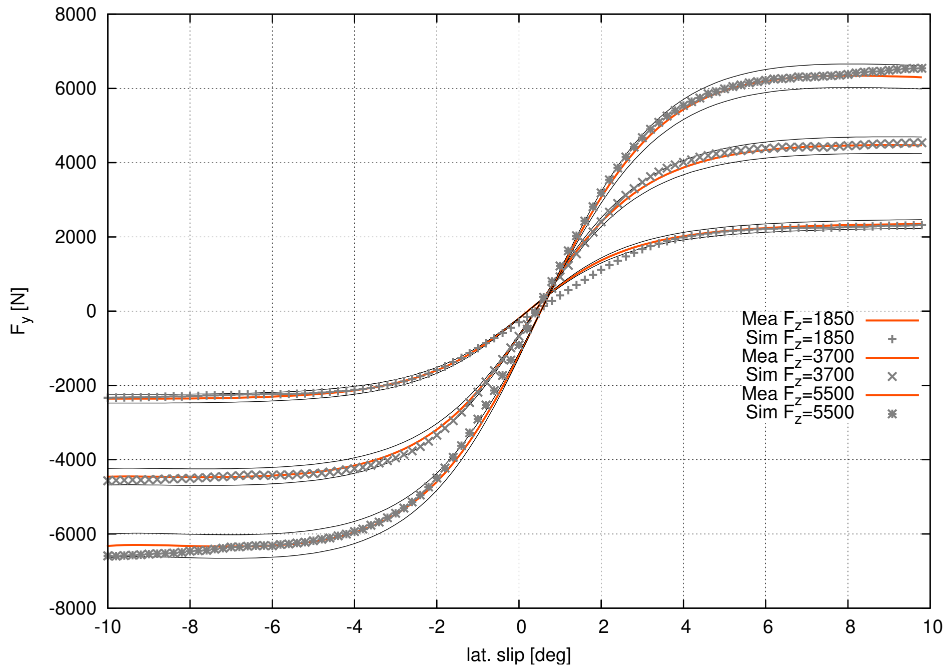


Figure 19: Optimization result F_y, discrete model and data table friction

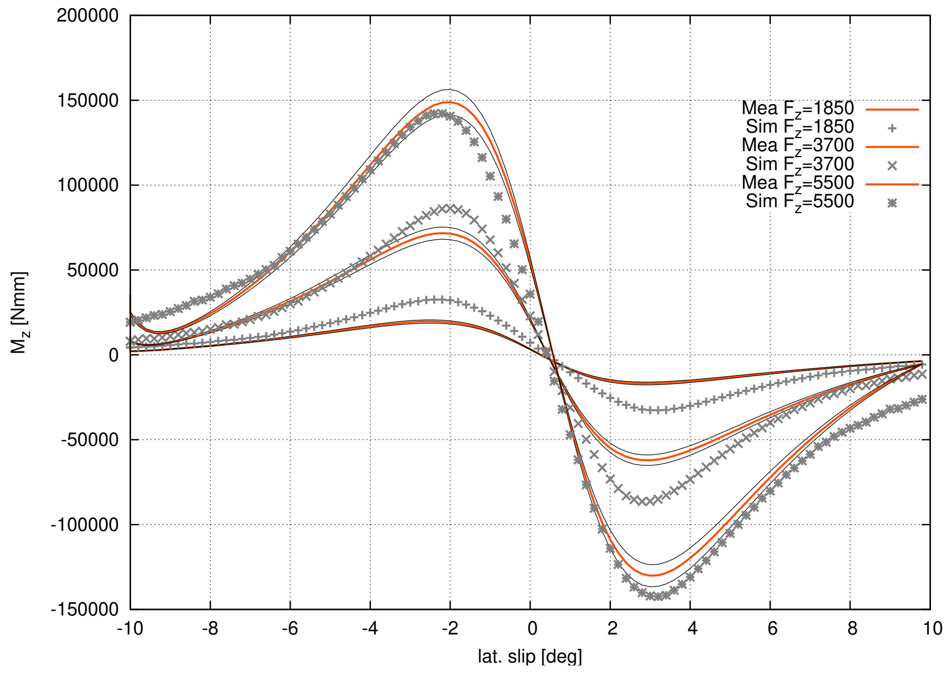


Figure 20: Optimization result M_z, discrete model and data table friction

3.5 Second example

A second tire data set is build for a 255/45-R18 passenger car tire. In the following figures, the results of the formula and discrete model with two friction functions are shown. Small differences between the results and the measurements are in magnitude of measuring accuracy.

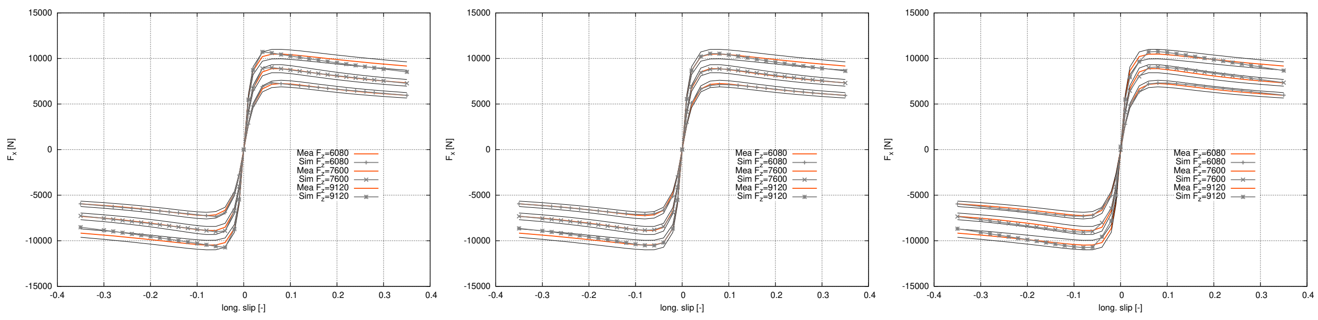


Figure 21: Optimization result F_x, Formula, discrete model with e-friction and table data

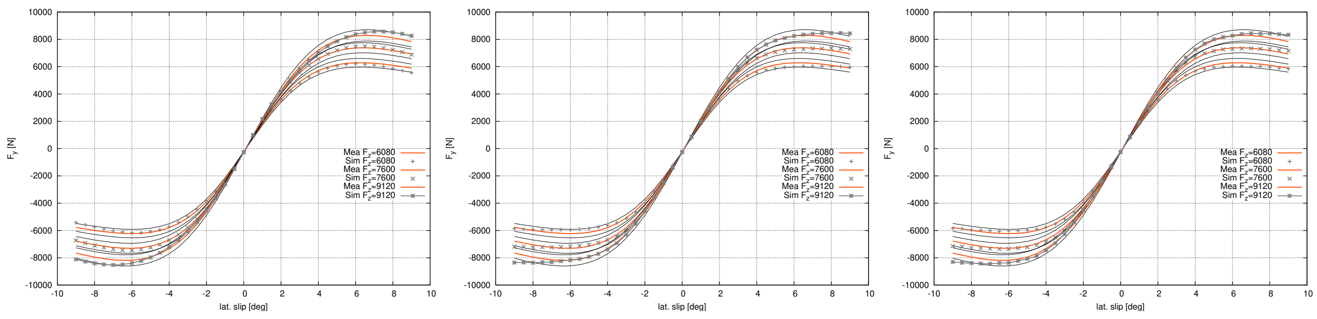


Figure 22: Optimization result F_y, Formula, discrete model with e-friction and table data

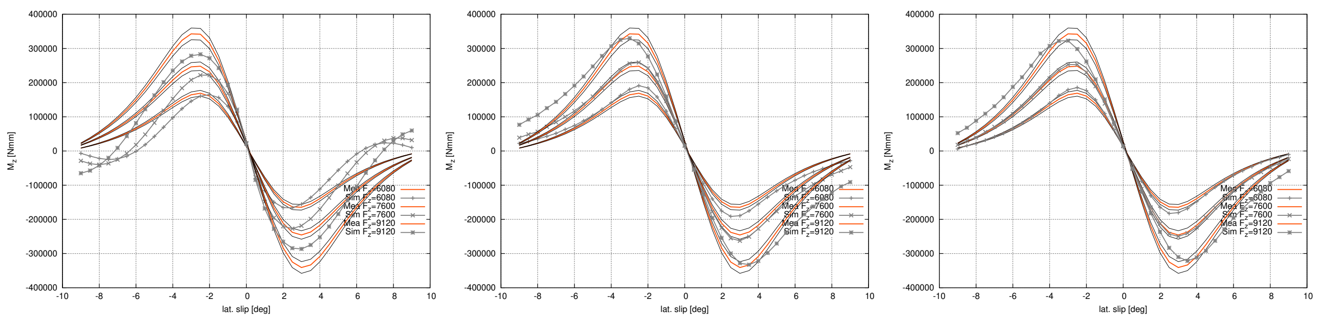


Figure 23: Optimization result M_z, Formula, discrete model with e-friction and table data

M_z was again not included in the optimisation target but the results of the discrete models are quite good, especially at smaller wheel loads with the table data friction function. Based on this data, results in typical combined slip situations with different camber angles (figure 24 and 25) are obtained with the data set from optimisation of the discrete model with table data friction. This shows the model's range of applications as well as the reasonable results under various rolling conditions in a steady state situation.

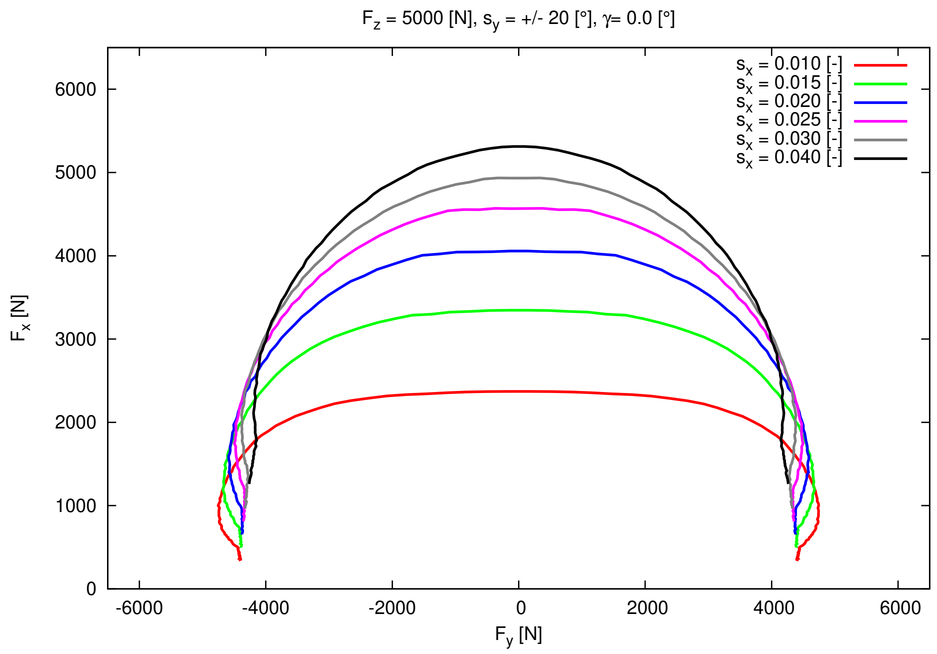


Figure 24: Combined slip result with $\gamma = 0[°]$

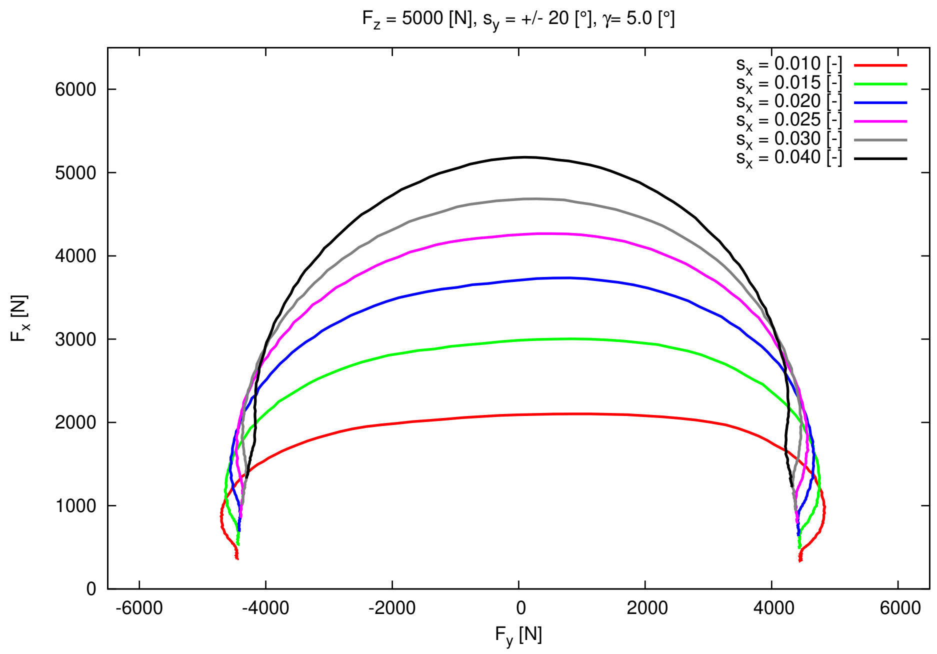


Figure 25: Combined slip result with $\gamma = 5[°]$

4 Softwaremodules

The five software modules can be used either without recompilation or – for instance in the case of operation system or software updates – recompiled. Tested versions are based on `windows 10` and

- Microsoft Visual C++ 2013 / 2017
- Scilab 6.0.2 - 64 bit
- Matlab R2020b - 64 bit.

4.1 Installation

The models described above are implemented in two `c++`-classes, which can by used by several drivers like SCILAB-functions and MATLAB-functions or SIMULINK-S-FUNCTIONS as well as in a stand-alone version. The structure of the software is illustrated in table 2. A graphical user interface (GUI) supports sensitivity analysis and other topics as described below.

object	driver	interface	model
SCILAB	`RMOD_K_Formula_SCILAB`	`RMOD_K_Formula_Opti`	`Discrete/ContinusFormula`
MATLAB	`RMOD_K_Formula_MATLAB`	`RMOD_K_Formula_Opti`	`Discrete/ContinusFormula`
GUI	`Formula-GUI`	`RMOD_K_Formula_Opti`	`Discrete/ContinusFormula`
SIMULINK	`RMOD_K_Formula_S_function`	`RMOD_K_Formula_Run`	`Discrete/ContinusFormula`
stand alone	`RMOD_K_Formula_Main`	`RMOD_K_Formula_Run`	`Discrete/ContinusFormula`

Table 2: Structure of software modules

According to this structure, five directories contain `VisualStudio 20xx` projects needed to build the executable or the libraries. Two projects build dynamic link libraries for parameter optimisation with SCILAB or MATLAB. Another project builds the dynamic link library `S-function` for SIMULINK models like the models in the section applications below. The stand alone executable is compiled and linked by the last project. All MATLAB dll-files have to be renamed to `.mexw64` instead of `.dll`.

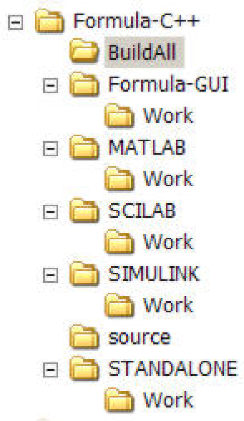


Figure 26: Directories

All projects are based on source code placed in the directory `source`. In an additional directory, the `BuildAll` project is located, intended to run a build of all five applications. If changes on the code of the continuous or discrete version of the formula are made, a new build of all libraries can by performed and the libraries and the executable are directly passed to the `Work`-folders. There some application examples are located, SCILAB or MATLAB files for optimisation and some SIMULINK models of a test rig.
Additional directories and projects may be created due to further development as for instance to build an interface to other optimisation solver like HOPSPACK (Hybrid Optimization Parallel Search PACKage), which solves derivative-free optimization problems [7].
The project files contain a 64-Bit version. Some windows system variables have been used to control the compile and link procedure. The first one points to the SCILAB folder, the second to the MATLAB folder and the third characterises the windows version.

```
- SCILAB_FOLDER = C:\program files\scilab

- MATLAB_FOLDER = C:\program files\matlab

- WIN_SYSTEM    = win64
```

The variables are used for instance in the linker option concerning additional dependencies

```
"\$(MATLAB_FOLDER)\extern\lib\$(WIN_SYSTEM)\microsoft\libmx.lib"
```

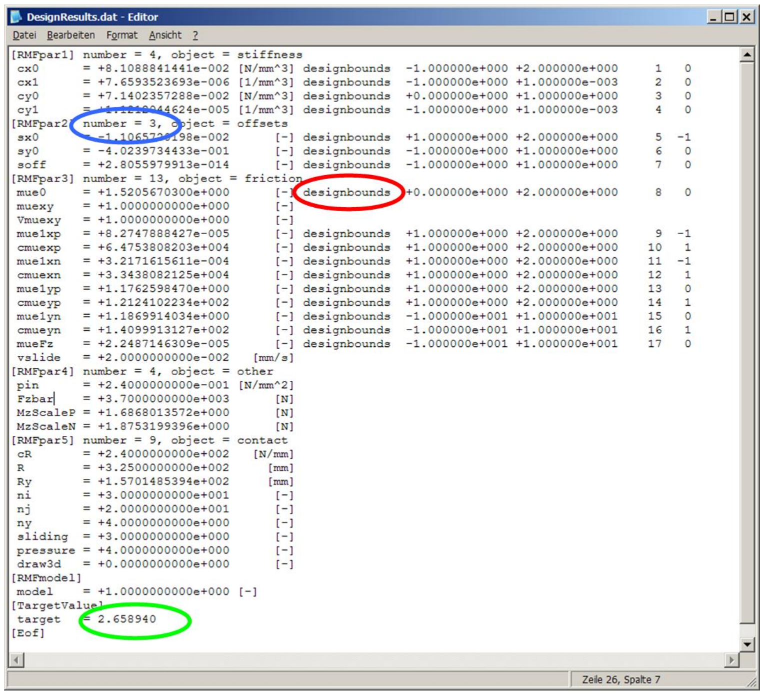
In the computers path, the path to the `scilab`-executables as for instance `wscilex.exe` must be added as well as the path needed to run `matlab`.

4.2 Data files

The drivers are controlled by three file types, the model parameter file, the optimisation target file and run files.

4.2.1 Model data file

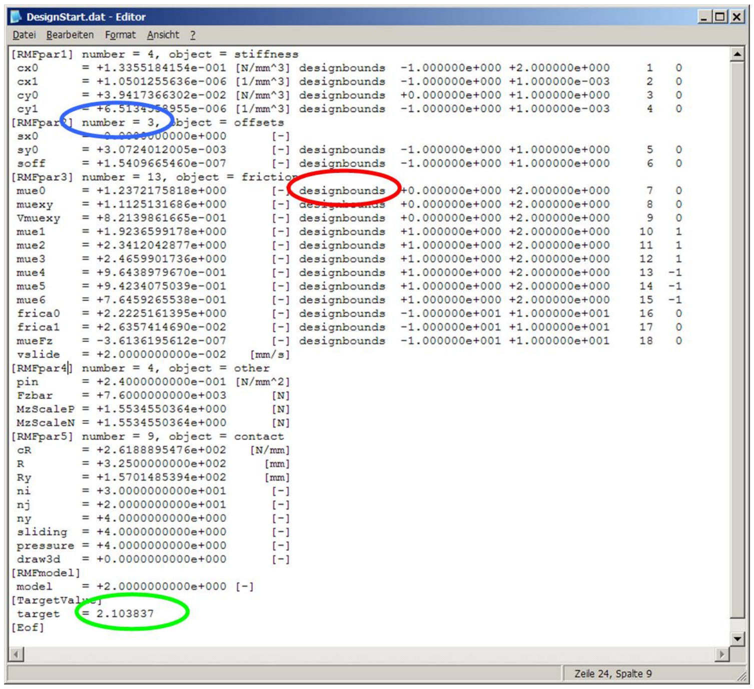
The model data file consists of several data categories, each started with `[RMFparX]`, the number of categories parameters (see blue mark in figure 27) and the parameter name. The name is followed by the `=` sign and the numeric value. If the parameter should by a design variable, then the unit is followed by the identifier **`designbounds`** (see red mark in figure 27) and upper and lower limit.



```
[RMFpar1] number = 4, object = stiffness
 cx0      = +8.1088841441e-002 [N/mm^3] designbounds  -1.000000e+000 +2.000000e+000    1   0
 cx1      = +7.6593523693e-006 [1/mm^3] designbounds  -1.000000e+000 +1.000000e-003    2   0
 cy0      = +7.1402357288e-002 [N/mm^3] designbounds  +0.000000e+000 +1.000000e+000    3   0
 cy1      = +1.1212044624e-005 [1/mm^3] designbounds  -1.000000e+000 +1.000000e-003    4   0
[RMFpar2] number = 3, object = offsets
 sx0      = -1.1065720198e-002       [-] designbounds  +1.000000e+000 +2.000000e+000    5  -1
 sy0      = -4.0239734433e-001       [-] designbounds  -1.000000e+000 +1.000000e+000    6   0
 soff     = +2.8055979913e-014       [-] designbounds  -1.000000e+000 +1.000000e+000    7   0
[RMFpar3] number = 13, object = friction
 mue0     = +1.5205670300e+000       [-] designbounds  +0.000000e+000 +2.000000e+000    8   0
 muexy    = +1.0000000000e+000       [-]
 Vmuexy   = +1.0000000000e+000       [-]
 mue1xp   = +8.2747888427e-005       [-] designbounds  +1.000000e+000 +2.000000e+000    9  -1
 cmuexp   = +6.4753808203e+004       [-] designbounds  +1.000000e+000 +2.000000e+000   10   1
 mue1xn   = +3.2171615611e-004       [-] designbounds  +1.000000e+000 +2.000000e+000   11  -1
 cmuexn   = +3.3438082125e+004       [-] designbounds  +1.000000e+000 +2.000000e+000   12   1
 mue1yp   = +1.1762598470e+000       [-] designbounds  +1.000000e+000 +2.000000e+000   13   0
 cmueyp   = +1.2124102234e+002       [-] designbounds  +1.000000e+000 +2.000000e+000   14   1
 mue1yn   = +1.1869914034e+000       [-] designbounds  -1.000000e+001 +1.000000e+001   15   0
 cmueyn   = +1.4099913127e+002       [-] designbounds  -1.000000e+001 +1.000000e+001   16   1
 mueFz    = +2.2487146309e-005       [-] designbounds  -1.000000e+001 +1.000000e+001   17   0
 vslide   = +2.0000000000e-002   [mm/s]
[RMFpar4] number = 4, object = other
 pin      = +2.4000000000e-001 [N/mm^2]
 Fzbar    = +3.7000000000e+003       [N]
 MzScaleP = +1.6868013572e+000       [N]
 MzScaleN = +1.8753199396e+000       [N]
[RMFpar5] number = 9, object = contact
 cR       = +2.4000000000e+002    [N/mm]
 R        = +3.2500000000e+002      [mm]
 Ry       = +1.5701485394e+002      [mm]
 ni       = +3.0000000000e+001       [-]
 nj       = +2.0000000000e+001       [-]
 ny       = +4.0000000000e+000       [-]
 sliding  = +3.0000000000e+000       [-]
 pressure = +4.0000000000e+000       [-]
 draw3d   = +0.0000000000e+000       [-]
[RMFmodel]
 model    = +1.0000000000e+000 [-]
[TargetValue]
 target   = 2.658940
[Eof]
```

Figure 27: Model data file, discrete model with e-friction

In figure 27, the variables of the discrete model with e-friction are shown. The first parameter category is *stiffness* and the parameters are described in equation (21). The second one is *offsets* and refers to equation (28). The *friction* category parameter meaning depends on the friction model used. If the continuous formula version or the discrete model with table data is addressed, some parameter names are changed or unused. If the continuous version of the formula or the discrete version with e-friction are used, the parameter values for `muexy, Vmuexy` and `vslide` are unused and the remaining values correspond to (17a) and (17b) splitted to positive and negative values. In the category *others*, inflation pressure, $\bar{F}_Z$ and the scaling values of M_z are to be found. Finally, in *contact*, the values used in equation (61) as well as the discretisation (if some) and identifiers for the friction function and normal force distribution are placed. The last one in this category **`draw3d`** controls whether output for 3-dimensional contact patch information is generated or not.
The model data file with friction table data in figure 28 has some different parameters in the *friction* category. The `mue1...6`-values refer to table 1 while the following two parameters `frica0` and `frica1` belong to equation (65).



```
[RMFpar1] number = 4, object = stiffness
 cx0      = +1.3355184154e-001 [N/mm^3] designbounds  -1.000000e+000 +2.000000e+000     1   0
 cx1      = +1.0501255636e-006 [1/mm^3] designbounds  -1.000000e+000 +1.000000e-003     2   0
 cy0      = +3.9417366302e-002 [N/mm^3] designbounds  +0.000000e+000 +1.000000e+000     3   0
 cy1      = +6.5134358955e-006 [1/mm^3] designbounds  -1.000000e+000 +1.000000e-003     4   0
[RMFpar2] number = 3, object = offsets
 sx0      = +0.0000000000e+000      [-]
 sy0      = +3.0724012005e-003      [-] designbounds  -1.000000e+000 +1.000000e+000     5   0
 soff     = +1.5409665460e-007      [-] designbounds  -1.000000e+000 +1.000000e+000     6   0
[RMFpar3] number = 13, object = friction
 mue0     = +1.2372175818e+000      [-] designbounds  +0.000000e+000 +2.000000e+000     7   0
 muexy    = +1.1125131686e+000      [-] designbounds  +0.000000e+000 +2.000000e+000     8   0
 Vmuexy   = +8.2139861665e-001      [-] designbounds  +0.000000e+000 +2.000000e+000     9   0
 mue1     = +1.9236599178e+000      [-] designbounds  +1.000000e+000 +2.000000e+000    10   1
 mue2     = +2.3412042877e+000      [-] designbounds  +1.000000e+000 +2.000000e+000    11   1
 mue3     = +2.4659901736e+000      [-] designbounds  +1.000000e+000 +2.000000e+000    12   1
 mue4     = +9.6438979670e-001      [-] designbounds  +1.000000e+000 +2.000000e+000    13  -1
 mue5     = +9.4234075039e-001      [-] designbounds  +1.000000e+000 +2.000000e+000    14  -1
 mue6     = +7.6459265538e-001      [-] designbounds  +1.000000e+000 +2.000000e+000    15  -1
 frica0   = +2.2225161395e+000      [-] designbounds  -1.000000e+001 +1.000000e+001    16   0
 frica1   = +2.6357414690e-002      [-] designbounds  -1.000000e+001 +1.000000e+001    17   0
 mueFz    = -3.6136195612e-007      [-] designbounds  -1.000000e+001 +1.000000e+001    18   0
 vslide   = +2.0000000000e-002   [mm/s]
[RMFpar4] number = 4, object = other
 pin      = +2.4000000000e-001 [N/mm^2]
 Fzbar    = +7.6000000000e+003      [N]
 MzScaleP = +1.5534550364e+000      [N]
 MzScaleN = +1.5534550364e+000      [N]
[RMFpar5] number = 9, object = contact
 cR       = +2.6188895476e+002   [N/mm]
 R        = +3.2500000000e+002     [mm]
 Ry       = +1.5701485394e+002     [mm]
 ni       = +3.0000000000e+001      [-]
 nj       = +2.0000000000e+001      [-]
 ny       = +4.0000000000e+000      [-]
 sliding  = +4.0000000000e+000      [-]
 pressure = +4.0000000000e+000      [-]
 draw3d   = +0.0000000000e+000      [-]
[RMFmodel]
 model    = +2.0000000000e+000 [-]
[TargetValue]
 target   = 2.103837
[Eof]
```

Figure 28: Model data file, discrete model with table data

4.2.2 Optimisation target file

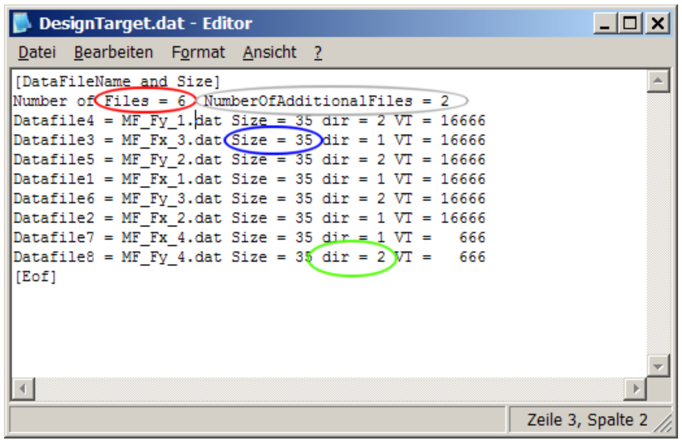
Usually model results and measurement are visualised in order to get an impression of the accuracy – either model or data set – or model results and measurement build the target function of an optimisation. In both cases, a set of measurements or reference data are needed. The subject of the optimisation – the measurements – are described in the design target file named `DesignTarget.dat`. The first line after the identifier gives the number of file to be included in the target function (red mark). One line per each file follows in which the file name, the number of data points (blue mark) and how to build the target function is given. The target function identifier (green mark) can be one, if the longitudinal force difference between model results and measurement shall be added to the target function. If this value is two, the lateral force difference is added while three takes longitudinal and lateral force difference into account, the track data mode. If additional data files – not to be included in the optimisation target evaluation – should be added, the key `NumberOfAdditionalFiles = xx` can be used to define the number of added files (gray mark).

4.2.3 Measurement or reference file

Measurement or reference data files contain the longitudinal slip, the yaw and the camber angle (both in degrees) followed by the longitudinal, lateral and normal force and the self aligning torque, each separated by blanks. Figure 30 shows an example, where only longitudinal slip acts as input. In the track data mode, combined slip situations as well as varying wheel loads may occur – as for instance if measurements from a vehicle should by used for optimisation.

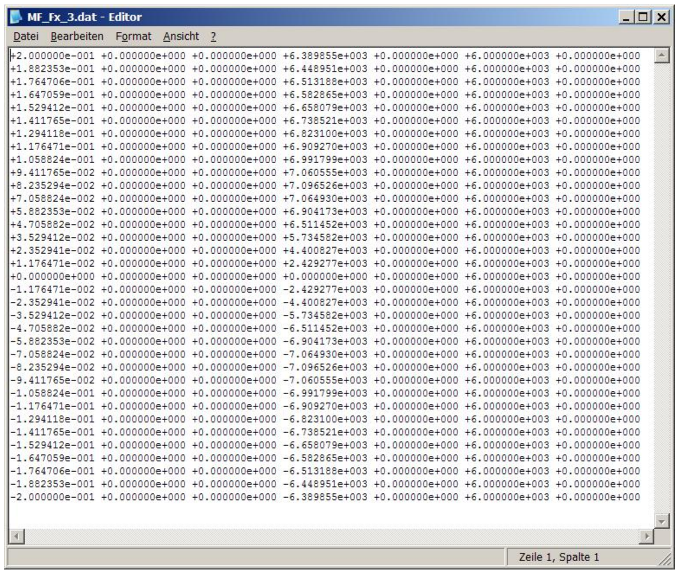
4.2.4 Run file examples

To produce results at different rolling conditions, the range of input values like slip or camber can be varied in some ranges as for instance with the file from figure 31. In the category `DataFileName`, a model data file must be specified (see red mark in figure 31). This is followed by an arbitrary number on ranges which follow the category identifiers `RangeInput`. Number of points to be computed, file name of the output file to be created and ranges of minimal as well a maximal inputs are declared in the lines of the block. For instance, the runs



```
[DataFileName and Size]
Number of Files = 6 NumberOfAdditionalFiles = 2
Datafile4 = MF_Fy_1.dat Size = 35 dir = 2 VT = 16666
Datafile3 = MF_Fx_3.dat Size = 35 dir = 1 VT = 16666
Datafile5 = MF_Fy_2.dat Size = 35 dir = 2 VT = 16666
Datafile1 = MF_Fx_1.dat Size = 35 dir = 1 VT = 16666
Datafile6 = MF_Fy_3.dat Size = 35 dir = 2 VT = 16666
Datafile2 = MF_Fx_2.dat Size = 35 dir = 1 VT = 16666
Datafile7 = MF_Fx_4.dat Size = 35 dir = 1 VT =   666
Datafile8 = MF_Fy_4.dat Size = 35 dir = 2 VT =   666
[Eof]
```

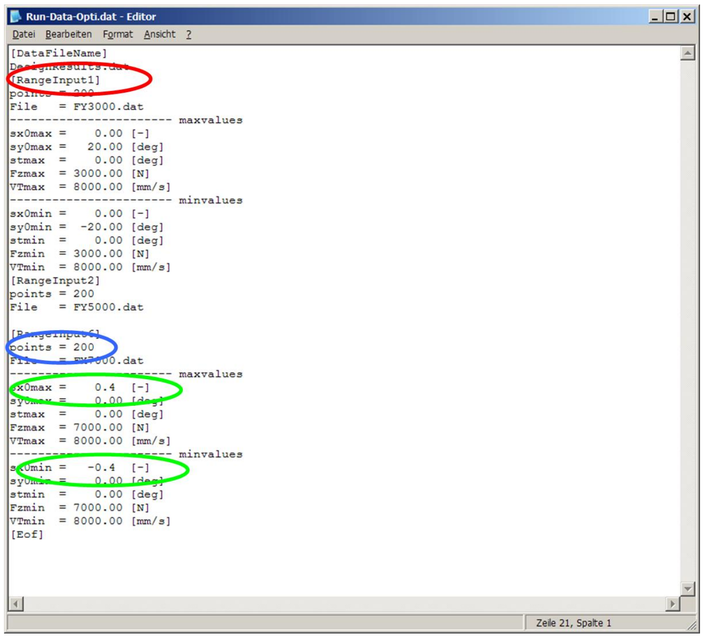
Figure 29: Target data file



```
+2.000000e-001 +0.000000e+000 +0.000000e+000 +6.389855e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
+1.882353e-001 +0.000000e+000 +0.000000e+000 +6.448951e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
+1.764706e-001 +0.000000e+000 +0.000000e+000 +6.513188e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
+1.647059e-001 +0.000000e+000 +0.000000e+000 +6.582865e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
+1.529412e-001 +0.000000e+000 +0.000000e+000 +6.658079e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
+1.411765e-001 +0.000000e+000 +0.000000e+000 +6.738521e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
+1.294118e-001 +0.000000e+000 +0.000000e+000 +6.823100e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
+1.176471e-001 +0.000000e+000 +0.000000e+000 +6.909270e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
+1.058824e-001 +0.000000e+000 +0.000000e+000 +6.991799e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
+9.411765e-002 +0.000000e+000 +0.000000e+000 +7.060555e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
+8.235294e-002 +0.000000e+000 +0.000000e+000 +7.096526e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
+7.058824e-002 +0.000000e+000 +0.000000e+000 +7.064930e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
+5.882353e-002 +0.000000e+000 +0.000000e+000 +6.904173e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
+4.705882e-002 +0.000000e+000 +0.000000e+000 +6.511452e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
+3.529412e-002 +0.000000e+000 +0.000000e+000 +5.734582e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
+2.352941e-002 +0.000000e+000 +0.000000e+000 +4.400827e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
+1.176471e-002 +0.000000e+000 +0.000000e+000 +2.429277e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
+0.000000e+000 +0.000000e+000 +0.000000e+000 +0.000000e+000 +0.000000e+000 +6.000000e+003 +0.000000e+000
-1.176471e-002 +0.000000e+000 +0.000000e+000 -2.429277e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
-2.352941e-002 +0.000000e+000 +0.000000e+000 -4.400827e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
-3.529412e-002 +0.000000e+000 +0.000000e+000 -5.734582e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
-4.705882e-002 +0.000000e+000 +0.000000e+000 -6.511452e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
-5.882353e-002 +0.000000e+000 +0.000000e+000 -6.904173e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
-7.058824e-002 +0.000000e+000 +0.000000e+000 -7.064930e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
-8.235294e-002 +0.000000e+000 +0.000000e+000 -7.096526e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
-9.411765e-002 +0.000000e+000 +0.000000e+000 -7.060555e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
-1.058824e-001 +0.000000e+000 +0.000000e+000 -6.991799e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
-1.176471e-001 +0.000000e+000 +0.000000e+000 -6.909270e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
-1.294118e-001 +0.000000e+000 +0.000000e+000 -6.823100e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
-1.411765e-001 +0.000000e+000 +0.000000e+000 -6.738521e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
-1.529412e-001 +0.000000e+000 +0.000000e+000 -6.658079e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
-1.647059e-001 +0.000000e+000 +0.000000e+000 -6.582865e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
-1.764706e-001 +0.000000e+000 +0.000000e+000 -6.513188e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
-1.882353e-001 +0.000000e+000 +0.000000e+000 -6.448951e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
-2.000000e-001 +0.000000e+000 +0.000000e+000 -6.389855e+003 +0.000000e+000 +6.000000e+003 +0.000000e+000
```

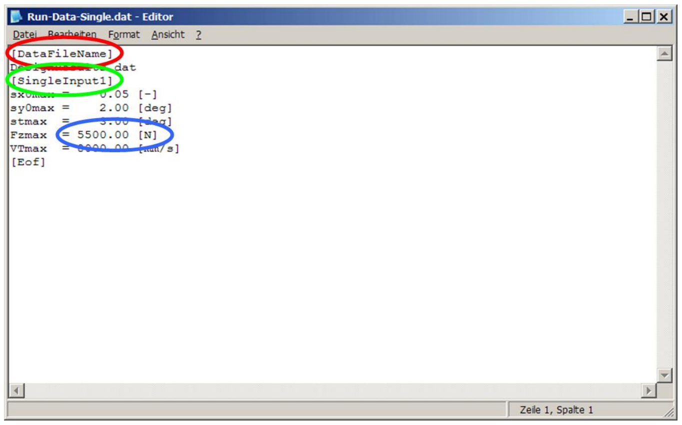
Figure 30: Measurement or reference data file

to prepare the plots in figure 24 have been made with this file: the range value of lateral slip is ±20° while the longitudinal slip, camber, wheel load and translational velocity is held constant (see green marks in figure 31). If only a single run is requested by the identifier `SingleInput`, the data file name and values for longitudinal and lateral slip, camber, wheel load and translational velocity form the input of the model. This may be used to get some insights in the contact patch mechanics, since 3-dimensional plots result from single input runs (see page 34).
The third run file example (figure 33) gets the input of the model from several files and calculates the models reaction (like a function evaluation within an optimisation). Here, the identifier `FileInput` tells about a data file name to follow containing seven columns $(s_x, s_y, s_t, F_x, F_y, F_z, M_z)$ and an arbitrary number of lines. The idea behind this possible input is to run parameter variations and get the comparison between measurements and model results.



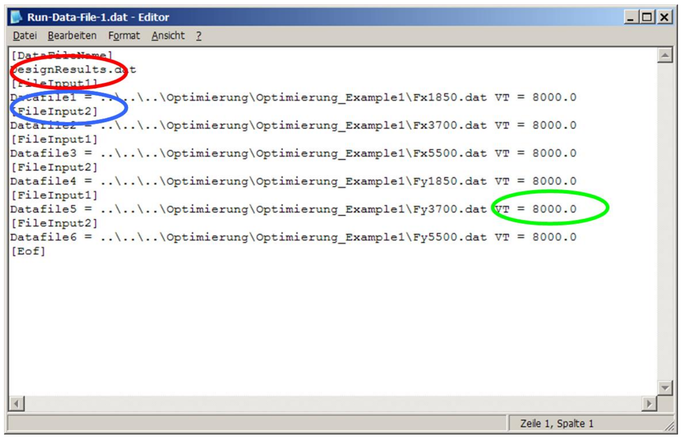
```
[DataFileName]
DesignResults.dat
[RangeInput1]
points = 200
File   = FY3000.dat
----------------------- maxvalues
sx0max =    0.00 [-]
sy0max =   20.00 [deg]
stmax  =    0.00 [deg]
Fzmax  = 3000.00 [N]
VTmax  = 8000.00 [mm/s]
----------------------- minvalues
sx0min =    0.00 [-]
sy0min =  -20.00 [deg]
stmin  =    0.00 [deg]
Fzmin  = 3000.00 [N]
VTmin  = 8000.00 [mm/s]
[RangeInput2]
points = 200
File   = FY5000.dat

[RangeInput6]
points = 200
File   = FY7000.dat
----------------------- maxvalues
sx0max =    0.4  [-]
sy0max =    0.00 [deg]
stmax  =    0.00 [deg]
Fzmax  = 7000.00 [N]
VTmax  = 8000.00 [mm/s]
----------------------- minvalues
sx0min =   -0.4  [-]
sy0min =    0.00 [deg]
stmin  =    0.00 [deg]
Fzmin  = 7000.00 [N]
VTmin  = 8000.00 [mm/s]
[Eof]
```

Figure 31: Range data file



```
[DataFileName]
DesignResults.dat
[SingleInput1]
sx0max =    0.05 [-]
sy0max =    2.00 [deg]
stmax  =    3.00 [deg]
Fzmax  = 5500.00 [N]
VTmax  = 8000.00 [mm/s]
[Eof]
```

Figure 32: Single run data file

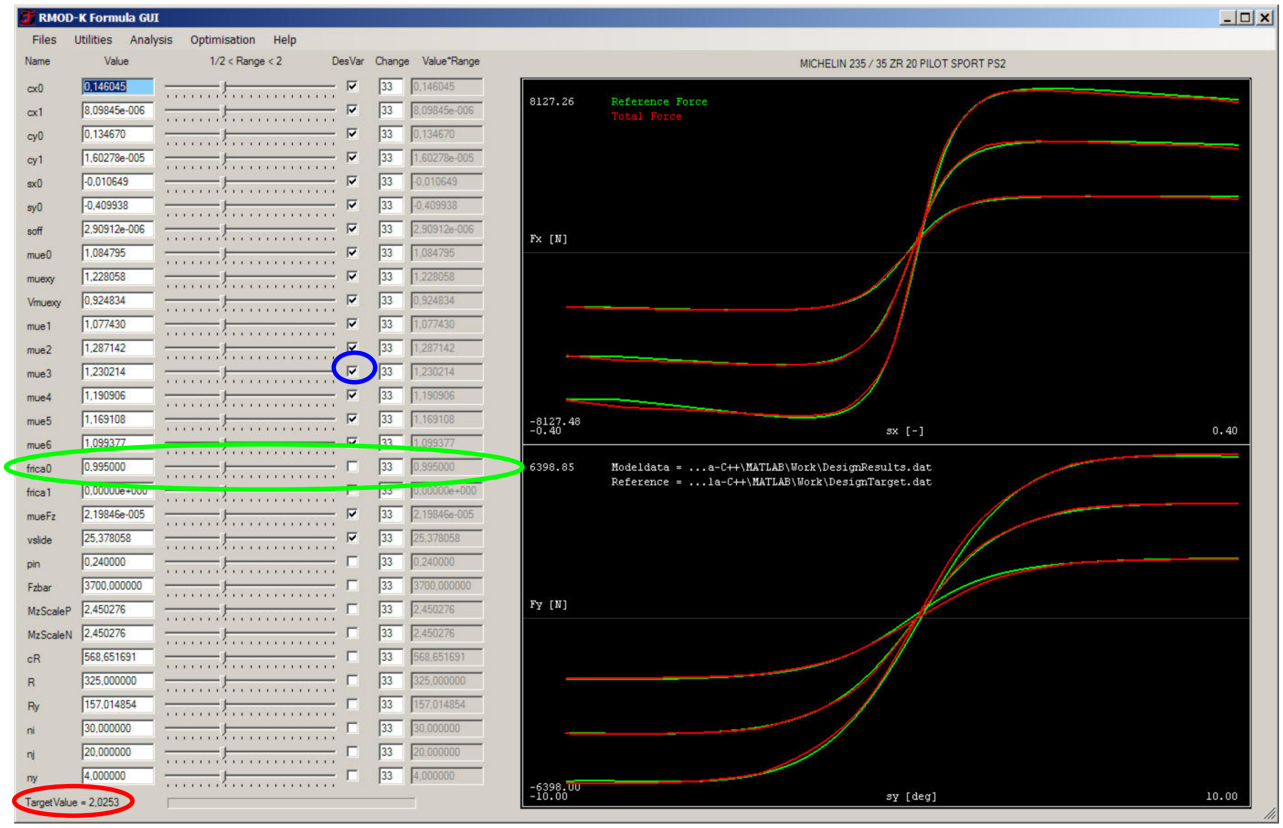


```
[DataFileName]
DesignResults.dat
[FileInput1]
Datafile1 = ..\..\..\Optimierung\Optimierung_Example1\Fx1850.dat VT = 8000.0
[FileInput2]
Datafile2 = ..\..\..\Optimierung\Optimierung_Example1\Fx3700.dat VT = 8000.0
[FileInput1]
Datafile3 = ..\..\..\Optimierung\Optimierung_Example1\Fx5500.dat VT = 8000.0
[FileInput2]
Datafile4 = ..\..\..\Optimierung\Optimierung_Example1\Fy1850.dat VT = 8000.0
[FileInput1]
Datafile5 = ..\..\..\Optimierung\Optimierung_Example1\Fy3700.dat VT = 8000.0
[FileInput2]
Datafile6 = ..\..\..\Optimierung\Optimierung_Example1\Fy5500.dat VT = 8000.0
[Eof]
```

Figure 33: File run data file

4.3 Graphical user interface

A graphical user interface (GUI) is shown in figure 34. The user is able to change the model parameter set and see immediately the results in comparison with given references results. This can be used to find a set of reasonable initial design values before starting an optimisation run, to analyse the parameter sensitivity or to study the effect of parameter changes in relation to the physical meaning of one parameter. The graphical user



Name	Value	1/2 < Range < 2	DesVar	Change	Value*Range
cx0	0,146045		✓	33	0,146045
cx1	8,09845e-006		✓	33	8,09845e-006
cy0	0,134670		✓	33	0,134670
cy1	1,60278e-005		✓	33	1,60278e-005
sx0	-0,010649		✓	33	-0,010649
sy0	-0,409938		✓	33	-0,409938
soff	2,90912e-006		✓	33	2,90912e-006
mue0	1,084795		✓	33	1,084795
muexy	1,228058		✓	33	1,228058
Vmuexy	0,924834		✓	33	0,924834
mue1	1,077430		✓	33	1,077430
mue2	1,287142		✓	33	1,287142
mue3	1,230214		✓	33	1,230214
mue4	1,190906		✓	33	1,190906
mue5	1,169108		✓	33	1,169108
mue6	1,099377		✓	33	1,099377
frica0	0,995000		☐	33	0,995000
frica1	0,00000e+000		☐	33	0,00000e+000
mueFz	2,19846e-005		✓	33	2,19846e-005
vslide	25,378058		✓	33	25,378058
pin	0,240000		☐	33	0,240000
Fzbar	3700,000000		☐	33	3700,000000
MzScaleP	2,450276		☐	33	2,450276
MzScaleN	2,450276		☐	33	2,450276
cR	568,651691		☐	33	568,651691
R	325,000000		☐	33	325,000000
Ry	157,014854		☐	33	157,014854
ni	30,000000		☐	33	30,000000
nj	20,000000		☐	33	20,000000
ny	4,000000		☐	33	4,000000

TargetValue = 2,0253

```
Modeldata = ...a-C++\MATLAB\Work\DesignResults.dat
Reference = ...la-C++\MATLAB\Work\DesignTarget.dat
```

Figure 34: Graphical user interface

interface is divided into the parameter manipulation area (left and green ellipse in figure 34) and the plot area (right). Additional information is display in the bottom area, where the target function value is shown (red ellipse in figure 34).

4.3.1 Parameter manipulation

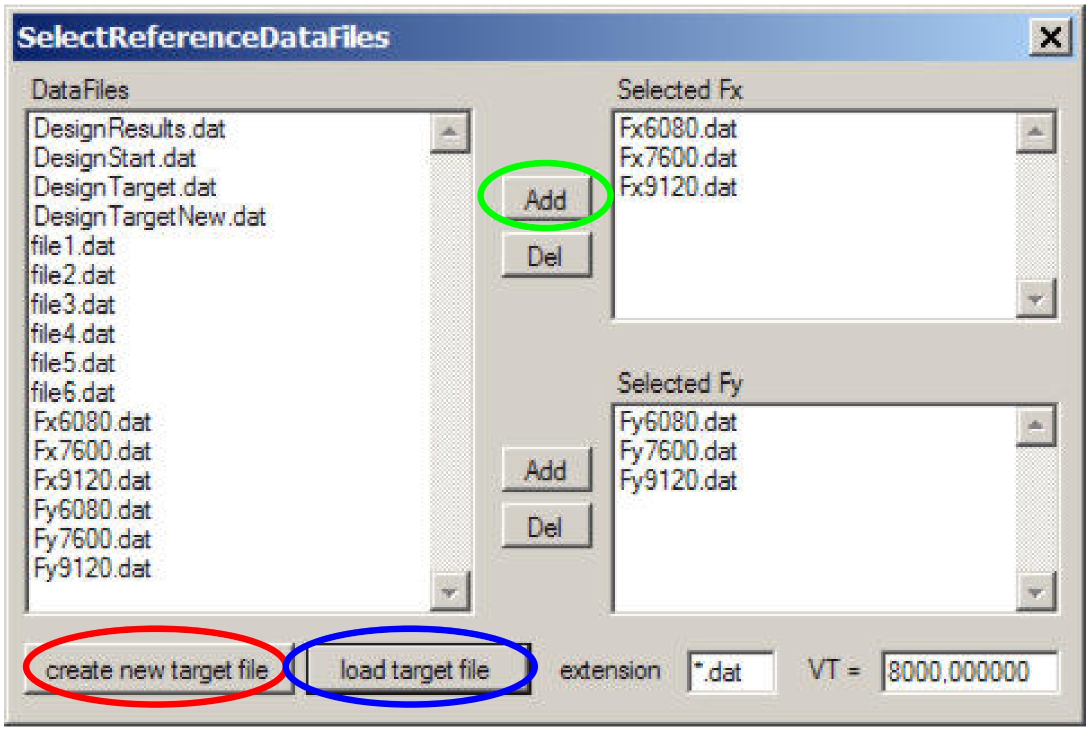
To load a data set, click on `LoadParameter` from the `Files` menue point and select the data file (for instance `DesignStart.dat`). The parameter manipulation area will be updated with parameter names and current values. The reference and model results will be shown, when a target data file (for instance `DesignTarget.dat`) is loaded using the `LoadMeasurements` menue item from the `Files` menue. Both result sets will by displayed in the right area of the GUI. To change a parameter value, two possibilities are supported be the GUI:

- change the slider position between left (0) and right position (100) in order to make changes of the parameter p_j in the range of $\frac{1}{2}\,p_j < p_j < 2\,p_j$
- or change the parameter by specifying a new value in the edit box besides the parameter name.

Due to parameter value change, the target function will change and the graphical results as well as the target function value are updated after each parameter change (slider move). This can be used as manual optimisation procedure. If one of the optimisation procedures described above is used, the check boxes (marked with the blue ellipse in figure 34) identify if a parameter should belong to the set of design variables or not.

4.3.2 Design target file

Under `Utilities` the `BuildTarget` menue item leads to the file selection box. On the left hand side, the contents of the current work directory is displayed (only files matching a given extension: see file filter extension in the box bottom area) and a group of files can be selected and added (green ellipse in figure 35) to the design target definition. With the button `create new target file` (red ellipse in figure 35) a dialog to save the new target file is activated. To look for details concerning an existing design target file, the `load target file`


SelectReferenceDataFiles

DataFiles
DesignResults.dat
DesignStart.dat
DesignTarget.dat
DesignTargetNew.dat
file1.dat
file2.dat
file3.dat
file4.dat
file5.dat
file6.dat
Fx6080.dat
Fx7600.dat
Fx9120.dat
Fy6080.dat
Fy7600.dat
Fy9120.dat

Selected Fx
Fx6080.dat
Fx7600.dat
Fx9120.dat

Selected Fy
Fy6080.dat
Fy7600.dat
Fy9120.dat

extension *.dat
VT = 8000,000000

Figure 35: Define a design target file

button (blue ellipse in figure 35) displays a dialog to load the target file. It contents is written into the lists of reference files on the right hand side of the box. Finally, the transport velocity v_T is set to a value by specifying it in the field in the right bottom area.

4.3.3 Plotting options

The `Utilities` menue point allows the manipulation of the curves display style concerning the color `CurveColor` and the line width `LineProperties`. To select a special results set component, click on `Visibility`. There, the

- reference curves (normally a combination of measurements in longitudinal and lateral direction)
- the total force curves (the sum of sticking and sliding force components)
- sticking force curves (the sticking part, see for instance equation (48))
- sliding force curves (the sticking part, see for instance equation (51))
- accuracy border curves (a tolerance relative to the reference curves)
- legends (add file names to the graphics)
- track data (forces plotted against the slip or against the row number of the data file)

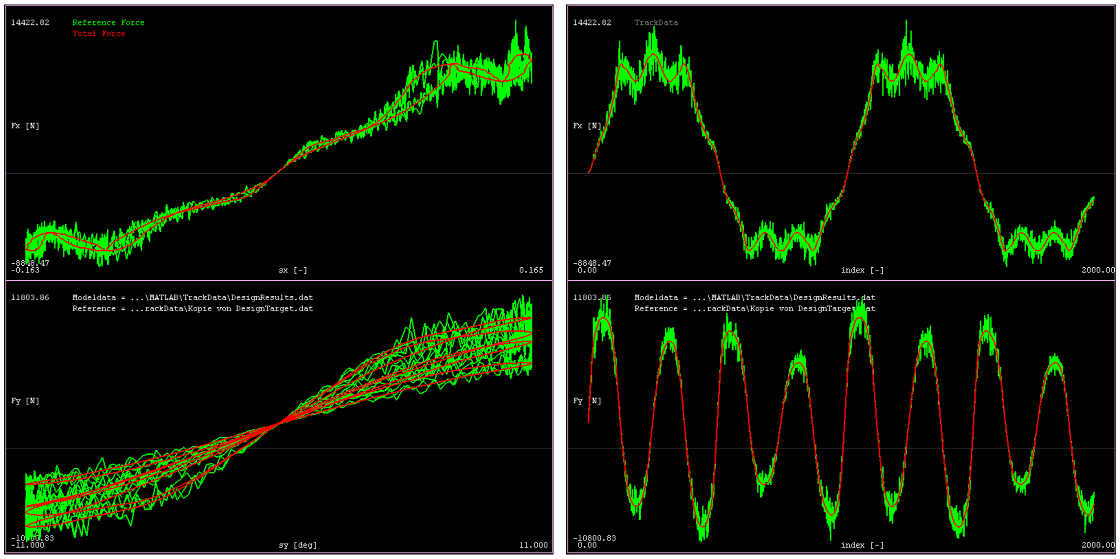
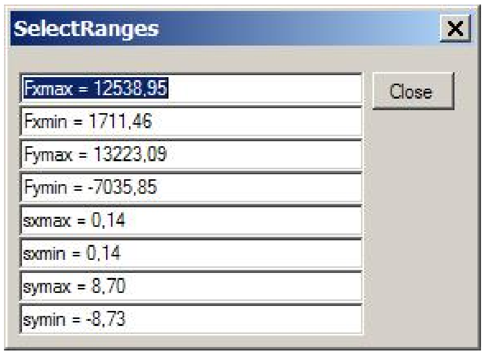


Figure 36: Track data mode view

can be selected for displaying. The two tack data modes are shown as example in figure 36.
The menue item `LineProperties` allows the user to select the line width or the accuracy border (unit %) used to draw the accuracy border curves. The user scaling is activated by the menue item `UserScale`, which opens the box to define minimum and maximum values for forces and slip. If the track data mode is set to plotting against the index, user scaling is only active concerning the forces scaling.



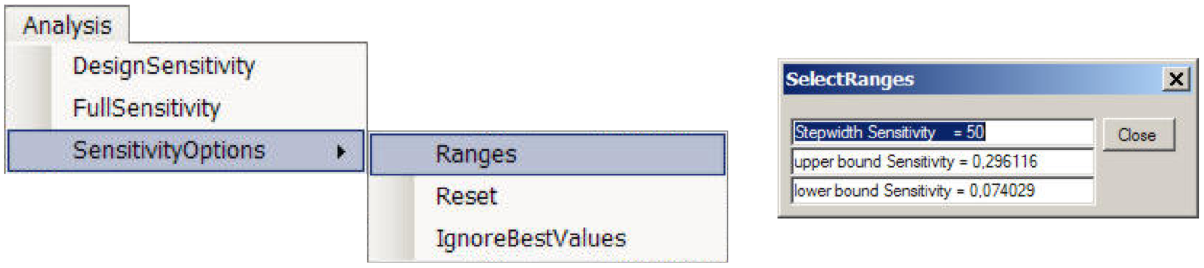
Fxmax = 12538,95
Fxmin = 1711,46
Fymax = 13223,09
Fymin = -7035,85
sxmax = 0,14
sxmin = 0,14
symax = 8,70
symin = -8,73

Figure 37: User scaling box

The `Save Graphics` menue item opens the save file box to set the file name of the screen shot. It contains the graphics part of the window as bitmap file.

4.3.4 Sensitivity analysis

The `Analysis` menue point allows the user the analyse the design sensitivity of a single parameter or all model parameters. To select a design variable, click on the corresponding slider - the edit box colour changes to blue (see figure 34). The design variable range is taken according to the sliding range $\frac{1}{2}\,p_j < p_j < 2\,p_j$ as default. To change the ranges or number of steps between minimum and maximum value, use the menue item `SensitivityOptions` - see figure 38. To reset minimum and maximum value to the default range, click on `Reset`. While the sensitivity analysis is running, the progress of the analysis is displayed in the bottom area of the GUI. The file `sensity.dat` contains the results in two columns, first column contains the design variable value and the second the target function values. By default, a file named `SensitySingle.gnu` produces a plot as shown in figure 39, wherein the best value of the target function and the corresponding variable value are given in the plot title. In the example of figure 39, two local minima where found. This illustrates the idea of the sensitivity analysis, which is to avoid initial values for the design variables in a local minimum with a larger target function value than other local minima.



Stepwidth Sensitivity = 50
upper bound Sensitivity = 0,296116
lower bound Sensitivity = 0,074029

Figure 38: Analysis menue

In the case of a full sensitivity analysis (`FullSensitivity`), the design variable values may remain unchanged (with respect to the current values) after each step (`IgnoreBestValues`) or the next step is done which a parameter value according to the smallest target function value in the step before. In this case, a sequence of `sensityXX.dat` files are generated as well as the corresponding file `SensityMulti.gnu`.

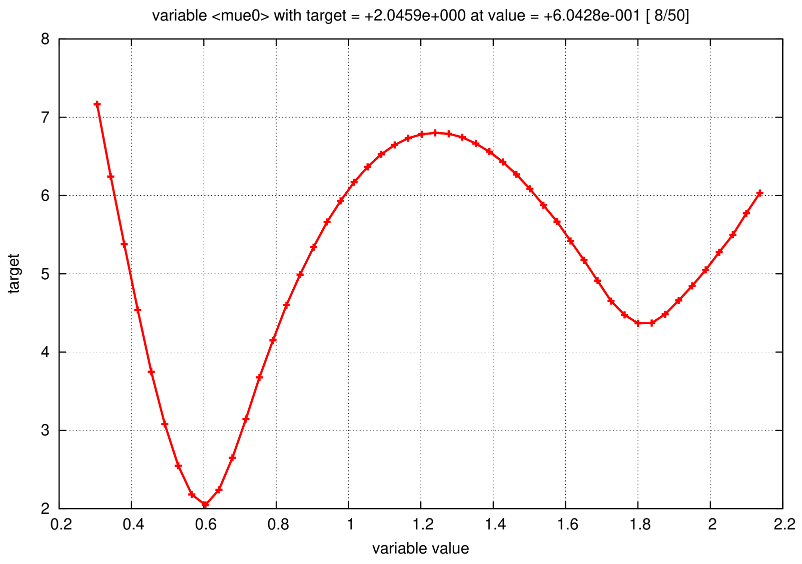


Figure 39: Analysis results

4.3.5 Optimisation

To support users with are not familiar with tools like MATLAB oder SCILAB, it is possible to start an optimisation run directly from the GUI. See section on page 39 for more details how to prepare the optimisation run. The `Optimisation` menue point contains

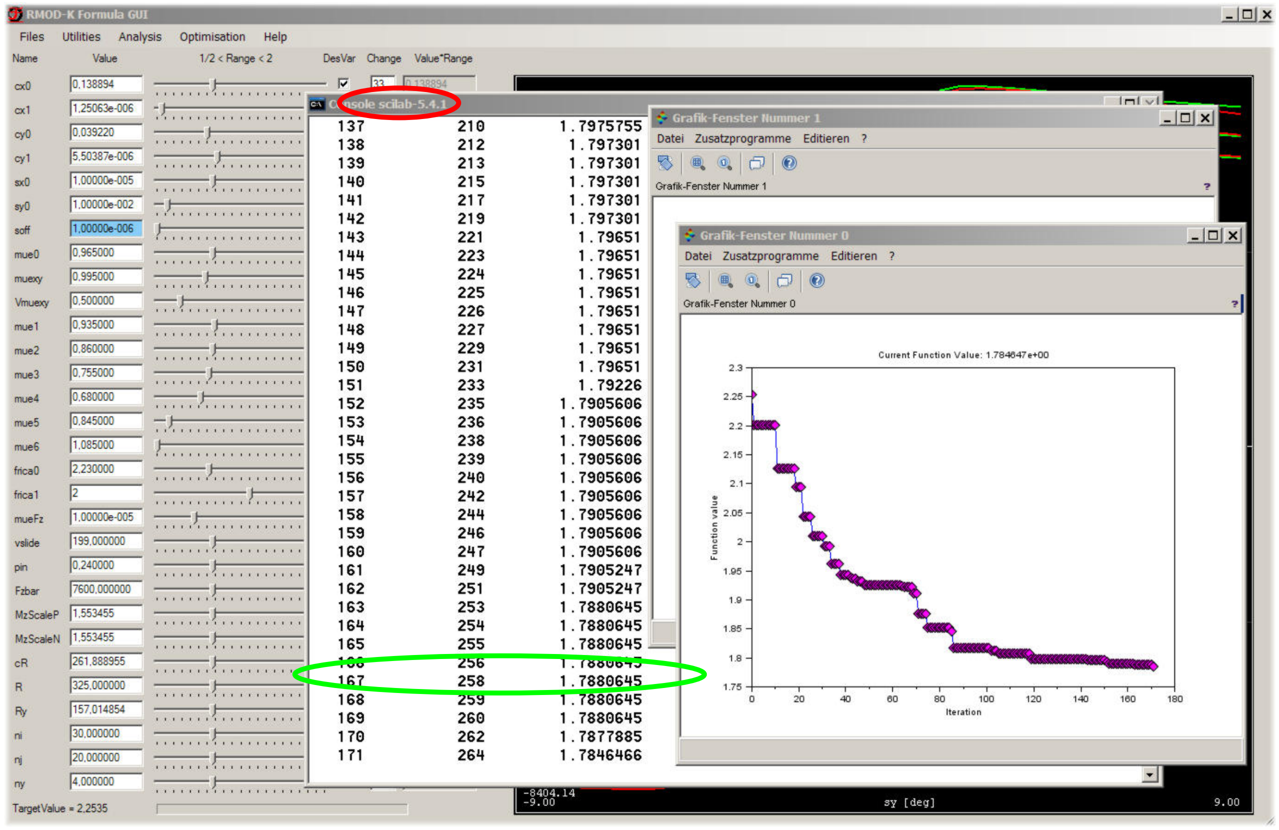
- the menue item `SetPath` to set the working directory for optimisation (with may be `....\Formula-C++\SCILAB\Work` following the standard installation according to figure 26)

- the menue item `SelectMethod` to select the optimisation method (SCILAB-script) (with may be `Optimisation_fminsearch.sce`) and

- the menue item `Run Scilab` to start a thread running the optimisation.

- the menue item `Run Matlab` to start a thread running the optimisation.

A SCILAB-version higher than 5.4.0 must be installed and – for instance `C:\Program Files\scilab-5.4.0\bin` – added to the system path variable. `Run Matlab` requires similar settings. In the typical screen shot in figure 40, the optimisation with SCILAB is currently running.



Name	Value
cx0	0,138894
cx1	1,25063e-006
cy0	0,039220
cy1	5,50387e-006
sx0	1,00000e-005
sy0	1,00000e-002
soff	1,00000e-006
mue0	0,965000
muexy	0,995000
Vmuexy	0,500000
mue1	0,935000
mue2	0,860000
mue3	0,755000
mue4	0,680000
mue5	0,845000
mue6	1,085000
frica0	2,230000
frica1	2
mueFz	1,00000e-005
vslide	199,000000
pin	0,240000
Fzbar	7600,000000
MzScaleP	1,553455
MzScaleN	1,553455
cR	261,888955
R	325,000000
Ry	157,014854
ni	30,000000
nj	20,000000
ny	4,000000

1/2 < Range < 2 DesVar ✓ Change 33 Value*Range 0,138894

TargetValue = 2,2535

Console scilab-5.4.1

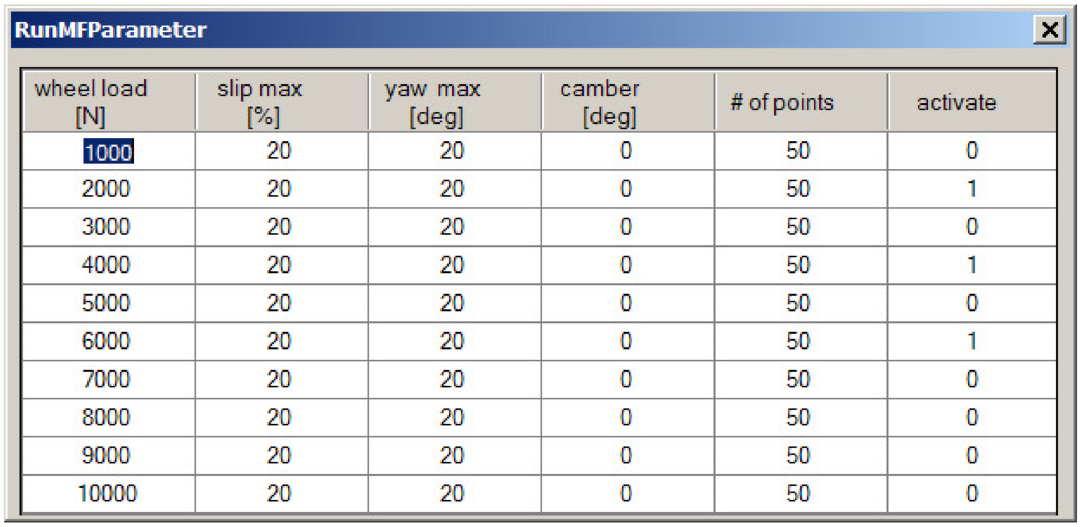
```
   137          210          1.7975755
   138          212           1.797301
   139          213           1.797301
   140          215           1.797301
   141          217           1.797301
   142          219           1.797301
   143          221            1.79651
   144          223            1.79651
   145          224            1.79651
   146          225            1.79651
   147          226            1.79651
   148          227            1.79651
   149          229            1.79651
   150          231            1.79651
   151          233            1.79226
   152          235          1.7905606
   153          236          1.7905606
   154          238          1.7905606
   155          239          1.7905606
   156          240          1.7905606
   157          242          1.7905606
   158          244          1.7905606
   159          246          1.7905606
   160          247          1.7905606
   161          249          1.7905247
   162          251          1.7905247
   163          253          1.7880645
   164          254          1.7880645
   165          255          1.7880645
   166          256          1.7880645
   167          258          1.7880645
   168          259          1.7880645
   169          260          1.7880645
   170          262          1.7877885
   171          264          1.7846466
```

Figure 40: Graphical user interface and optimisation thread

When the thread has finished loading the file `DesignResults.dat` will display the new results and the corresponding values of the design variables. The user can control the optimisation by changing initial design values or by including or excluding variables from the design variable set and restore the file under `DesignStart.dat` before starting a new run.

4.3.6 Magic Formula import

The magic formula tire model ([5], [6]) is widely used in vehicle dynamic simulation. Therefore, model data for a lot of different tires are available and can be used as semi-measurement source. To import magic formula, use `LoadMFParameter` from the `Files` menue point and select the desired running conditions with `DefineTargetFromMF` from the `Utilities` menue point, where the wheel load, the slip range in longitudinal and lateral direction as well as the camber angle will be defined – see figure 41.
The number of steps for each curve can be chosen in column five while in column six, the number zero deactivates the row will the number one actives it. For each activate row, two data sets (with slip input in longitudinal and lateral direction) are generate by `RunTargetFromMF` from the `Utilities` menue point and the results are


RunMFParameter

wheel load [N]	slip max [%]	yaw max [deg]	camber [deg]	# of points	activate
1000	20	20	0	50	0
2000	20	20	0	50	1
3000	20	20	0	50	0
4000	20	20	0	50	1
5000	20	20	0	50	0
6000	20	20	0	50	1
7000	20	20	0	50	0
8000	20	20	0	50	0
9000	20	20	0	50	0
10000	20	20	0	50	0

Figure 41: Parameter for building a target file from MF data

displayed as reference data. Running the optimisation procedure as described above requires renaming the file `TargetMF.dat` in `DesignTarget.dat`.

5 Application examples

Several application examples illustrate the models fields of applications and the methods how to use the present version of the software.

5.1 SIMULINK example

As a first application example, a model of a tire trailer with three degrees of freedom in figure 42 is build.

5.1.1 Model description

The degrees of freedom are the trailer longitudinal velocity V, the angular velocity of the rim φ_R and the relative rotation of the belt φ_B. Written in state space form and treating the tire force as external force, the equations are

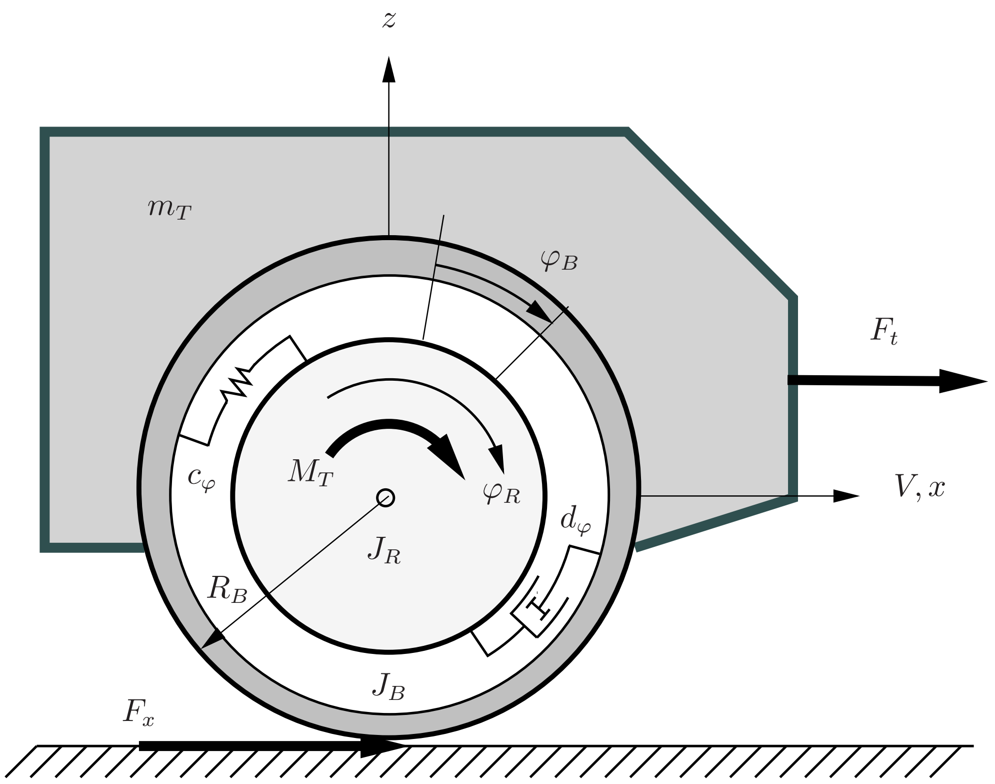


Figure 42: Tire trailer model with three degrees of freedom

$$\begin{bmatrix} m_T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & J_B & J_B \\ 0 & 0 & 0 & 0 & J_R \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\varphi}_B \\ \dot{\varphi}_R \\ \ddot{\varphi}_B \\ \ddot{\varphi}_R \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & -c_\varphi & 0 & -d_\varphi & 0 \\ 0 & c_\varphi & 0 & d_\varphi & 0 \end{bmatrix} \begin{bmatrix} V \\ \varphi_B \\ \varphi_R \\ \dot{\varphi}_B \\ \dot{\varphi}_R \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -R_B & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} F_x \\ M_T \\ F_T \end{bmatrix} . \quad (68)$$

The equations of motion contain the torque M_T and the trailing force F_T. If the trailer mass is large or the trailing force ensures $V = const.$, this degree of freedom can be omitted. A controller is added to control the angular velocity of the rim and in consequence the slip s_x. The torque M_T in the closed loop version follows with the given time depended target angular velocity Ω_{R_T} from

$$M_T = K_P\left(\Omega_{R_T} - (\dot\varphi_R + \dot\varphi_B)\right) + K_I \int \left(\Omega_{R_T} - (\dot\varphi_R + \dot\varphi_B)\right) dt. \tag{69}$$

This leads to

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & J_B & J_B \\ 0 & 0 & 0 & J_R \end{bmatrix}}_{\underline{A}_1} \begin{bmatrix} \dot\varphi_B \\ \dot\varphi_R \\ \ddot\varphi_B \\ \ddot\varphi_R \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -c_\varphi & 0 & -d_\varphi & 0 \\ c_\varphi - K_I & -K_I & d_\varphi - K_P & -K_P \end{bmatrix}}_{\underline{A}_2} \begin{bmatrix} \varphi_B \\ \varphi_R \\ \dot\varphi_B \\ \dot\varphi_R \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -R_B\,F_x \\ f(\Omega_{R_T}) \end{bmatrix}. \tag{70}$$

Introducing the slip

$$s_x = (R_{dyn}\,(\dot\varphi_R + \dot\varphi_B) - V)/V. \tag{71}$$

with data provided from output equation due to *normal* measurement by an angular velocity sensor

$$y = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \varphi_B \\ \varphi_R \\ \dot\varphi_B \\ \dot\varphi_R \end{bmatrix} = \dot\varphi_R \tag{72}$$

a linearisation of the longitudinal tire force is

$$F_x = F_{x_0} + F'_{x_0}\,R_{dyn}\,(\Delta\dot\varphi_R + \Delta\dot\varphi_B)/V = F_{x_0} + \underbrace{\frac{F'_{x_0}\,R_{dyn}}{V}}_{d_{F_x}}(\Delta\dot\varphi_R + \Delta\dot\varphi_B) = F_{x_0} + d_{F_x}\,(\Delta\dot\varphi_R + \Delta\dot\varphi_B) \tag{73}$$

and using this in (70), $\underline{A}$ becomes

$$\underline{A} = \underline{A}_1^{-1}\,\underline{A}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{J_B} & -\frac{1}{J_B} \\ 0 & 0 & 0 & \frac{1}{J_R} \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -c_\varphi & 0 & -d_\varphi - R_B\,d_{F_x} & -R_B\,d_{F_x} \\ c_\varphi - K_I & -K_I & d_\varphi - K_P & -K_P \end{bmatrix}. \tag{74}$$

If d_{F_x} is large negative, at least one of the eigenvalues of $\underline{A}$ has a positive real part in the case of the open loop system ($K_I = K_P = 0$), which becomes unstable. Concerning the closed loop system, from (74) controller gain stability maps are available by the *Hurwitz*-criterion.

5.1.2 Simulations

Two simulations models compare two test rigs with and without a slip controller. In the first one, a harmonic function as motion constraint pushes the angular velocity of the rim φ_R out of the degrees of freedom and the only remaining degree of freedom is the relative rotation of the belt with respect to the rim. The related block diagram in figure 43 has no closed control loop. A RMOD-K-Formula `S-function` block is build with a mask, having the data file name and a tire number as input. In the model block, slip, normal force and longitudinal velocity are inputs while the forces and the self aligning moment are outputs.

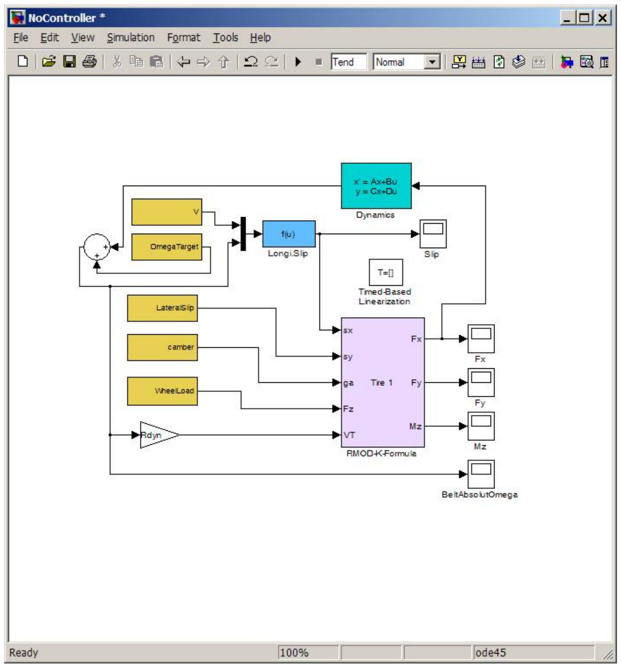


Figure 43: Test rig without slip controller

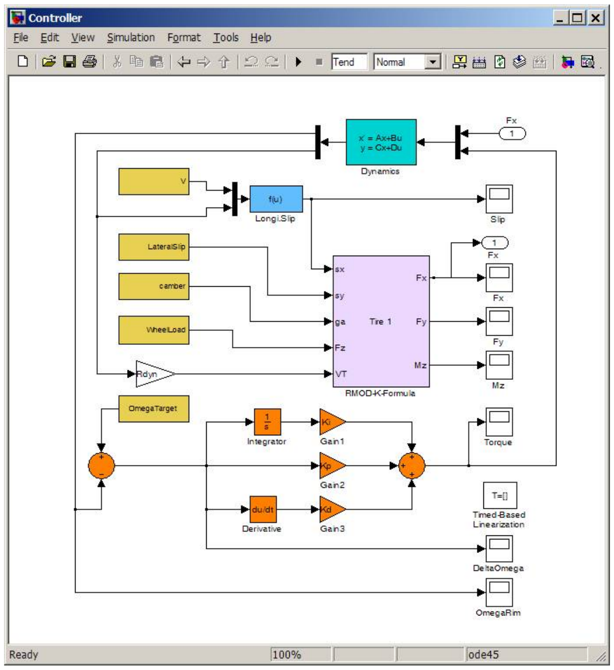


Figure 44: Test rig with slip controller

In the second model version, a controller is used to control the slip. The *PI*-controllers output signal drives for instance an electric engine with the torque M_T. This model is more closely to what a real test rig should be able to do. The closed loop system is shown in figure 44.
The same input to both models result in similar output F_x with one big difference: while the controlled model behaves stable, some oscillations occur without control. These oscillations are visible in F_x (figure 45) and can

be observed in real measurements also. The reason for instability is to be seen in the unstable oscillations of the belt caused by the decreasing shape of $F_x(s_x)$ at large slip angles – see for instance figure 21, this data set was used in the example. The unstable behaviour depends strongly on the sign of the slip. This is explained in [1].

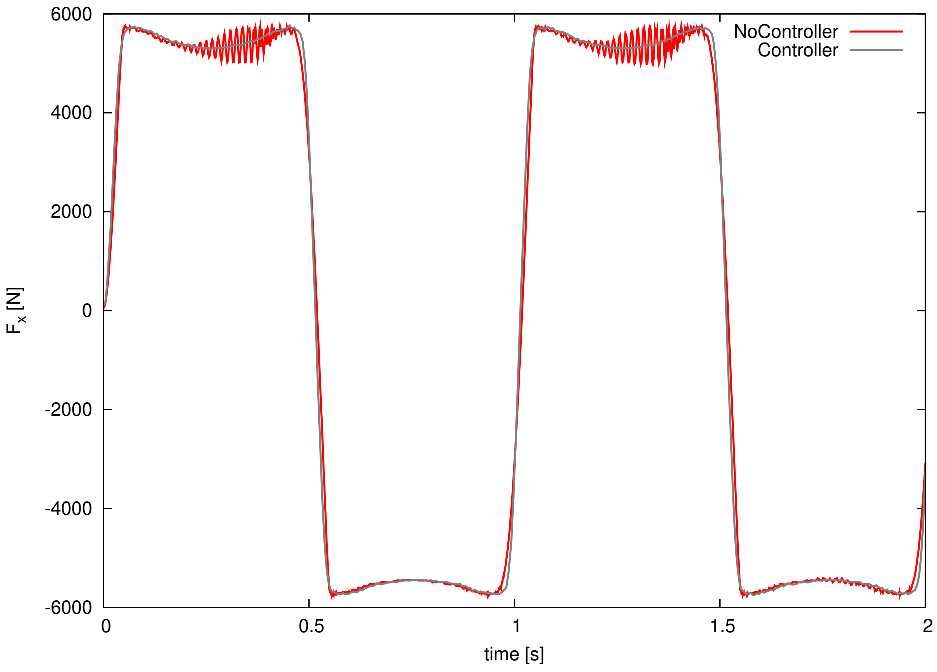


Figure 45: Test rig results: longitudinal force

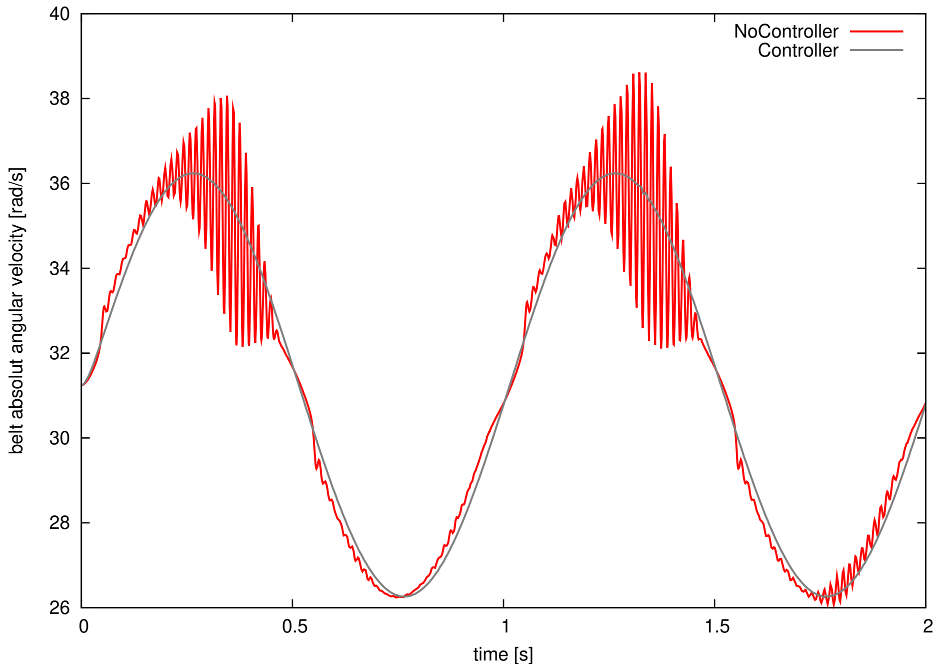


Figure 46: Test rig results: belt absolute angular velocity

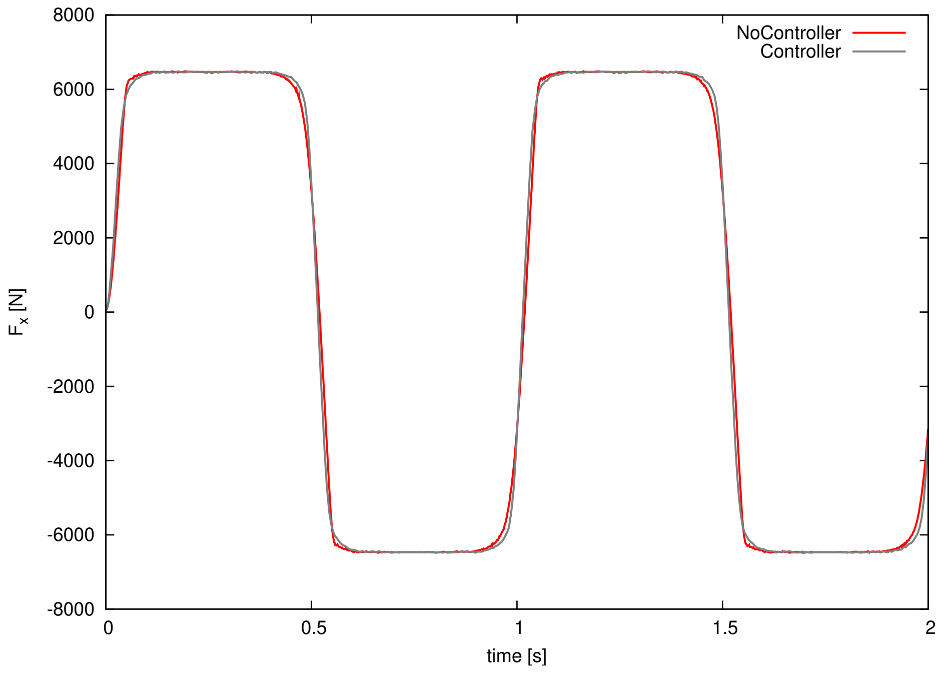


Figure 47: Test rig results: longitudinal force without decrease at higher sliding velocities

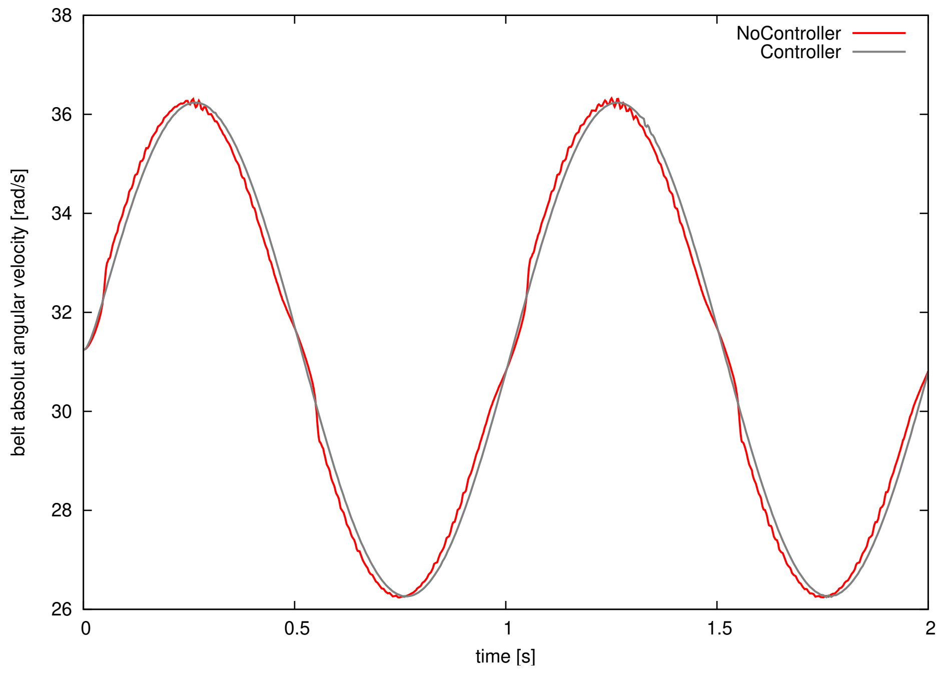


Figure 48: Test rig results: belt absolute angular velocity without decrease at higher sliding velocities

5.2 MATLAB example

The interface to the MATLAB-optimisation file – `RMOD_K_Formula_Opti` – uses the information of the model data file (figure 27) to get the number of design variables and the related parameters. If a parameter shall by included in the set of design variables, simply add the string `designbounds` as well as upper and lower bound or remove this, if the parameter should no longer belong to the set. The optimisation results – the target value – is included in the data file if any optimisation has been carried out. The optimisation looks for such data file named `DesignStart.dat` and produces a results file of the same format with the name `DesignResults.dat`. The target is given by the optimisation target file, see section above.

5.2.1 Optimisation m-file

In the MATLAB file for optimisation, the variable `UseLastValues` should be set to false, if the parameter values in the file `DesignStart.dat`, otherwise the values from `TheBestValues.mat` are used as initial values. The first run in a new directory must be done with `UseLastValues` set to false, since the file `TheBestValues.mat` does not exist. `MaxNumberEvals` and `MaxNumberIterations` are limiting the number of steps in the optimisation – see MATLAB for more information. The first call to `MATLAB_RMOD_K_Formula` does the allocations inside the C++-code and return the number of design variables while the last call does writung and deallocations.

```
%---------------------------------------------------------------------------
% MATLAB optimization interface to RMOD-K-Formula OPTI code
%---------------------------------------------------------------------------
clc;
%
% --- set job type information
%
UseLastValues        = false;
%UseLastValues       = true;
MaxNumberEvals       = 3000;
MaxNumberIterations  = 3000;
time                 = cputime;
%
% --- set options of the optimization
%
options = optimset(@fminsearch);
options = optimset(options,'Display','iter','MaxFunEvals',MaxNumberEvals,'MaxIter',MaxNumberIterations);
%
% --- init model and target, get number of variables
%
design0 = zeros (100,1);
ndes    = MATLAB_RMOD_K_Formula(design0,0);
%
% --- copy design start variables and resize
%
design  = zeros (int8(ndes),1);
for i=1:int8(ndes)
    design(i) = design0(i);
end
%
% --- or init design with values from file
%
if ( UseLastValues )
    load ('TheBestValues.mat')
    design = TheBestValues;
end
%
% --- start optimization
%
[TheBestValues,TheTargetValue] = fminsearch(@TargetFunction, design, options);
%
% --- save best solution and prepare gnuplots plots
%
save('TheBestValues.mat','TheBestValues');
MATLAB_RMOD_K_Formula(TheBestValues,-1)
%
% --- get the calculation time
%
sprintf('cpu time %7.2f s',cputime - time)
```

The function `targetfunctionvalue` is very short. Here again, `MATLAB_RMOD_K_Formula` is called.

```
function targetfunctionvalue = TargetFunction(bet)
%---------------------------------------------------------------------------
% MATLAB optimization interface to RMOD-K-Formula OPTI code
%---------------------------------------------------------------------------
%
% --- call the model
%
targetfunctionvalue = MATLAB_RMOD_K_Formula(bet,1);
```

5.3 SCILAB example

SCILAB offers some optimisation tools like a genetic algorithm to solve the task of parameter optimisation. The same description of the target (DesignTarget.dat) and the initial values (DesignStart.dat) are used like previously described.

5.3.1 SCILAB interface

The interface to SCILAB runs with version 5.3.3 and put into two SCILAB-files. The first one is Optimisation_Start.sci, where the dll is linked and some wrapper functions are supplied.

```
//
   // --- clean memory and screen, NArraySize is the dimension of the workspace (> 3*NumberOfModelParameters)
   //
   clear all;
   clc;
   funcprot(0);
   format('v');
   ulink();
   link("SCILAB-RMOD-K-Formula.dll","RMOD_K_SCIFunction","c")
   NArraySize = 200;
   //
   // --- get information about the optimsation problem, initial values and bounds from input file
   //
   function [Target,DesignStart,DesignMin,DesignMax] = SCILAB_RMOD_K_INIT (design)
       global NArraySize NumberOfDesvars
       mymode = 0.0;
       mysize = NArraySize;
       OutputArray = ones(mysize,1);
       [Target,OutputArray] = fort("RMOD_K_SCIFunction",design,1,"d",mymode,2,"d",...
                                   "out",[1,1],3,"d",[mysize,1],4,"d");
        NumberOfDesvars = int32(Target);
        DesignStart = ones (NumberOfDesvars,1);
        DesignMin   = ones (NumberOfDesvars,1);
        DesignMax   = ones (NumberOfDesvars,1);
        for i=1:NumberOfDesvars
           DesignStart(i) = OutputArray(i);
           DesignMin(i)   = OutputArray(i+NumberOfDesvars);
           DesignMax(i)   = OutputArray(i+NumberOfDesvars*2);
        end
   endfunction
   //
   // --- terminate run a free memory
   //
   function Target = SCILAB_RMOD_K_TERMINATE (design)
       global NArraySize
       mymode = -1.0;
       mysize = NArraySize;
       CurrentDesign = ones(mysize,1);
       [Target,CurrentDesign] = fort("RMOD_K_SCIFunction",design,1,"d",mymode,2,"d",...
                                     "out",[1,1],3,"d",[mysize,1],4,"d");
   endfunction
   //
   // --- call the function to get target values
   //
   function Target = SCILAB_RMOD_K (thedesign)
       global NArraySize method
       global BestDesign
       global BestTarget
       mymode = 1.0;
       mysize = NArraySize;
       CurrentDesign = ones(mysize,1);
       [Target,CurrentDesign] = fort("RMOD_K_SCIFunction",thedesign,1,"d",mymode,2,"d",...
                                     "out",[1,1],3,"d",[mysize,1],4,"d");
       if ( Target < BestTarget ) then
           BestTarget = Target;
           BestDesign = thedesign;
           if ( messages == 2 )
               mprintf ("\n Current best target value = %g\n",Target)
           end
       end
   endfunction
```

This is used by three optimisation drivers, which are the general optimisation macro of SCILAB providing different solvers by setting the solver flag in Optimisation_Optim.sce, the genetic algorithm with design bounds in `Optimisation_Optim_ga.sce` and *fminsearch* in `Optimisation_Optim_fminsearch.sce`. The methods addressed by OPTIM in the general optimisation macro need the derivate of the target function, which is done by numerical differentiation. This might by the reason why OPTIM and OPTIM_GA where not successful in some experiments with a large number of design variables. Splitting the task of optimisation in several subtasks with freeze operations (first optimize stiffness, then freeze stiffness and optimize friction related parameters) performed much better. For more details see the SCILAB documentation.

```
global NumberOfDesvars NArraySize messages
   global BestTarget
   global DesignStart BestDesign
   //
   // --- initialise the functions
   //
   exec('Optimisation_Start.sci');
   //
   // --- initialise the data
   //
   DesignStart = ones (100,1);
   [Target,DesignStart,DesignMin,DesignMax] = SCILAB_RMOD_K_INIT(DesignStart);
   BestTarget = 1.0e+10;
   messages   = 3;
   //
   // --- initialise the optimisation
   //
   myplotfun = list ( optimplotfval , optimplotx  );
   opt = optimset ("Display","iter",...
                  "FunValCheck","on",...
                  "MaxFunEvals",400,...
                  "MaxIter",400,...
                  "PlotFcns",myplotfun,...
                  "TolFun",1.e-12,...
                  "TolX",1.e-13)
   //
   // --- perform the optimisation
   //
   [x fval] = fminsearch ( SCILAB_RMOD_K ,DesignStart ,opt );
   //
   // --- terminate the function
   //
   Target2 = SCILAB_RMOD_K_TERMINATE(BestDesign)
   dos ('copy DesignResults.dat DesignStart.dat')
```

5.3.2 SCILAB optimisation

In this example, a data set was changed to get a large initial value of the target function. The friction values in the table function where disturbed and the data from figure 21 to 23 was used as *pseudo* measurements.

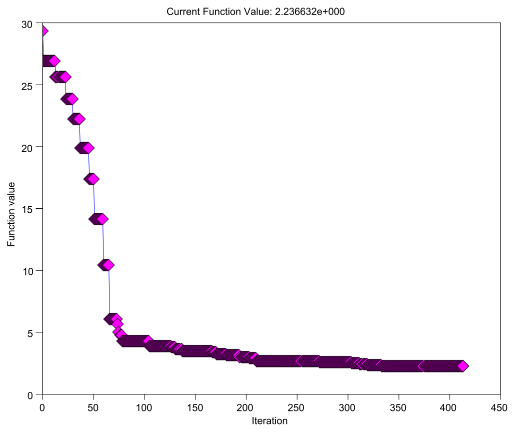


Figure 49: Optimisation target history

From the target function plot in figure 49 it can be seen that the target value has reached after 350 iterations a certain limit and the comparison between start and end of the optimisation is shown in figure 50.

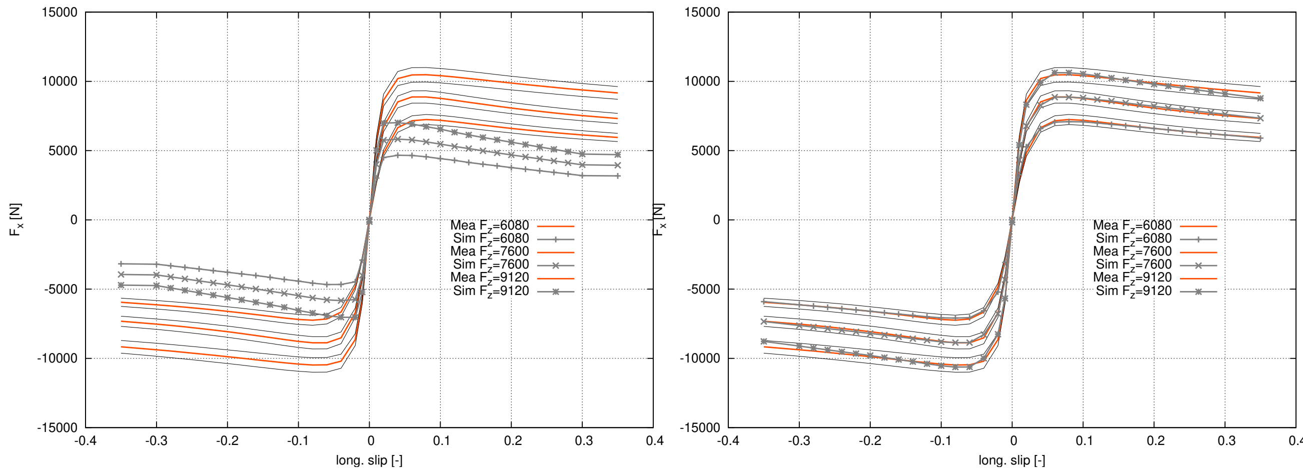


Figure 50: Start and optimization result F_x, Formula, discrete model with table data

5.4 Insight contact example

As mentioned previously, the identifier `SINGLE INPUT` produces the contact information for a given rolling condition. The resulting contact force distribution gives some insights in steady state rolling contact with respect to the parametrisation of the model. The border between sticking and non sticking area becomes visible in figure 51, where non zero longitudinal, lateral and camber input are combined.

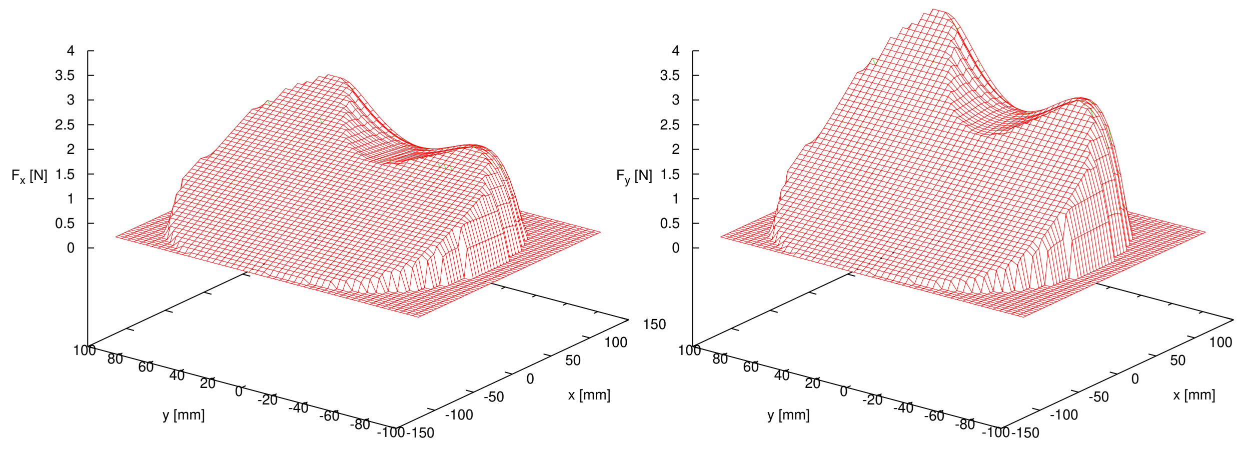


Figure 51: Contact force distribution, F_x and F_y

By comparison between tangential and normal force distribution (figure 52), the sensitivity of a certain parameter in the context of contact force distribution can be studied. Due to camber input, the normal force distribution depends on the lateral direction of the contact patch as well as the sticking border.

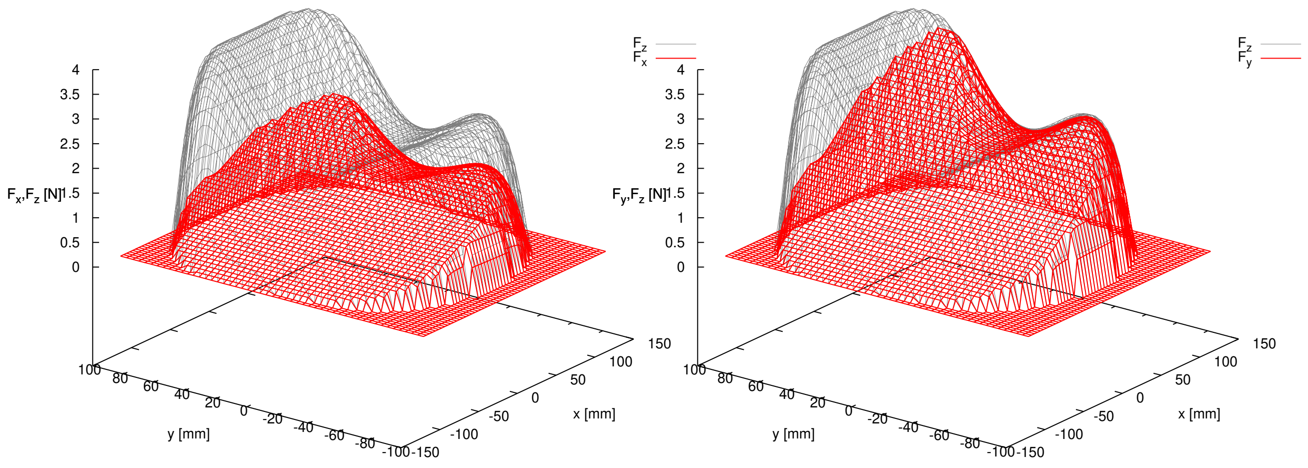


Figure 52: Contact force distribution, F_x, F_z and F_y, F_z

References

[1] Ch. Oertel. Eigensystem of tangential contact in tyre models. *Vehicle System Dynamics*, 38:245–260, 2002.

[2] Ch. Oertel, A. Fandre. Tire Model RMOD-K 7 and Misuse Load Cases. *SAE International*, 2009. SAE-Paper 2009-01-0582.

[3] E. Fiala. Seitenkräfte am rollenden Luftreifen. *Verein Deutscher Ingenieure – VDI Zeitschrift*, 96(29):973–979, 1954.

[4] G.Gim, Y. Choi, S. Kim. A semiphysical tyre model for vehicle dynamics analysis of handling and braking. *Vehicle System Dynamics*, 43:267–280, 2005.

[5] H. B. Pacejka, E. Bakker. The Magic Formula Tyre Model. Tyre models for vehicle dynamics analysis, Proceedings of the 1st International Colloquium on Tyre Models for Vehicle Dynamic Analysis, Delft, 1991.

[6] H. B. Pacejka. *Tire and Vehicle Dynamics*. Butterworth-Heinemann, Oxford, 2002.

[7] Todd D. Plantenga. Hopspack 2.0 user manual. Technical Report SAND2009-6265, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, October 2009.

[8] Y.T. Wei, Y. Liu. Three dimensional nonlinear finite element analysis for pneumatic tire structures. *Automotive Engineering*, 20(5):272–278, 1998.

Index