FTire Software: Advances in Modelization and Data Supply

Michael Gipser*
mg@ftire.com
Esslingen University of Applied Sciences
Kanalstrasse 33
73728 Esslingen
GERMANY

Presented at the
September 2006
Meeting of the Tire Society

*Presenter / Corresponding Author

M. Gipser

FTire Software: Advances in Modelization and Data Supply

REFERENCE: M. Gipser, "*FTire* Software: Advances in Modelization and Data Supply," submitted for presentation at the 2006 Tire Society meeting, and for consideration for publication in the journal *Tire Science and Technology*.

ABSTRACT: Being developed and continuously improved for more than 7 years now, the *FTire* simulation software has become one of the widely used and generally accepted tire models for ride comfort, handling, and road load prediction in durability applications. Strength of *FTire* is its strict physical background, taking into account most of the relevant sources and non-linear transfer mechanisms of excitations up to very high frequencies and short wavelengths. The model's level of detail is accompanied by a numerically robust and efficient solver, and by a very comfortable program interface. This allows simulating even extreme maneuvers and situations with moderate computation time. *FTire* can be used together with most of the important MBS packages. The contribution discusses recent advances in the *FTire* software. This discussion comprises enhancements in the model, like motorcycle tire simulation with very large camber angles, detailed lateral belt bending, distributed tire imperfections, new detailed and efficient road models, as well as improved tools for parameterization and validation.

KEY WORDS: Tire simulation, *FTire*, motorcycle tires, belt flexibility, cross section geometry, tire imperfections, tire model parameterization, road models.

Motorcycle Tires

One of the most important recent improvements of *FTire* is the ability to simulate motorcycle tires even at very large camber angles. To achieve this, both the belt geometry model and the contact processor have been enhanced.

The cross section geometry of a motorcycle tire (as well as that of any other tire), cf. figure 1, can be described in *FTire* in two alternative ways. Either a constant lateral curvature radius is assumed for the belt plies, or the belt plies and the tread surface is given by two tables of *x*/*y* data pairs. The latter, new approach is discussed in greater detail in the third chapter. It is well known in tire design that the cross-section geometry plays a major role for the handling characteristics of the tire. Consequently, it should be specified in an *FTire* model as accurate as possible.

Second change that had been necessary for motorcycle tires was to place all tread elements normal to the belt surface, rather than in constant radial direction. The difference between these two approaches is negligible as long as the belt is nearly flat. But it is vital

for the correct contact computation at extremely large camber angles when the belt shows large curvature, and thus the surface normal considerably deviates from radial direction.

Thirdly, a more accurate description of the 'kinematic' coupling between belt out-of-plane bending and belt torsion about the circumferential axis had been introduced, cf. figure 2. This is because out-of-plane bending is not negligible at large camber angles. The coupling to the belt torsion in turn affects the position and shape of the contact patch, with all well-known consequences. The implementation of this coupling is simple but effective: the local coordinate system of a belt segment, in which in-plane and out-of-plane bending angles and resulting bending moments are computed, is rotated by a constant fraction of the belt torsion angle. The respective proportionality has value 1 for complete coupling and value 0 for no coupling, any intermediate value being allowed. This is one of the new optional model parameters, being called **`belt_torsion_oop_bend_coupl`** in data files.

Finally, a more detailed description of the belt bending in lateral direction turned out to be indispensable. The underlying numerical approach is discussed in the second chapter. The original approach of using only a single quadratic function for the radial belt displacement created unacceptable large geometrical errors near the belt boundaries.

Figures 3 to 5 show some results of motorcycle tire simulations. The first image is a snapshot of a camber thrust simulation on a drum, at 35 deg camber angle. The resulting contact patch shape and pressure distribution is indicated in a small contour plot which can be dynamically displayed during a running simulation. However, the exact shape of the contact patch can be seen more accurately in image 4. It shows simulated and measured contact patch shapes at 3 different vertical loads for a rear motorcycle tire, running at 20 deg camber angle. Apparently, the simulation results are in good accordance to the respective measured footprints. Finally, figure 5 shows simulated camber thrust diagrams for a front and a rear wheel motorcycle tire, both reaching the extreme value of 60 deg camber angle.

Improved Belt Flexibility Model

The most recent *FTire* version introduces a detailed belt flexibility model, taking more accurately into account the shape of the lateral bending curves (for a description of *FTire*'s basic belt model, we refer to [3] and [4]). This is accomplished by introducing new degrees of freedom for each belt segment. These degrees of freedom are the coefficients of certain shape functions, which are orthogonal eigensolutions of the bending beam equation.

Clearly, this approach is simplified and can by no means compete with a detailed FE model. Its mere purpose is to approximate the belt distortion as accurate as it is required to predict road contact forces and enveloping properties. This road load prediction has to be sufficiently accurate, even with extremely short-waved and high obstacles.

If we disregard the longitudinal coupling for the moment, the lateral bending of the belt layers, in terms of the radial belt displacement function *u(s)*, can be described by the beam equation (cf. figure 6):

$$EI \cdot u^{(IV)} = f_{cont} \qquad (1)$$

Here, f_{cont} is the distributed unilateral contact force, induced by the radial deflection of the tread rubber:

$$f_{cont}(s) = \max\left[0, c_{tread} \cdot \left(z_{road}(s) - z_{contour}(s) - u(s) + h_{tread}(s)\right)\right] \qquad (2)$$

$z_{contour}(s)$ denotes the center line of the unloaded belt, including translation and rotation of the rim. Clearly, the equations have to be completed by certain boundary conditions. Moreover, (1) and (2) are only valid in steady-state conditions. For dynamic simulation, a mass and a damping term has to be added. Note that equation (2) introduces a severe nonlinearity.

In *FTire*, the equations are solved iteratively. Expressed somewhat simplified, the belt displacement function of the previous time step is used to approximately compute f_{cont} for the actual time step. This requires solving the equation (1) with a right hand side that no longer depends on *u(s)*; the modified equation is linear now. The solution is approxi-

mated by a Ritz-Galerkin approach, using orthonormal eigenfunctions of the beam equation:

$$u^{approx}(s) = \alpha + \beta \cdot s + \sum_{i=1}^{N} \gamma_i \cdot q_i(s) \tag{3}$$

where

$$EI \cdot q_i^{(IV)} = \lambda_i q_i \tag{4}$$

The shape functions q_i fulfill appropriate homogenous boundary conditions. These functions only depend on the belt width and can easily be computed, cf. figure 7. The advantage of using orthonormal eigenfunctions is the decoupling property: the computation of the generalized coordinates α, β, γ_i means nothing but dividing the respective 'generalized load' f_i through the lateral belt bending stiffness:

$$\gamma_i = \frac{1}{EI} \int_{-0.5w}^{0.5w} f_{cont}(s) \cdot q_i(s) ds = \frac{1}{EI} f_i \tag{5}$$

The first two generalized coordinates, which are defining $u^{linear}(s) = \alpha + \beta \cdot s$, are treated separately. The radial shift α already is included in the detailed (and highly nonlinear) radial belt segment stiffness sub-model, and the rotation term $\beta \cdot s$ is covered by the belt torsion degree of freedom. Both degrees of freedom take into account mass (or moment of inertia, respectively), and damping.

Of course, the spatial belt distortion is governed by an anisotropic 2-dimensional partial differential equation, rather than the one-dimensional beam equation. However, the complete solution of this shell equation would take unacceptably long computation time.

For this reason, the longitudinal coupling only is *approximated*

1. by the effect of the in–plane bending stiffness, smoothing the mean belt segment displacements described by α,
2. by the 'twist' stiffness between two adjacent belt segments, smoothing the term $\beta \cdot s$, and

3. by additionally smoothing the remaining generalized bending coordinates γ_i^k (k denotes the belt segment number).

The latter is achieved by adding a coupling term to (5), using a new parameter κ_i:

$$EI \cdot \left\{ \gamma_i^k + \kappa_i \cdot \left(-\gamma_i^{k-1} + 2\gamma_i^k - \gamma_i^{k+1} \right) \right\} = f_i^k \qquad (6)$$

The effect of this kind of smoothing is similar to the one of a second order low-pass filter in time domain. Note that (6) establishes another set of linear systems of equations, which are to be solved in each time step. However, the system matrices are cyclic band matrices, which can be treated numerically in a very efficient way. In the current version, the smoothing factors κ_i are all set to the same value which is defined by the parameter **`belt_lat_bend_stiffn_long_coupl`**.

In figure 8, the effect of the lateral belt bending model is demonstrated by means of an extreme example. The tire is deflected on a 20x20 mm high obstacle, located 30 mm laterally off the footprint center. The images show, in 3-fold magnification, the distorted belt geometry from different points of view. In the front view, the smoothened lateral belt bending functions can clearly be seen. Note that such small obstacles require a sufficiently large number of contact points and of tread strips, in order to resolve the obstacle geometrically.

The bar graphs in figure 9 are related to the question on how to choose the number N of eigenfunctions. One has to find a compromise between solution accuracy and extra computational effort. Apparently, in this example there is no benefit in terms of accuracy when using more than 8 shape functions. On the other hand, computing time only grows moderately with the number of shape functions. So, it is concluded that $N = 8$ seems to be an acceptable compromise. This has also turned out in a couple of other experiments.

Detailed Geometry and Tire Imperfections

Clearly, belt and carcass shape, as well as tread gauge variation in lateral direction are most important tire design parameters. They do not only influence the internal stresses, but also affect the contact pressure distribution. This happens for two reasons: the geometrical properties determine the tire's unloaded outer contour, and the tread gauge varia-

tion causes tread stiffness variations. If *FTire* is used to study these effects, it must be able to represent the detailed geometry.

As mentioned in the first chapter, this can be done by optionally entering tables of *x*/*y*-data pairs for (1) the tire's outer contour (not following grooves) in a cross-section view, and for (2) the outer hull of all carcass and belt plies, cf. figure 1. There are several alternative ways and options available in doing so:

1. one may specify data for the total cross section, or only for left or right half, letting *FTire* mirror them automatically,
2. one may specify original geometry data, or allow the data to be shifted and stretched such that the outer contour exactly matches the theoretical tire size,
3. one may specify piecewise linear interpolation or smooth parametric spline interpolation,
4. as an option, one can additionally smoothen the data by higher order polynomial regression.

It is essential to specify the data as accurate as possible, or to smoothen them by one of the methods mentioned above. A final check on whether data are really smooth enough should be the inspection of the resulting contact pressure, by using the tool *FTire/static*.

FTire will automatically sort the data, and decide whether they are to be mirrored. The data points for the two curves are treated completely independent on each other. Their spacing may be different and non-constant. The number of data points per curve is limited to 100. However, this limit can be easily extended for special investigations. The table data format is simple enough to import data by using a text editor.

The package *FTire/fit*, which will be discussed in the next chapter, provides two comfortable tools in the framework of geometry data input:

(1) a digitizing tool to import scanned cross section images, and

(2) a single-click checking tool for the cross-section geometry as 'seen' in *FTire* after pre-processing, cf. figure 10.

In the checking tool's graphical output, the grey shaded regions denote the theoretical dimensions of rim and tire, as they are specified by the tire size string. Moreover, the tool plots the curves for outer contour and belt / carcass surface after final processing, and sketches the actual number and placement of the contact elements.

Another recent extension of *FTire* introduces more detailed specification and treatment of general non-harmonic imperfections. Table data of the respective non-constant variables can be specified versus the angle about wheel spin axis. *FTire* will interpolate these data either piecewise linearly, or by smooth spline interpolation. Both interpolation approaches use cyclic boundary conditions. Data points may be unequally spaced. The imperfections that can be treated this way are:

1. radial non-uniformity (variation of the radial stiffness),
2. tangential non-uniformity (variation of the tangential stiffness),
3. run-out (variation of the belt radius),
4. mass variation, and
5. tread gauge variation.

In the case of run-out, mass variation, and tread gauge variation, *FTire* will automatically balance the tire. This is convenient if the higher-order impact of mass or geometry variation is to be separated from the 'classical' first-harmonic effect of static or dynamic imbalance. Both kinds of imperfection may be specified simultaneously.

FTire/tools contains another single-click program which allows to display all specified imperfections in terms of diagrams. In a next version, *FTire* will be able to process distributed tread gauge variation data not only in circumferential direction, but over the full 2D tread surface.

Experiences in Parameterization. The New *FTire/fit*

Lately, the evolution of *FTire* mainly focused on *FTire/fit*, the toolbox for measurement-based parameterization. The development of *FTire/fit* is nearly solely driven by own experiences during its application. This has lead to a better understanding

1. of the usefulness of certain measurement types,
2. of an appropriate sequence in which measurements are evaluated,
3. of an appropriate sequence in which parameters are determined,
4. of the relevance of certain measurements for certain parameters,
5. of plausible ranges of parameter values,
6. of the sensitivity between parameter values and model properties,
7. and finally on how to improve the user-friendliness of the program.

One important experience is about the role of modal data. They seem to contain less relevant information than static measurements (in contrast to what had been assumed in the early days of *FTire*), and typically they are more laborious to get. There is one obvious cause for the lack of relevant information: during modal measurements on an unloaded tire, only small amplitudes will be reached. However, these measurements are then used to parameterize *FTire* for load cases with large to extreme deflection values, that is, in completely different operating points.

Another experience is the amount of valuable information contained in footprint bitmaps. The same holds for several kinds of static deflection curves without and with camber angle, on flat surface or on certain well-defined obstacles. Moreover, handling properties like cornering stiffness and pneumatic trail show a high correlation with certain out-of-plane stiffness data. In many cases, after a thorough analysis of static and steady-state behavior, there remain only few dynamically relevant parameters to be adjusted in order to get also a good correlation in cleat tests.

The procedure roughly sketched in the following was successful in many cases, and is well supported by *FTire/fit* in all stages:

(1) Prepare the identification process:

1. create a new data file with *FTire/estim* (cf. fig. 11), by specifying tire and rim size, load index, speed range, mass, and inflation pressure(s). As a reference tire, use one which is as close as possible to the new tire. *FTire/estim* can be launched via the *FTire/fit* GUI,

2. specify all drum diameters and cleat geometries used during identification and validation. *FTire/fit* provides example road data files and functions to manage such obstacle-defining files,

3. specify ('check-in') all static, steady-state, and dynamic measurement files used in the sequel. If these files are given in the TYDEX file format (cf. [9]), a single mouse-click to check them in is sufficient in many cases. *FTire/fit* will automatically recognize what kind of measurement they contain, will determine constant operating conditions like inflation pressure, wheel load, camber angle, etc., and will save information on how the validation or identification is to be performed. Moreover, depending on the kind of measurement, it will occasionally extract relevant information like radial, longitudinal, lateral, and torsional stiffness, cornering stiffness, slip stiffness, pneumatic trail, camber thrust, sliding friction, etc., and insert this information in terms of 'nominal data' into the tire data file, or save it elsewhere. If the files are given in any other ASCII format, *FTire/fit* assists in importing the files and creates TYDEX files out of it. If measurements are only given in terms of scanned images, *FTire/fit* provides a digitizing tool based upon MSPaint™, which also assists in creating TYDEX files,

4. specify ('check-in') all footprint bitmap files. *FTire/fit* will automatically calibrate these files and save the relevant information for later validation.

(2) Import or digitize tread and carcass contour geometry data. As already mentioned in the third chapter, *FTire/fit* provides a respective digitizing tool, based upon MSPaint™.

(3) Identify dynamic rolling radius on basis of the measurement of the angular velocity of a freely rolling tire at different drum speeds and wheel loads (or roughly estimate the rolling radius by subtracting tread gauge from maximum radius).

(4) Identify/validate the following static properties:

1. vertical stiffness on flat surface (which is merely a validation of the two deflection values for half and full LI load that have been automatically inserted

in the data file in step (1.3)). The respective simulation had been prepared by *FTire/fit*; a single mouse-click is sufficient to launch the validation and save all results for later report generation. If the actual stiffness deviates from the predicted one, adjust the respective deflection values. This might happen if there is a discrepancy between the static and the steady-state kind of simulation, caused by different treatment of hysteresis and friction properties,

2. longitudinal and lateral stiffness on flat surface (same remark as in 1.),
3. torsional stiffness (turning the standing tire about the vertical axis). Adjust belt torsional stiffness about radial axis accordingly, if simulation deviates from measurement,
4. vertical stiffness on longitudinal and transversal cleat. Adjust lateral belt bending stiffness and belt in-plane bending stiffness accordingly, if simulation deviates from measurement,
5. vertical stiffness at large camber angle on flat surface and on transversal cleat. Adjust belt torsion and twist stiffness about circumferential axis, if simulation deviates from measurement,
6. footprint size and shape at different wheel loads. *FTire/fit* provides an automatic simulation preparation and superimposes the simulated footprint boundary over the measured contact patch bitmap. Again, all this is done by a single mouse-click. If there is a mismatch in size or shape, adjust in-plane and lateral bending stiffness. Run 4. again and find a compromise.

(5) Identify/validate the following steady-state properties:

1. longitudinal slip stiffness. Either activate the measured nominal value directly, by replacing tread rubber stiffness, or identify tread rubber stiffness manually. *Fire/fit* has prepared the validation, provided a respective measurement file is available,
2. cornering stiffness and pneumatic trail. Either activate the measured nominal values directly, by replacing lateral stiffness and out-of-plane bending stiff-

ness, or (re-)identify these two values manually. If there is a discrepancy to the value of lateral stiffness determined in (4.2), find a compromise.

(6) Identify sliding friction coefficients. During check-in of the measurement files, *FTire/fit* has automatically collected all available and relevant cases. Ideally, this identification is performed by one mouse-click only. After the identification, validate the relevant measurement cases (these cases, which have been automatically detected and collected by *FTire/fit*, are: (a) pulling the tire in longitudinal and (b) in lateral direction, (c) turning the tire about vertical axis, and (d) running at large longitudinal slip or (e) at large side-slip). If the identified values of stiction and sliding friction differ much, stick-slip-phenomena might occur in lateral and longitudinal stiffness simulation. In this case, find a compromise by relaxing the differences in friction coefficients.

(7) Run in-plane cleat-test identification (or just validation), determining few remaining parameters like the percentage of free mass, the structural damping (expressed in terms of the modal damping), longitudinal coupling of tread shear stiffness, tread rubber damping, etc.,

(8) Run out-of-plane cleat-test identification (or just validation), determining few more parameters like conicity, modal out-of-plane damping, the coupling between belt torsion and lateral displacement, etc.

In all the phases listed above, *FTire/fit* will automatically collect diagrams showing the comparison between simulation and measurement. Finally, with another mouse-click, *FTire/fit* generates a comprehensive report file, containing all these comparisons and more.

Road Models

Seven years of experience with industrial applications of *FTire* have shown there are two concerns which most users share: (1) the ease, cost, accuracy, and reliability of model parameterization as discussed in the previous chapter, and (2) the ability of using road surface data stemming from sources outside the *FTire* environment.

The latter aspect is vital especially in durability applications. Here, a tire model like *FTire* only is used to predict dynamic road loads. Thus, the tire simulation itself is not in the focus of the interest. Rather, expressed somewhat simplified, the vehicle/tire MBS simulation is nothing but a pre-processing phase to generate loads out of geometry data. The results are used in the durability prediction in the strong sense, which is based upon FEA computations.

In other words: the ability to import and use a wide variety of road data files is one of the most important features of a dynamic tire model.

During the development of *FTire*, this fact was taken into account by defining and consequently using a simple interface to road models. This interface allows using:

1. all MSC.ADAMS™ road data files (rdf-files), including all 2D and 3D methods, xml-files, ARC models, and special motorsports models,
2. all road models that are programmed externally, using the TYDEX/STI Standard Road Description Interface, cf. [9],
3. all road models that are programmed externally, using a simplified interfacing convention ('user-defined road models', cf. [4]),
4. triangularization-based 3D data in WaveFront (obj) format,
5. all *COSIN/ev* (Cosin Road Model) data files, cf. [4],
6. several 'in-house' road models of different customers.

Note that some of these road models need a separate license. In either case, all roads of type (2) to (5) can be used with all implementations of *FTire* in different MBS packages. This is because the road model is not evaluated by the MBS package (except for a potential evaluation in an animation scene), but rather by *FTire* itself.

An important aspect of road models is the efficiency of their evaluation. *FTire* needs to know the road height in each time step near each tread element which is an 'assumed candidate' for road contact. Thus, the number of necessary road evaluations might easily sum up to far more than 1 million per second.

On this background, it turns out that triangularization-based road models, despite their flexibility with respect to local refinement, are not always optimal for high-end tire-models. This is true even if *FTire* uses optimized strategies to finding the active triangle, first searching in the neighborhood of the previous evaluation.

As a consequence, an optimized road model is proposed, which is called the 'Regular Grid Road' (RGR), cf. fig. 12 and [4]. This model does not require searching active triangles. Rather, it can easily compute the actual location in the mesh, due to constant mesh spacing. Moreover, the data file format is less complex because there is no need to store any topology of the mesh.

Data files for this model come in three variants, even though *FTire*'s internal implementation is the same in all cases:

1. *FTire*'s intrinsic RGR version, which is discussed below,
2. FTR road data files, an implementation developed and commercialized together with *VI-grade GmbH*,
3. CRG road data files, based upon a common development of *DaimlerChrysler AG (Research)* and *Tuev Sued AG*, Germany (www.tuev-sued.de/3D-Track).

Regular grid road data files contain the road surface's z-values over an equally spaced x/y-grid. The number format may be binary or formatted, and a non-lossy compression algorithm is available. Thus, not only evaluation performance, but also accuracy and disc-space amount is optimal.

RGR roads allow using a curvilinear center line (cf. fig. 13). This optional line is defined by a separate table of arbitrarily spaced x/y-data, and is evaluated by smooth parametric spline interpolation.

The combination with the grid data is realized as follows: the center line is used as curvilinear x-axis of the grid. x data points are assumed to be located equidistantly along the center line, with a spacing of Δx. The x-coordinate starts with value 0 at the beginning of the road, which is defined by the first data point in the center-line table. Note that the spacing of the center-line data points is completely independent on the spacing of the curvilinear regular grid.

The grid's *y*-coordinate is chosen to be perpendicular to the center-line. That is, it coincides with the center-line's normal in the road surface plane. Positive *y*-values define grid points that are located on the left of the center-line, if viewed along the direction of increasing *x*-values.

In order to make the regular grid data evaluation well-defined and unique, the center line's minimal curvature radius is required to be greater than half the grid width.

Furthermore, the center-line itself might be equipped with z-vales and even lateral banking angles. In this case, the *z*-values of the center-line and those of the grid are superimposed.

RGR files have proven their efficiency in many applications in different companies. For the next version, an extension is planned to allow to 'swap' different portions of the road during a running simulation. This function would help to simulate even longer roads or roads with even finer resolution.

Conclusion

In this paper, several new features of the general purpose tire model *FTire* have been addressed. These features (1) enable the simulation of motorcycle tires, (2) provide more accurate load predictions on very small and high obstacles, (3) allow studying the effects of tire imperfection in greater detail, (4) refine the geometrical properties of the model, (5) facilitate the parameterization process, and (6) make *FTire* compatible with a wide variety of new road models used in industrial practice.

However, this report is nothing but an intermediate report about some recent activities. The development of *FTire* will continue, in order to further increase the prediction quality in existing applications, and to meet the demands of more users and new applications.

References

[1] Dorfi, H.R., "A Study of the In-Plane Force Transmission of Tires", *Tire Science and Technology, TSTCA* Vol. 32, Nr. 4, 2005, pp. 188-213.

[2] Dorfi, H.R., "Tire Non-Uniformities And Steering Wheel Vibrations", *Tire Science and Technology, TSTCA* Vol. 34, Nr. 4, 2005, pp. 64-102.

[3] Gipser, M., "ADAMS/FTire – A Tire Model for Ride & Durability Simulations", *Proc. ADAMS User's Conf.*, Tokyo, 2000 (available for download on www.ftire.com).

[4] Gipser, M., *FTire Documentation and FTire Demo Versions*. http://www.ftire.com

[5] Lieven, W., Warnecke, U., Tran-Quoc, T., "Virtuelle Dauerlaufauslegung von PKW-Karosserien (Virtual Durability Dimensioning of Passenger Car Bodies)", *VDI-Berichte Band 1846*, 2004, pp. 437-448.

[6] Lugner, P., Pacejka, H., Plöchl, P., "Recent Advances in Tyre Models and Testing Procedures", *Vehicle System Dynamics* Vol. 43, 2005, pp. 413-436.

[7] Pacejka, H.B., *Tyre and Vehicle Dynamics*, Butterworth and Heinemann, Oxford, 2002.

[8] Riepl, A., Reinalter, W., Fruhmann, G., "Rough Road Simulation with Tire Model RMOD-K and *FTire*", *Proc. 18th IAVSD Symposium*, Kanagawa Inst. of Tech., Atsugi-shi, Japan, 2003.

[9] Van Oosten J.J.M., Unrau H.J., Riedel G., Bakker E., "TYDEX Workshop: Standardisation of Data Exchange in Tyre Testing and Tyre Modelling", *Tyre Models for Vehicle Dynamic Analysis, Supplement to Vehicle System Dynamics*, Vol. 27. Swets&Zeitlinger Lisse, 1997.

List of Figure Captions

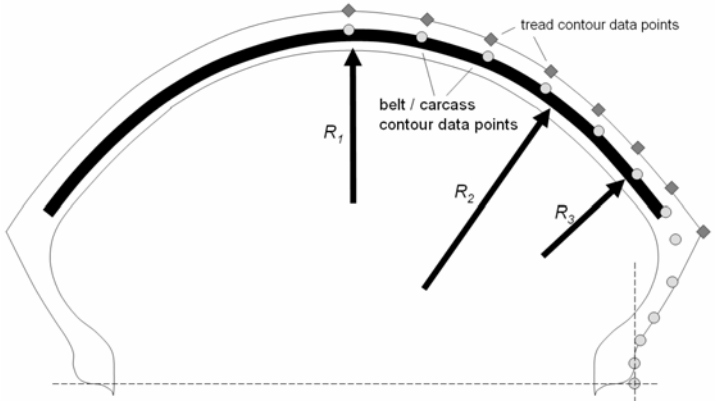

FIG. 1 - *Motorcycle tire cross-section, with describing data points.*

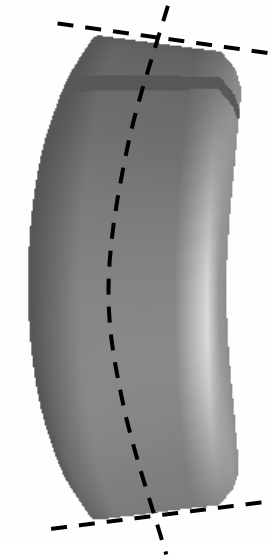

FIG. 2 - *Kinematic coupling of out-of-plane bending and circumferential torsion.*

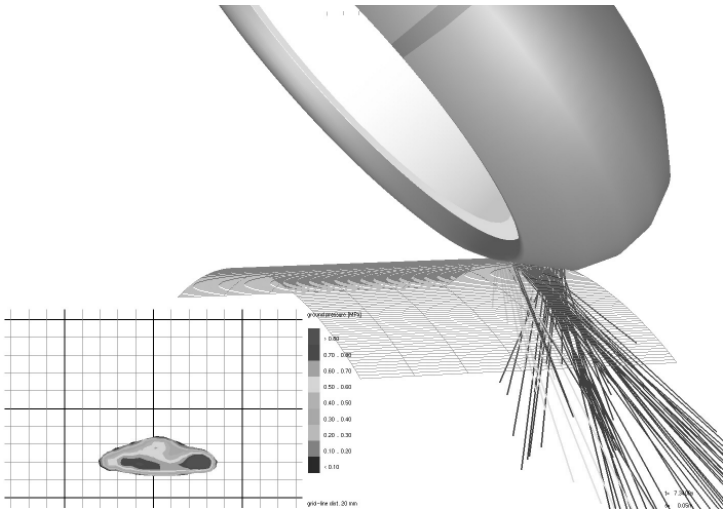

FIG. 3 - *FTire motorcycle model on drum, at 35 deg camber angle.*

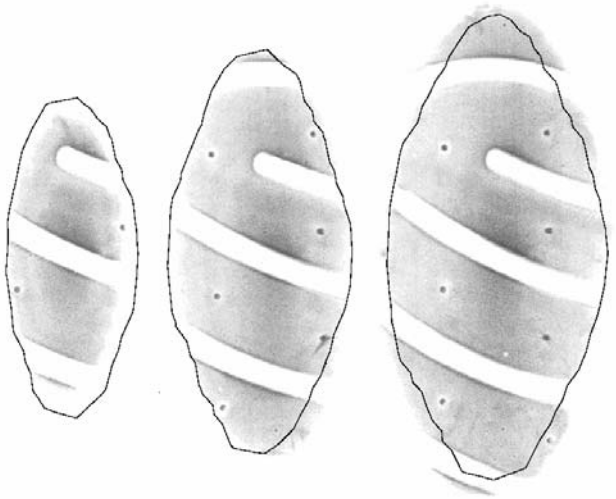

FIG. 4 – Comparison between measured (bitmap) and simulated (line) foot print shapes of a rear motorcycle tire at 1000/2000/3000 N and 20 deg camber angle.

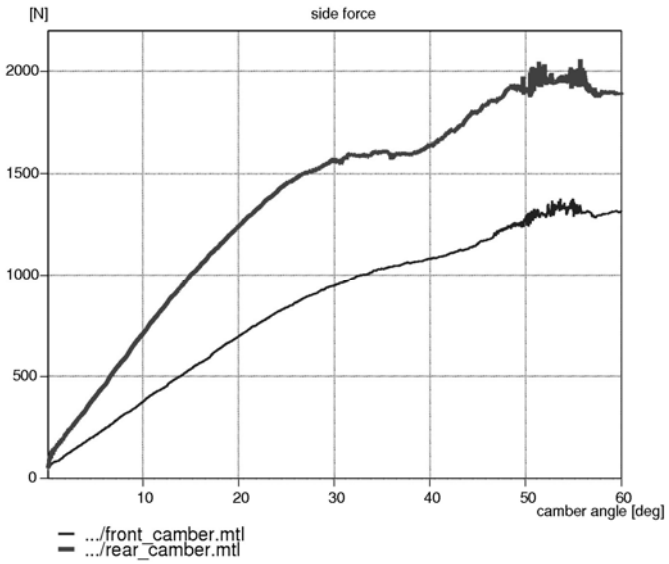

FIG. 5 - Simulated camber thrust (steady-state side force vs. camber angle at zero side-slip angle) of a motorcycle front tire at 1500 N (lower curve), and a rear tire at 2000 N (upper curve).

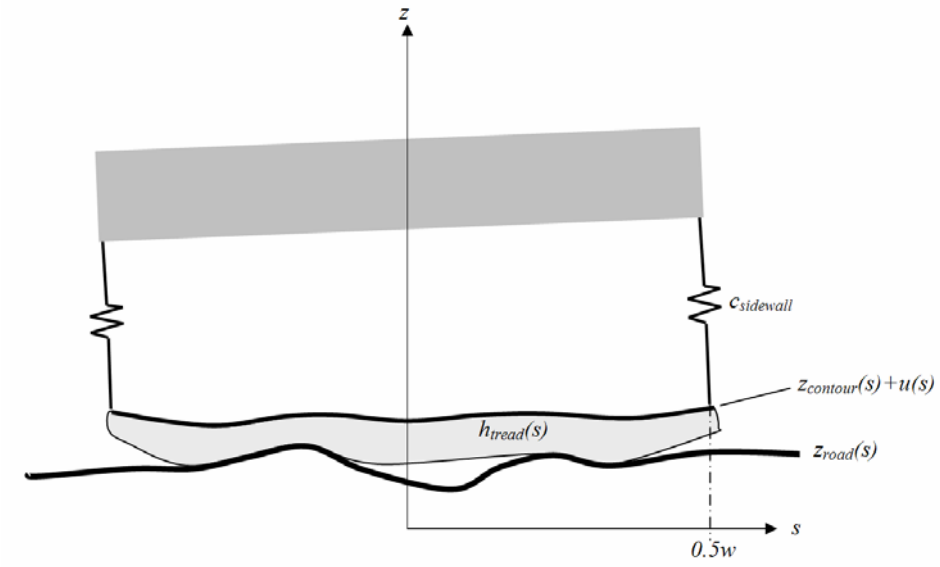

FIG. 6 - *Variables of the lateral belt bending model.*

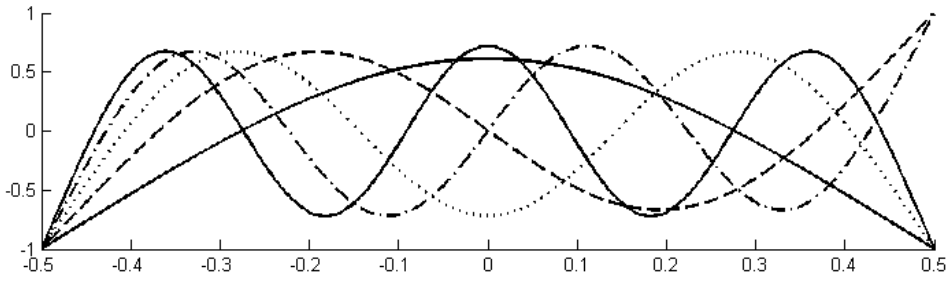

FIG. 7 - *First eigensolutions of beam equation with or without elastic support.*

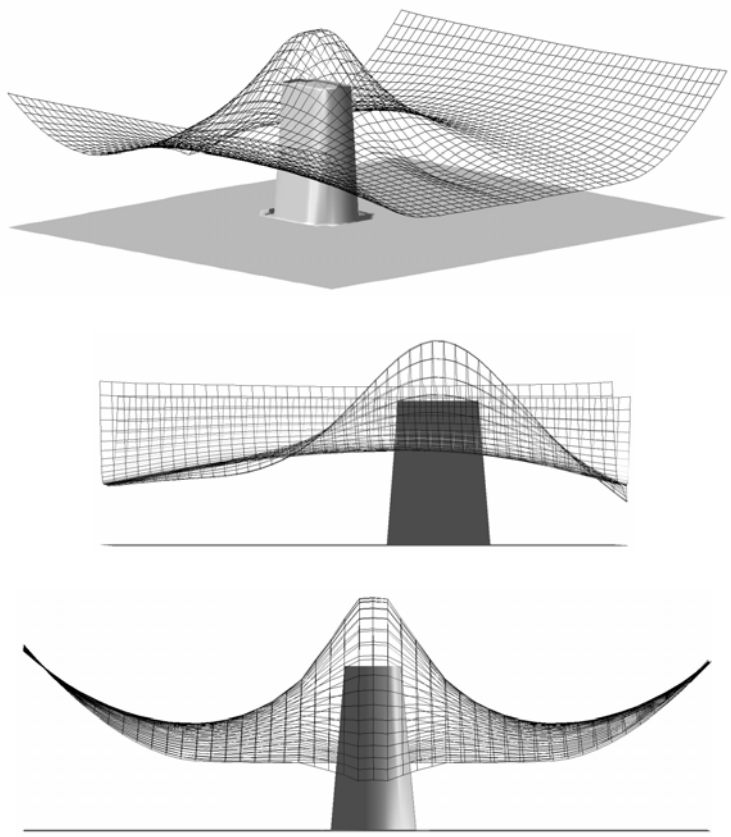

FIG. 8 - *FTire belt distortion when tire envelopes small out-of-center obstacle (vertical displacement 3-fold magnified). Perspective view, front view, and side view.*

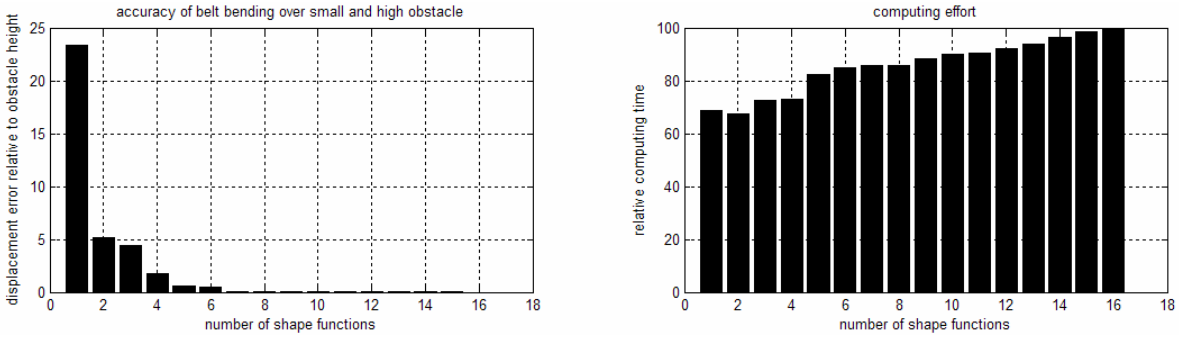

FIG. 9 - *Accuracy and computing effort of belt bending model vs. number of shape functions (beam eigensolutions).*

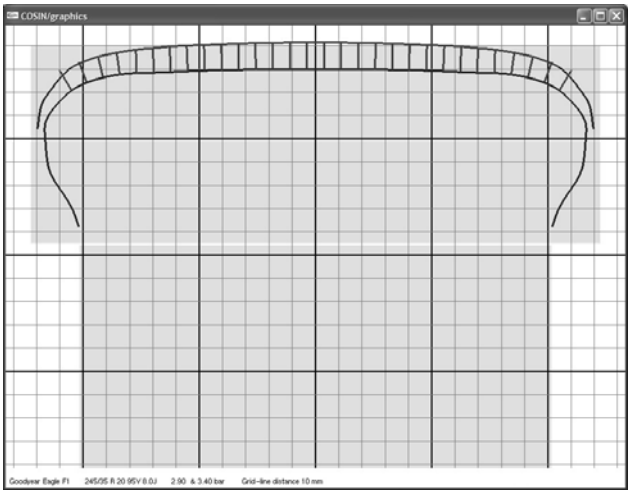

FIG. 10 - *Visualization of cross section geometry and contact element placement in FTire/fit*

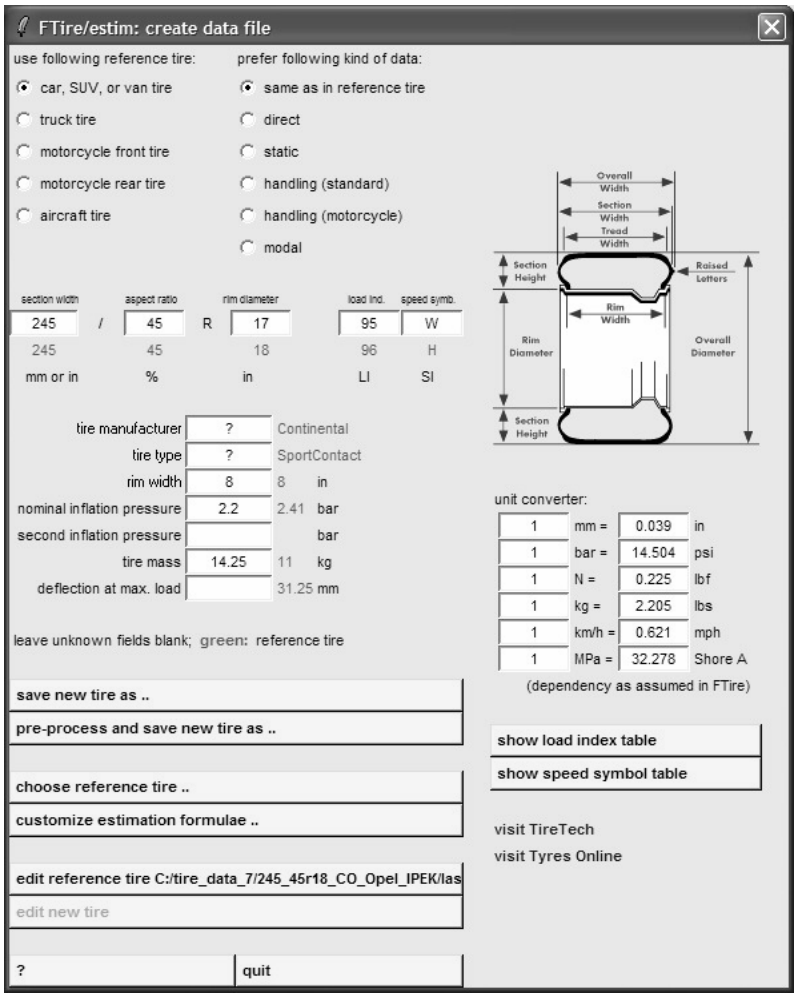

FIG. 11 - *FTire/estimate user interface.*

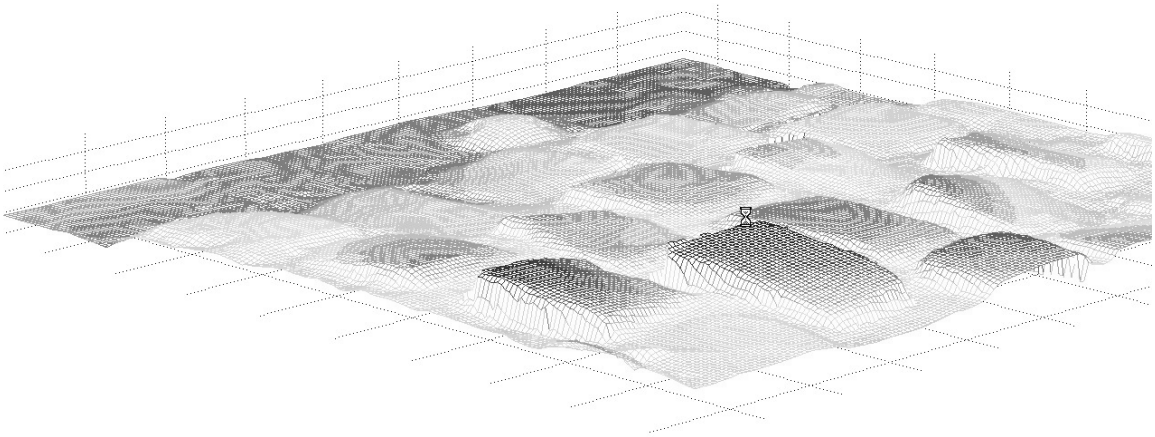

FIG. 12 - *Section of a Belgian block road, described by an RGR road data file.*

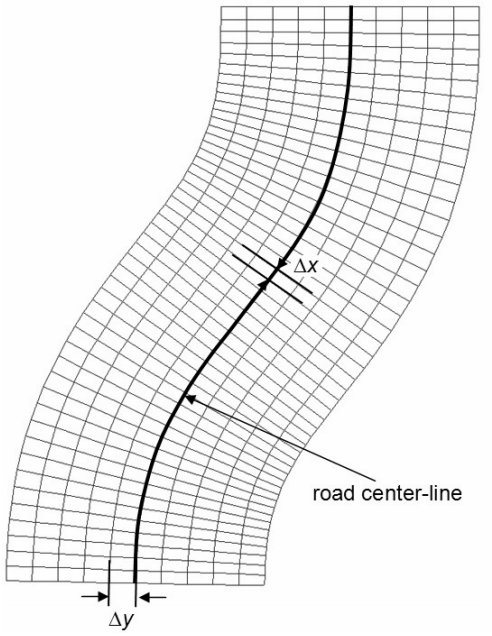

FIG. 13 – *Curvilinear mesh of RGR roads.*