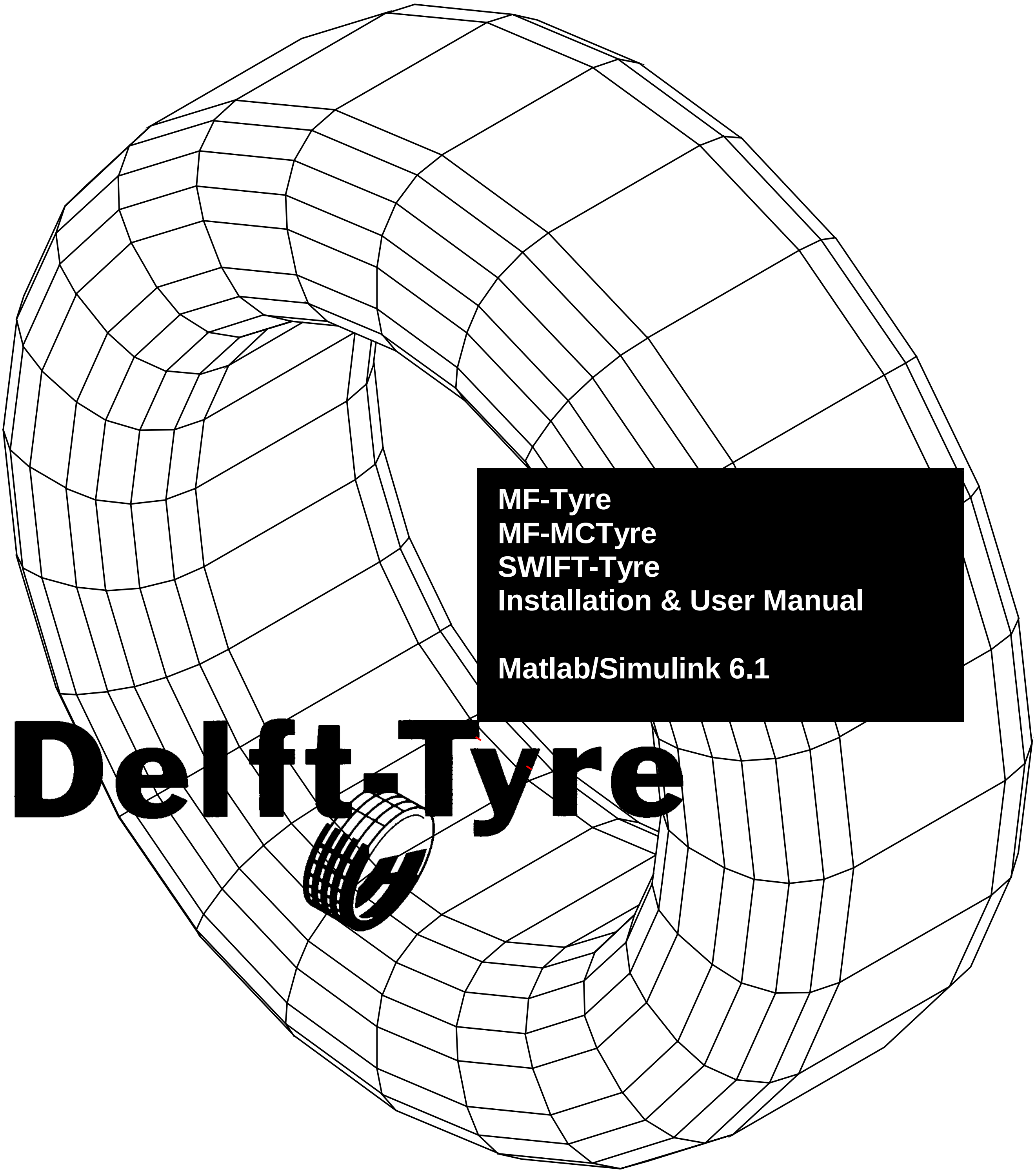


MF-Tyre
MF-MCTyre
SWIFT-Tyre
Installation & User Manual

Matlab/Simulink 6.1

Delft-Tyre

MF-Tool(+), MF-MCTool(+), MF-(MC)Tool(+), MF-Tyre, MF-MCTyre & SWIFT-Tyre are part of the DELFT-TYRE product line, developed at TNO Automotive, Delft, The Netherlands.

Delft-Tyre

MF-Tyre
MF-MCTyre
SWIFT-Tyre

TNO Automotive

Installation and Users manual

Matlab/Simulink

Date
April 2003

Versions

MF-Tyre	5.2
MF-MCTyre	1.1
SWIFT-Tyre	1.1
MATLAB	6.1

Document revision
2.1

Number of pages
19

Number of appendices
0

Number of figures
4

Number of tables
3

Netherlands Organization for
Applied Scientific Research (TNO)

CONTACT

For questions concerning this manual or support for using MF-Tyre/MF-MCTyre/SWIFT-Tyre users can contact the DELFT-TYRE representative via the address stated below.

TNO Automotive
Advanced Chassis & Transport Systems

attn. Mr. Jan J.M. van Oosten
PO box 6033
2600 JA Delft
The Netherlands

Telephone:	+31 15 269 6420
Fax:	+31 15 269 7314
E-mail:	delfttyre@wt.tno.nl
Url:	www.delft-tyre.com

Information about TNO Automotive:
Url: www.automotive.tno.nl

Contents

1 Installation software

1.1 Introduction

This chapter is intended to assist you in setting up and using the DELFT-TYRE tyre models with Matlab. The installation can be divided into two parts: set-up of the software and activation of the license mechanism.

The software installation is described in this document. An overview of each step of the procedure is given in chapter 1.2.

For information regarding the licensing mechanism, you are referred to the document entitled '*License Mechanism Installation Manual – UNIX and Windows*' [2].

1.2 Installing the software

1.2.1 Overview

The installation procedure can be divided into four steps:

1. extracting the DELFT-TYRE software
2. extracting the license software (see ref. [2])
3. obtaining a license and setting up the license mechanism (see ref. [2])
4. testing the software installation

Each of these steps will be described in detail in the following paragraphs.

1.2.2 Extracting the DELFT-TYRE software

The MF-Tyre/MC-Tyre and SWIFT-Tyre software is contained in a zipped archive named `'dt_matlab6_xxxx.zip'` (where xxxx = version number).

The licensing files for using parts or all of the software can be found in a separate file, see ref. [2].

It is advised to place the DELFT-TYRE software in a new empty directory, preferably under the 'Toolbox' directory of your Matlab installation directory. For example in the directory
`C:\MatlabR12.1\Toolbox\DelftTyre\`

This directory must be added to the Matlab search path in order for Matlab to be able to find the model files of the Delft-Tyre software. You can use the Matlab path browser for this, or you can use the Matlab command 'addpath':

```
addpath  'C:\Matlab\Toolbox\DelftTyre'
```

Surround the name in quotes if it contains any spaces. Please note that additional files like tyre property files (used with the models) will not be searched in the Matlab path, so these files need to be copied to the current work directory from where you start your models, if your working directory differs from the Delft-Tyre installation directory.

1.2.3 Obtaining a license and setting up the license mechanism

The licensing mechanism makes use of a license file, named 'mftool.lic'. You can obtain the license keys from TNO Automotive or from your local DELFT-TYRE representative by returning the completed and signed Receipt form.

In order to obtain a license, some information about the machine on which the software is going to be used is needed.

For a counted license (limited number of users), the license manager daemon 'lmgrd.exe' must be running.

You can choose to start the daemon by hand each time you login (or call it automatically from a batch file), or install it as a Windows NT service. For the latter option, administrator privileges are needed.

These items are discussed further in ref. [2].

1.2.4 Testing the software installation

After the software is in place and the licensing has been set up, it is strongly advised to test the installation. A number of Matlab test files (`test_*.m`) are available, try running these before building your own models. Detailed descriptions can be found in chapter 2.1

2 Users info

2.1 Introduction

This manual describes the implementation of the DELFT-TYRE MF-Tyre (car/truck) & MF-MCTyre (motorcycle) tyre and SWIFT-Tyre models in MATLAB/Simulink.

Tyre Models

MF-Tyre models for both car/truck and for motorcycle tyres are included in the library functions. The differences between the car/truck tyre model (MF-Tyre 5.2) and the motorcycle tyre model (MF-MCTyre 1.1) are minor from a practical point of view: the required model is selected automatically based on the set of tyre parameters that is supplied to the tyre model. Please read “MF-MCTyre” for “MF-Tyre” whenever appropriate.

SWIFT-Tyre is suitable for car/truck tyre simulation, and consists of a rigid ring model for the tyre dynamics and the Magic Formula model for contact slip behaviour. Next to the full SWIFT functionality, Steady State and Transient behaviour as modelled in MF-Tyre can be selected.

It is possible to use a mix of different tyre models in a SIMULINK simulation model.

Available commands and models

The following command line functions are at your disposal in MATLAB:

- `mfread` - for reading tyre property files into the Matlab workspace
- `mfeval` - used for Magic Formula evaluation

For on-line help on these commands, type `help mfeval` or `help mfread`

In Simulink, the library with MF-Tyre and Swift blocks can be opened with the command `dtlib`. Four blocks are available:

- `mftyre_cpi` - MF-Tyre using the contact point interface (CPI)
- `mftyre_sti` - MF-Tyre using the standard tyre interface (STI)
- `swift_sti` - SWIFT-Tyre using the standard tyre interface (STI)
- `position and orientation` - transformation block to calculate position and transformation matrix from wheel velocities.

Online help for can be obtained by selecting a block and selecting ‘help’ in the right mouse popup menu.

For testing purposes a number of additional command line functions are available:

- `test_mfteval` – plot of tyre cornering characteristics
- `test_mftyre_cpi` – simulation of a side slip angle sweep
- `test_mftyre_sti` – simulation of a side slip angle sweep
- `test_swift_sti` – traversing an uneven road surface with constant axle height

You can run these models to see how the delft-tyre blocks can be used and what output they generate. **Please note that the tyre property files used with these models need to be in your current working directory, otherwise the models will not be able to run properly.**

Units and sign convention
As with all DELFT-TYRE products the ISO sign convention for slip definitions and tyre forces is used. The default units used in the calculations are SI (kg, N, m, s, radians). Consequently the side slip angle α and the camber angle γ for the Magic Formula evaluation (`mfeval,mftyre_cpi`) have to be specified in radians. For the longitudinal slip κ the dimensionless definition is used, where κ = -1 corresponds to wheel lock.

ADAMS compatibility
The tyre property files (e.g. `mft_car175_70R13.tir`) and road data file for the SWIFT model (`swt_road.rdf`) are the same for ADAMS and all other DELFT-Tyre implementations.

2.2 MFREAD function

The Magic Formula coefficients can be exported in a tyre property file using MF-Tool. To read the tyre data from this file you can use `mfread`. This command reads the contents of a tyre property file, and stores this data in an array, which is accessible in your Matlab workspace.

The command syntax is:

```
typarr = mfread (<tyre_property_file>).
```

The data stored in `typarr` can be used for the function `mfeval` and Simulink blocks `mftyre_cpi` and `mftyre_sti`. Please note that the tyre property file must be available in your current working directory, or that the file location also includes the absolute path, otherwise `mfread` will not be able to locate the file.

Please remember the following:

- Using strings with MATLAB requires single quotes before and after the string (‘).
- File location of a tyre property file needs to be absolute if the file does not exist in the current working directory.
- The Simulink block `swift_sti` does not require using `mfread;` the tyre property file name has to be specified directly in the block mask.

2.3 MFEVAL function

The function `mfeval` evaluates the Magic Formula for a series of input variables. The input consists of vertical tyre load, slip angle, longitudinal slip and camber and the forces and moments in the tyre contact area are calculated as a function of these input variables. The function will prove to be particularly useful for visualisation of the tyre characteristics, as is shown in the example `test_mfeval`. The function `mfeval` accepts vectorised input; working in this way speeds up the calculation significantly with respect to a user programmed do-loop.

The syntax is:

```
forces = mfeval(inputs, typarr, use_mode)
```

where:

- `inputs` is a matrix containing the input signals for the Magic Formula. It has to consist of four columns. In the first column the normal load *Fz* should be specified, in the second column the slip angle α, in the third column the longitudinal slip κ, and in the fourth column the camber angle γ.
- `typarr` is the variable containing the Magic Formula coefficients, which can be obtained using the `mfread` command.
- `use_mode` specifies the operating mode. For the function `mfeval` no transient calculations are possible, the allowed values are –4 to 4. For more details on `use_mode` see section 2.9.

For every row of the matrix `inputs` the forces are calculated. The matrix `forces` will have six columns and as many rows as the matrix `inputs`. The first column contains the longitudinal force *Fx*, second column: lateral force *Fy*, third column: normal force *Fz*, fourth column: overturning moment *Mx*, fifth column: rolling resistance torque *My*, sixth column: self aligning moment *Mz*.

2.4 Simulink blocks: the CPI and STI interface

In Simulink two different interfaces are available:

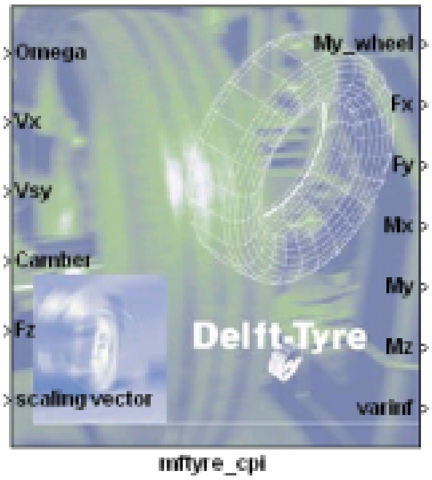
- ***"Contact Point Interface" (CPI), MF-Tyre***
  The interface is defined at the tyre contact point center. Tyre normal force, camber angle, wheel angular velocity, etc. are used to calculate the tyre forces. This interface is aimed at the development of relatively simple vehicle models, without unnecessary overhead in the calculations. Also the experienced user who wants to have more control over the tyre model implementation may select this interface.

- ***"Standard Tyre Interface" (STI), MF-Tyre and SWIFT-Tyre***
  The interface is defined at the wheel center. Orientation and velocity of the wheel center are used to calculate slip and camber angle. The tyre normal force is calculated as a function of wheel center position and road parameters. The slip forces, moments generated by the tyre are translated again to the wheel center.

  The STI interface requires that the velocities, positions and transformation matrices are consistent. This is particularly important for the SWIFT model, which may give incorrect results if this condition is not met. The additional block `position and orientation` performs an integration of the local (angular) velocities and calculates the corresponding position and transformation matrix for the wheel centre.

2.5 MFTYRE_CPI block

An illustration of the block MFTYRE_CPI in MATLAB/Simulink is shown in figure 1. The purpose of this block is to calculate the forces generated by the tyre in the tyre contact point center as a function of slip, vertical load and camber.



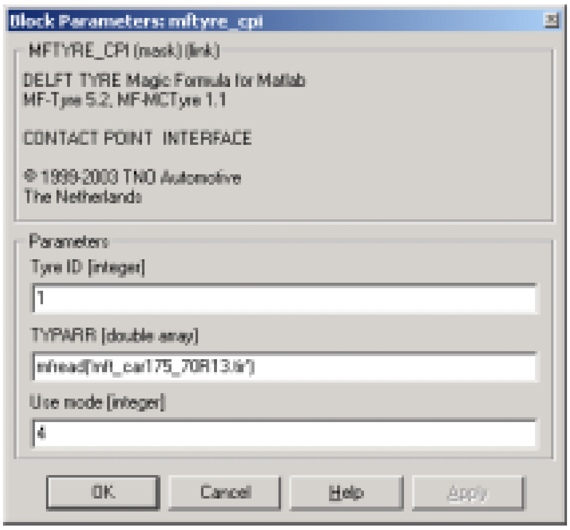


figure 1: mftyre_cpi block

The following inputs have to be supplied:

- **Omega**: wheel angular velocity about the wheel spin axis
- **Vx**: forward velocity of the wheel centre in the wheel plane, parallel to the road surface.
- **Vsy**: lateral velocity at the ground contact point parallel to the road surface and perpendicular to the wheel plane.
- **camber**: inclination angle of the wheel with respect to the road surface
- **Fz**: tyre normal load
- **scaling vector**: contains 7 coefficients to **scale** the following scaling factors (which are defined in the tyre property file):

Vector index	scaling factor	scales
1	LMUX	peak longitudinal friction coefficient
2	LKX	longitudinal slip stiffness
3	LMUY	peak lateral friction coefficient
4	LKY	cornering stiffness
5	LGAY	camber stiffness lateral force
6	LTR	pneumatic trail
7	LGAZ	Camber stiffness aligning moment

Table 1: definition of the scaling vector

Note: The values in the scaling vector **do not replace** the scaling factors from the tyre property file. The scaling factor that will be used in simulation is the **product** of the scaling factor in the tyre property file and the corresponding coefficient in the scaling vector. So, if no additional scaling - other than defined in the tyre property file - needs to be applied, a vector of [1 1 1 1 1 1 1] should be used.

The outputs are:
- **My_wheel**: resultant moment on the rim about the wheel spin axis
- **Fx**: longitudinal force at the tyre contact point
- **Fy**: lateral force at the tyre contact point
- **Mx**: overturning moment at the tyre contact point
- **My**: rolling resistance moment at the tyre contact point
- **Mz**: self aligning moment at the tyre contact point
- **varinf**: extra output variables concerning tyre behaviour (e.g. slip angle), see section 2.10.

By double clicking on the `mftyre_cpi` block in Simulink, the following parameters should be provided for each tyre:
- **Tyre ID** is an integer value identifying the tyre block. This number will be printed when messages related to the tyre block are printed. Please take care that the numbers within a model are unique.
- **TYPARR** is the array containing the Magic Formula parameters. It is also possible to call the `mfread` function directly, e.g. `mfread('mft_car175_70R13.tir').`
- **Use mode** defines the operating mode of the tyre model, see section 2.9.

2.6 MFTYRE_STI block

An illustration of the block MFTYRE_STI in MATLAB/Simulink is shown in figure 2. The purpose of this block is to calculate the forces and moments at the wheel center generated by the tyre as a function of the kinematic properties of the wheel center: position, orientation, and motion: (angular) velocity. The interface follows the definitions of the Standard Tyre Interface as developed by the TYDEX Workgroup. The rotation of the wheel about the wheel spin axis may be introduced by a relative rotation angle of the wheel with respect to a carrier axis system (**angtwc**, **omegar**) and/or by having a rotating carrier frame. Road irregularities with a wavelength in excess of the tyre diameter may be introduced by specifying height and slope as a function of the wheel position.

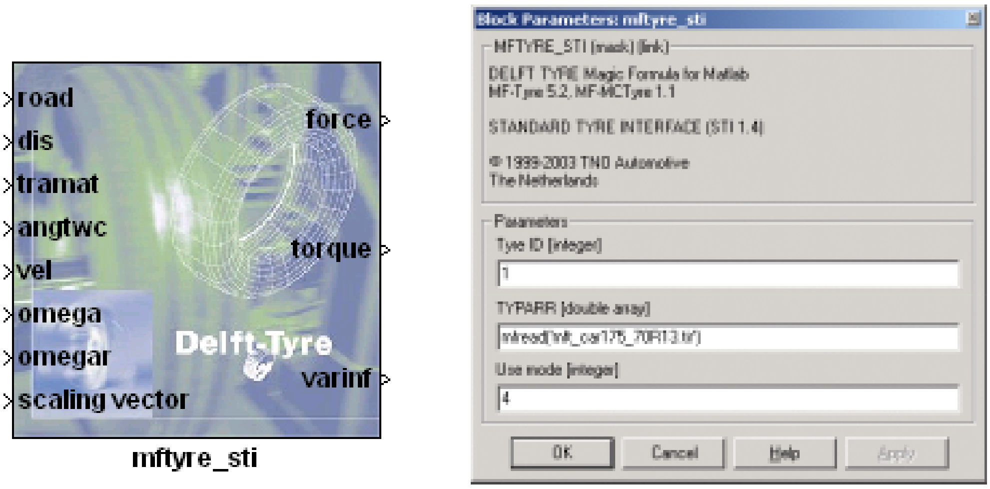


figure 2: mftyre_sti block

The following inputs have to be supplied:

- **road**: a vector with three components:
  1- z-value of the road surface
  2- dz/dx: slope of the road in x-direction
  3- dz/dy: slope of the road in y-direction
  All road data has to be specified in the global frame. If road undulations are to be included these values can be made a function of the location of the wheel. The contact algorithm is not suited for obstacles with a short wavelength and requires that the z-location of the road be below the wheel centre. For short wavelength obstacles the use of SWIFT -Tyre is recommended, see section 2.7.
- **dis**: a vector with three components: the x, y and z-coordinate of the wheel carrier in the global axis system.
- **tramat**: a vector with nine components to specify the transformation from wheel carrier axis system to the global axis system. Following formula applies for a transformation from the local frame to the global frame:

$$\begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} = \begin{bmatrix} tramat(1) & tramat(4) & tramat(7) \\ tramat(2) & tramat(5) & tramat(8) \\ tramat(3) & tramat(6) & tramat(9) \end{bmatrix} \cdot \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} \qquad (1)$$

  So tramat(1..3) is the representation of the unit x-vector of the carrier axis system in the global frame. The same applies to tramat(4..6) being the unit y-vector and tramat(7..9) being the unit z-vector of the carrier axis system expressed in the global frame.

- **angtwc**: rotation angle of the wheel with respect to the wheel carrier about the wheel spin axis (currently not used by MF-Tyre/MF-MCTyre).
- **vel**: a vector with three components: the global velocity of the wheel centre expressed in the wheel carrier local frame (x, y and z component).
- **omega**: a vector with three components: the global angular velocity of the wheel centre expressed in the wheel carrier local frame (x, y and z component).
- **omegar**: relative angular velocity of the wheel with respect to the wheel carrier about the wheel spin axis.
- **scaling vector**: contains 7 coefficients to **scale** the following scaling factors (which are defined in the tyre property file):

vector index	scaling factor	scales
1	LMUX	peak longitudinal friction coefficient
2	LKX	longitudinal slip stiffness
3	LMUY	peak lateral friction coefficient
4	LKY	cornering stiffness
5	LGAY	camber stiffness lateral force
6	LTR	pneumatic trail
7	LGAZ	camber stiffness aligning moment

Table 2: definition of the scaling vector

The outputs are:
- **force**: forces applied by the tyre onto the rim at the centre of the wheel. The force vector has three components in the accompanying x, y and z direction of the carrier frame.
- **torque**: moments applied by the tyre onto the rim at the centre of the wheel, expressed in the carrier axis system. Torque as three components representing the moments in the x, y and z direction
- **varinf**: extra output variables concerning tyre behaviour (e.g. slip angle), see section 2.10.

By double clicking on the `mftyre_sti` block in Simulink, the following parameters should be provided for each tyre:
- **Tyre ID** is an integer value identifying the tyre block. This number will be printed when messages related to the tyre block are printed. Please take care that the numbers within a model are unique.
- **TYPARR** is the array containing the Magic Formula parameters. It is also possible to call the `mfread` function directly, e.g. `mfread('mft_car175_70R13.tir').`
- **Use mode** defines the operating mode of the tyre model, see section 2.9.

2.7 SWIFT_STI block

If you want to set the tyre property file for a SWIFT Simulink block, the file location must be entered in the block parameters setting of the block itself. The data is read from within the corresponding S-Function and is not dependent on the `typarr` array. `mfread` however can still be used to load the SWIFT parameters in the current working directory for your own viewing using `mfeval`.

An illustration of the block SWIFT_STI in MATLAB/Simulink is shown in figure 3. The purpose of this block is to calculate the forces and moments at the wheel center generated by the tyre as a function of the kinematic properties of the wheel center: position, orientation, and motion: (angular) velocity. The interface follows the definitions of the Standard Tyre Interface as developed by the TYDEX Workgroup. The rotation of the wheel about the wheel spin axis may be introduced by a relative rotation angle of the wheel with respect to a carrier axis system (**angtwc**, **omegar**) and/or by having a rotating carrier frame. Road irregularities with a wavelength in excess of the tyre diameter may be introduced by specifying height and slope as a function of the wheel position, otherwise effective inputs should be used.

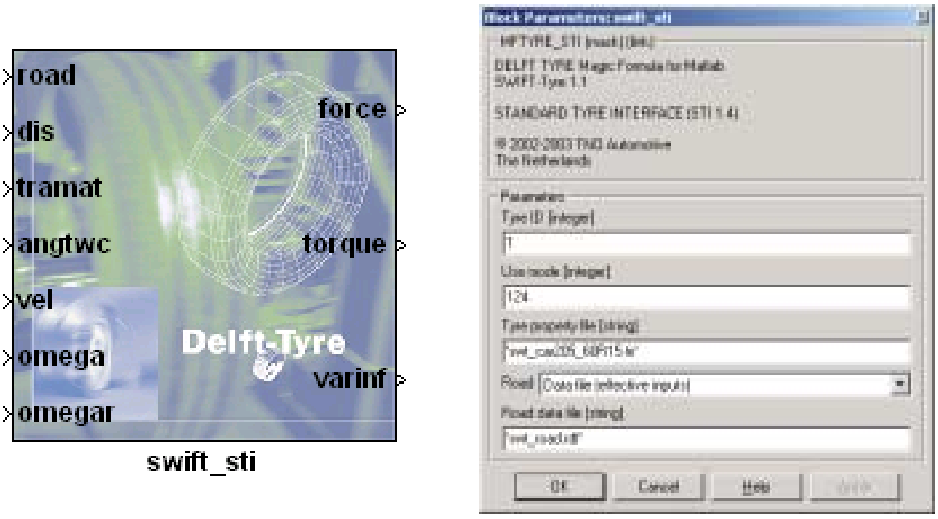


figure 3: swift_sti block

The following inputs have to be supplied:

- **road:** present for compatibility with MF-Tyre. In SWIFT-Tyre, the road profile must be defined in an ASCII file format when the check box 'Effective inputs' is selected (see e.g. file `swt_road.rdf`) In the road property file the road height is specified as a function of travelled distance. In the road data file a left and right track data are specified. The appropriate track data is selected depending on the value of TYRE ID: even right track, uneven left track.
  SWIFT uses a zero-order sample and hold when evaluating the road profile. Changes in the height of the road profile are interpreted as steps. For maximum accuracy it is important that the sample points coincide with the data provided by the user, otherwise interpolated data will be used. So you should use road data with a fixed sample interval and specify this value for ROAD_INCREMENT in the [MODEL] section of the tyre property file. Minimum value of the road increment is 0.01 m.
- **dis**: a vector with three components: the x, y and z-coordinate of the wheel carrier in the global axis system.
- **tramat**: a vector with nine components to specify the transformation from wheel carrier axis system to the global axis system. Following formula applies for a transformation from the local frame to the global frame:

$$\begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} = \begin{bmatrix} tramat(1) & tramat(4) & tramat(7) \\ tramat(2) & tramat(5) & tramat(8) \\ tramat(3) & tramat(6) & tramat(9) \end{bmatrix} \cdot \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} \qquad (2)$$

So tramat(1..3) is the representation of the unit x-vector of the carrier axis system in the global frame. The same applies to tramat(4..6) being the unit y-vector and tramat(7..9) being the unit z-vector of the carrier axis system expressed in the global frame.

- **angtwc**: rotation angle of the wheel with respect to the wheel carrier about the wheel spin axis
- **vel**: a vector with three components: the global velocity of the wheel centre expressed in the wheel carrier local frame (x, y and z component).
- **omega**: a vector with three components: the global angular velocity of the wheel centre expressed in the wheel carrier local frame (x, y and z component).
- **omegar**: relative angular velocity of the wheel with respect to the wheel carrier about the wheel spin axis.

The outputs are:

- **force**: forces applied by the tyre onto the rim at the centre of the wheel. The force vector has three components in the accompanying x, y and z direction of the carrier frame.
- **torque**: moments applied by the tyre onto the rim at the centre of the wheel, expressed in the carrier axis system. Torque as three components representing the moments in the x, y and z direction
- **varinf**: extra output variables concerning tyre behaviour (e.g. slip angle), see section 2.10.

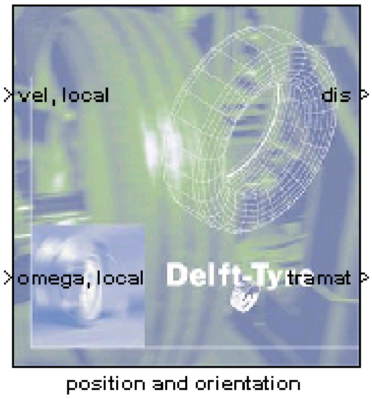
When running the SWIFT-Tyre block, an `error.txt` file is created in the current working directory. This file will contain some information and errors which may occur during simulation run.

By double clicking on the SWIFT-Tyre block in Simulink, the following parameters should be provided for each tyre (see figure 3).

- **Tyre ID** is an integer value identifying the tyre block. This number will be printed when messages related to the tyre block are printed. Please take care that the numbers within a model are unique. In case effective inputs is used the following applies: Odd tyre id's will use the left track data in the data file while even tyre id's will use the right track data from the given data file.
- **Use mode** defines the operating mode of the tyre block. See section 2.9.
- **Tyre property file** is the file name of the tyre property file, absolute or relative to the current working directory.
- **Road** enables you to choose road input via file [Data file (effective inputs)] or as a smooth road [external inputs (smooth road)].
- **Road data file** contains the file location, absolute or relative to the current working directory.

2.8 POSITION AND ORIENTATION block

In addition to the STI interface blocks, a Position and orientation block is available in the MF-Tyre library, which transforms the local translational and angular velocity of the wheel center to the global position and transformation matrix for use with STI interface blocks.



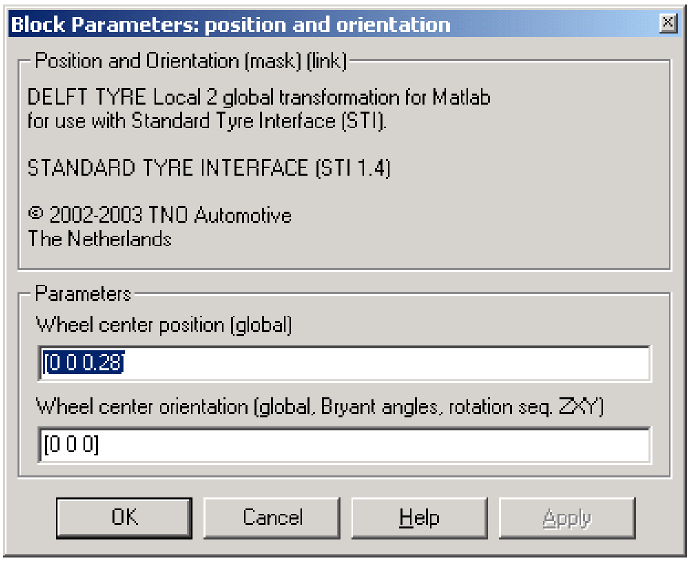


figure 4: Position and orientation block

The following inputs have to be supplied:

- **vel, local(3)**: local velocity of the wheel center expressed in the carrier frame: x, y and z component
- **omega, local (3)**: local angular velocity of the wheel carrier expressed in the carrier frame: x, y and z component

The outputs are:

- **dis(3)**: location of the wheel center in the global frame: x, y and z coordinate
- **tramat(9)**: transformation matrix from wheel carrier to the global frame
  tramat(1..3): representation of the unit x-vector of the carrier axis in the global frame
  tramat(4..6): representation of the unit y-vector of the carrier axis in the global frame
  tramat(7..9): representation of the unit z-vector of the carrier axis in the global frame

By double clicking on the Simulink block, the following parameters should be provided**:**

- **init_pos**: Initial wheel center position (global x, y and z coordinate)
- **init_angle**:Initial wheel center orientation (global, Bryant angles, rotation seq. ZXY)

2.9 Tyre model operating mode: USE_MODE

The possible values for USE_MODE are given in table 3.

Operating mode	steady-state	transient	SWIFT
Vertical spring only	0	0	20
Longitudinal only (Fx, My)	1	11	21
Lateral only (Fy, Mx, Mz)	2	12	22
Uncombined forces, moments (Fx, Fy, Mx, My, Mz)	3	13	23
Combined forces, moments (Fx, Fy, Mx, My, Mz)	4 (default)	14	24 (default)
Mirroring of lateral characteristics	multiply USE_MODE with -1		
SWIFT statics			+ 100

table 3: Definition of USE_MODE

So use_mode = -124 means
- combined forces and moments
- rigid ring dynamics included (SWIFT)
- initial static calculation
- mirrored tyre characteristics

2.10 Simulink tyre model output: VARINF

The definition of the VARINF array is given in table 4. Note that not all values are currently in use. Furthermore there is a difference between de CPI and STI interface: this originates form the fact that in the CPI-tyre interface no road contact point calculation is made.

Nr.	Description	MFTYRE_CPI	MFTYRE_STI	SWIFT_STI
1	Slip angle (steady state definition)	√	√	√
2	Longitudinal slip (steady state definition)	√	√	√
3	Camber angle	√	√	√
4	Longitudinal force Fx in contact point	√	√	√
5	Lateral force Fy in contact point	√	√	√
6	Vertical force Fz in contact point	√	√	√
7	Overturning moment Mx in contact point	√	√	√
8	Rolling resistance moment My in contact point	√	√	√
9	Self aligning moment Mz in contact point	√	√	√
10	Radial deflection	√	√	√
11	Deflection velocity (normal to ground)		√	√
12	Effective rolling radius	√	√	√
13	x-coordinate contact point position (global frame)		√	√
14	y-coordinate contact point position (global frame)		√	√
15	z-coordinate contact point position (global frame)		√	√
16	x-component of the normal to the road (global frame)		√	√
17	y-component of the normal to the road (global frame)		√	√
18	z-component of the normal to the road (global frame)		√	√
19	Longitudinal relaxation length	√	√	√
20	Lateral relaxation length	√	√	√
21	Friction coefficient x	√	√	√
22	Friction coefficient y	√	√	√
23	Not used			
24	Longitudinal slip velocity Vsx		√	√
25	Lateral slip velocity Vsy		√	√
26	Wheel forward velocity Vx		√	√
27	Derivative of longitudinal deformation du/dt	√	√	√
28	Derivative of lateral deformation dv/dt	√	√	√
29	Longitudinal slip (used in Magic Formula evaluation)	√	√	√
30	Slip angle (used in Magic Formula evaluation)	√	√	√
31	Pneumatic trail	√	√	√
32	Residual moment Mz,res	√	√	√
33	Moment arm of Fx	√	√	√
34	Gyroscopic moment Mz,gyr	√	√	√
35	Braking induced plysteer force	√	√	√
36	Distance traveled			√
37	Effective plane height			√
38	Effective plane angle			√
39	Effective plane curvature			√
40	contact length			√

table 4: Definition of the VARINF-array

3 References

[1] A. Riedel, J.J.M. van Oosten [1997] – *Standard Tyre Interface, Release 1.4.* Presented at 2nd International Colloquium on Tyre Models for Vehicle Dynamics Analysis, February 20-21, 1997. Issued by the TYDEX Working group.

[2] *License Mechanism Installation Manual – UNIX and Windows*.
February 2002, TNO Automotive, Delft, The Netherlands.