

SIMPACK Code Export

Frank Kohlschmied

Code Export & Simulink Interfaces

Intec GmbH



What Are You
Intending To Do?

SIMPACK Code Export



What Are You
Intending To Do?

Agenda

- ? What is Code Export ?
- ? Functionality
- ? Fields of Application
- ? Example
- ? Outview

SIMPACK Code Export

Generating equations of motion



What Are You
Intending To Do?



What is Code Export ?

Functionality

Fields of Applications

Example

Outview



Internally: equations are not visible for user

- + easy model access for GUI - pre-/post processor, solver
- + various analysing tools available
(linear system analysis, kinematics, equilibrium calc)
- + no compile or link necessary
- dependent on software installation
- difficult to integrate to other systems
- higher calculation time
- > model verification, analysing subsystems → offline simulation



Externally: equations are written to file in programming language

- + direct access to equations for user
- + easy to integrate into other systems
- + reduced calculation time
- external pre-/postprocessor necessary
- costly model modification
- > integration of verified subsystems in global simulation → online simulation



SIMPACK combines both possibilities of processing equations of mechanical systems and with that offers a unique process patency

SIMPACK Code Export

Code Generation



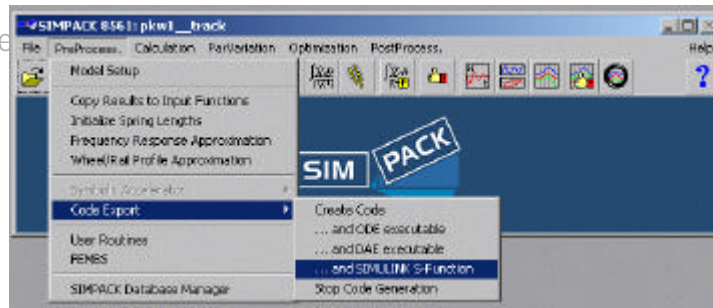
What is Code Export ?

Functionality

Fields of Applications

Example

Outview



```

SIMPACK Echo Area: Build Symbolic Code
test
for EXPLICIT DAE
** Generate code for MBS data
** Generate code for general RHS
** Generate code for general Y-Vector
** Generate code for ODE-Interface
** Generate code for MBS-Element parameters
** CPU-time for generating code: = 9.0999998E-02 [sec]

*****
* Backward-Code-Optimization: remove unused statements *
*****

** File =
symbolic_code_Rhs.f

** File =
symbolic_code_Y.f

** End of backward-optimization

*****
*** Forward-Optimization of symbolic Code ***
*****

** File =
symbolic_code_Rhs.f.opt
- optlev = 0: split in subroutines

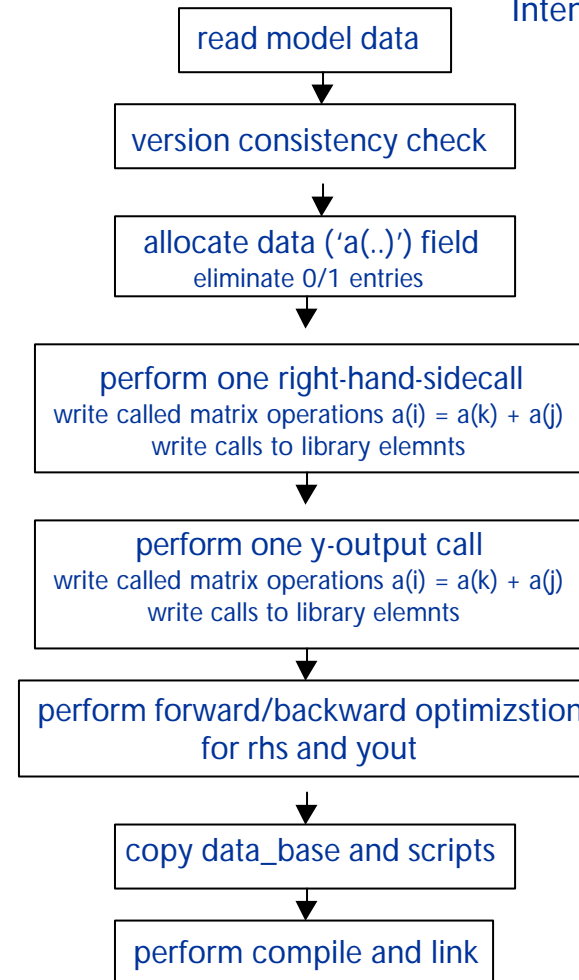
** File =
symbolic_code_Y.f.opt
- optlev = 0: split in subroutines

** End of code-forward-optimization

** Code-Optimization Statistics:
-----
for file: symbolic_code_Rhs.f.opt
number of removed lines = 362
number of codestaments = 76
number of subroutines = 1
for file: symbolic_code_Sen_Pos.f.opt
number of removed lines = 0
number of codestaments = 0
number of subroutines = 0
for file: symbolic_code_Y.f.opt
number of removed lines = 1
number of codestaments = 57
number of subroutines = 1
    
```

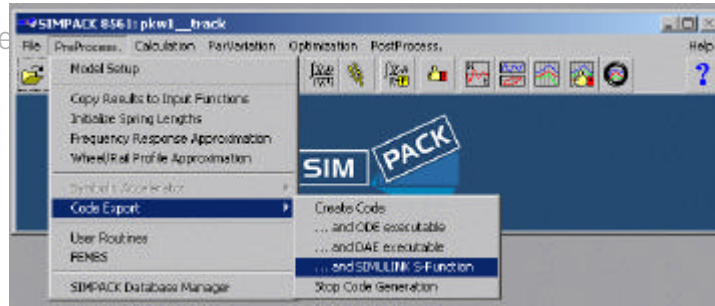


What Are You Intending To Do?



SIMPACK Code Export

Code Generation



What Are You Intending To Do?



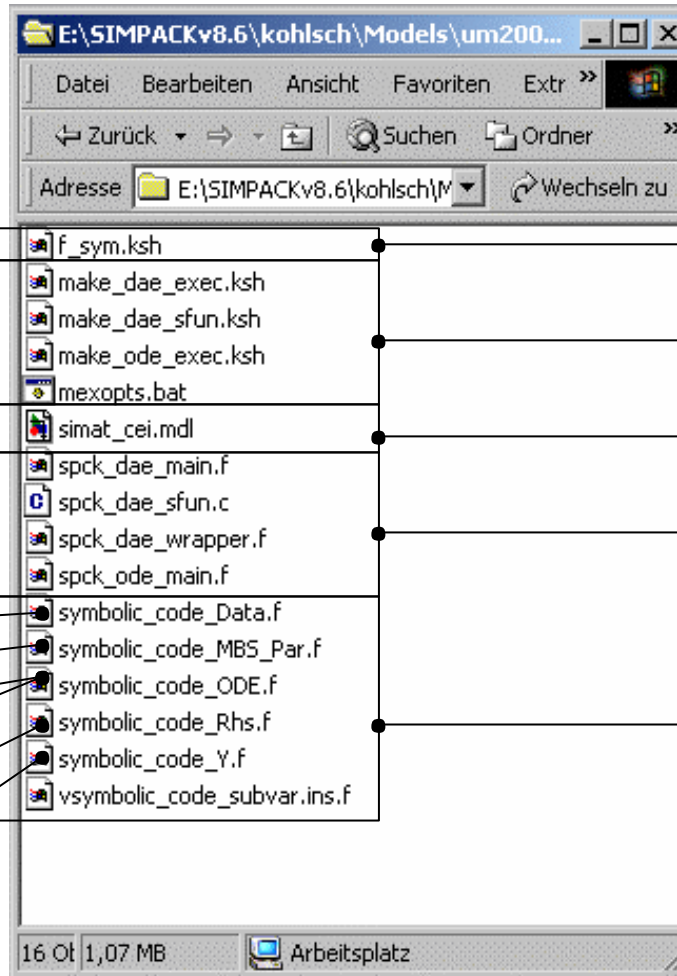
What is Code Export ?

Functionality

Fields of Applications

Example

Outview



new created directory:
 <model>.sym

[Examples for compile link & solver couplings](#)

platform-independent F90 compile script

application specific link scripts

default SIMULINK model

application specific frame routines

[model specific symbolic code](#)

m, I_{ij}, com

$P(i)$

$x(t=0)$

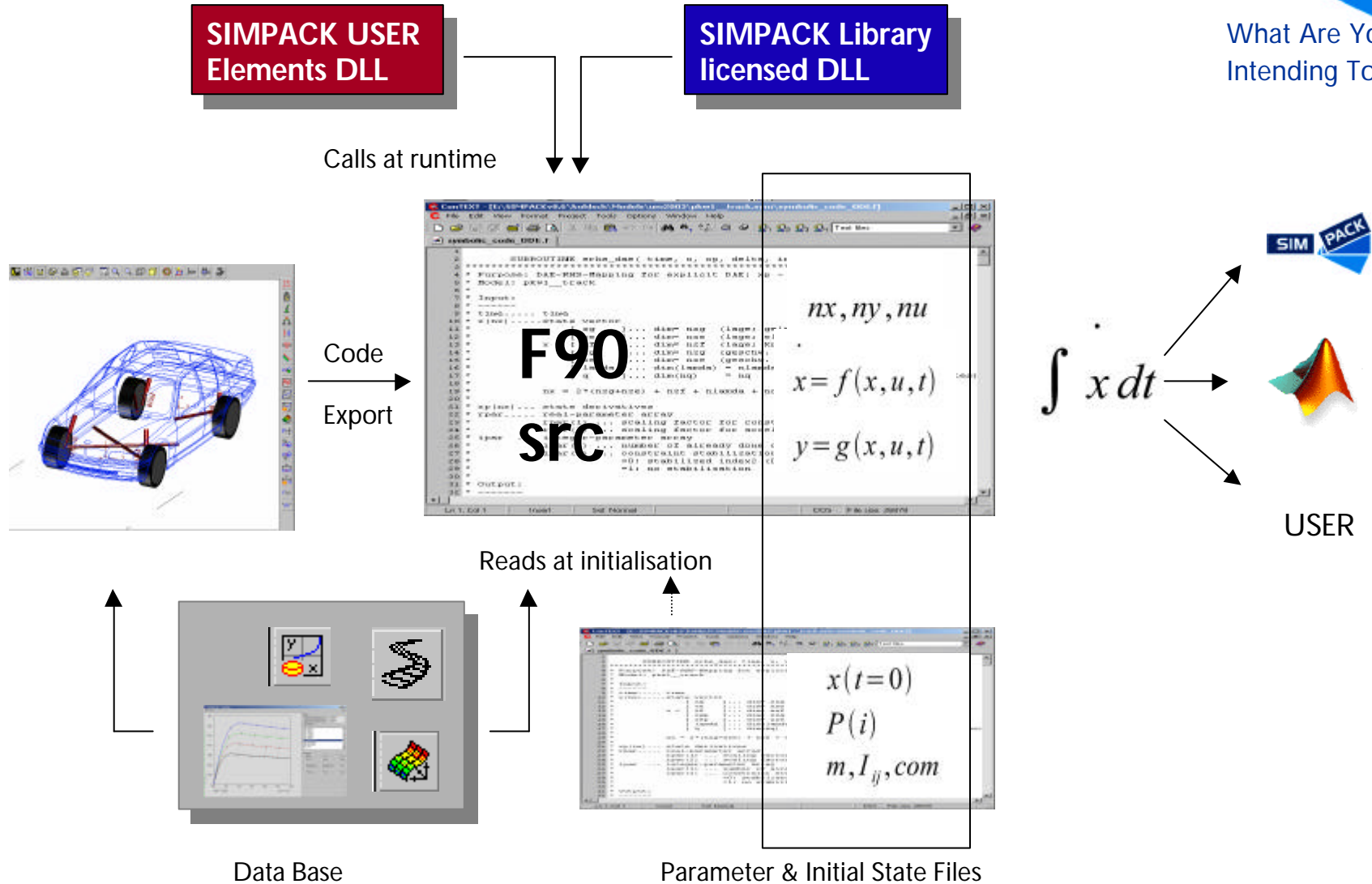
nx, nv, nu

$x = f(x, u, t)$

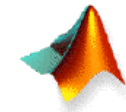
$y = g(x, u, t)$

SIMPACK Code Export

Code Execution



What Are You Intending To Do?



USER



What Are You
Intending To Do?

SIMPACK Code Export

What is Code Export ?



Functionality

Fields of Applications

Example

Outview

Code Export functionality was originally designed to handle time intensive applications like Parameter Variations and Optimisations

▶ SIMPACK Symbolic Code (ceased with v8.0)

▶ SIMPACK Code Export

- replaces external Symbolic Code to be used with arbitrary solver systems
- offered in several stages of functionality range
- functionality will be expanded according to customer needs
- Stage 0 available since 11/2002 with v8.5
- Stage 1 first minor release available since 04/2003 with v8.6

▶ SIMPACK Symbolic Accelerator (not yet available)

- replaces internal Symbolic Code for time intensive applications



What Are You
Intending To Do?

SIMPACK Code Export

Code Export Stages

What is Code Export ?

Code Export will be offered in different stages

▶ Functionality

Fields of Applications

Example

Outview

▶ Stage 0: functionality of v8.0 Symbolic Code

- + standard SIMPACK elements supported
- + static parameters
- + locally defined Input Functions
- + ...

▶ Stage 1: expanded Stage 0 functionality according to customer requests

- + Automotive+ Elements (Tyre 49, Track Joint, ...)
- + Functions from Data Base (Input Functions, IPF Arrays, Track)
- + dynamic parametrisation (elements, rigid body data, initial state)
- + elastic bodies
- + array base description of axis (elasto) kinematics
- + DAE-free macro-description of axis (elasto) kinematics
- + ...
- Wheel Rail
- Contact
- Root Functions



What Are You
Intending To Do?

SIMPACK Code Export

Special functionality Stage 1

What is Code Export ?

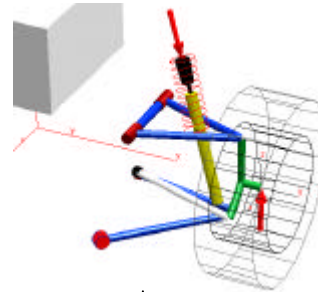


Functionality

Fields of Applications

Example

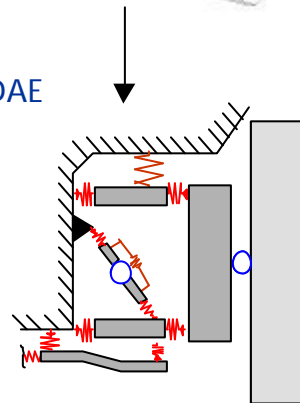
Outview



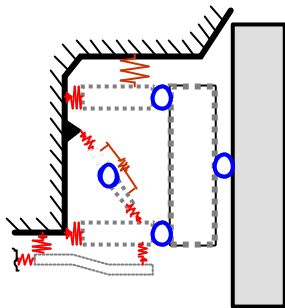
Physical wheel
(elasto) kinematic

*)

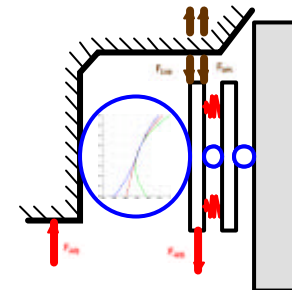
Conventional
SIMPACK ODE/DAE
Model



DAE – free Macro joint
for (elasto) kinematics



DAE – free
(elasto) kinematics
using pre-calculated
arrays



SIMPACK Code Export

Licensing

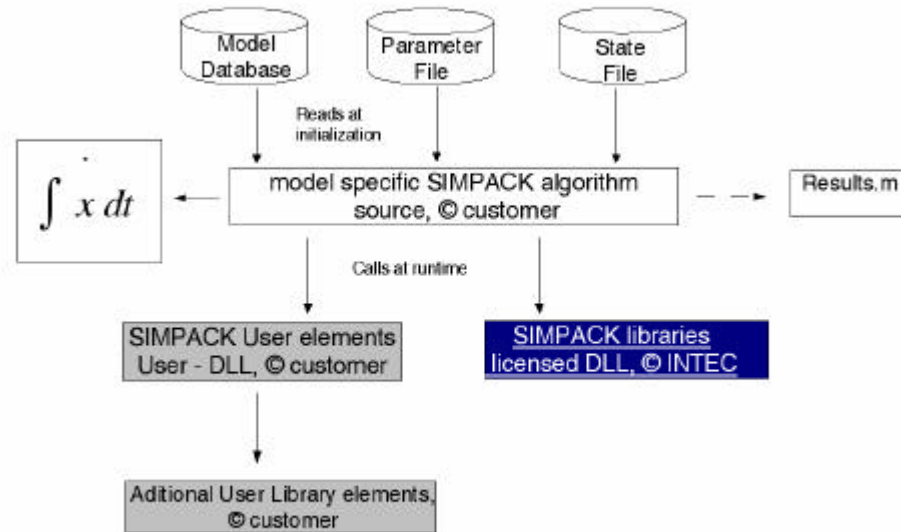
What is Code Export ?

Functionality

Fields of Applications

Example

Outview



- ▶ Model specific exported code is customer's property
- ▶ Code Generation: generation of code depending on stage level
- ▶ Code Execution: Usage of SIMPACK element library depending on stage level
- ▶ Source Code of Element Library: available within special agreement
- ▶ Solver: may be used with solver license



What Are You Intending To Do?



What Are You
Intending To Do?

SIMPACK Code Export

Advantages of model based on exported Code

What is Code Export ?

Functionality

▶ Fields of Application

Example

Outview

- ▶ Independent from any SIMPACK installation
- ▶ For execution no SIMPACK modelling/user experience is necessary
- ▶ Expandable to user needs, e.g Post Processing
- ▶ Reduced calculation time according to code optimisation
- ▶ No modelling know how is disclosed when model is handed on
- ▶ Code can be linked to arbitrary solvers:
 - User specific
 - external SIMPACK Solvers
 - comercial Solver systems, e.g. MATLAB
- ▶ Several options of coupling and communication to solver system are possible:
 - static or dynamic linking
 - function call or co-simulation
 - one or two process communication (socket, shared memory)
- ▶ Integrated modeling in offline (SIMPACK) and online (exported code) simulation

SIMPACK Code Export



Environment
Cartographic Road Track
Road Surface



What Are You
Intending To Do?

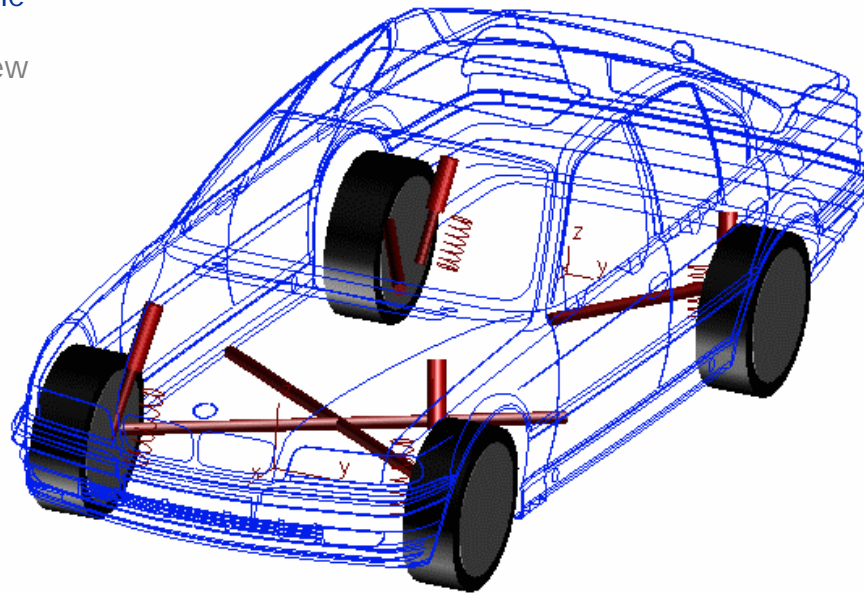
What is Code Export ?

Functionality

Fields of Application

▶ Example

Outview



Joints
19: Automotive Track Joint

...



Constraints
28: Massless Link



Force Elements
49: Automotive Tyre
Pacejka Similarity

...



Control Elements
168: Automotive Track Sensor



Input Function Set
Suspension Stiffness



Input Function Array
Tyre Friction Coefficient



SIMPACK Code Export



Environment
Cartographic Road Track
Road Surface



What Are You
Intending To Do?

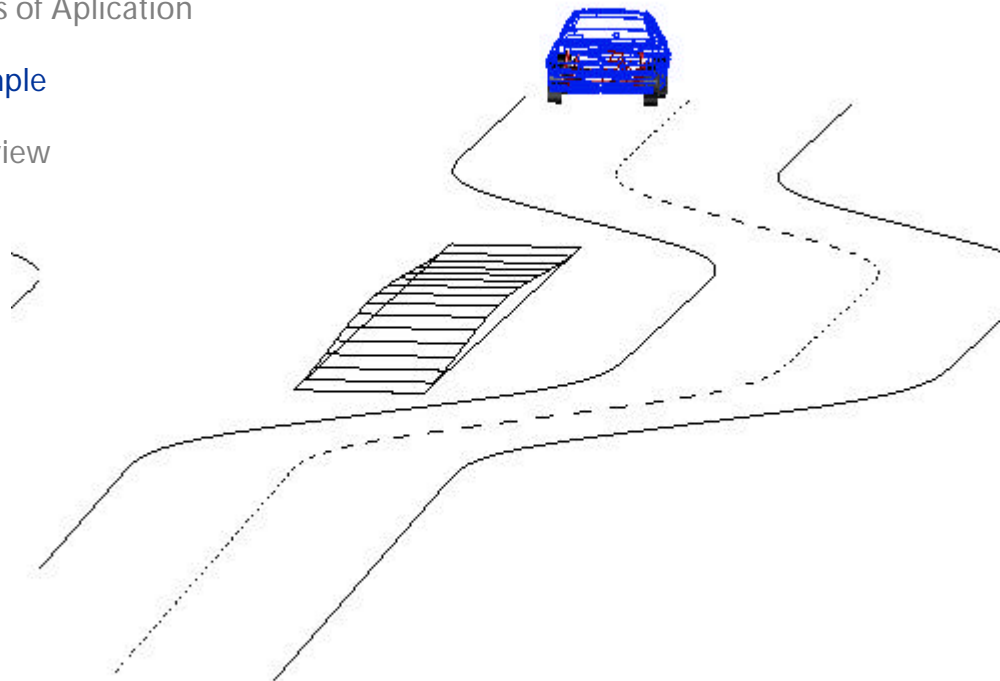
What is Code Export ?

Functionality

Fields of Application

▶ Example

Outview



Joints
19: Automotive Track Joint
...



Constraints
28: Massless Link



Force Elements
49: Automotive Tyre
Pacejka Similarity
...



Control Elements
168: Automotive Track Sensor



Input Function Set
Suspension Stiffness



Input Function Array
Tyre Friction Coefficient

SIMPACK Code Export

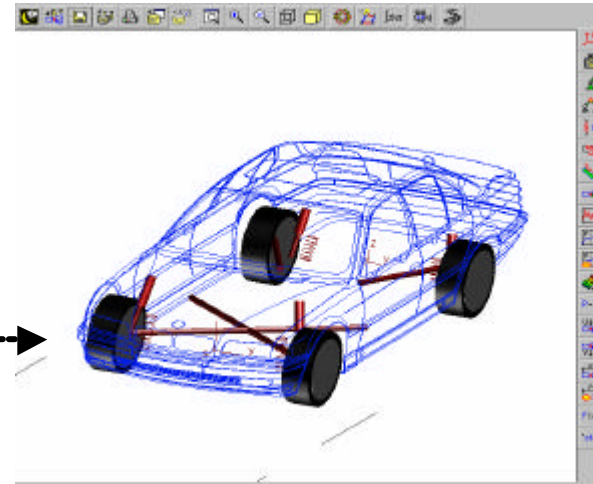
What is Code Export ?

Functionality

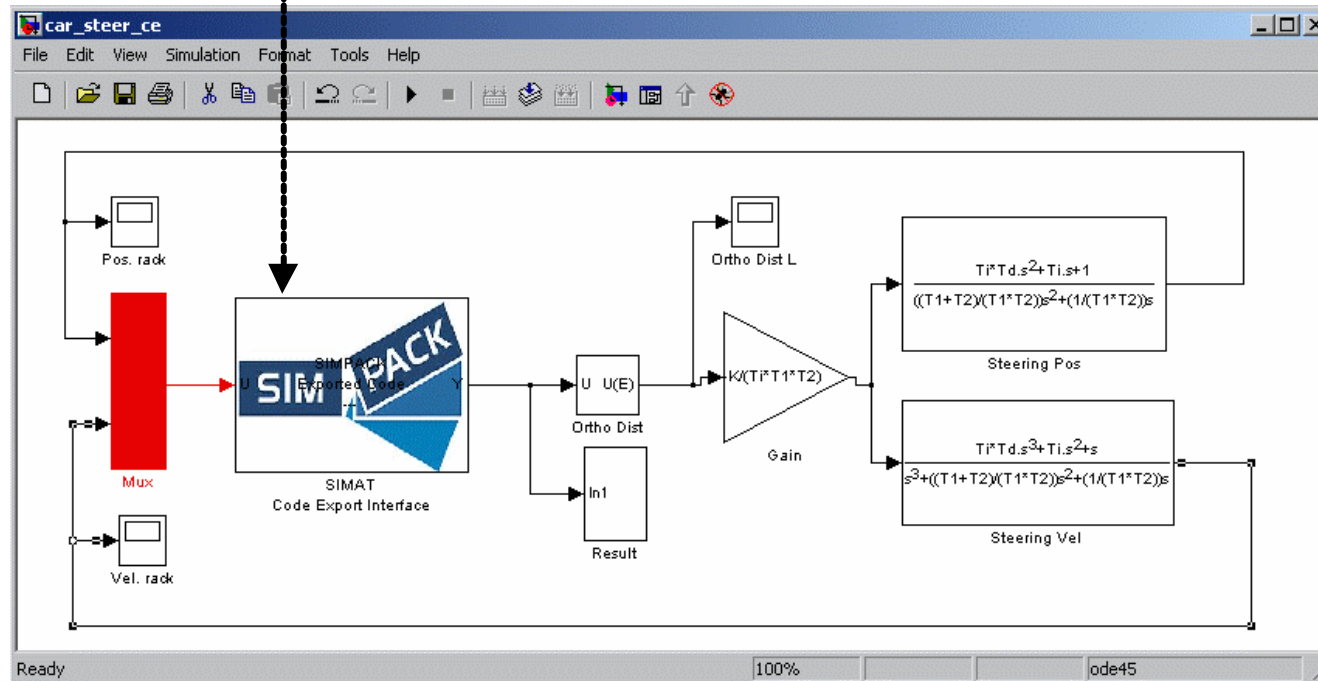
Fields of Application

▶ Example

Outview



MATLAB Steering control sensor using SIMPACK
Automotive track sensor in exported code



What Are You
Intending To Do?



What Are You
Intending To Do?

SIMPACK Code Export

What is Code Export ?

Functionality

Fields of Application

Example

▶ Overview

Outview

- ▶ Supported Platforms
 - > by now: Windows, HP UX, Silicon Graphics
 - > soon: IBM, SUN;
 - > possible Linux/ i386
- ▶ Code Export is offered in several stages of functionality.
Stage 1 shipped in minor releases
- ▶ Source Code available within special agreements
- ▶ Options for Hardware in the Loop Platforms:, still to be investigated
 - > Linux based HIL platforms now supported
 - > re-build to f77 -> f2c -> f90
 - > creating f90 objects/binaries on target platform
 - * which CPU/OS combinations ?
 - * binary compatibility ?

