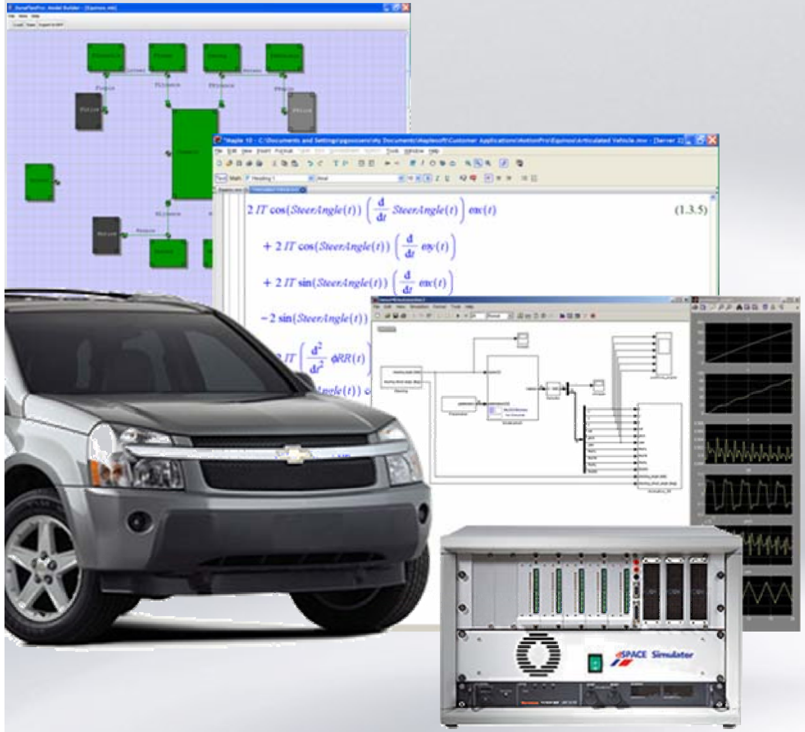
# DynaFlexPro

**Symbolic modeling and real-time simulation of multibody vehicle dynamics**



Maplesoft™
command the brilliance

**Paul Goossens, Maplesoft**
**Dr Chad Schmitke, MotionPro**

# Overview

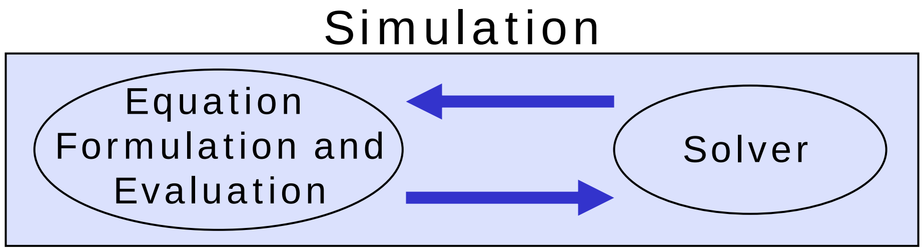
- Symbolic computing and multibody dynamics
- Linear graph theory and multibody dynamics
- DynaFlexPro
- Test case: Chevy Equinox

# Benefits of symbolic computing

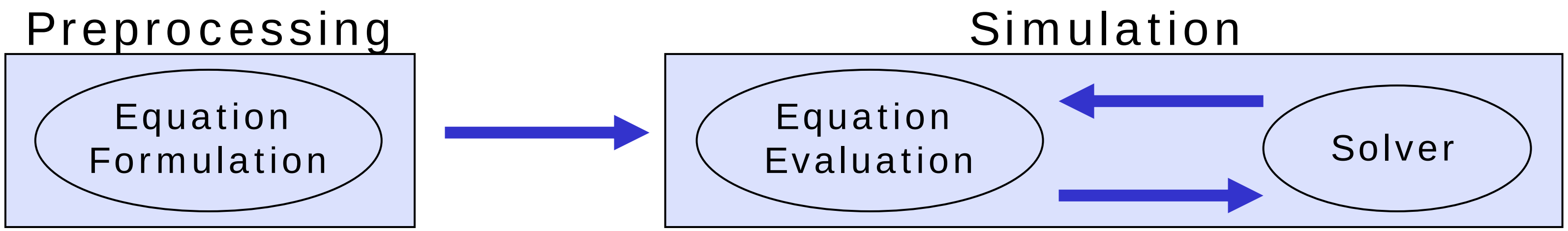
- Equations of motion can be visually examined, leading to physical insight
- Symbolic equations are very portable
- Equations can be symbolically differentiated to obtain sensitivity information
- *Faster simulations* than numerical computing approaches

# Why simulations are faster

- Simulation efficiency
  - Numerical formulations couple equation formulation with equation evaluation, requiring re-formulation at each time step

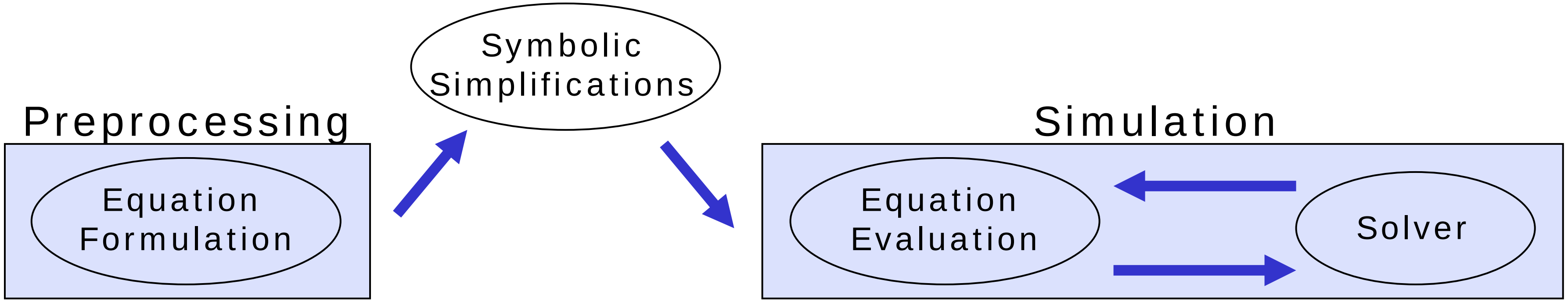



  - Symbolic approaches perform equation formulation once (off-line), requiring only equation evaluation during simulation



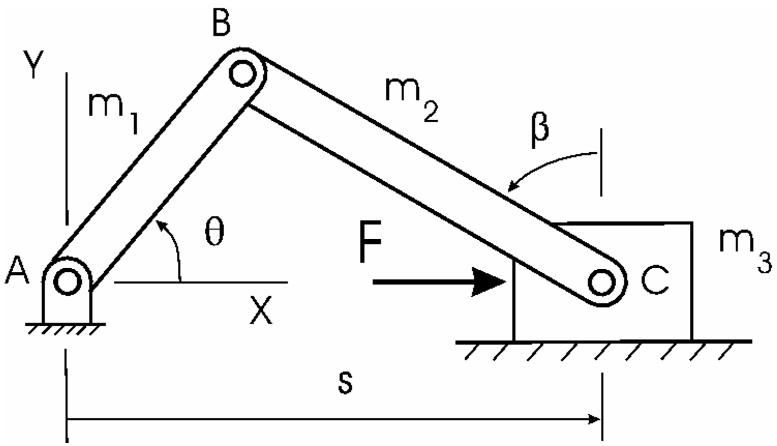
# Why simulations are faster

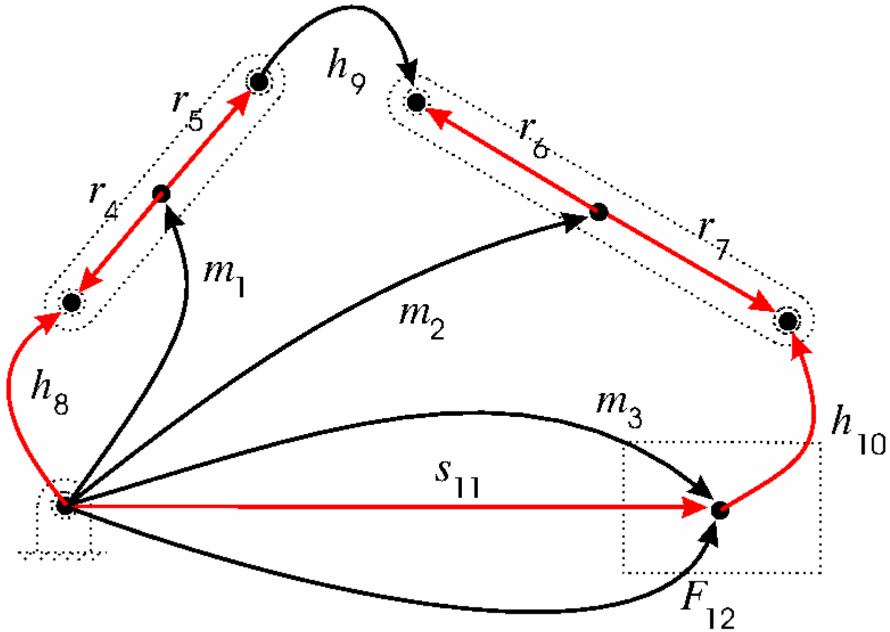
- Equation Simplification
  - Removal of unnecessary calculations (e.g. terms cancelling out, simplification of terms multiplied by 0 or 1)
  - Trigonometric simplifications
  - Opportunities for code optimization (e.g. repeated expressions identified and calculated only once)



# Linear graph theory

- Freedom to choose topology *and* variables:




$$Tree = \{r_4 - r_7, h_8, h_{10}, s_{11}\} \Rightarrow \mathbf{q} = [\theta_8, \beta_{10}, s_{11}]^T$$

$$Tree = \{r_4 - r_7, m_1, m_2, m_3\} \Rightarrow \mathbf{q} = [x_1, y_1, \theta_1, x_2, y_2, \theta_2, x_3, y_3, \theta_3]^T$$

# Linear graph theory

•Systematic algorithms for generating equations for any topology and variables:

$$\mathbf{M}\dot{\mathbf{p}} + \mathbf{\Psi}_{\mathbf{p}}^{T}\boldsymbol{\lambda} = \mathbf{F}$$

$$\dot{\mathbf{q}} = \mathbf{h}(\mathbf{p}, \mathbf{q}, t)$$

$$\mathbf{\Phi}(\mathbf{q}, t) = \mathbf{0}$$

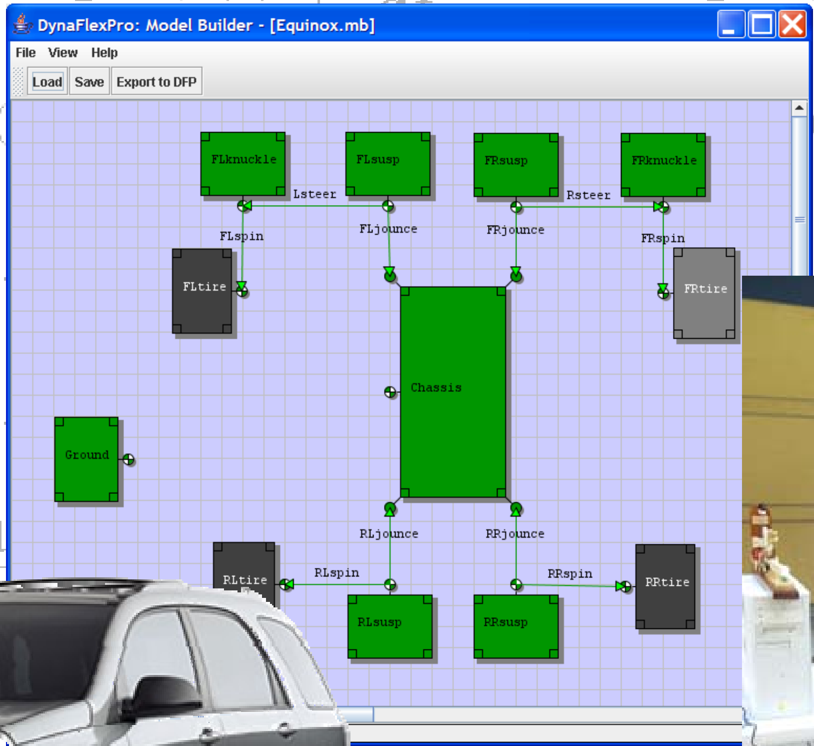
e.g. for slider-crank:

$$\mathbf{\Phi}(\mathbf{q}, t) = \begin{Bmatrix} L_1 \cos\theta_8 + L_2 \sin\beta_{10} - s_{11} \\ L_1 \sin\theta_8 - L_2 \cos\beta_{10} \end{Bmatrix} = \mathbf{0}$$

# DynaFlexPro

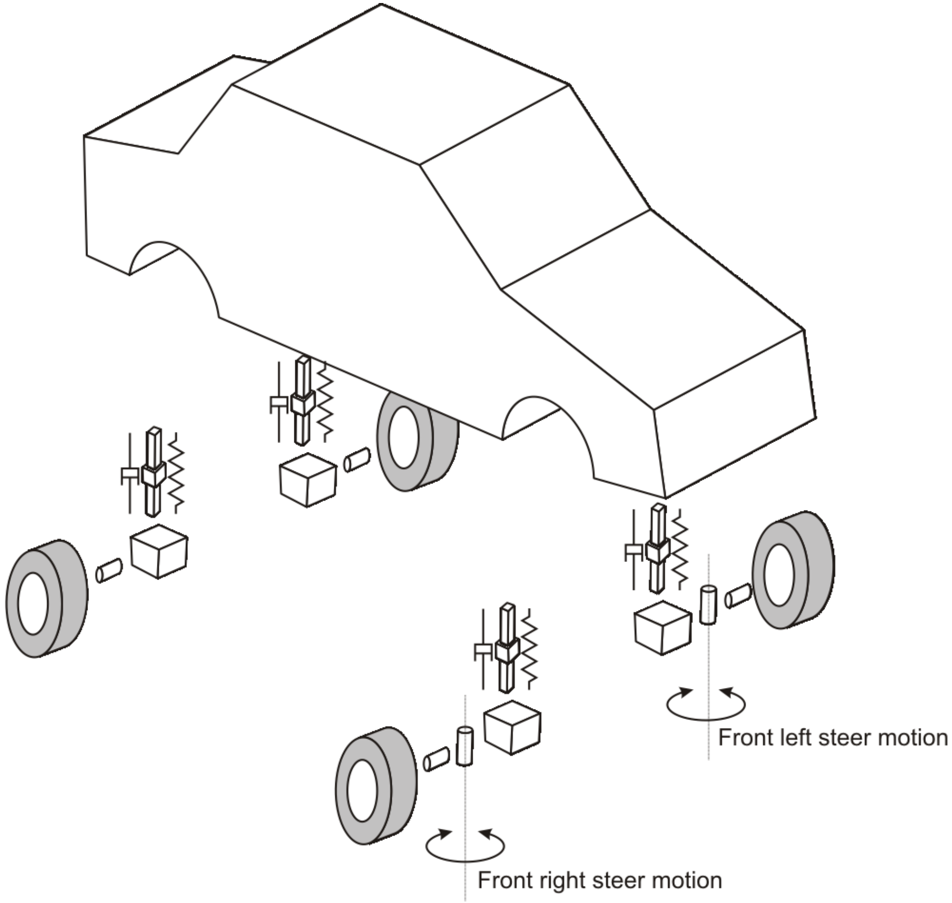
- Commercial program for multibody dynamics
- Linear graph theory + symbolic computing
- Four basic steps:
  - *Define the Model*: Create a block diagram of the system model
  - *Equation Generation*: Automatically generate the system's governing equations in Maple
  - *Simulation and Plotting*: Analyze the system within Maple
  - *Code Generation*: Automatically generate optimized procedures and functions for export

# Real-Time Simulation of Full Chassis

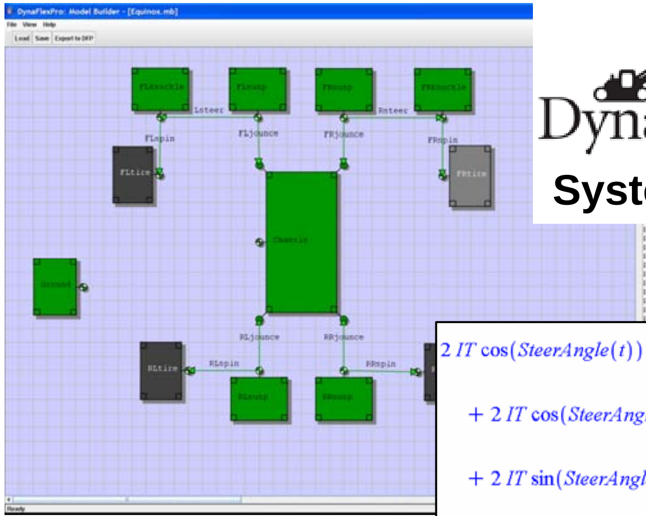



# Model details

- 4 prismatic joints to represent the jounce motion of each wheel

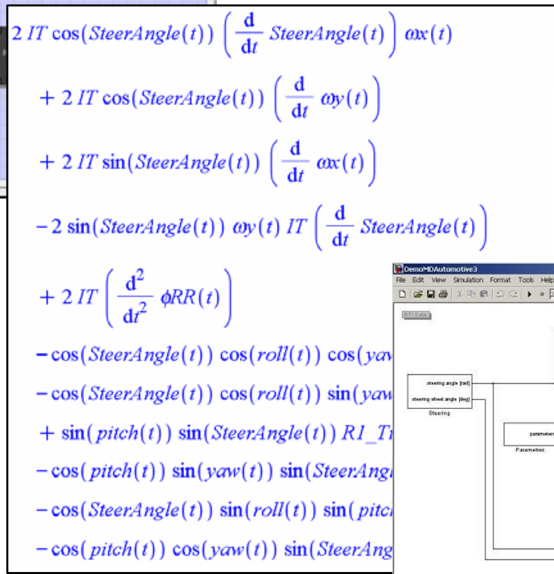


# Modeling Tool-chain



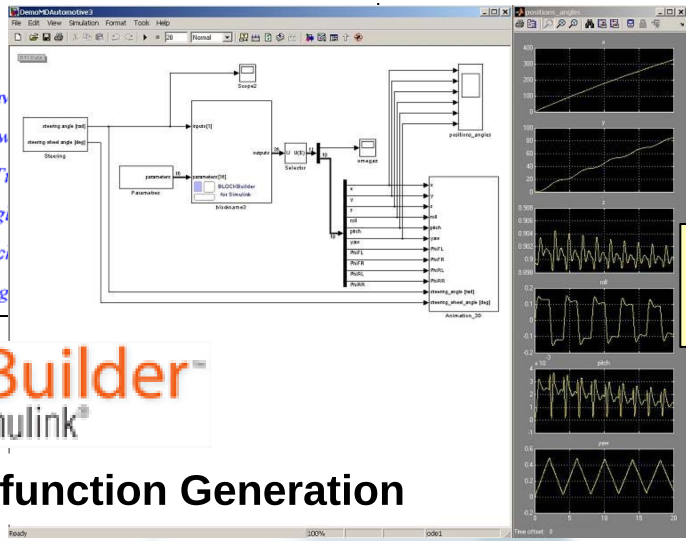
DynaFlexPro

**System Model Description**




Maple 11

**Analysis/Validation/Documentation**
**- Rapid Kinematic and Dynamic Equation Generation**



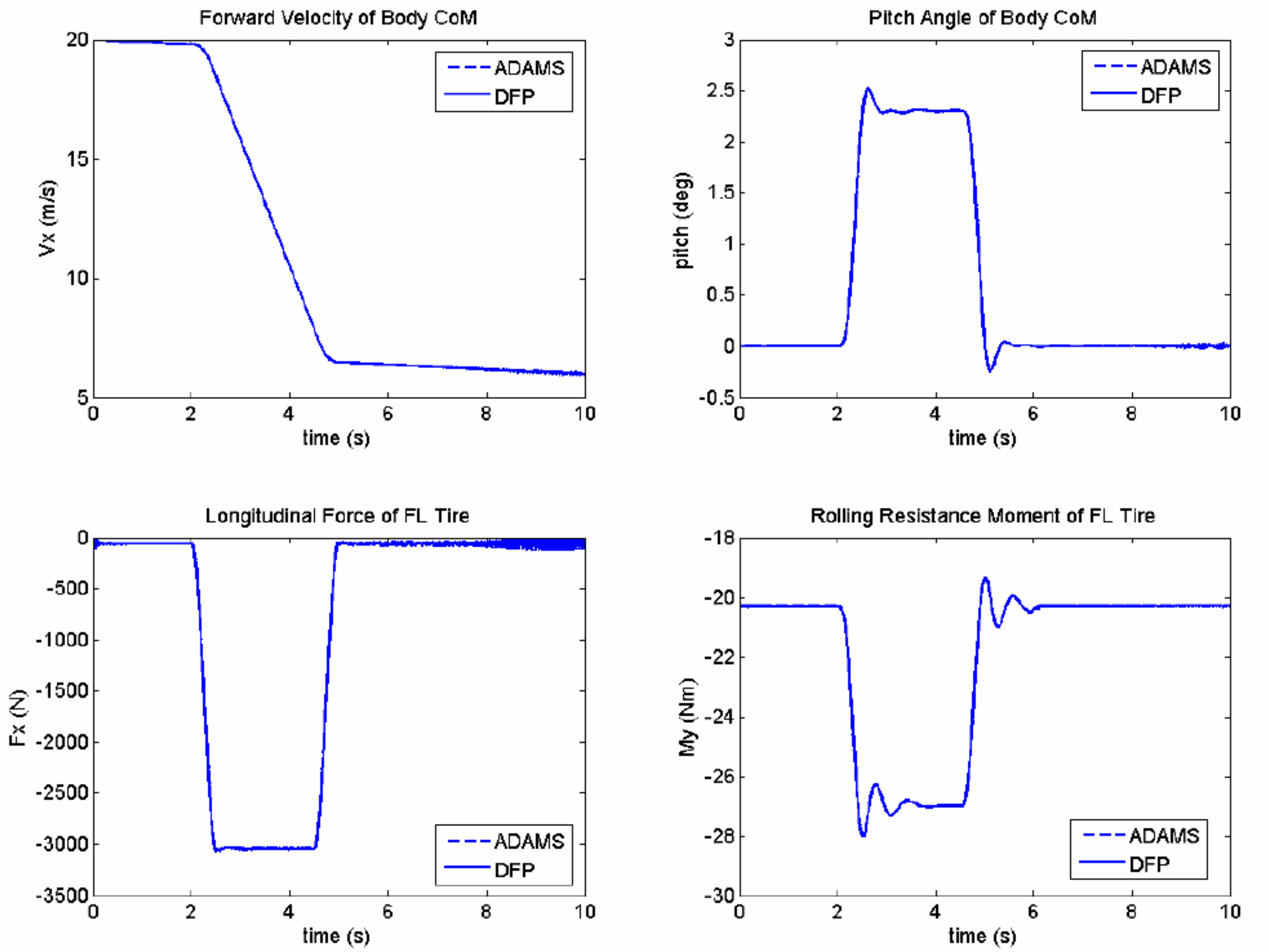
BLOCKBuilder for Simulink

**"One-Click" S-function Generation**

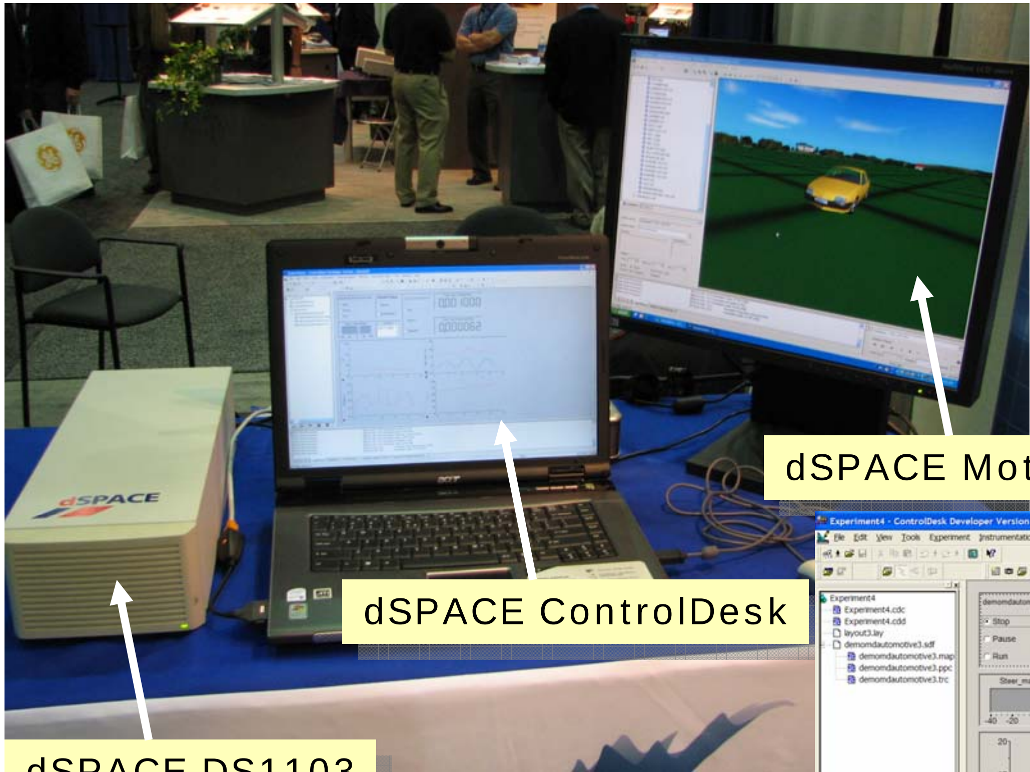
Real-Time Workshop

Maplesoft command the brilliance

# Model Accuracy



# Real-Time Execution

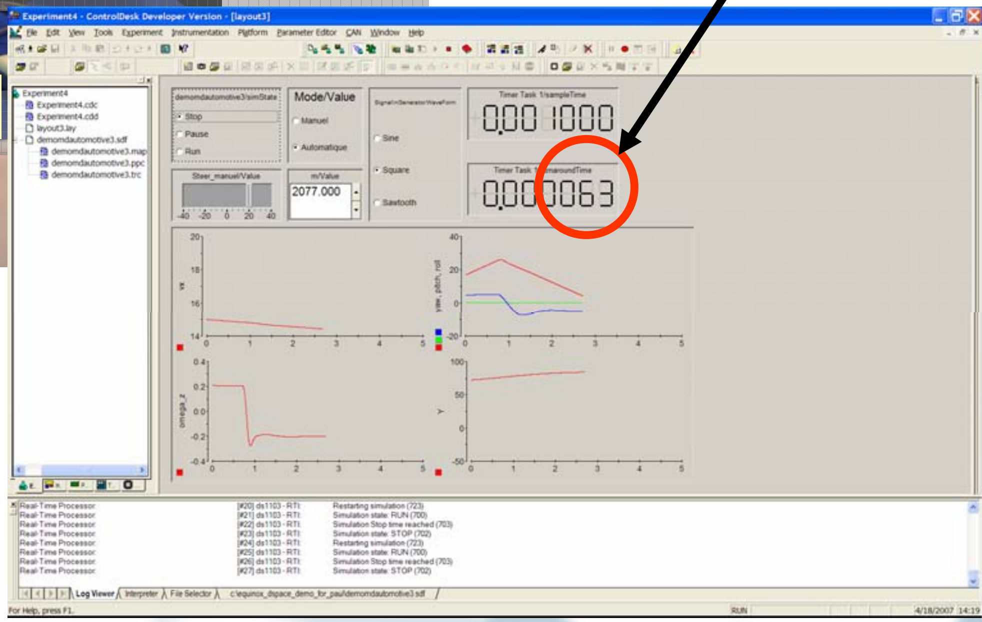



dSPACE MotionDesk

dSPACE ControlDesk

dSPACE DS1103

63μs Cycle Time



# Thank you!

# Booth #3310