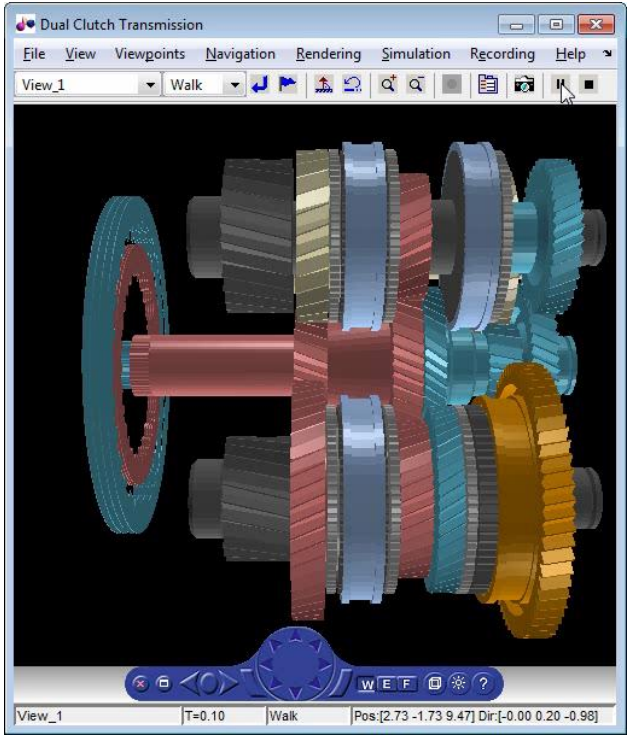
Introduction to Simulink, Stateflow, and Simscape

By Paul Peeling
MathWorks

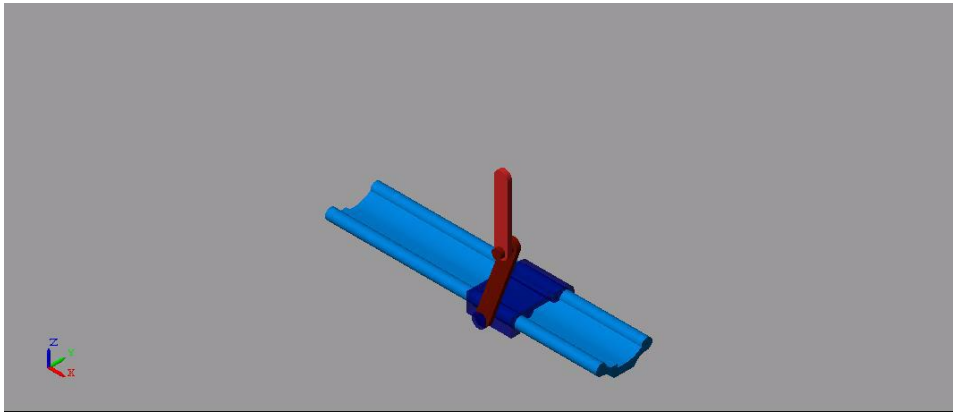
Key Technologies for Embracing Complexity

- Model-Based Design
- Multi-Domain Modelling
- Code Generation

Two Engineering Challenges

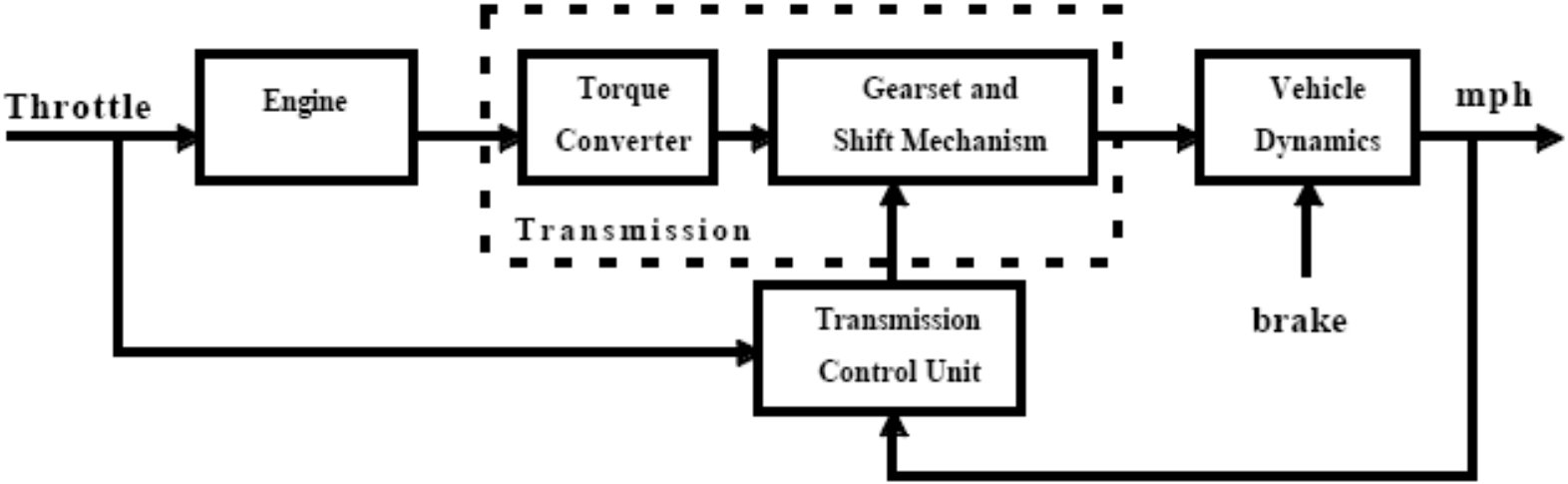


Shift schedule optimisation of an automatic transmission controller

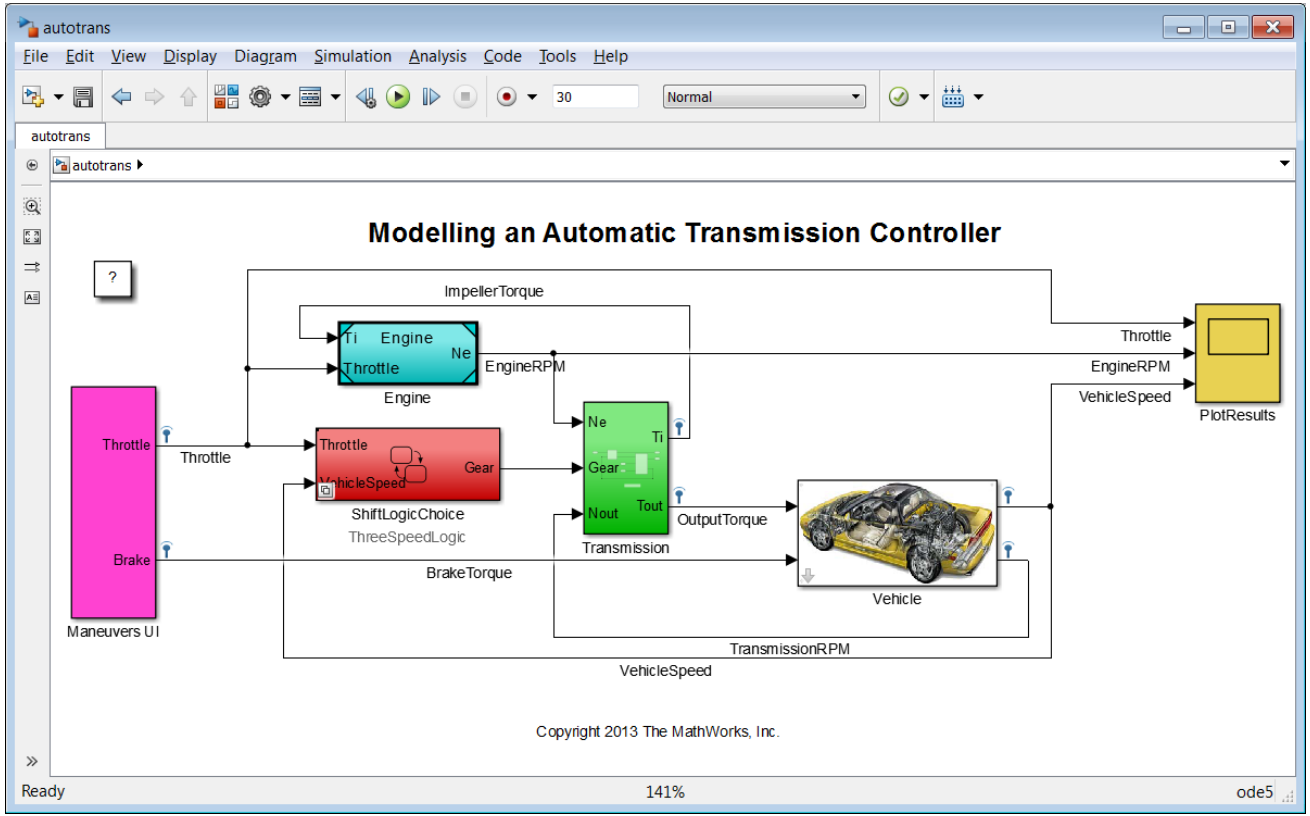


Modelling and control of an inverted double pendulum

Modelling Automatic Transmission

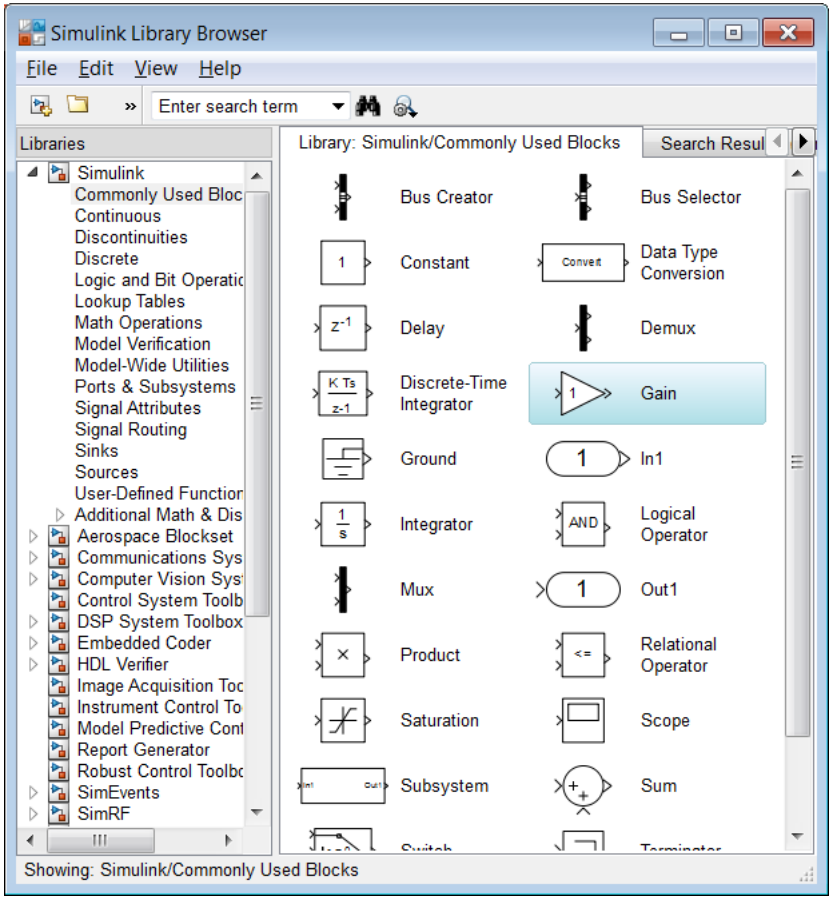


Modelling Automatic Transmission in Simulink



Blocks in Simulink

- Fundamental blocks
- Subsystems
- Other Simulink Models
- MATLAB Code
- DLLs
- Stateflow Charts
- Simscape Components
- …



Simulation of dynamic systems

$$I_{ei}\dot{N}_e = T_e - T_i$$

$N_e =$ engine speed (RPM)

$I_{ei} =$ moment of inertia of the engine and the impeller

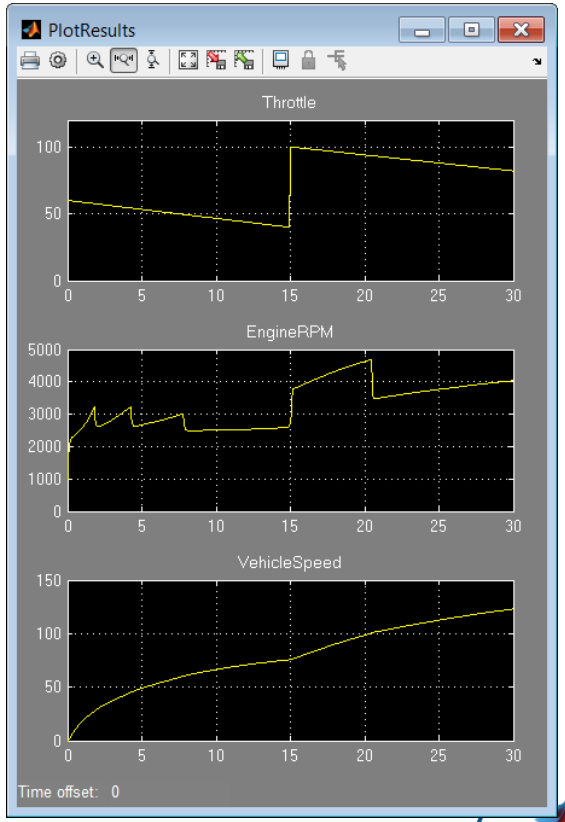
$T_e, T_i =$ engine and impeller torque

$$T_i = \frac{N_e^2}{K^2}$$

$$K = f_2\frac{N_{in}}{N_e} = \text{K-factor (capacity)}$$

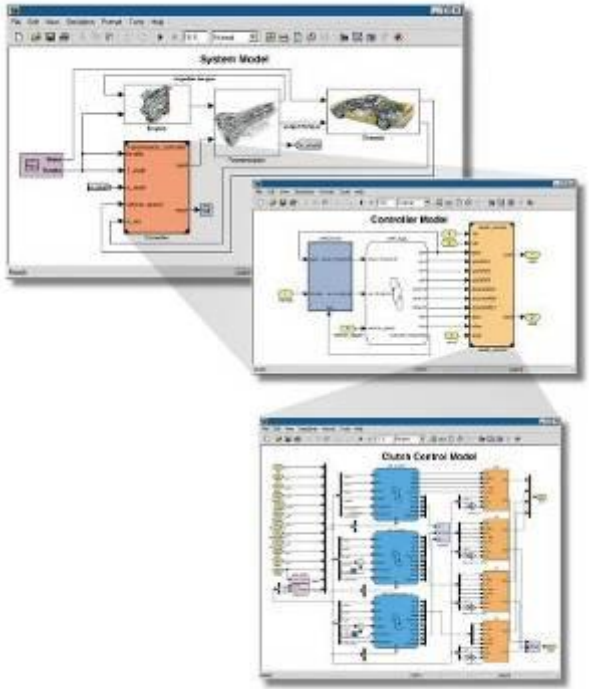
$N_{in} =$ speed of turbine (torque converter output) = transmission input speed (RPM)

$$R_{TQ} = f_3\frac{N_{in}}{N_e} = \text{torque ratio}$$



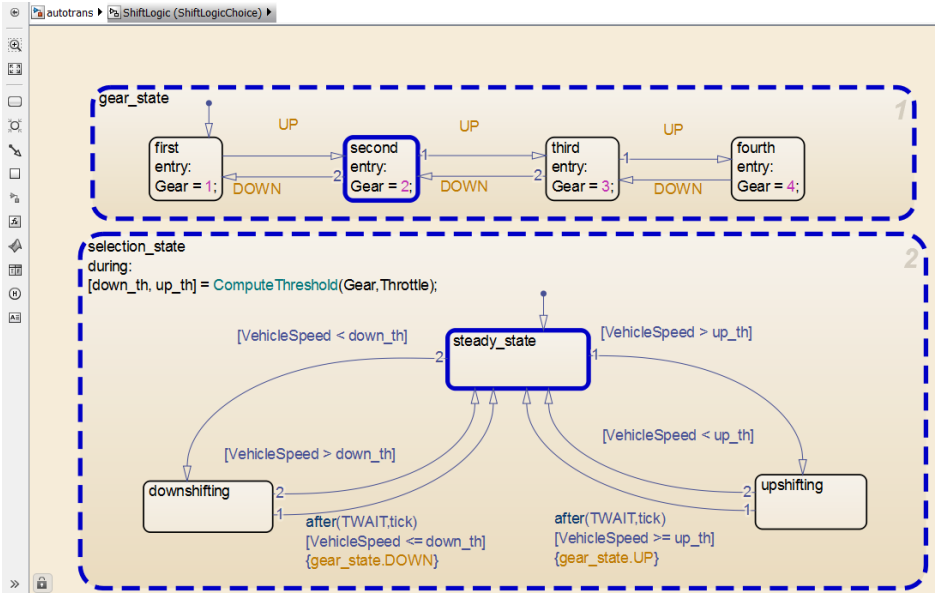
Simulink: Key Features

- Visual, block-diagram, environment
- Hierarchical, component-based modelling
- Extensive and expandable libraries of pre-defined blocks
- Open Application Program Interface (API)
- Full MATLAB® integration
- Multi-domain



Stateflow for Complex Logic

- When to use Stateflow?
- Model **instantaneous** changes in dynamic systems
  - Changes in state
  - Events
- Finite state machines
- Flow diagrams

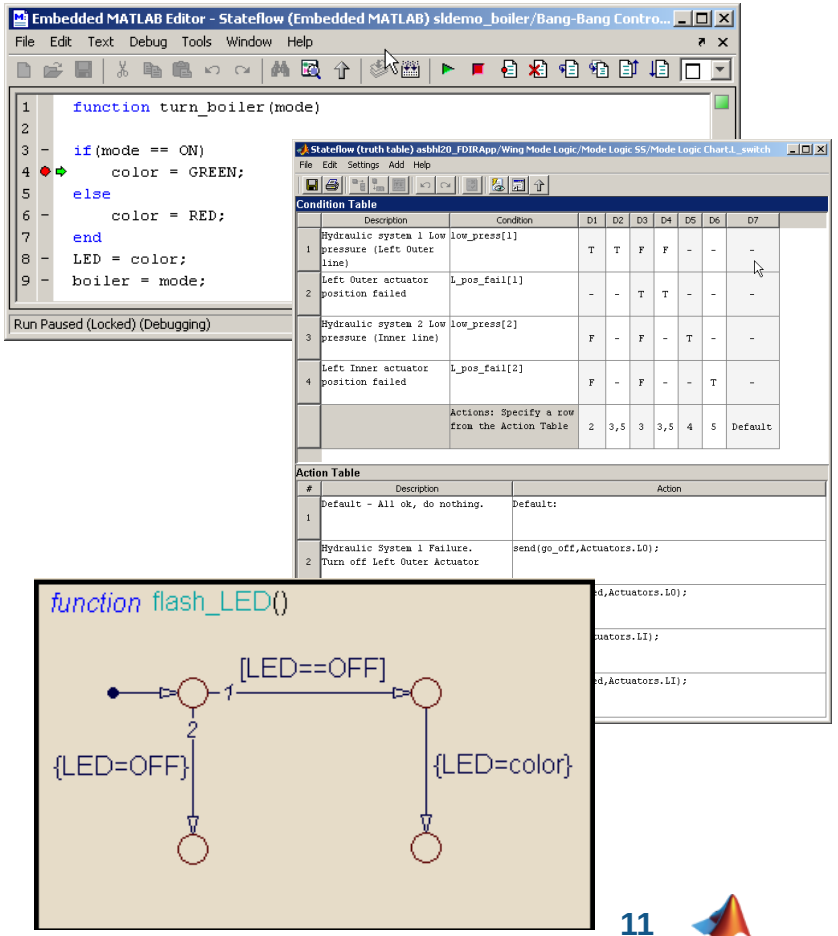


Stateflow Overview

- Extend Simulink with a design environment for developing state machines and flow charts
- Design systems containing control, supervisory, and mode logic
- Describe logic in a natural and understandable form with deterministic execution semantics

Stateflow: Key Features

- Defines functions
  - Procedurally, using MATLAB
  - Graphically, using flow diagrams
  - In tabular form, with truth tables
- Provides language elements, hierarchy, and parallelism
- Animates Stateflow® charts
- Incorporates custom and legacy C code
- Performs static and run-time checks

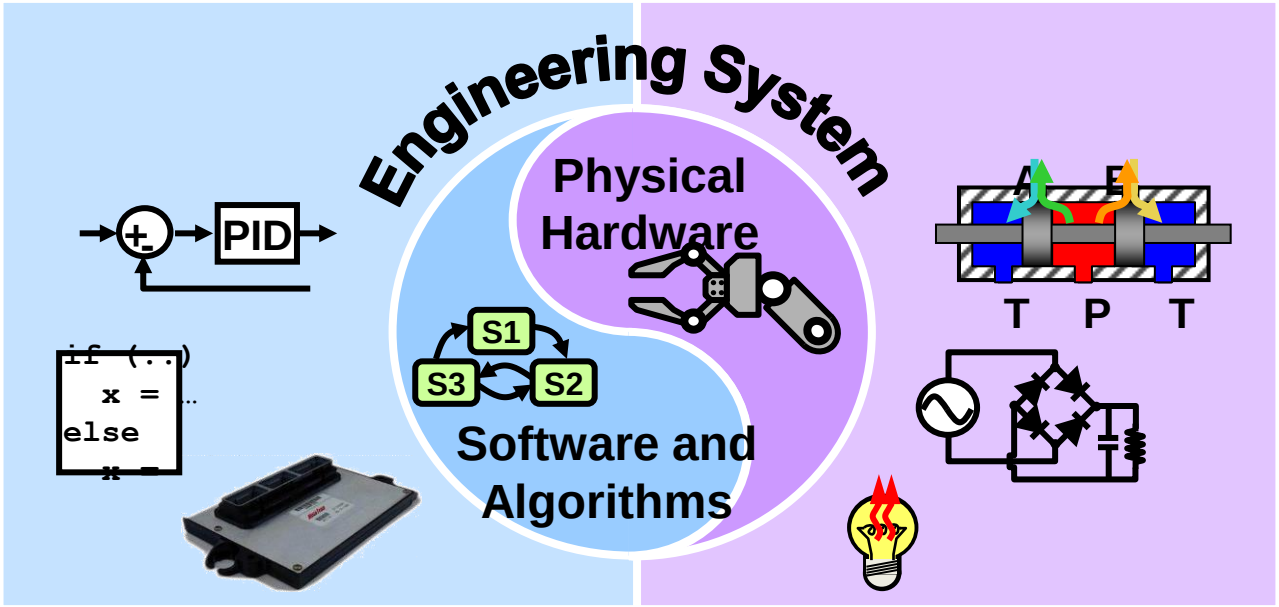


Conclusions

Simulink and Stateflow provide:

- A powerful environment for modelling real processes...
- in a modular fashion...
- and are fully integrated with the MATLAB environment for extensive design & analysis capability

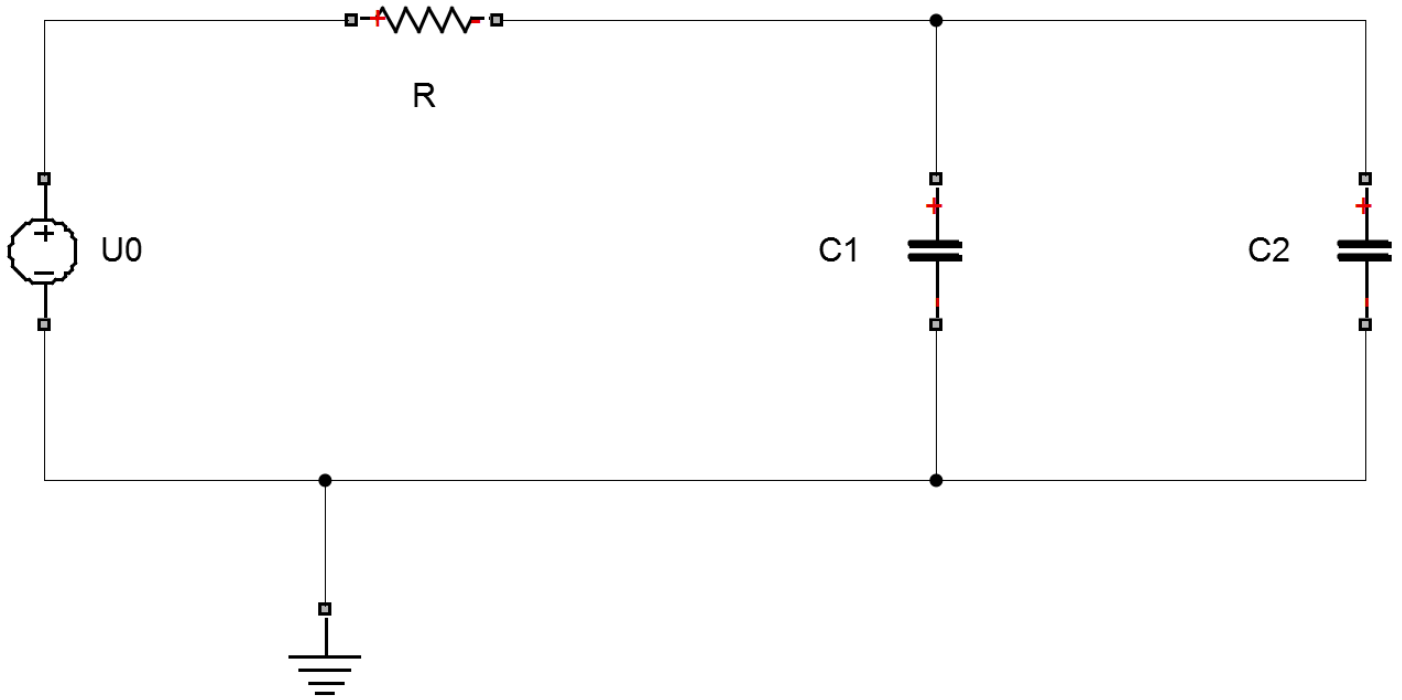
Physical Modelling



Inputs and outputs, state charts, algorithms, …

Physical devices

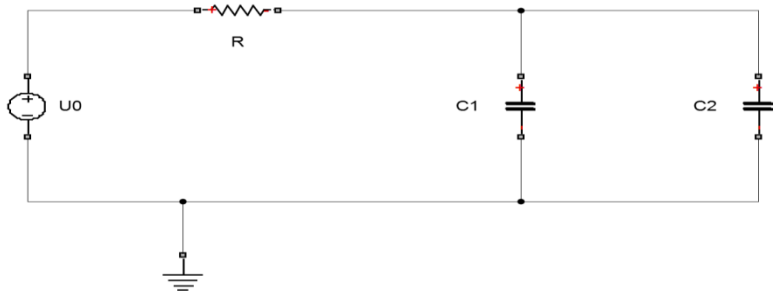
What does this model represent?



Modelling an electrical circuit in Simulink

Step 1: figure out the equations

Step 2: build the model



$$U_0 = f(t)$$

$$U_R = R.i_0$$

$$i_1 = C_1.\frac{dU_1}{dt}$$

$$i_2 = C_2.\frac{dU_2}{dt}$$

$$U_0 = U_R + U_1$$

$$U_2 = U_1$$

$$i_0 = i_1 + i_2$$

Differential Algebraic Equation

$$U_0 = f(t)$$

$$U_R = R.i_0$$

$$i_1 = C_1.\frac{dU_1}{dt}$$

$$i_2 = C_2.\frac{dU_2}{dt}$$

$$U_0 = U_R + U_1$$

$$U_2 = U_1$$

$$i_0 = i_1 + i_2$$

Algebraic equation

Differential equation

Modeling an electrical circuit in Simscape

$$U_0 = f(t)$$

$$U_R = R.i_0$$

$$i_1 = C_1.\frac{dU_1}{dt}$$

$$i_2 = C_2.\frac{dU_2}{dt}$$

Component equations

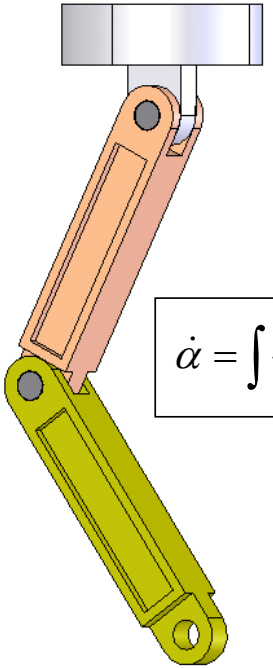
$$U_0 = U_R + U_1$$

$$U_2 = U_1$$

$$i_0 = i_1 + i_2$$

Constructed by the Simscape solver

Mathematical Modelling of Mechanical Systems

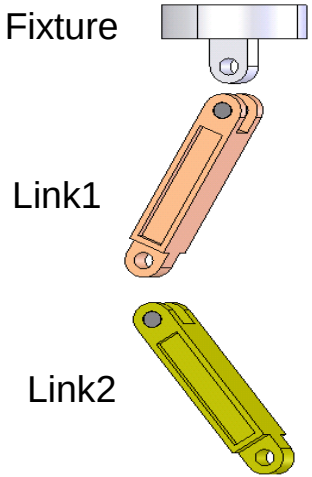


Derivation of the equations of motion requires extensive knowledge and great effort.

$$\dot{\alpha} = \int \frac{-L_2 \sin(\alpha) + nw_2(-\sin(\alpha-\gamma))\sin(\gamma) - ne(-\sin(\alpha-\gamma))\cos(\alpha-\gamma)\alpha^2 - n\cos(\alpha-\gamma)\gamma^2}{1 - ne\sin^2(\alpha-\gamma)} \, d\gamma$$

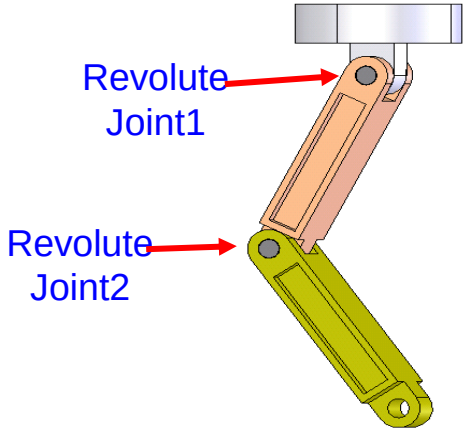
With SimMechanics

Bodies



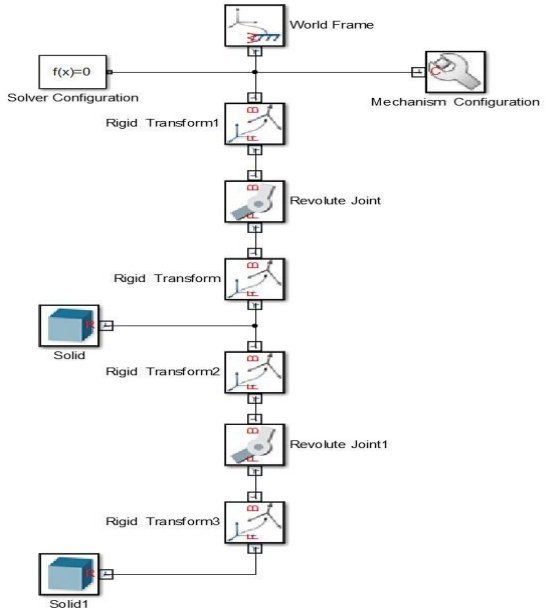
+

Joints



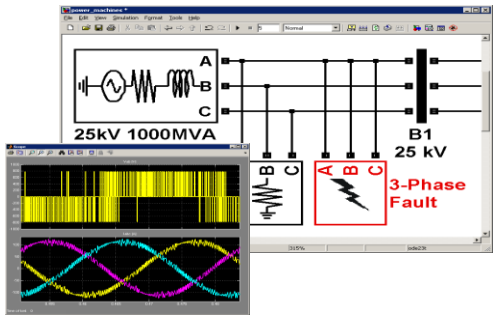
→

SimMechanics Model



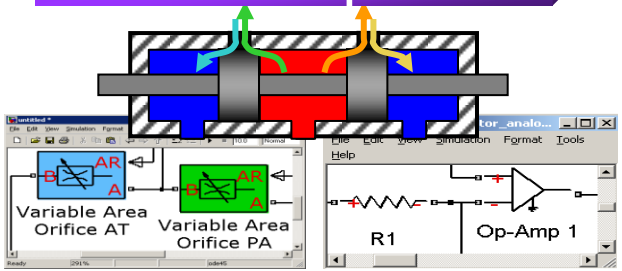
Physical Systems in Simulink®

SimPowerSystems™



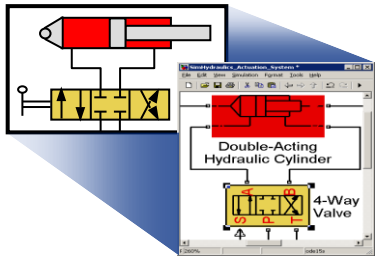
Electrical power systems

Simscape™



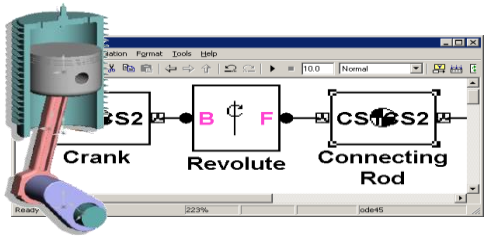
Multidomain physical systems

SimHydraulics®



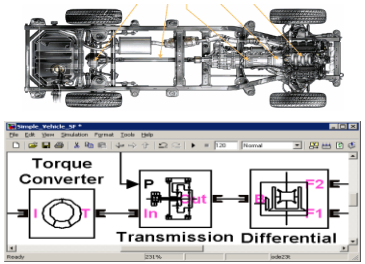
Fluid power and control

SimMechanics™



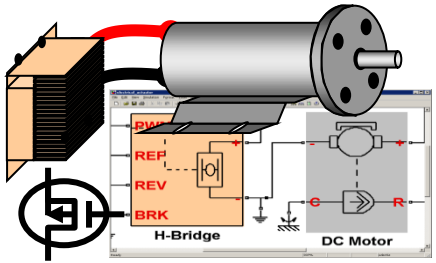
Mechanical dynamics (3-D)

SimDriveline™



Drivetrain systems (1-D)

SimElectronics™



Electromechanical and electronic systems

Simscape Key Features

- Library of foundation physical modelling building blocks
  - Mechanical, electrical, hydraulic,…
- Simscape language source provided
- Signals and parameters with units, and automatic unit conversion
- Physical network solver technology designed for physical systems
- Integrated with Simulink to support complete system modelling (physical system plus algorithms)
- Convert to C code for deployment