Introduction to Physical Modelling

Rory Adams
Senior Application Engineer
Creating Reusable Physical Models

What you should walk away with…

Increased knowledge of:

- What is meant by physical modelling and why do it?
- Using Simscape and add-ons for multi-domain physical modelling
- Using Parameter Estimation to match test data
Outline

- Introduction to Simscape and benefits of network based modelling

- Overview of Simscape add-ons

- Example: Modelling a Robot Arm
  - Early requirements validation
  - Match real world measurements with parameter estimation
  - Integrated plant and controller
Simulink

Simulink provides:

- A powerful environment for modelling real processes...
- in a modular fashion...
- and is fully integrated with the MATLAB environment for extensive design & analysis capability
Inputs and outputs, state charts, algorithms, …
Designing physical systems

- Multi-domain
  - Mechanical
  - Electrical
  - Hydraulic/pneumatic
  - Thermal

- Modelling environment
  - Multi-domain
  - Mix with algorithms/control
  - Multiple levels of fidelity
  - Optimization tools
  - Code generation
Data-driven versus physics-based modelling

**Data-driven**
- Data sources
  - Experimental data
  - External tool (complexity-reduction exercise)
- Complex behaviour with many unknowns
- Examples
  - Combustion engine
  - Aerodynamics

**Physics-based**
- Mathematical paradigms
  - ODEs & DAEs
  - State charts
  - Physical networks
- Behaviour is adequately approximated by limited set of equations
- Examples
  - Algorithms / control
  - Gears & clutches

\[ E(x, \alpha) \dot{x} = f(x, u, \alpha) \]
Modelling a Physical System in Simulink
Case study: Simple Electric Circuit

Step 1: figure out the equations
Step 2: build the model

\[ U_0 = f(t) \]
\[ U_R = R \cdot i_0 \]
\[ i_1 = C_1 \cdot \frac{dU_1}{dt} \]
\[ i_2 = C_2 \cdot \frac{dU_2}{dt} \]
\[ U_0 = U_R + U_1 \]
\[ U_2 = U_1 \]
\[ i_0 = i_1 + i_2 \]
Extending Simulink® using Simscape™

Simulink

- Equation set
  - $\dot{x} = f_e(x, u, \alpha)$
  - Explicit equation

- Relevance
  - Single body motion
  - Multiple-body motion when there is compliance
  - Most algorithms (control)

- Signal-flow based

---

Simscape extension

- Equation set
  - $E(x, \alpha)\dot{x} = f(x, u, \alpha)$
  - Implicit equation

- Relevance
  - 1-D multi-body systems e.g. drivelines
  - Electrical networks
  - Hydraulic/pneumatic network

- Network based
Representation of Physical Systems
Modelling an electrical circuit in Simscape

Component equations

\[ U_0 = f(t) \]
\[ U_R = R \cdot i_0 \]
\[ i_1 = C_1 \cdot \frac{dU_1}{dt} \]
\[ i_2 = C_2 \cdot \frac{dU_2}{dt} \]

Network equations

(Automatically constructed by Simscape solvers)

Simscape solves the system of equations \textit{simultaneously}, so it \textit{intrinsically} solves algebraic loops.
DC Motor: Simulink

\[ v = K_e \omega + i_m R_{wind} + L_{wind} \frac{di_m}{dt} \]

\[ T = K_t i_m - D \omega - J \frac{d\omega}{dt} \]
DC Motor: Simscape

- Simscape model advantages
  - Easier to read than equations
  - Quicker to create
  - More intuitive – easier to explain to other engineers
DC Motor: Simscape Language

- Simscape language advantages
  - For custom components, often easier to express equations
  - Flexibility & extensibility

\[ v = K_e \omega + i_m R_{\text{wind}} + L_{\text{wind}} \frac{di_m}{dt} \]

\[ T = K_t i_m - D \omega - J \frac{d\omega}{dt} \]
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
Physical Systems in Simulink®

- **SimPowerSystems™**
  - Electrical power systems

- **SimMechanics™**
  - Mechanical dynamics (3-D)

- **SimDriveline™**
  - Drivetrain systems (1-D)

- **SimHydraulics®**
  - Fluid power and control

- **SimElectronics™**
  - Electromechanical and electronic systems

- **Simscape™**
  - Multidomain physical systems
Extensive Component Libraries

- SimElectronics > 90 component models
  - Actuators, drivers
  - Sensors
  - Semiconductors
  - Integrated circuits

- Models look like schematics
  - Easy to read and interpret
Derivation of the equations of motion requires extensive knowledge and great effort.

\[ \dot{\alpha} = \int \left( -L_2 \sin(\alpha) + n \omega_2 (-\sin(\alpha - \gamma)) \sin(\gamma) - n \epsilon (-\sin(\alpha - \gamma)) \cos(\alpha - \gamma) \alpha^2 - n \cos(\alpha - \gamma) \gamma^2 \right) \frac{d\gamma}{1 - n \epsilon \sin^2(\alpha - \gamma)} \]
\[ \dot{\alpha} = \int \frac{-L_2 \sin(\alpha) + n\omega_2 (-\sin(\alpha - \gamma))\sin(\gamma) - n\varepsilon (-\sin(\alpha - \gamma))\cos(\alpha - \gamma)\alpha^2 - n\cos(\alpha - \gamma)\gamma^2}{1 - n\varepsilon \sin^2(\alpha - \gamma)} \, d\gamma \]
With SimMechanics

Bodies

- Fixture
- Link1
- Link2

Joints

- Revolute Joint1
- Revolute Joint2

SimMechanics Model
6 DOF Robot Arm Model

1. Refine electronic actuator requirements
2. Automatically tune actuator parameters
3. Integrate controls, electric actuators and mechanical robot arm in the same environment
Robot Arm CAD Import

**Model:** CAD

**Problem:** Perform dynamic simulation of robot arm modeled in CAD

**Solution:** Import CAD model into SimMechanics™ with SimMechanics Link
SimMechanics Link – Importing CAD Models

- Automatically create SimMechanics models from a CAD assembly
  - Converts mass and inertia into rigid body definitions
  - Converts mate definitions to constraints and joints
  - Creates STL files for use with SimMechanics visualization

- CAD Tools
  - Directly connects SolidWorks, ProEngineer and Autodesk Inventor
  - Public API for other CAD systems

- Free download from www.mathworks.com

- Requires MATLAB
Refine Actuator Requirements

Model:

Small / Medium / Large

Problem: Determine size requirements for electric actuator

Solution: Use SimMechanics to determine force required for prescribed motion
Problem: Simulation data does not match measured data for the motors because the parameters are incorrect

Solution: Use Simulink Design Optimization to automatically tune model parameters

<table>
<thead>
<tr>
<th>R</th>
<th>L</th>
<th>K</th>
<th>J</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>27.2</td>
<td>0.025</td>
<td>0.997</td>
<td>0.036</td>
<td>0.341</td>
</tr>
</tbody>
</table>
Estimating Parameters Using Measured Data

- Advantages of parameter estimation with Simulink Design Optimization

1) Enables **quick and easy comparison** of simulation results and measured data to ensure **simulation matches reality**

2) **Automatic** tuning of parameters **saves time**

3) Optimization algorithms reveal parameter sensitivity and help **improve model parameterization**
System Level: Integrated Plant and Controllers

Model:

Problem: Test and analyse behaviour of complete system (plant + controllers)

Solution: Integrate SimElectronics model of actuators with SimMechanics model of robot arm with Simulink & StateFlow model of control algorithm
Developing Control Systems

- Implement nonlinear plant models
- Extract linear model for use with linear control theory
- Explore interaction between control system and plant

\[ A \times x + B \times u = 0 \]
Hybrid Electrical Vehicle Model
Balance Fidelity and Speed

- **Electrical**
  - System Level
    - Test integration, optimise system
  - Mean Value
    - Three-phase electrical system
  - Detailed
    - Test power quality

- **Battery**
  - Generic, predefined, and custom models

- **Vehicle**
  - Inertial & Aero Effects
  - Tyre models
Generating Code

- Use Simulink Coder to convert models into C code
  - Distribute as .DLL to co-workers for parameter studies
  - Distribute as .DLL to customers to protect intellectual property
  - Run in real-time
  - Run hardware-in-the-loop simulations
Optimise System-Level Performance

- Simulating plant and controller in one environment allows you to optimise system-level performance.
Physical Modelling Methods Ideal are for Plant Models

- Build accurate models quickly
  - System-level equations derived automatically

- Model is easier to read
  - Reflects structure of system

- Easier to update model
  - New technologies or designs can be easily incorporated into the model