Physical Modelling with Simscape™

Rick Hyde
Outline

- Part 1: Introduction to Simscape
  - Review approaches to modelling
  - Overview of Simscape-based libraries
  - Introduction to physical networks
  - Simscape tutorial: DC motor

- Part 2: Application examples
  - PMSM and drive
  - Landing gear extension/retraction
  - Engine cooling system
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

\[ E(x, \alpha) \dot{x} = f(x, u, \alpha) \]
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)

**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 networks
What does this model represent?
What does this model represent?
Modelling an electrical circuit in Simulink

Step 1: figure out the equations

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

Step 2: build the model
Modelling an electrical circuit in Simscape

Build the model
Modelling an electrical circuit in Simscape

\[ U_0 = f(t) \]

\[ U_R = R \cdot i_0 \]

\[ i_1 = C_1 \cdot \frac{dU_1}{dt} \]

\[ i_2 = C_2 \cdot \frac{dU_2}{dt} \]

\[ U_0 = U_R + U_1 \]

\[ U_2 = U_1 \]

\[ i_0 = i_1 + i_2 \]

Component equations from library blocks

Additional constraints from circuit topology
Algebraic loops in Simscape

Simscape converts entire network to equations and solves them *simultaneously*, so it *intrinsically* solves algebraic loops.
Signal-based modelling methods pose challenges

- Very flexible, but can be difficult to use efficiently
- Requires expertise in several areas
  - Physics, math, programming
- Deriving system level equations is difficult and error prone
- Resulting model can be difficult to read and maintain
Physical modelling methods ideal 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
Simscape libraries

- SimPowerSystems
- SimMechanics
- SimHydraulics
- SimDriveline
- SimElectronics

Simscape

MATLAB, Simulink
Extensive Component Libraries

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

- Models look like schematics
  - Easy to read and interpret
Generate Code

- Use Simulink Coder to convert models into C code
  - Share with other users (Model Reference)
  - Protect intellectual property
    - Model Reference Protected Mode
    - Alternative: ssc_protect
  - Run hardware-in-the-loop simulations
Outline

- Part 1: Introduction to Simscape
  - Review approaches to modelling
  - Overview of Simscape-based libraries
  - Introduction to physical networks
  - Simscape tutorial: DC motor

- Part 2: Application examples
  - PMSM and drive
  - Landing gear extension/retraction
  - Engine cooling system
Through & across variables

- Abstract to a physical network
- All nodes have the same pressure (across variable)
- Sum of flows (through variables) at a node is zero
- Each component must specify an equation involving the through and/or across variables at its boundary
## Through & across variables by domain

<table>
<thead>
<tr>
<th>Domain</th>
<th>Through</th>
<th>Across</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical</td>
<td>Current</td>
<td>Voltage</td>
</tr>
<tr>
<td>Hydraulic</td>
<td>Flow rate</td>
<td>Pressure</td>
</tr>
<tr>
<td>Rotational mechanical</td>
<td>Torque</td>
<td>Angular speed</td>
</tr>
<tr>
<td>Translational mechanical</td>
<td>Force</td>
<td>Translational speed</td>
</tr>
<tr>
<td>Pneumatic</td>
<td>Mass flow</td>
<td>Pressure</td>
</tr>
<tr>
<td></td>
<td>Heat flow</td>
<td>Temperature</td>
</tr>
<tr>
<td>Magnetic</td>
<td>Magnetic flux</td>
<td>Magneto motive force</td>
</tr>
<tr>
<td>Thermal</td>
<td>Heat flow</td>
<td>Temperature</td>
</tr>
</tbody>
</table>
## Components

<table>
<thead>
<tr>
<th>Domain (across, through)</th>
<th>Resistance</th>
<th>Inductance</th>
<th>Capacitance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical (v, i)</td>
<td>Resistor</td>
<td>Inductor</td>
<td>Capacitor</td>
</tr>
<tr>
<td>Rotational mechanical</td>
<td>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Translational mechanical</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hydraulic</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Resistance (R) dissipates power.
- $R = \text{voltage/current} \ \text{(ratio of across to through)}$
# Components

<table>
<thead>
<tr>
<th>Domain</th>
<th>Resistance</th>
<th>Inductance</th>
<th>Capacitance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical (v, i)</td>
<td>Resistor</td>
<td>Inductor</td>
<td>Capacitor</td>
</tr>
<tr>
<td>Rotational mechanical (w, trq)</td>
<td>Rotational damper</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Translational mechanical</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hydraulic</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Components

<table>
<thead>
<tr>
<th>Domain</th>
<th>Resistance</th>
<th>Inductance</th>
<th>Capacitance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical (v, i)</td>
<td>Resistor</td>
<td>Inductor</td>
<td>Capacitor</td>
</tr>
<tr>
<td>Rotational mechanical (w, trq)</td>
<td>Rotational damper</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Translational mechanical (v, F)</td>
<td>Translational</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>damper</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hydraulic</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Components

<table>
<thead>
<tr>
<th>Domain</th>
<th>Resistance</th>
<th>Inductance</th>
<th>Capacitance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical (v, i)</td>
<td>Resistor</td>
<td>Inductor</td>
<td>Capacitor</td>
</tr>
<tr>
<td>Rotational mechanical (w, trq)</td>
<td>Rotational damper</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Translational mechanical (v, F)</td>
<td>Translational damper</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hydraulic (p, Q)</td>
<td>?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Components

<table>
<thead>
<tr>
<th>Domain</th>
<th>Resistance</th>
<th>Inductance</th>
<th>Capacitance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical (V, i)</td>
<td>Resistor</td>
<td>Inductor</td>
<td>Capacitor</td>
</tr>
<tr>
<td>Rotational mechanical (w, trq)</td>
<td>Rotational damper</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Translational mechanical (v, F)</td>
<td>Translational damper</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hydraulic (p, Q)</td>
<td>Hydraulic orifice</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>p=kQ^n</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Components

<table>
<thead>
<tr>
<th>Domain</th>
<th>Resistance</th>
<th>Inductance</th>
<th>Capacitance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical (v, i)</td>
<td>Resistor</td>
<td>Inductor</td>
<td>Capacitor</td>
</tr>
<tr>
<td>Rotational mechanical (w, trq)</td>
<td>Rotational damper</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>Translational mechanical (v, F)</td>
<td>Translational damper</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hydraulic (p, Q)</td>
<td>Hydraulic orifice</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>p=kQ^n</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Inductance keeps through variable flowing (current)
## Components

<table>
<thead>
<tr>
<th>Domain</th>
<th>Resistance</th>
<th>Inductance</th>
<th>Capacitance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical (v, i)</td>
<td>Resistor</td>
<td>Inductor</td>
<td>Capacitor</td>
</tr>
<tr>
<td>Rotational mechanical (w, trq)</td>
<td>Rotational damper</td>
<td>Rotational Spring</td>
<td></td>
</tr>
<tr>
<td>Translational mechanical (v, F)</td>
<td>Translational damper</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hydraulic (p, Q)</td>
<td>Hydraulic orifice</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>p=kQ^n</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Inductance keeps through variable flowing (current)
## Components

<table>
<thead>
<tr>
<th>Domain</th>
<th>Resistance</th>
<th>Inductance</th>
<th>Capacitance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical (v, i)</td>
<td>Resistor</td>
<td>Inductor</td>
<td>Capacitor</td>
</tr>
<tr>
<td>Rotational mechanical (w, trq)</td>
<td>Rotational damper</td>
<td>Rotational Spring</td>
<td>?</td>
</tr>
<tr>
<td>Translational mechanical (v, F)</td>
<td>Translational damper</td>
<td>Translational spring</td>
<td></td>
</tr>
<tr>
<td>Hydraulic (p, Q)</td>
<td>Hydraulic orifice</td>
<td></td>
<td>Accumulator</td>
</tr>
<tr>
<td></td>
<td>p=kQ^n</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Capacitor maintains across variable (voltage)
## Components

<table>
<thead>
<tr>
<th>Domain</th>
<th>Resistance</th>
<th>Inductance</th>
<th>Capacitance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical (v, i)</td>
<td>Resistor</td>
<td>Inductor</td>
<td>Capacitor</td>
</tr>
<tr>
<td>Rotational mechanical (w, trq)</td>
<td>Rotational damper</td>
<td>Rotational Spring</td>
<td>Inertia</td>
</tr>
<tr>
<td>Translational mechanical (v, F)</td>
<td>Translational damper</td>
<td>Translational spring</td>
<td>Mass</td>
</tr>
<tr>
<td>Hydraulic (p, Q)</td>
<td>Hydraulic orifice</td>
<td>Accumulator</td>
<td>Fluid inertia</td>
</tr>
<tr>
<td></td>
<td>$p = kQ^n$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Capacitor maintains across variable (voltage)
The inerter

- Invented by Prof Malcolm Smith, Cambridge University.
- Force is proportional to *relative* acceleration.
Outline

- Part 1: Introduction to Simscape
  - Review approaches to modelling
  - Overview of Simscape-based libraries
  - Introduction to physical networks
  - Simscape tutorial: DC motor

- Part 2: Application examples
  - PMSM and drive
  - Landing gear extension/retraction
  - Engine cooling system
DC Motor: Modelling options

1. Use SimElectronics DC Motor block
2. Build an equivalent circuit using Simscape foundation library
3. Define a custom component using Simscape language
   - Equation-based
   - Composite
SimElectronics® pre-built component
SimElectronics DC motor parameterization options

- Parameterized according to datasheet standards
- Multiple methods for assigning parameters
- Description, functionality and formulation provided in Help dialog
SimElectronics pre-built component example
Equivalent circuit based on Simscape foundation library components
Equation-based Simscape language component

1. Define equations
2. Define ports/connections
3. Define variables
4. Define parameters
5. Write block help
6. Add icon [optional]
7. Publish

\[ v = Ri + L \frac{di}{dt} + k\omega \]
Equation-based Simscape language example
Equation-based Simscape language component

\[ v = Ri + L \frac{di}{dt} + k\omega \]

\[ \tau = ki \]
Composite Simscape language component

- Connect components in Simscape language file
  - Refer to existing components
  - Overwrite parameter values if necessary

```matlab
components (Hidden=true)

rotorResistor = found.component.resistor(R = rotor_resistance);
rotorInductor = found.component.inductor(l = rotor_inductance, i0
rotEMechConv = found.component.rotational_converter(K = back_emf
friction = found.component.friction(brkwy_trq = breakaway
motorInertia = found.component.inertia(inertia = motor_inertia

end

connections

connect(p, rotorResistor.p);
connect(rotorResistor.n, rotorInductor.p);
connect(rotorInductor.n, rotEMechConv.p);
connect(rotEMechConv.n, n);
connect(rotEMechConv.R, friction.R, motorInertia.I, R);
connect(rotEMechConv.C, friction.C, C);

end
```
Outline

- Part 1: Introduction to Simscape
  - Review approaches to modelling
  - Overview of Simscape-based libraries
  - Introduction to physical networks
  - Simscape tutorial: DC motor

- Part 2: Application examples
  - PMSM and drive
  - Landing gear extension/retraction
  - Engine cooling system
PMSM and drive modelling

- **Components**
  - PMSM
  - Switching devices (IGBTs)
  - Controller

- **Tools**
  - SimPowerSystems
    - System-level to support circuit and controller design
  - SimElectronics
    - Detailed switching device models – fine-tuning and losses prediction
IGBT modelling

I-V characteristic
- Piecewise linear
- Lookup table
- Physics-based
- Composite/complex

Dynamics
- No dynamics
- Behavioural
- Constant capacitance
- Nonlinear charge model

Complexity:
- Slower simulations, detailed parameterization required
- Higher-fidelity/better predictions
## IGBT modelling

<table>
<thead>
<tr>
<th>Complexity</th>
<th>SimPowerSystems</th>
<th>SimElectronics</th>
<th>SPICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Piecewise linear</td>
<td>Nonlinear, physics-based, simple</td>
<td>Composite from complex physics-based components</td>
<td></td>
</tr>
<tr>
<td>No dynamics or constant capacitance</td>
<td>Datasheet parameterization</td>
<td>Manufacturer NETLIST</td>
<td></td>
</tr>
</tbody>
</table>
pmsm_drive
System-level equivalent of PMSM and drive

- Example pmsm_drive suitable for
  - Component design
  - Electrical system design

- What about the system designer e.g. for a hybrid vehicle?
  - Model the whole system and predict efficiency/performance
  - Multiple drive cycles
  - HIL test
Abstracted PMSM drive

- Energy-based
  - Mechanical power out = power in minus losses
  - Losses (physics-based analysis)
    - Copper losses proportional to the torque
    - Iron losses dependent on voltage and switching frequency
  - Losses (empirical approach)
    - Tabulate as function of torque and speed (and possibly DC volts)
    - Table derived from detailed PMSM and drive model

- Assume torque is closed-loop controlled
Abstracted PMSM drive
sm_landing_gear
Landing Gear Design

Model:

Problem: Determine the force required of the hydraulic cylinder to meet system specifications

Solution: Use Simulink, SimMechanics, SimHydraulics, and SimElectronics to design the system.
Modeling Thermal Fluid Systems

Thermal Liquid Library

- New domain and library for thermal liquid systems
  - Single-phase liquids
  - Fluid properties vary with temperature

- Applications
  - Heat exchangers
  - Pipelines
Engine cooling example
Challenges in Thermal Fluid Simulation

- Restrictions on block combinations
  - Some combinations require fluid volumes at nodes
  - Blocks sharing differential variables makes initialization difficult

- Poor behavior at regime changes, low flow rates, and flow reversals
  - Zero crossing difficulties, solver issues
Addressing challenges with full-flux method

- Each element contains a volume of fluid
  - No fluid volumes at nodes

- Control volume approach used in each element
  - Internal nodes defined
  - Network nodes have no differential variables

- Power flux includes convection+conduction
  - Smooth temperature variation of node temperature at zero flow
Simscape language best practice

- Build incrementally
- Write test scripts/harnesses
- Avoid discontinuities
- Use appropriate level of fidelity
- Write out equations for custom blocks before building
Conclusions

- Simscape extends Simulink to
  - Support a network approach
  - Create models that topologically match the physical system

- Create component models by
  - Using ready-made blocks from MathWorks’ libraries
  - Constructing composite components from Simscape foundation library blocks
  - Writing custom equations in Simscape language

- All the benefits of Simulink apply e.g.
  - Code generation
  - Tight integration with MATLAB®
LINKS

Physical modelling central web-page:
http://www.mathworks.co.uk/physical-modeling/

Essentials of physical modelling:
http://www.mathworks.co.uk/help/physmod/simscape/ug/essential-physical-modeling-techniques.html
Viewing Simscape Simulations Results

ssc_explore

- Explore simulation results from entire physical network
  - Select multiple signals
  - Overlay or separate plots
  - Arrange plots
  - Extract plot to separate window
- Spend more time analyzing, less time simulating
- Download from MATLAB Central

Zero-Crossing Statistics

- Log zero-crossing statistics for Simscape networks
  - Shows when ZCs occur
  - Can help indicate location of simulation bottlenecks

```matlab
if (abs(w) <= vel_thr)
  % Linear region
  t = brkwy_trq_th * w / vel_thr;
elseif w > 0
  t = visc_coef * w + Col_trq + ...  
      (brkwy_trq - Col_trq) * exp(-
else
  t = visc_coef * w - Col_trq - ...  
      (brkwy_trq - Col_trq) * exp(-
end
```
Local Solvers Improve Simulation Performance

- Use fixed-step, implicit solvers locally on Simscape networks
  - Use implicit solvers only where necessary
- Configure solver per physical network
  - Run different physical networks at different sample rates
    - Electrical (higher), mechanical (lower)
- Primary benefit is to speed-up simulations where fixed-step solvers are required, like HIL