MATLAB EXPO 2017

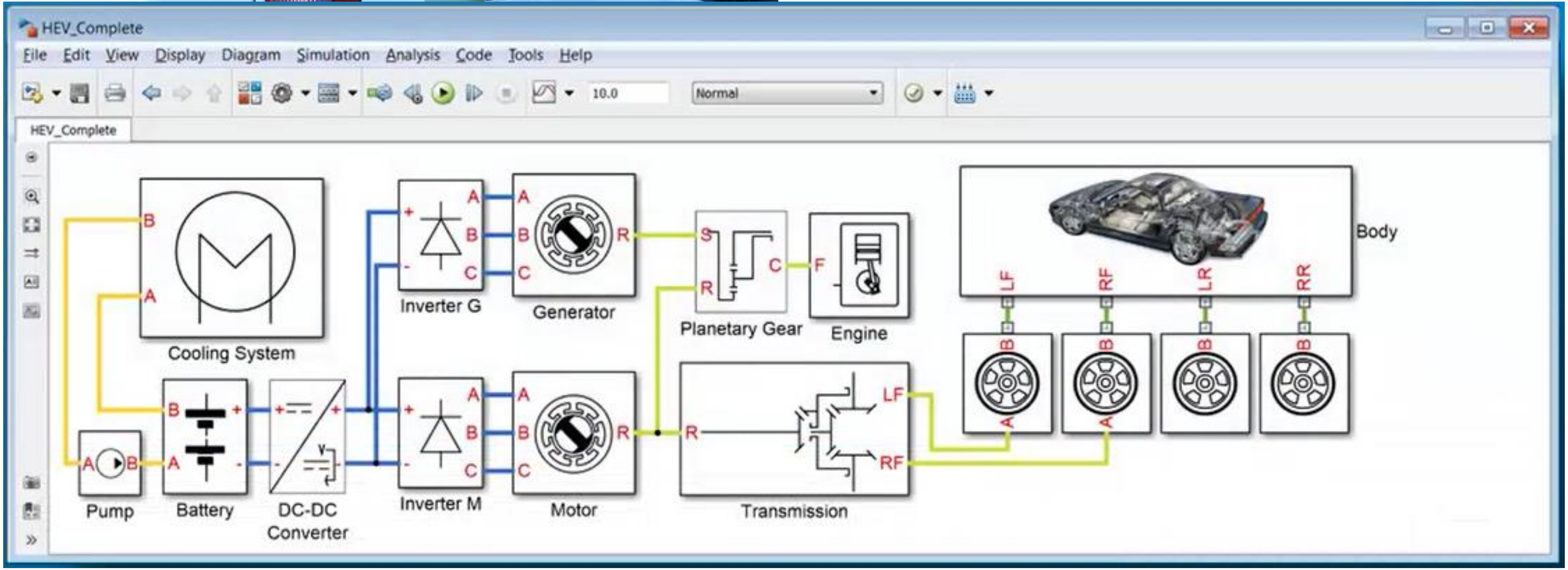
How Simscape™ Supports Innovation for Cyber-Physical Systems

Rick Hyde

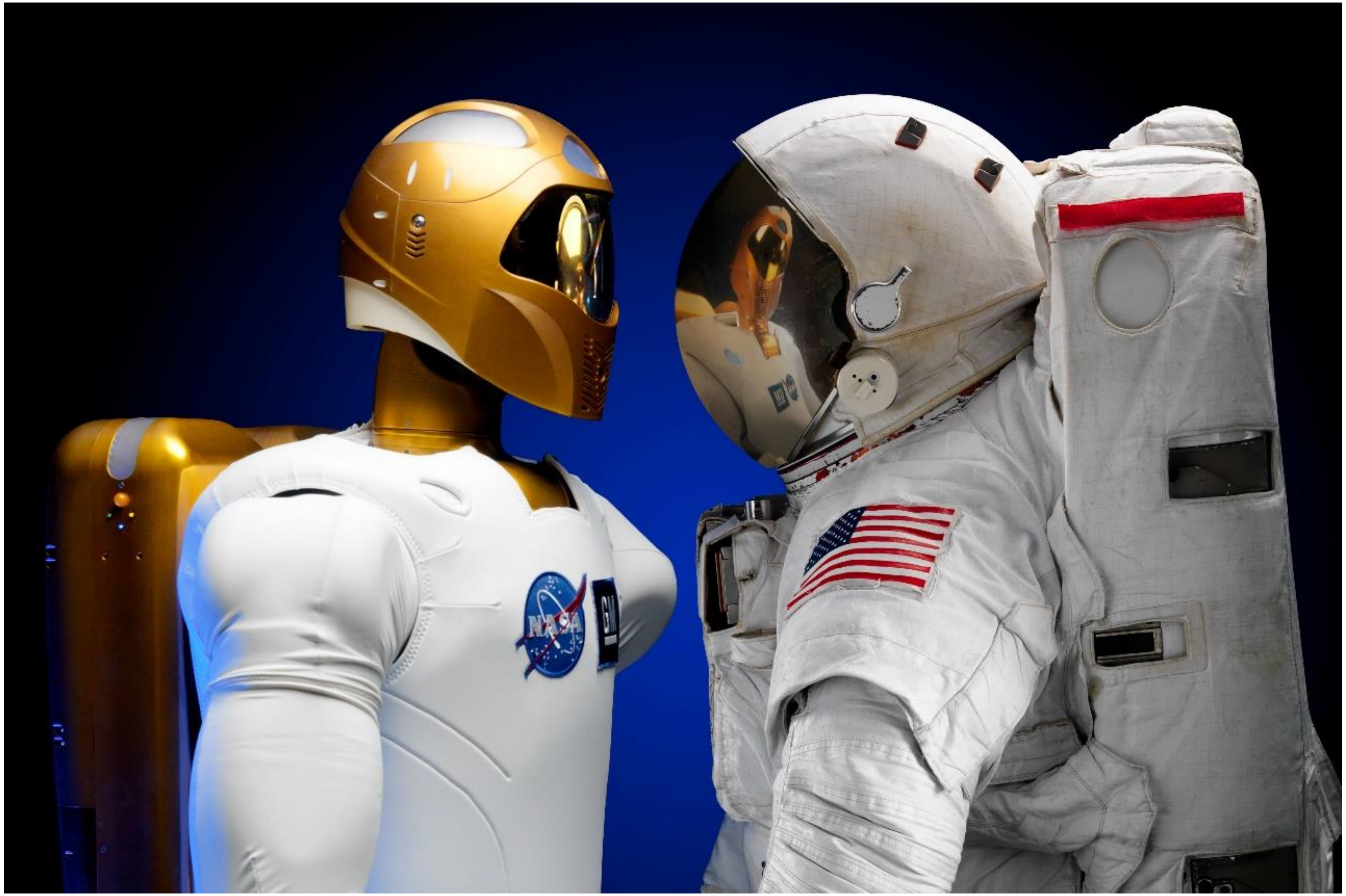
How can we use system-level modelling to support *innovative product design*?

Innovation in electric and hybrid vehicles

Electrical
Mechanical
Fluid



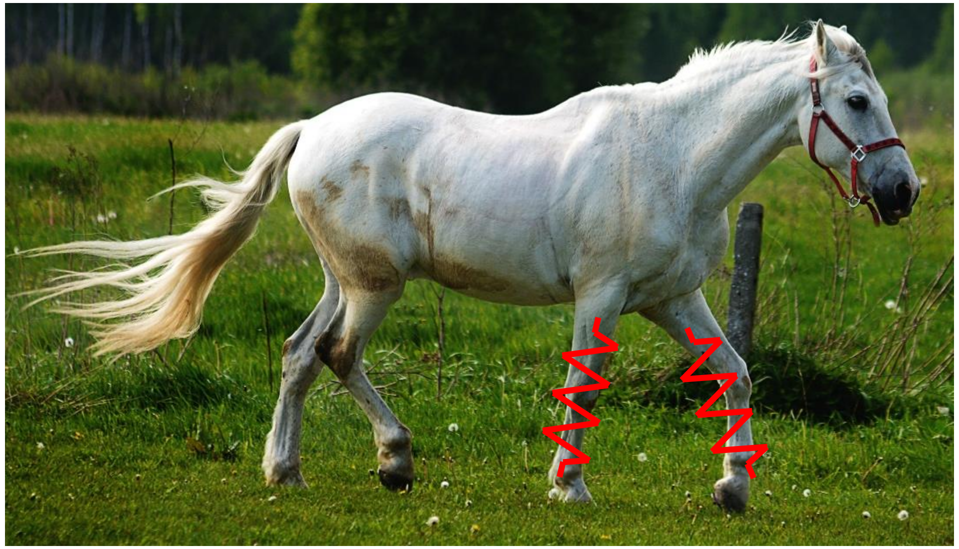
Innovation in robotics

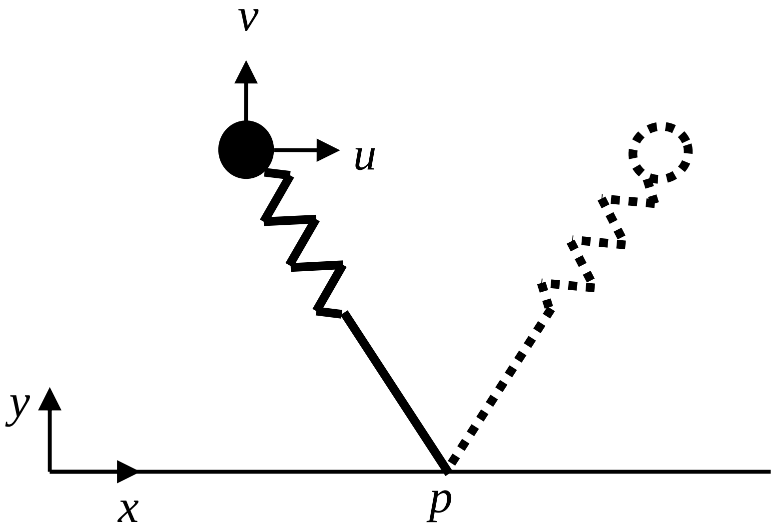


Example: Quadruped running robot
Biologically-inspired design (Biomimetics)

- Animal terrestrial motion
  - Muscles are inefficient (30%)
  - Muscles are also the energy store
  - Running gait uses kinetic energy recovery
  - The leg is well modelled by a linear spring

- Innovation: Use equivalent inverted pendulum model as basis for robot

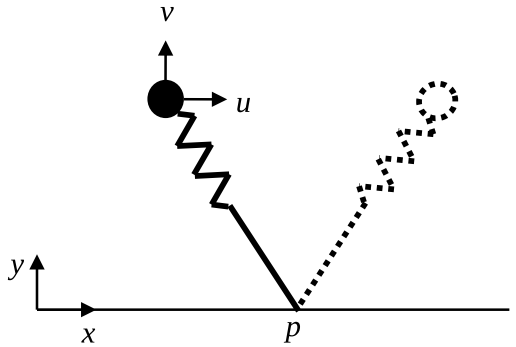


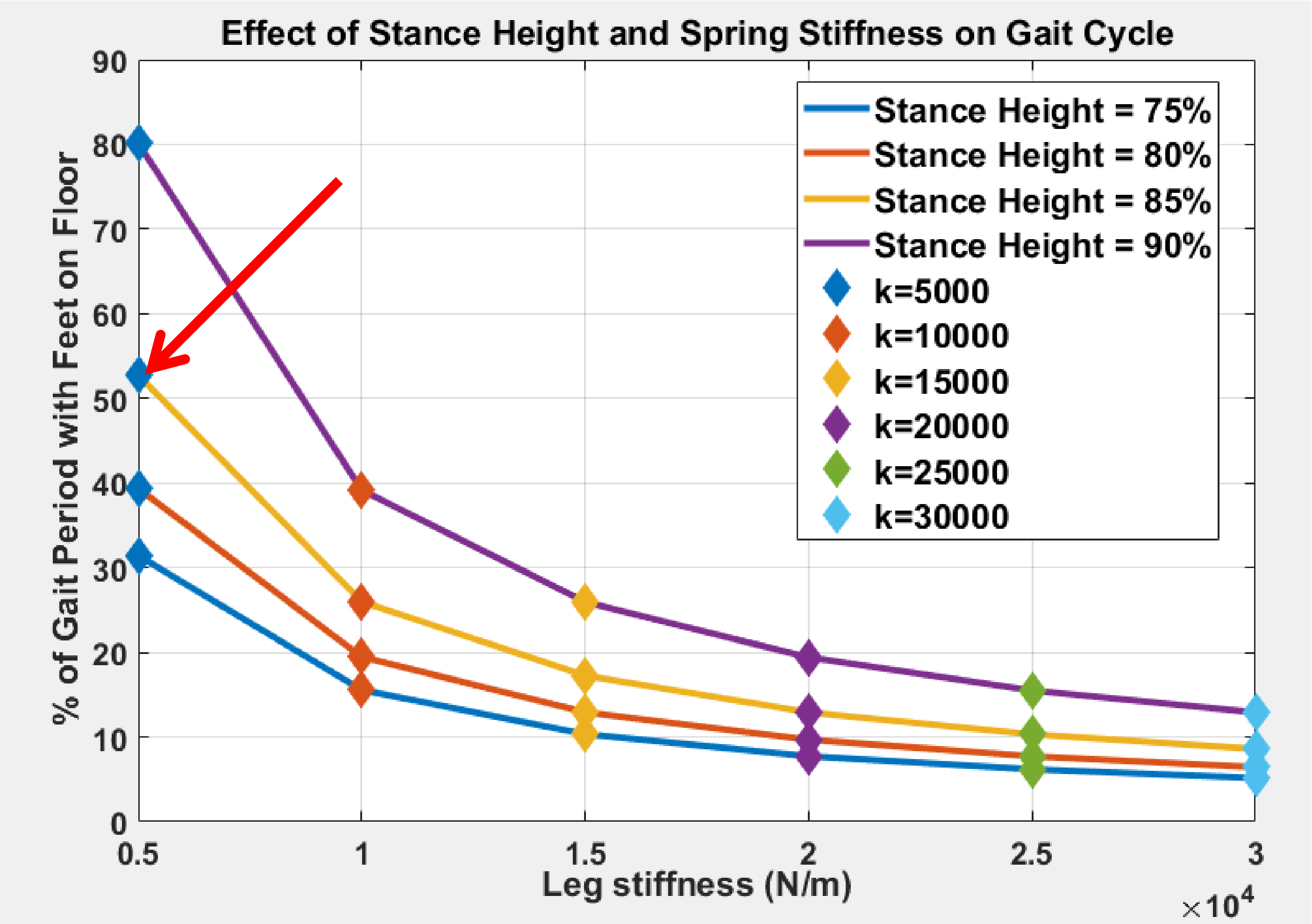


Running robot design example

Design step #1 – gait selection

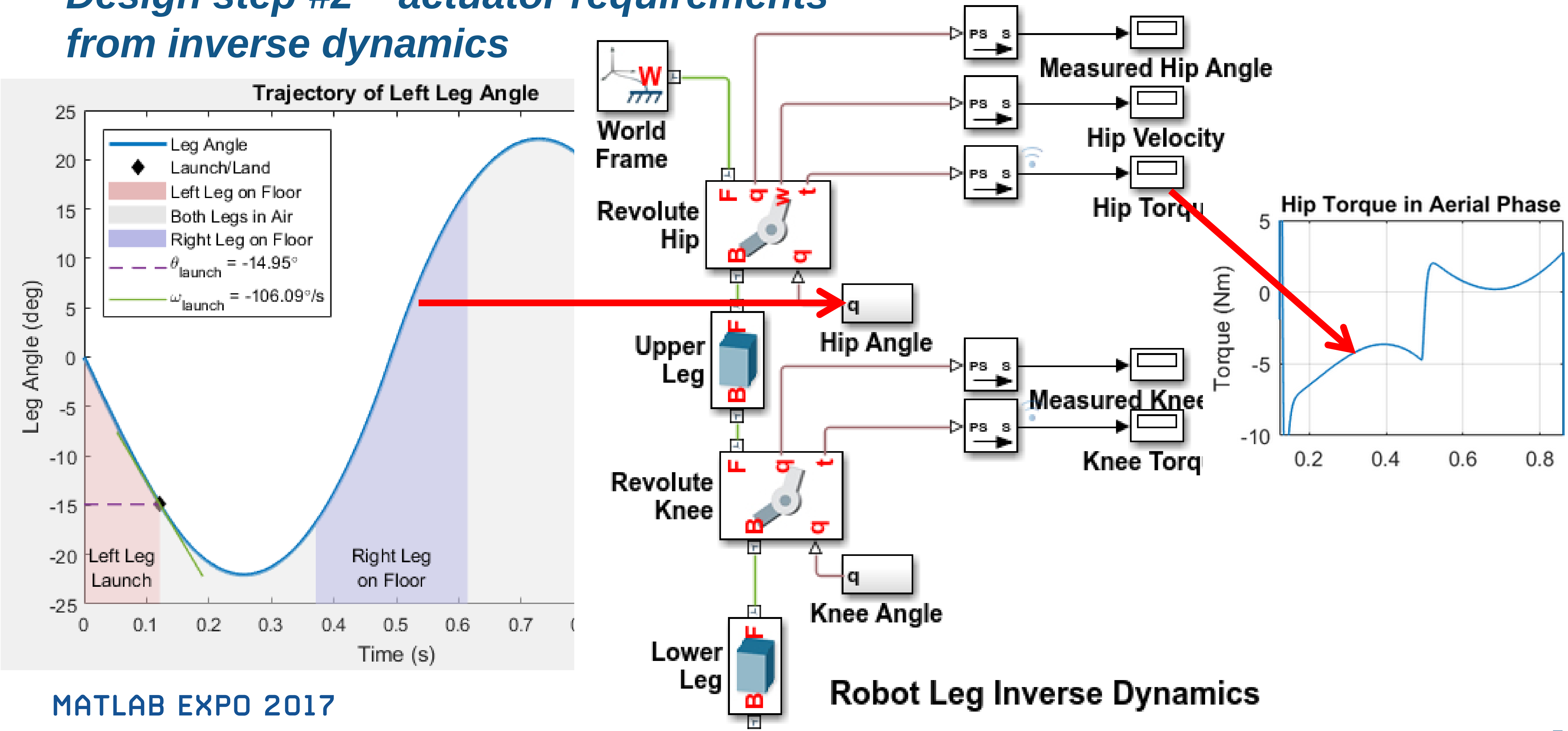
- Fixed parameters
  - Leg length
  - Running speed
  - Mass
- Design parameters
  - Leg (spring) stiffness
  - Stance height
- Simple point-mass model
  - MATLAB script for trade-off





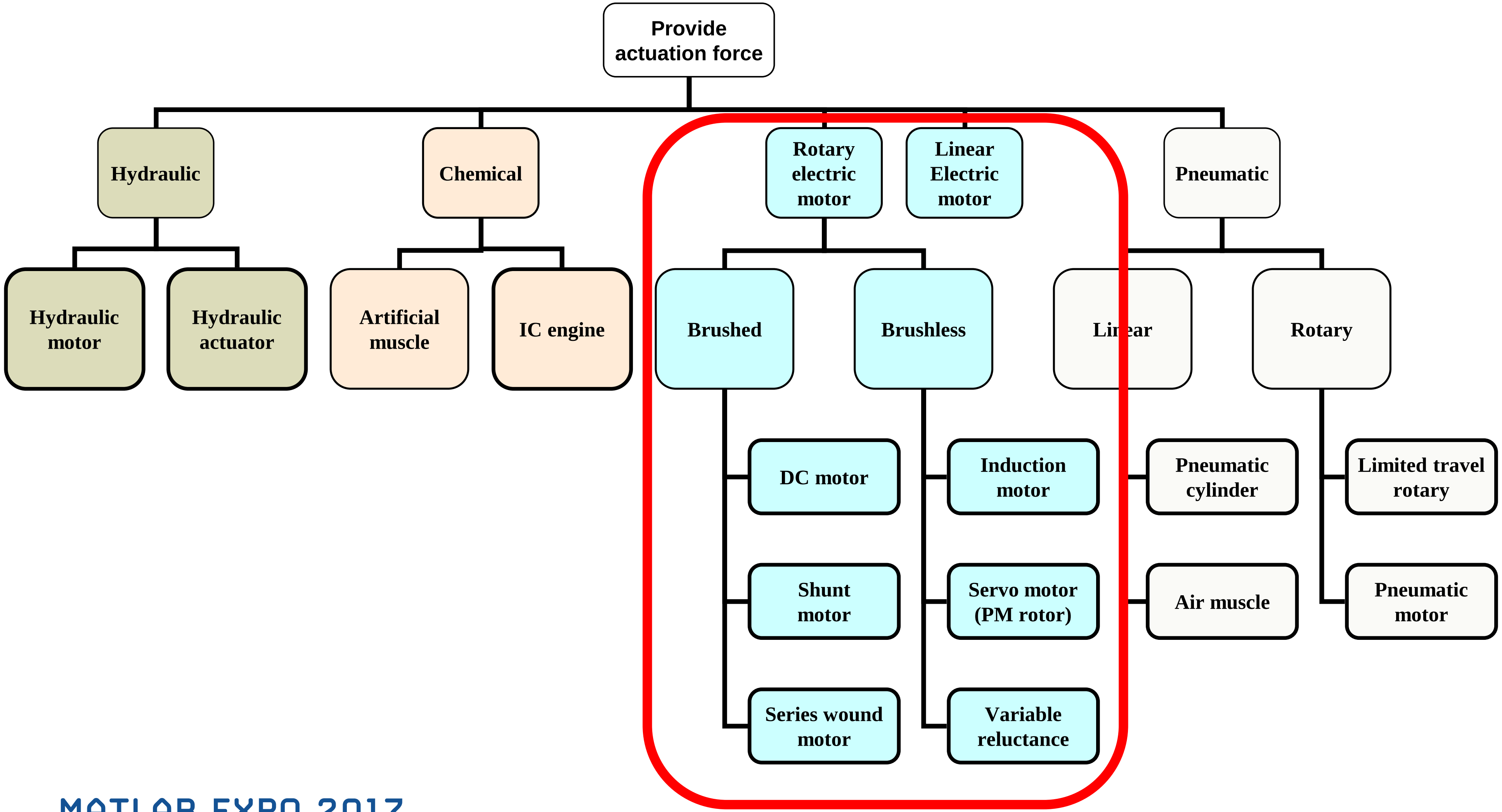
Running robot design example

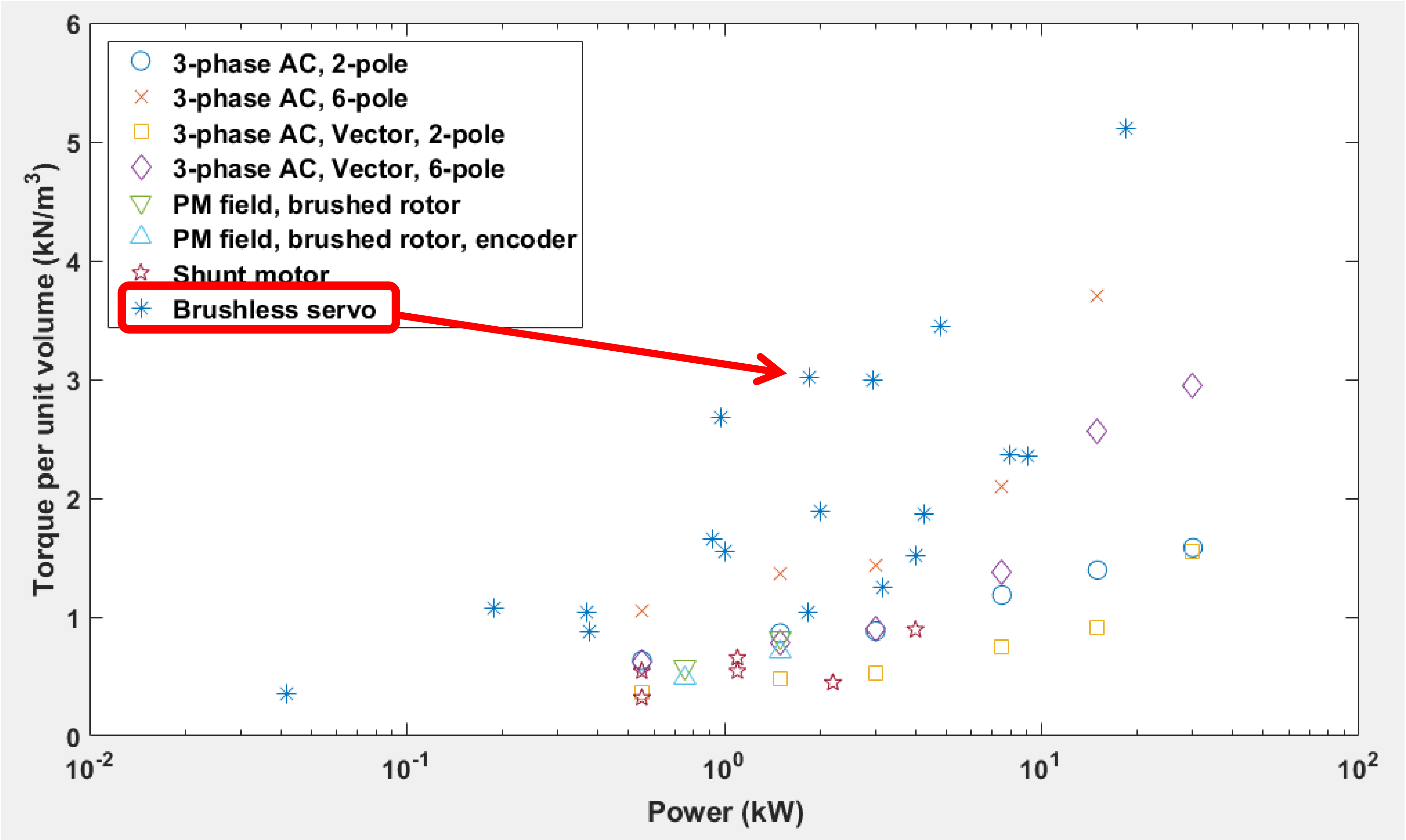
Design step #2 – actuator requirements from inverse dynamics



Running robot design example

Design step #3 – actuator selection

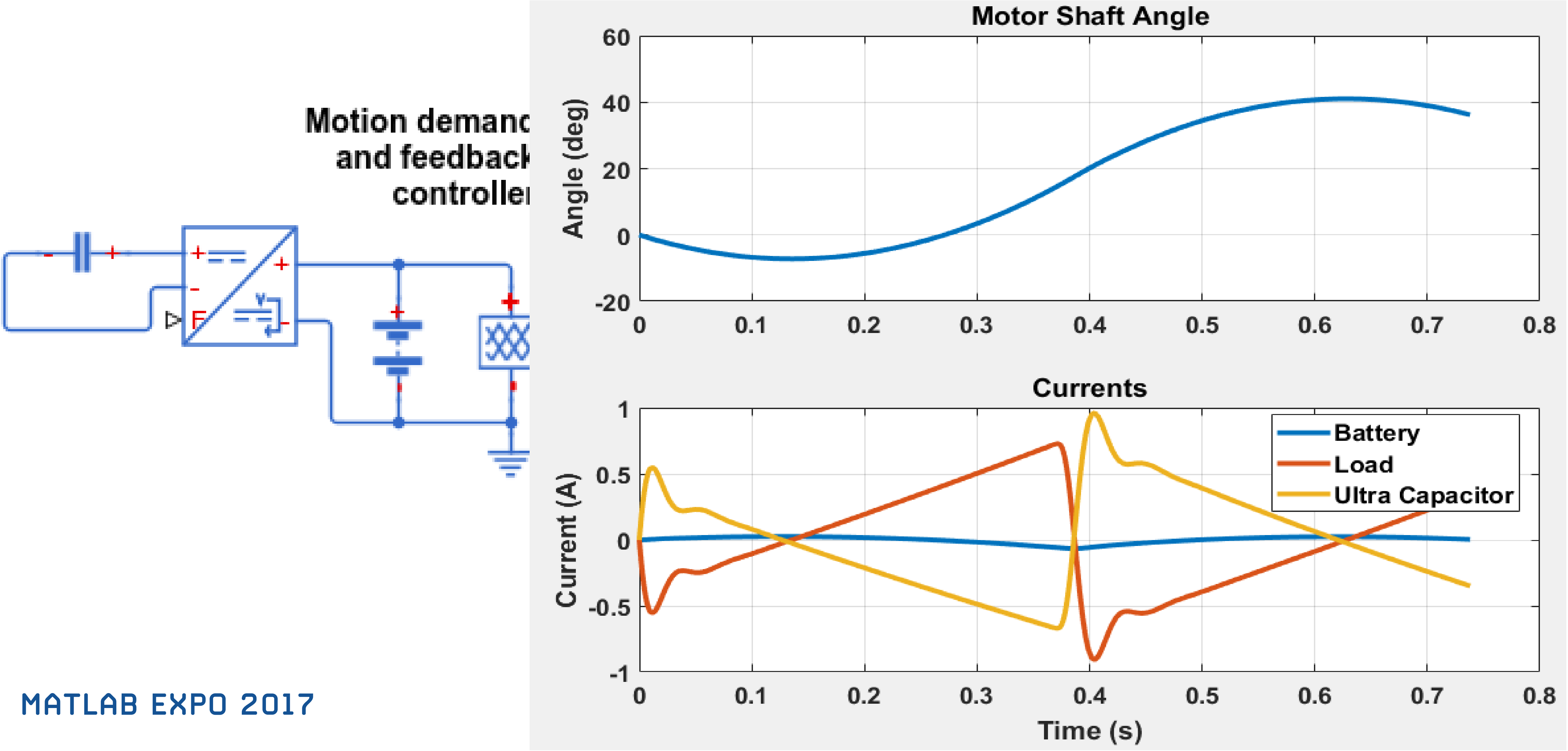




3-phase AC, 2-pole
3-phase AC, 6-pole
3-phase AC, Vector, 2-pole
3-phase AC, Vector, 6-pole
PM field, brushed rotor
PM field, brushed rotor, encoder
Shunt motor
Brushless servo
Torque per unit volume (kN/m^3)
Power (kW)
0
1
2
3
4
5
6
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}

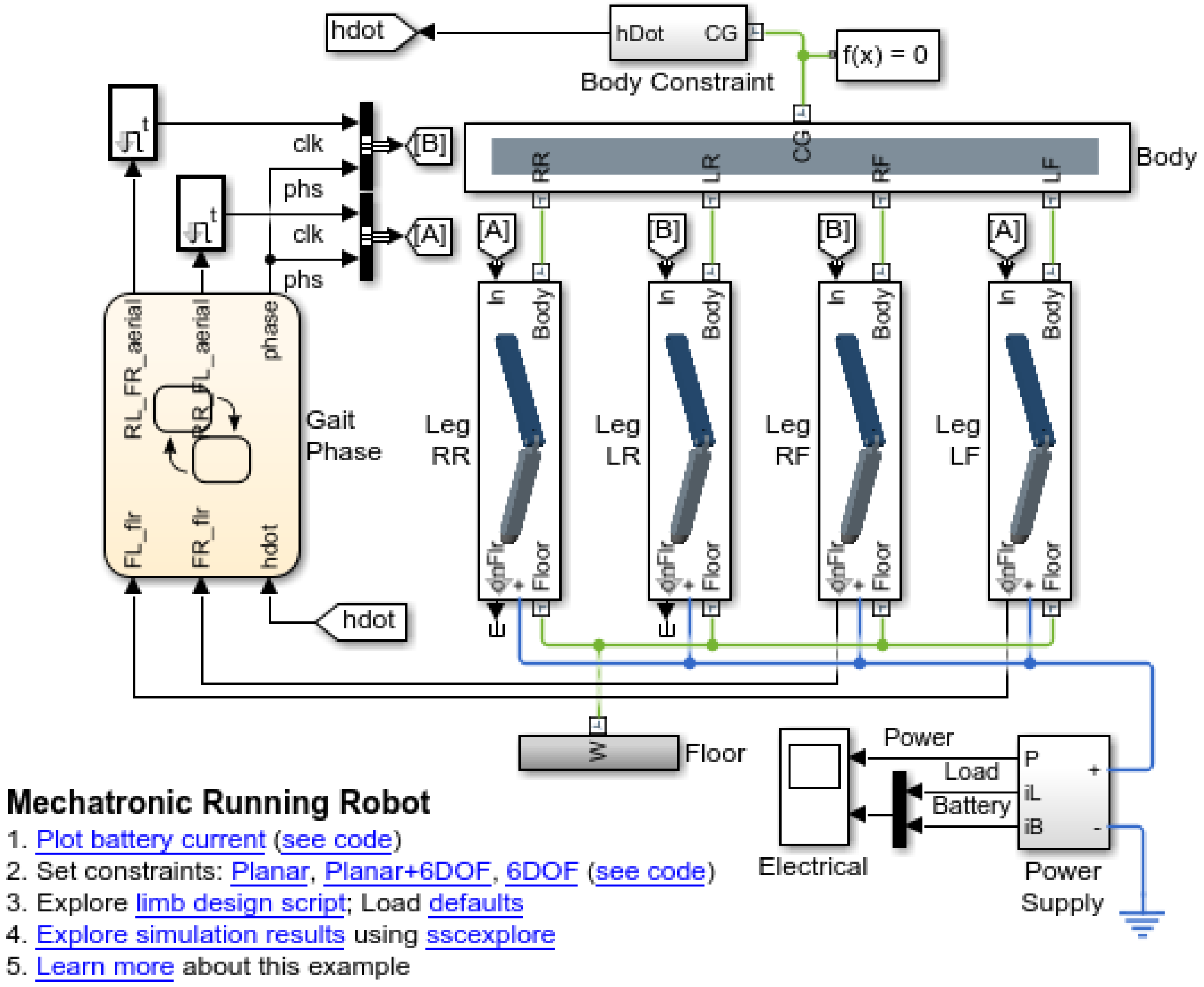
Running robot design example

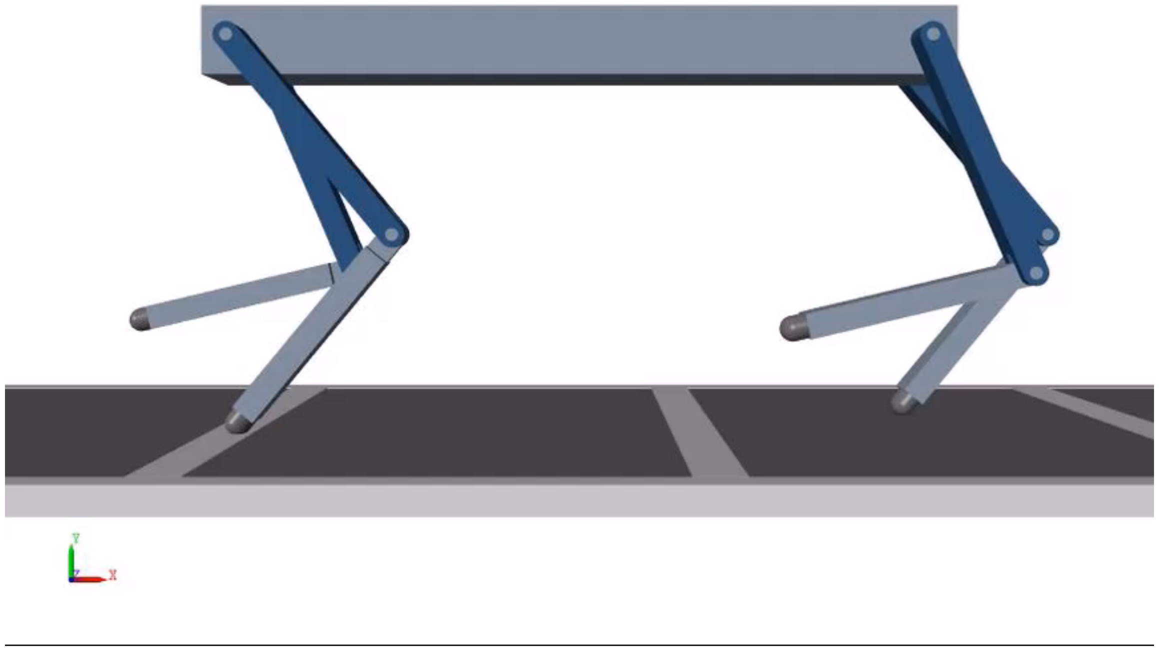
Design step #4 – actuation validation



Running robot design example

Design step #5 – evaluation





Motor efficiency at rated load = 95%
Motor efficiency for trotting gait = 84%

http://www.mathworks.com/matlabcentral/fileexchange/64237-running-robot-model-in-simscape

Running robot design example

Design step automation using MATLAB scripting

```
%% Generate nominal gait, leg length and payload mass

% Biomechanical parameters
L = 1.0;     % Leg length (m)
m = 25;      % Mass (Kg)
k = 5315;    % Leg stiffness (N/m)

% Initial conditions for normalized positions and speeds
x0 = 0.0;     % Horizontal position of mass in middle of stance phase ()
y0 = 0.85*L; % Height of mass in middle of stance phase ()
u0 = 2.0;     % Horizontal speed in middle of stance phase (/s)
```

- Automation permits greater understanding of design trade-offs
  - e.g. see effect of gearbox ratio on efficiency

Gear ratio	80	100	120
Efficiency	84%	81%	78%

Running robot design example

Key points

1. Multiple models
2. Each model matched to a design task
3. Design data passed between models
4. Automation to support analysis & optimisation
5. Code generation for HIL testing

Enables product design innovation in a way that starting with the CAD tool could never do

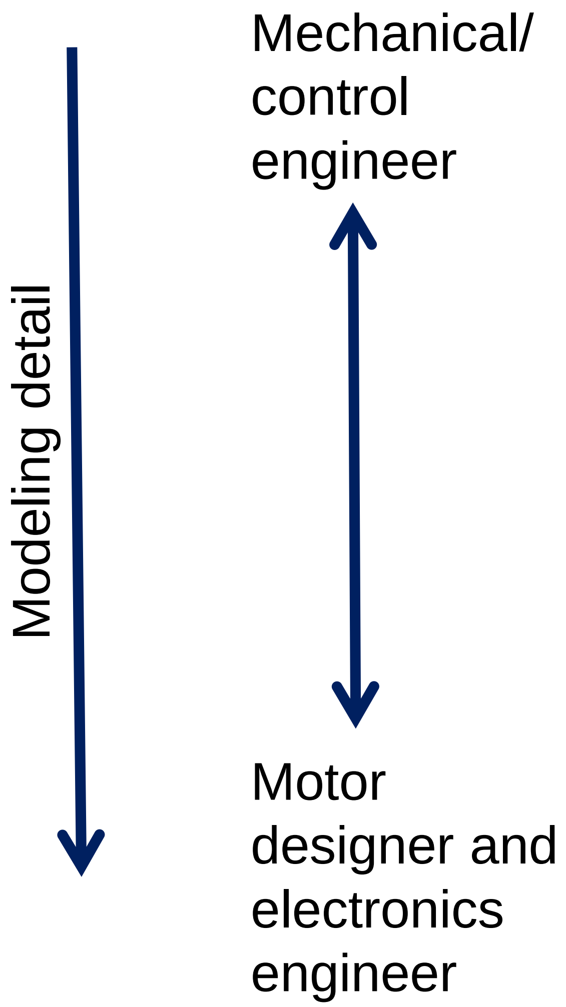
Building the right model for the task at hand can be challenging

Requirements not understood by project management

Identification of required modelling detail

Identify required modelling detail for *PMSM drives*

1. **System-level simulation**
   - Torque-speed behaviour
   - Model motor losses as part of overall efficiency calculation
   - Thermal & fault modelling
2. **Component validation**
   - Ensure motor stays within manufacturer operating limits
   - Detailed analysis of impact on other components e.g. power harmonics
3. **Component design**
   - Motor and/or drive circuitry
   - Determine overall actuation losses
   - Understand/predict fault behaviour



Building the right model for the task at hand can be challenging

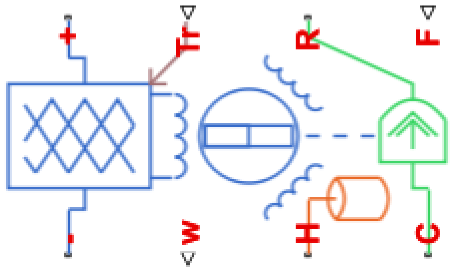
Requirements not understood by project management

Identification of required modelling detail

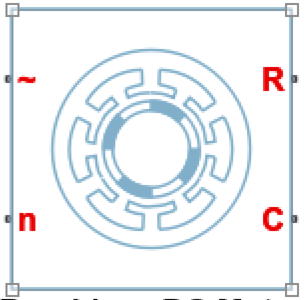
Limited time and nothing to build on – starting from scratch

Lacking domain knowledge

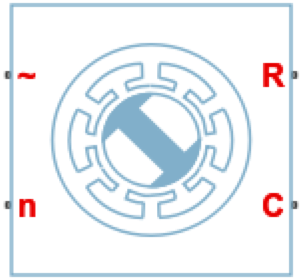
Simscape library components provide a useful starting point and encapsulate some domain knowledge



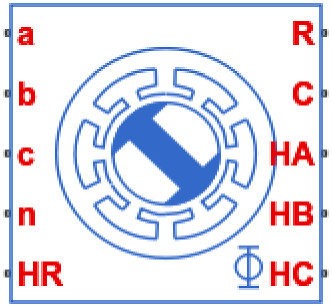
Servomotor



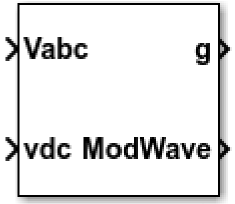
Brushless DC Motor



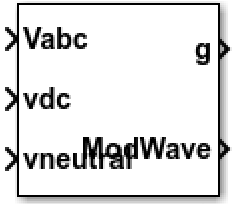
Permanent Magnet Synchronous Motor



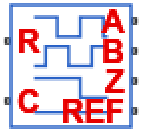
FEM-Parameterized PMSM



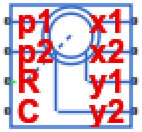
PWM Generator (3-phase, 2-level)



PWM Generator (3-phase, 3-level)



Encoder



Resolver

Modeling detail

Building the right model for the task at hand can be challenging

Requirements not understood by project management

Identification of required modelling detail

Limited time and nothing to build on – starting from scratch

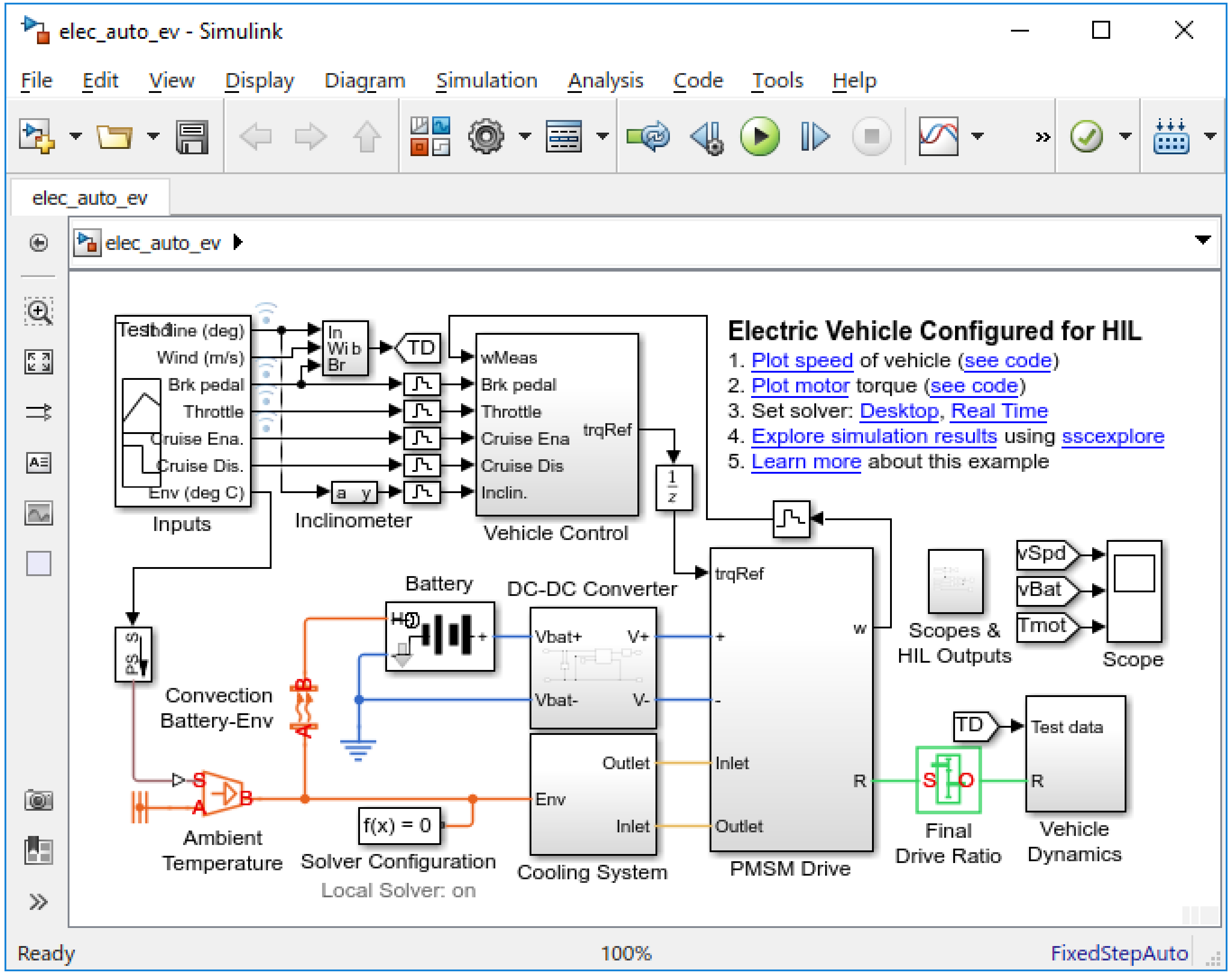
No data

Lacking domain knowledge

Modelling a PMSM with limited supplier data
Tune to measurement data

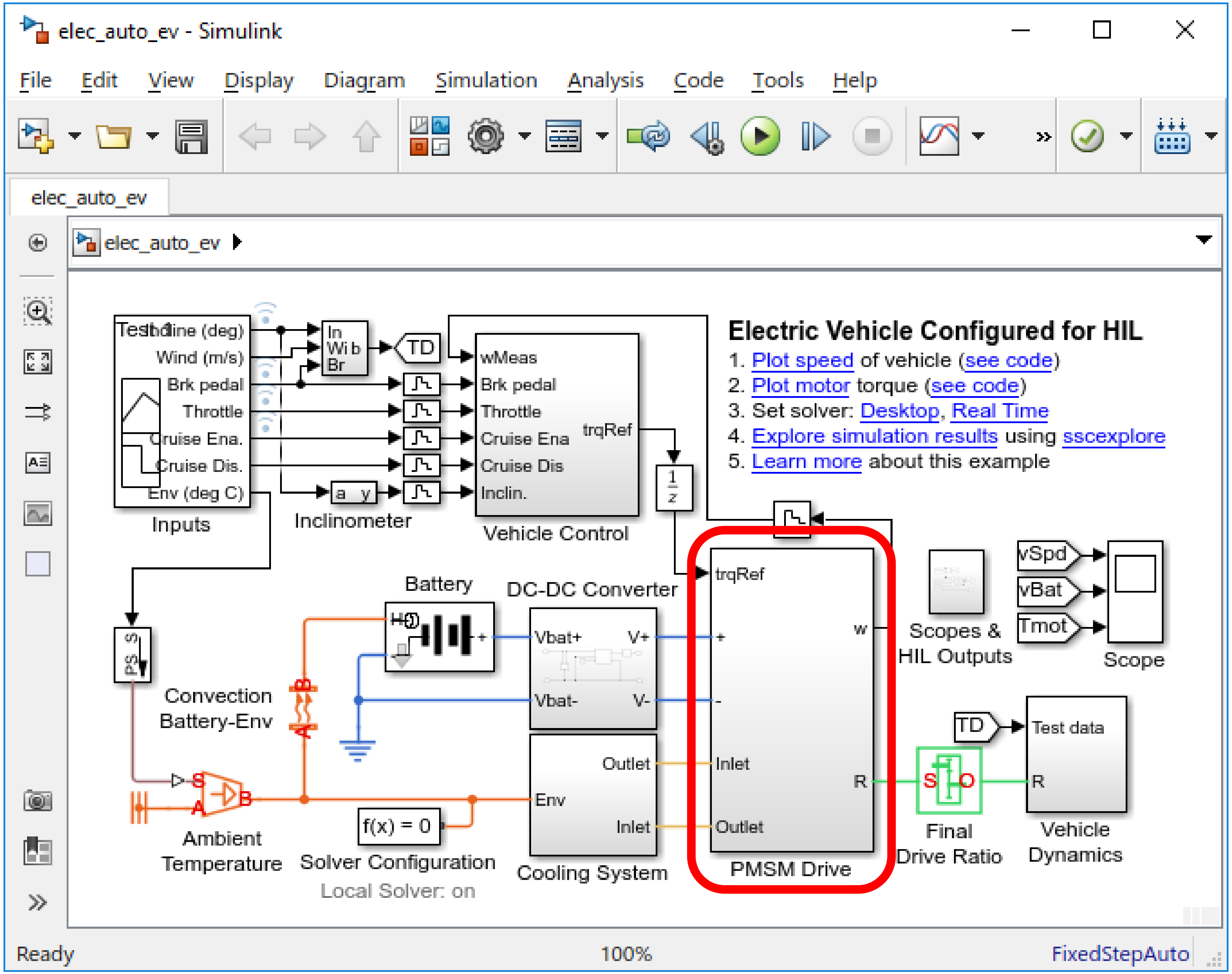
See PMSM parameter identification example in Track 2 at 16:15pm

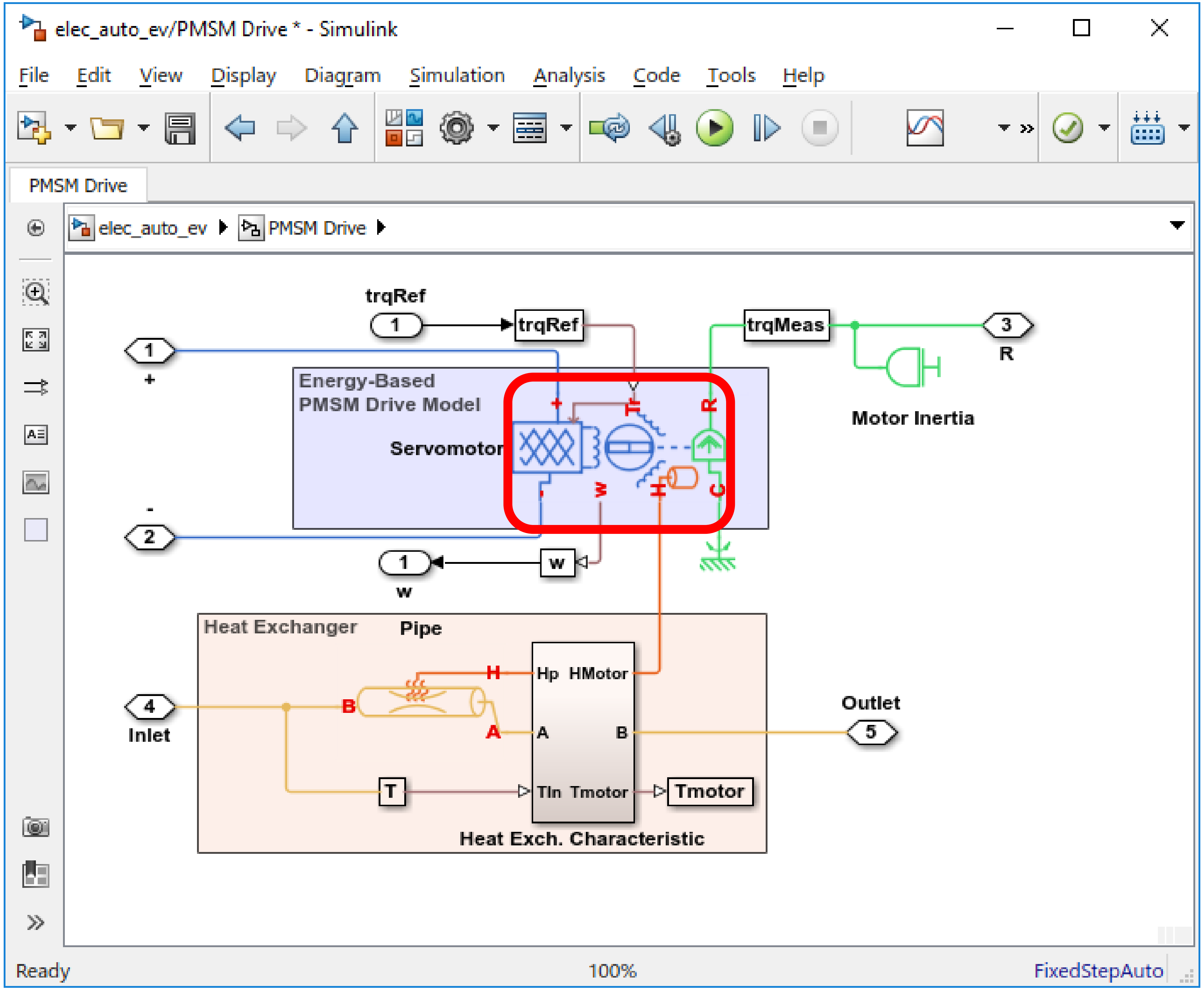
Using abstraction to deal with limited data



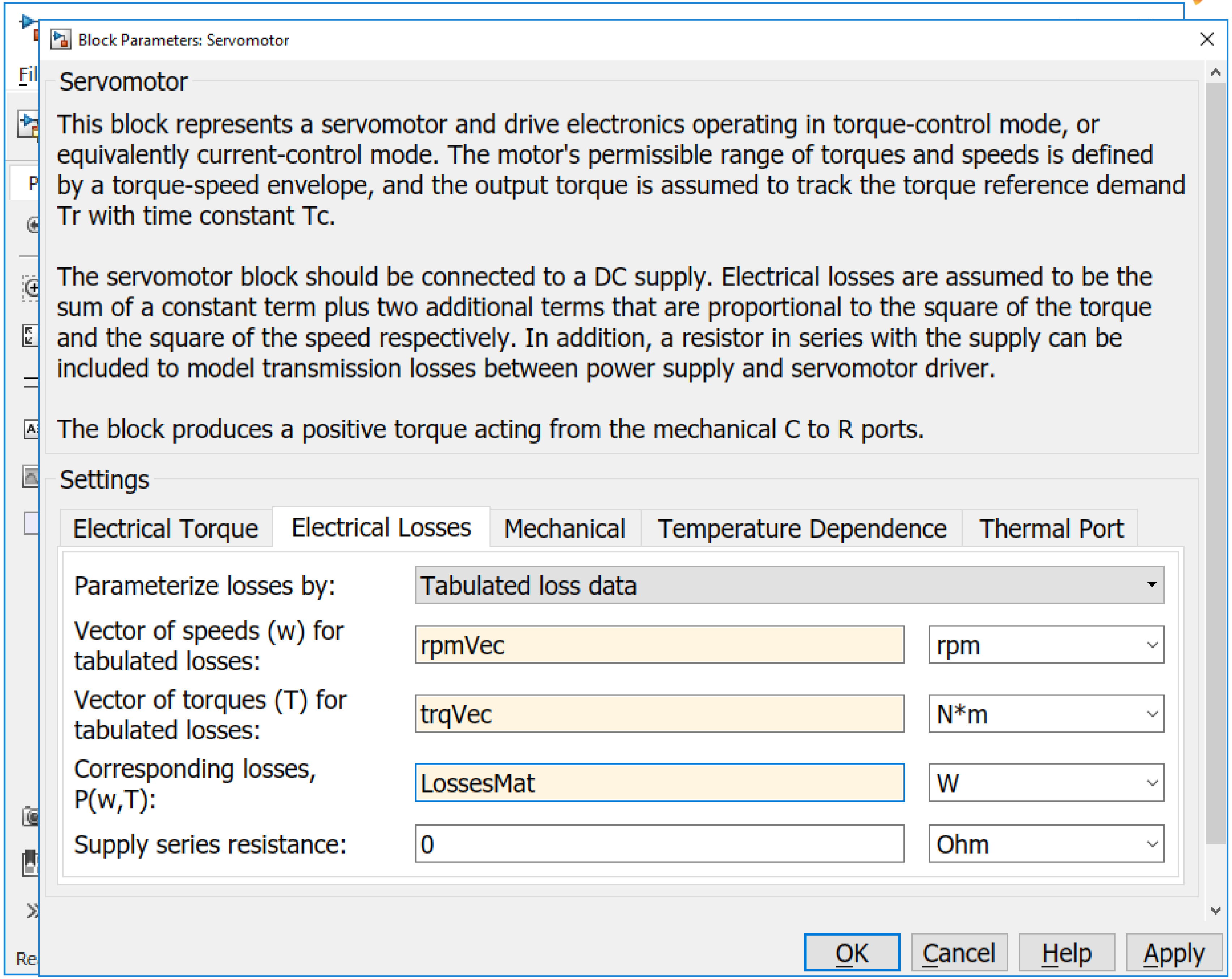
R2017a/R2017b:
elec_auto_ev.slx

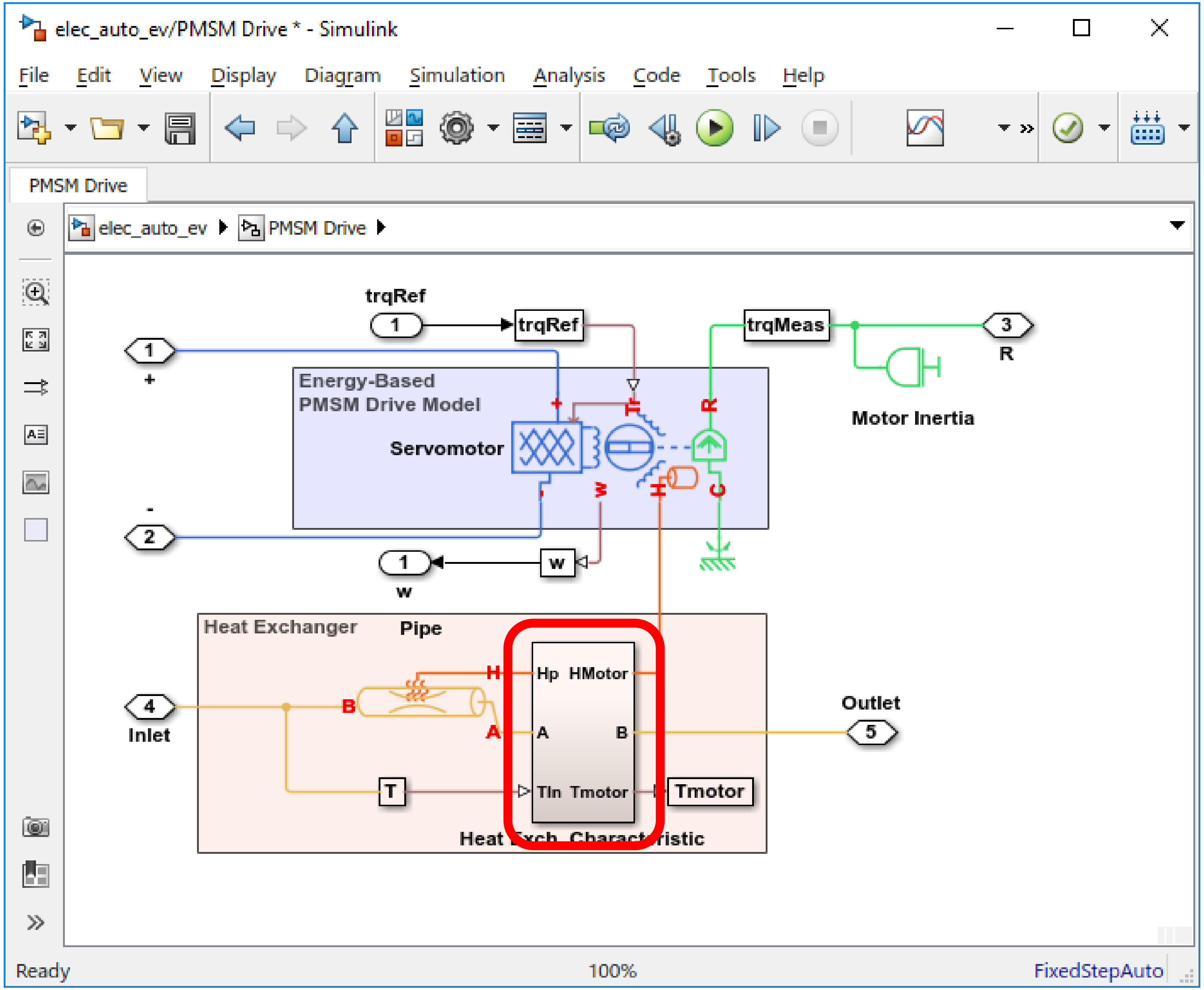
R2016b/R2016a/R2015b:
elec_electric_vehicle.slx


elec_auto_ev - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
elec_auto_ev
Electric Vehicle Configured for HIL
1. Plot speed of vehicle (see code)
2. Plot motor torque (see code)
3. Set solver: Desktop, Real Time
4. Explore simulation results using sscexplore
5. Learn more about this example
Wind (m/s)
Brk pedal
Throttle
Cruise Ena.
Cruise Dis.
Env (deg C)
Inputs
Inclinometer
wMeas
Brk pedal
Throttle
Cruise Ena
Cruise Dis
Inclin.
trqRef
Vehicle Control
TD
Battery
DC-DC Converter
Vbat+
V+
Vbat-
V-
Convection Battery-Env
Ambient Temperature
f(x) = 0
Solver Configuration
Local Solver: on
Outlet
Env
Inlet
Cooling System
trqRef
w
Inlet
R
Outlet
PMSM Drive
Scopes & HIL Outputs
vSpd
vBat
Tmot
Scope
TD
Test data
R
Final Drive Ratio
Vehicle Dynamics
Ready
100%
FixedStepAuto

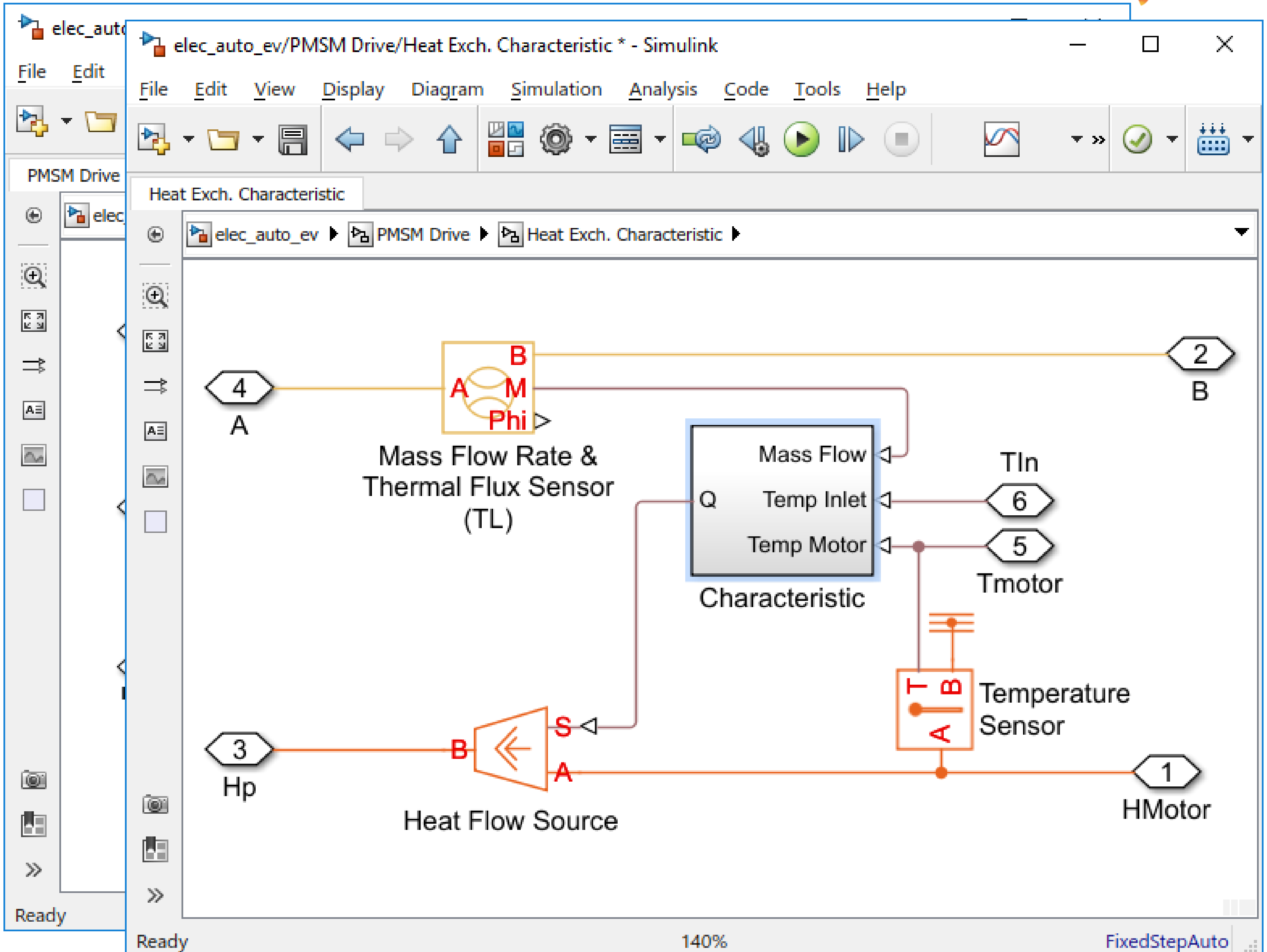


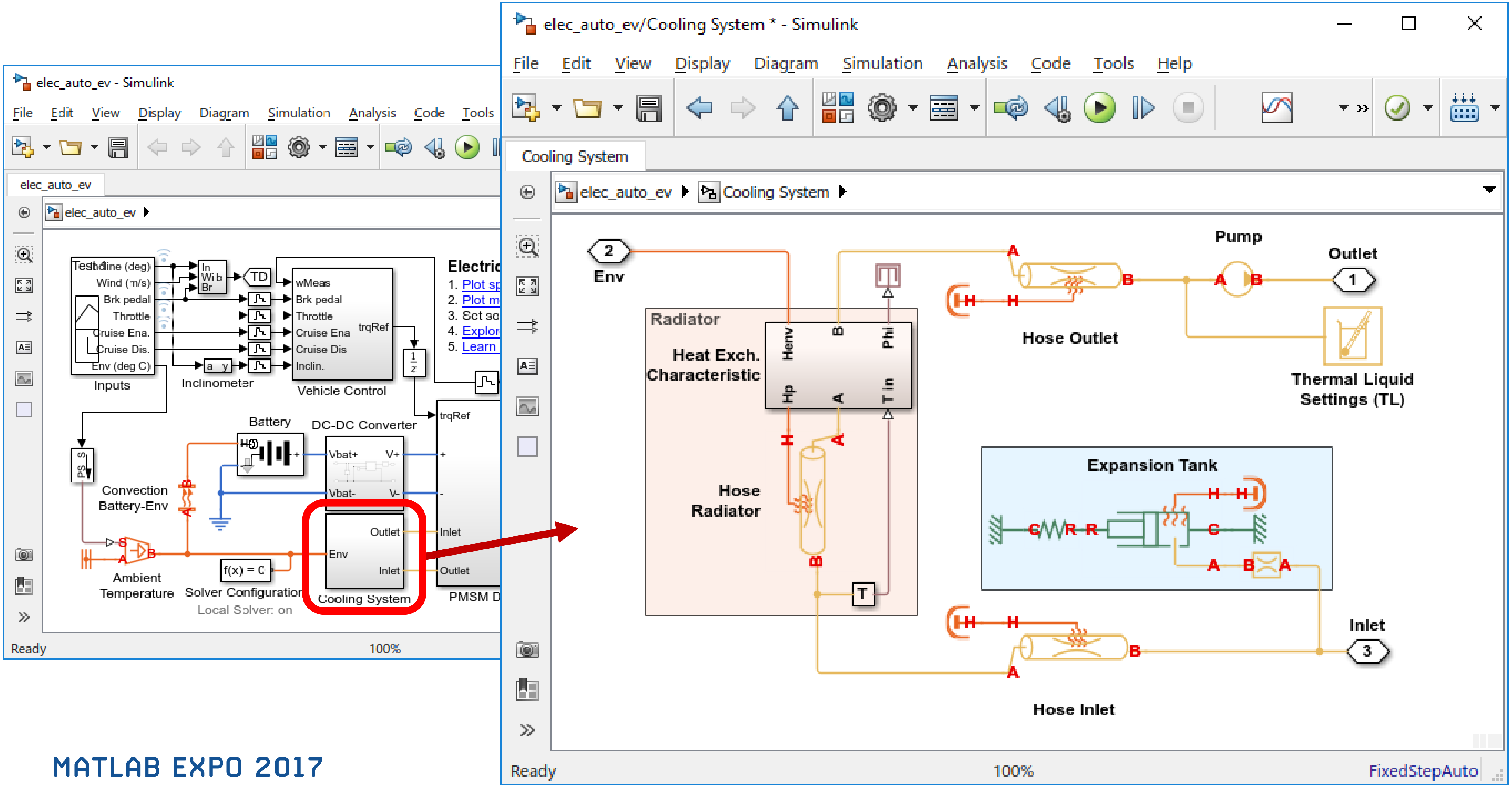
elec_auto_ev/PMSM Drive * - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
PMSM Drive
elec_auto_ev
PMSM Drive
trqRef
trqRef
trqMeas
R
Motor Inertia
Energy-Based
PMSM Drive Model
Servomotor
w
Heat Exchanger
Pipe
Hp HMotor
Inlet
A
B
Outlet
TIn Tmotor
Tmotor
T
Heat Exch. Characteristic
Ready
100%
FixedStepAuto

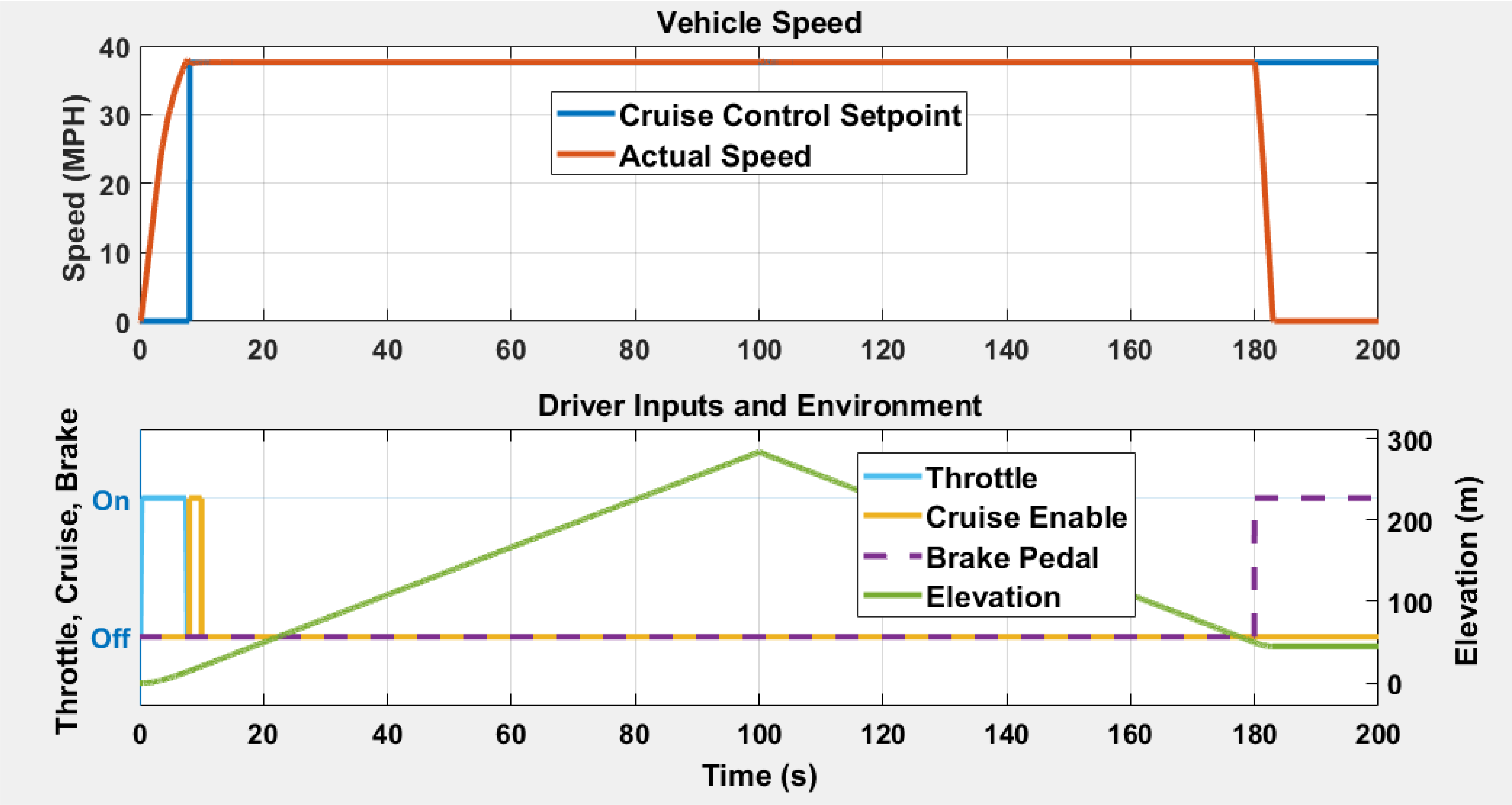

Block Parameters: Servomotor
Servomotor
This block represents a servomotor and drive electronics operating in torque-control mode, or equivalently current-control mode. The motor's permissible range of torques and speeds is defined by a torque-speed envelope, and the output torque is assumed to track the torque reference demand Tr with time constant Tc.
The servomotor block should be connected to a DC supply. Electrical losses are assumed to be the sum of a constant term plus two additional terms that are proportional to the square of the torque and the square of the speed respectively. In addition, a resistor in series with the supply can be included to model transmission losses between power supply and servomotor driver.
The block produces a positive torque acting from the mechanical C to R ports.
Settings
Electrical Torque
Electrical Losses
Mechanical
Temperature Dependence
Thermal Port
Parameterize losses by:
Tabulated loss data
Vector of speeds (w) for tabulated losses:
rpmVec
rpm
Vector of torques (T) for tabulated losses:
trqVec
N*m
Corresponding losses, P(w,T):
LossesMat
W
Supply series resistance:
0
Ohm
OK
Cancel
Help
Apply

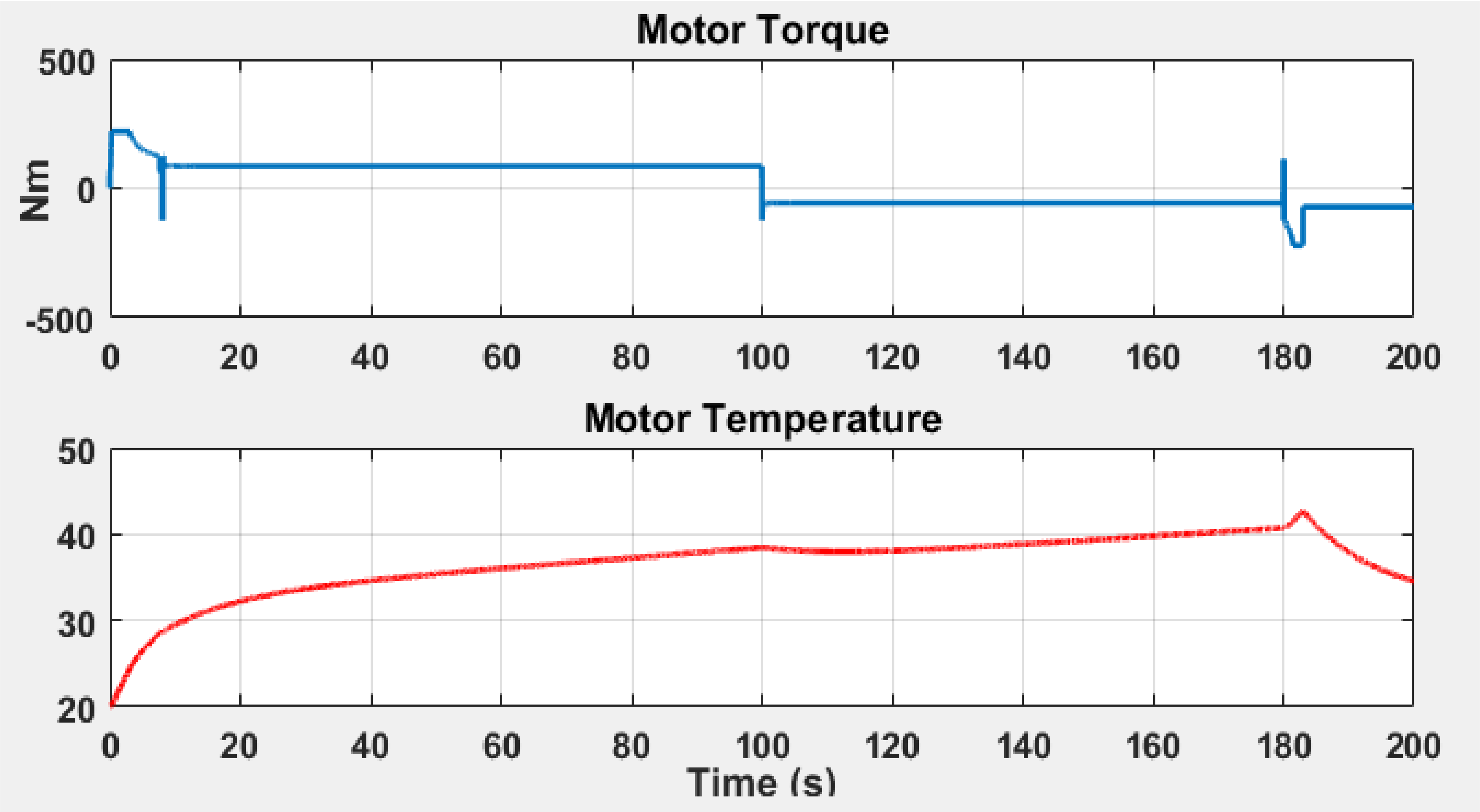


elec_auto_ev/PMSM Drive * - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
PMSM Drive
elec_auto_ev
PMSM Drive
trqRef
1
trqRef
trqMeas
3
R
Motor Inertia
1
+
Energy-Based
PMSM Drive Model
Servomotor
-
2
1
w
w
Heat Exchanger
Pipe
Hp HMotor
4
Inlet
A
B
Outlet
5
T
TIn Tmotor
Tmotor
Heat Exch. Characteristic
Ready
100%
FixedStepAuto


elec_auto_ev/PMSM Drive/Heat Exch. Characteristic * - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
Heat Exch. Characteristic
elec_auto_ev ▸ PMSM Drive ▸ Heat Exch. Characteristic ▸
4
A
B
A
M
Phi
Mass Flow Rate &
Thermal Flux Sensor
(TL)
2
B
Mass Flow
Q
Temp Inlet
Temp Motor
Characteristic
TIn
6
5
Tmotor
T
B
A
Temperature
Sensor
3
Hp
B
S
A
Heat Flow Source
1
HMotor
Ready
140%
FixedStepAuto


elec_auto_ev - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools
elec_auto_ev
Inputs
Inclinometer
Vehicle Control
Battery
DC-DC Converter
Convection Battery-Env
Ambient Temperature
Solver Configuration
Local Solver: on
Cooling System
Outlet
Env
Inlet
trqRef
PMSM D
Ready
100%
elec_auto_ev/Cooling System * - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
Cooling System
elec_auto_ev
Env
Radiator
Heat Exch. Characteristic
Hose Radiator
Pump
Outlet
Hose Outlet
Thermal Liquid Settings (TL)
Expansion Tank
Inlet
Hose Inlet
Ready
100%
FixedStepAuto


Vehicle Speed
Speed (MPH)
40
30
20
10
0
0
20
40
60
80
100
120
140
160
180
200
Cruise Control Setpoint
Actual Speed
Driver Inputs and Environment
Throttle, Cruise, Brake
On
Off
Elevation (m)
300
200
100
0
Throttle
Cruise Enable
Brake Pedal
Elevation
0
20
40
60
80
100
120
140
160
180
200
Time (s)


Motor Torque
500
0
-500
Nm
0
20
40
60
80
100
120
140
160
180
200
Motor Temperature
50
40
30
20
0
20
40
60
80
100
120
140
160
180
200
Time (s)

Building the right model for the task at hand can be challenging

Requirements not understood by project management

Identification of required modelling detail

Limited time and nothing to build on – starting from scratch

No data

Lacking domain knowledge

Building the right model for the task at hand can be challenging

Requirements not understood by project management

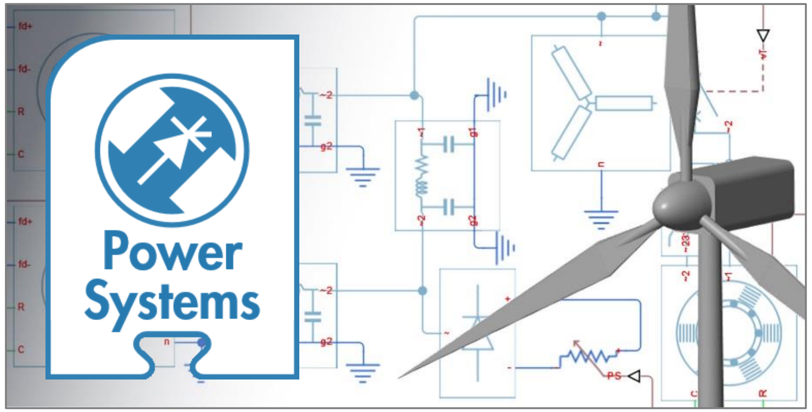
Identification of required modelling detail

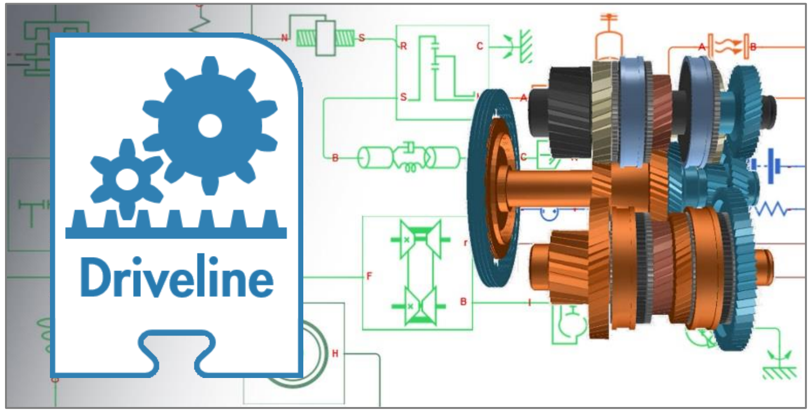
Limited time and nothing to build on – starting from scratch

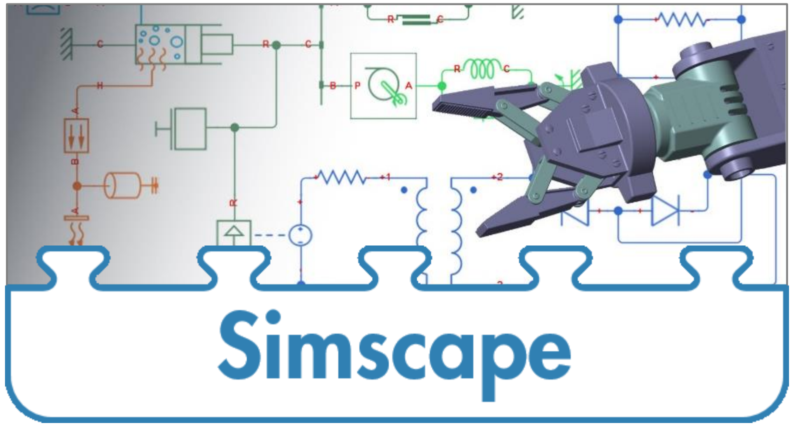
No data

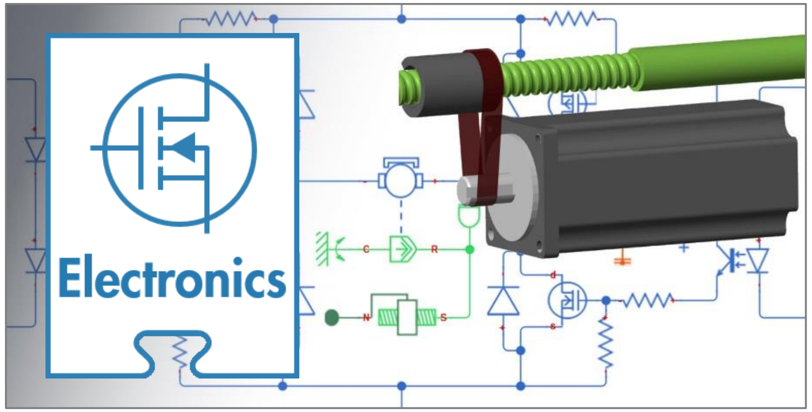
Lacking domain knowledge

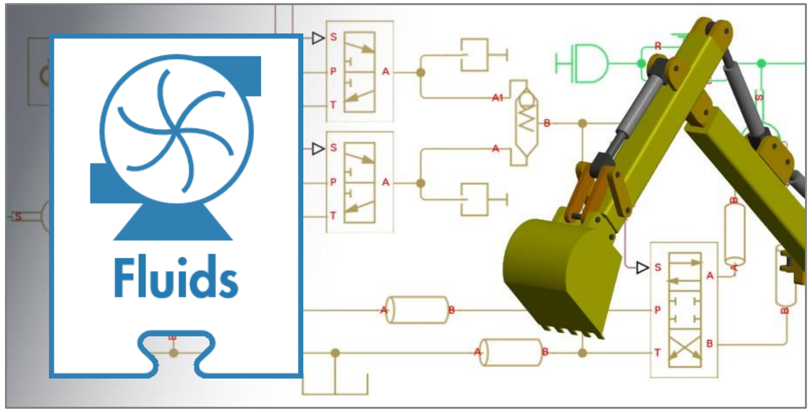
Simscape libraries enable you to build representative models fast

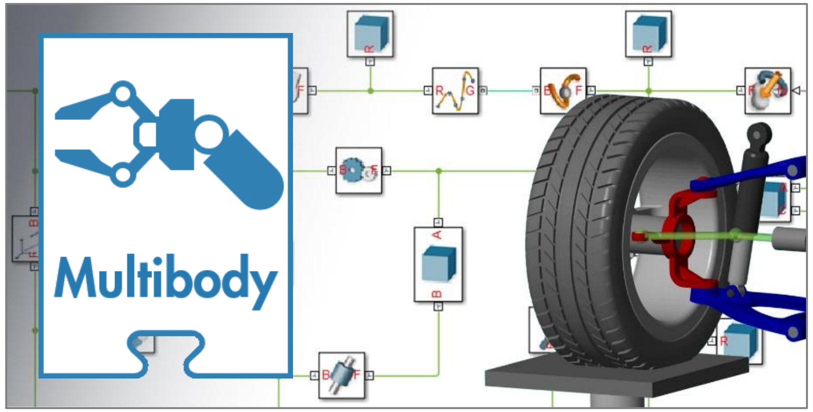










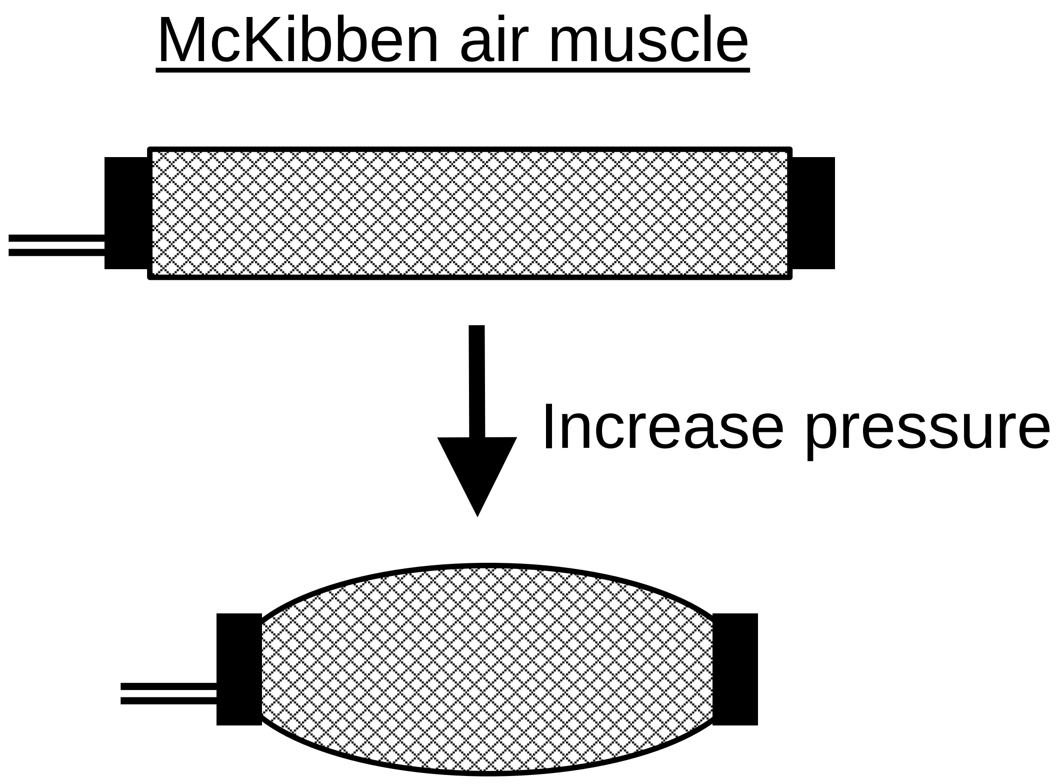


Creating custom Simscape components

Example: McKibben air muscle

Steps:

1. Write out defining equations
2. Find starting point in Simscape foundation library
3. Incrementally add functionality, testing as you go



Creating custom Simscape components

Step 1: Write out equations

L_u = Un-stretched length

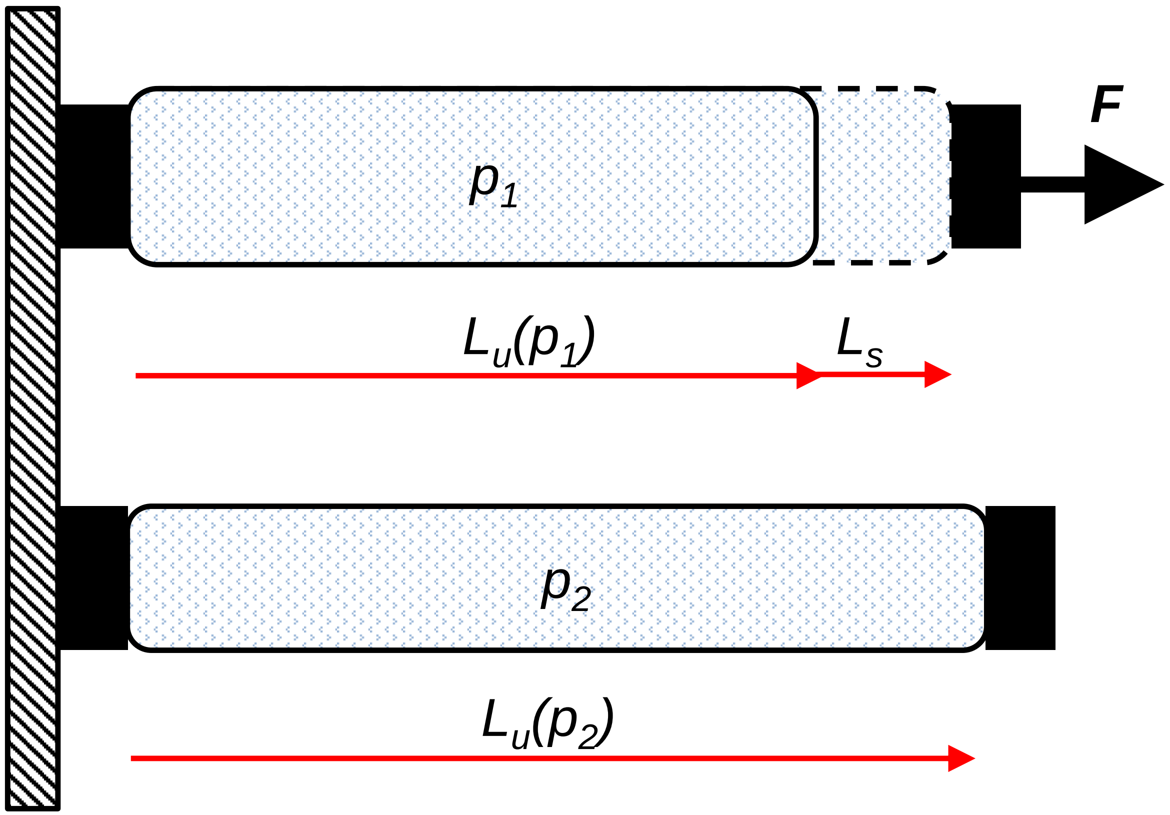
L_s = Additional stretch due to force, F

Assumptions:

- Volume is approximately constant
- Stretch force is proportional to L_s

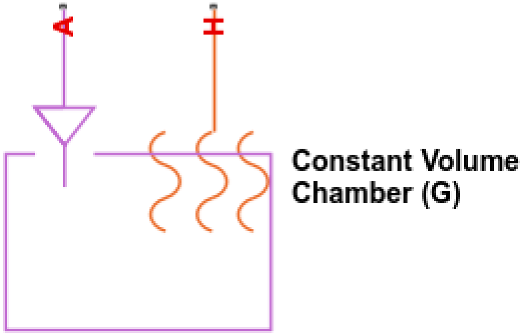
Equations:

- $L = L_u(p) + L_s$
- $F = k \times L_s$
- $pV = nRT$

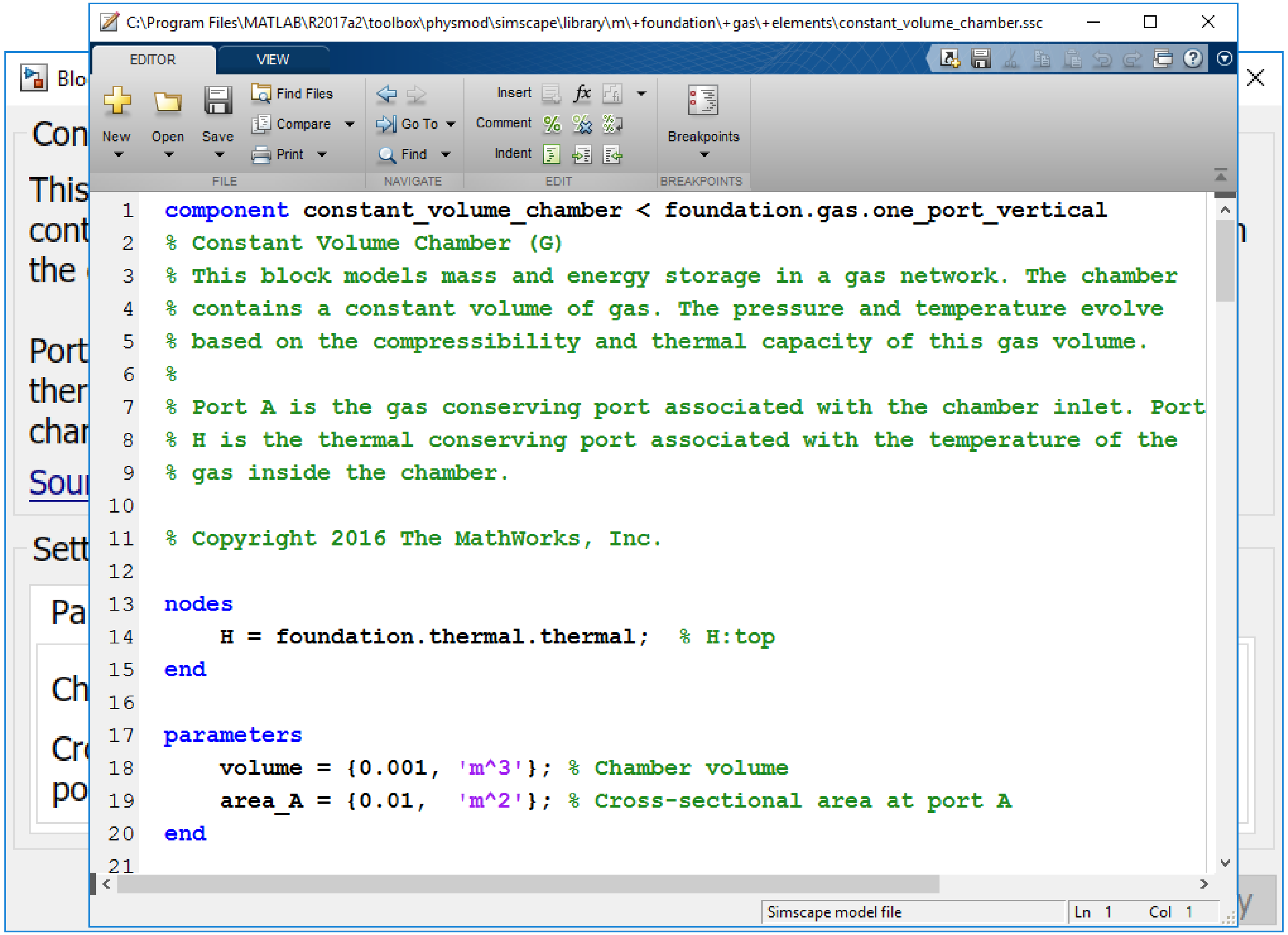


Creating custom Simscape components

Step 2: Find starting point from foundation library



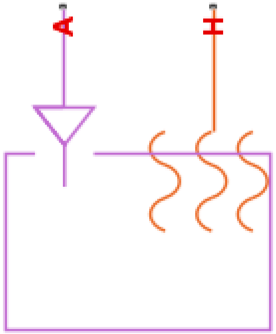
- Has equation of state
- Need to add mechanical ports & equations




```
component constant_volume_chamber < foundation.gas.one_port_vertical
% Constant Volume Chamber (G)
% This block models mass and energy storage in a gas network. The chamber
% contains a constant volume of gas. The pressure and temperature evolve
% based on the compressibility and thermal capacity of this gas volume.
%
% Port A is the gas conserving port associated with the chamber inlet. Port
% H is the thermal conserving port associated with the temperature of the
% gas inside the chamber.

% Copyright 2016 The MathWorks, Inc.

nodes
    H = foundation.thermal.thermal;   % H:top
end

parameters
    volume = {0.001, 'm^3'}; % Chamber volume
    area_A = {0.01,  'm^2'}; % Cross-sectional area at port A
end
```

Creating custom Simscape components

Step 3: Incrementally add functionality



Constant Volume Chamber (G)

```
component air_muscle < foundation.gas.one_port_vertical
% Air Muscle (G)
% This block models a McKibben air muscle.

% Copyright 2016-2017 The MathWorks, Inc.

nodes
    H = foundation.thermal.thermal; % H:top
    R = foundation.mechanical.translational.translational; % R:bottom
    C = foundation.mechanical.translational.translational; % C:top
end
```

Add:

- Two mechanical ports
- Two additional new equations

$$L = L_u(p) + L_s$$

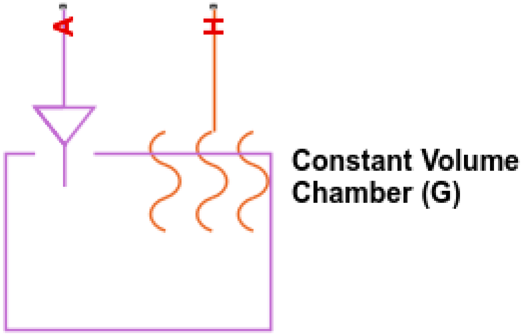
$$F = k \times L_s$$

```
L     == Ls + Lu;
force == K * Ls;
```

```
Lu = tablelookup(pVec,LuVec,p_chamber,
```

Creating custom Simscape components

Step 3: Incrementally add functionality




```
variables
    % Mechanical variables
    force = {0, 'N'}; % Force
    Ls = {0, 'm'};    % Stretch
end
```

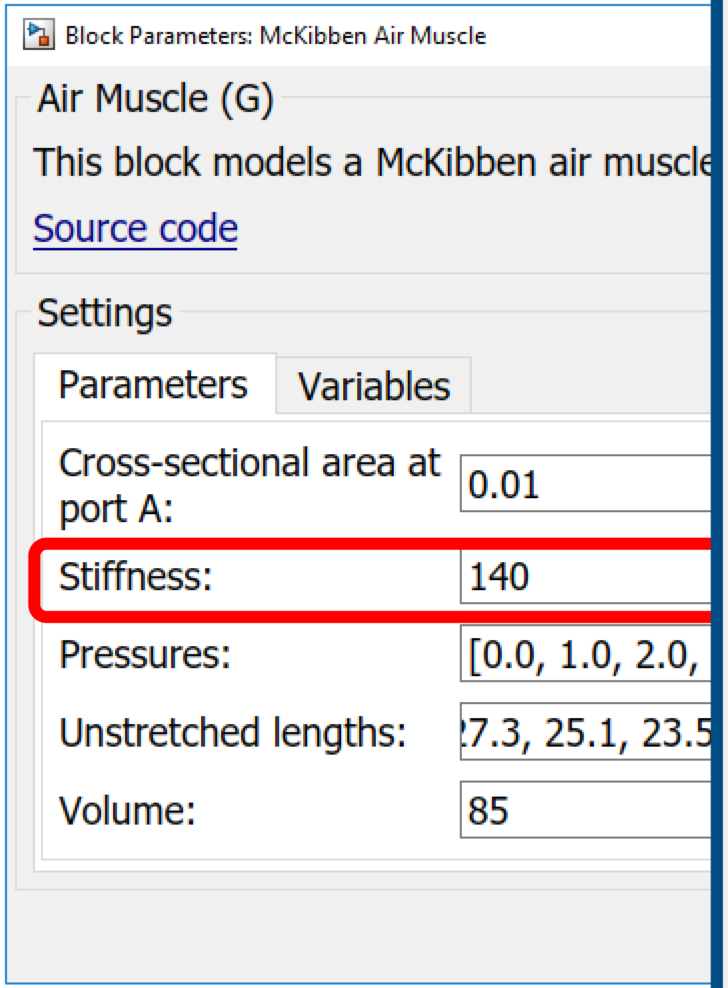
Add definitions for:

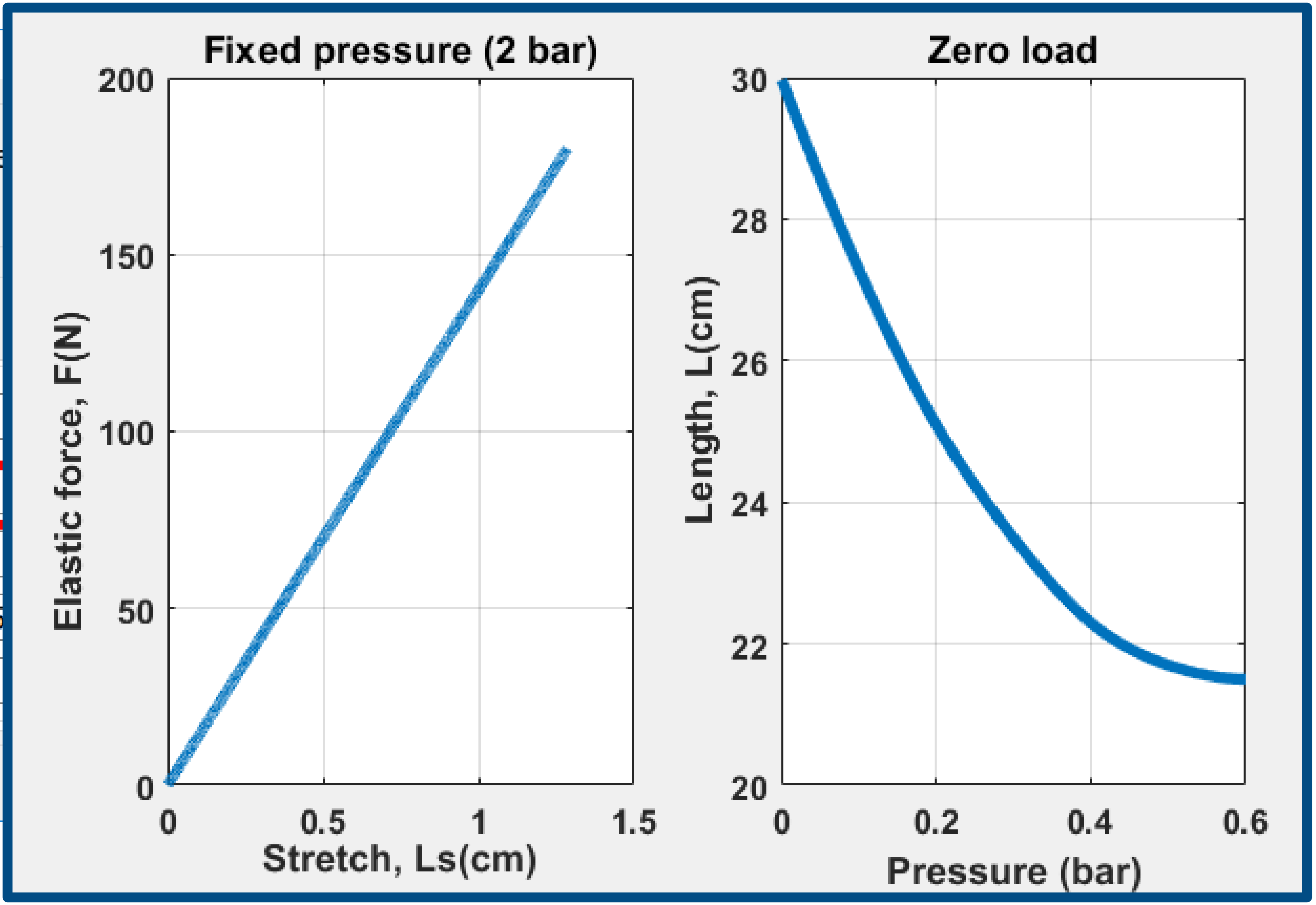
- Variables
- Parameters

```
parameters
    K = {140,'N/cm'}; % Stiffness
    pVec = {[0 1 2 3 4 5 6] ,'bar'}; %
    LuVec  = {[30 27.3 25.1 23.5 22.3
end
```

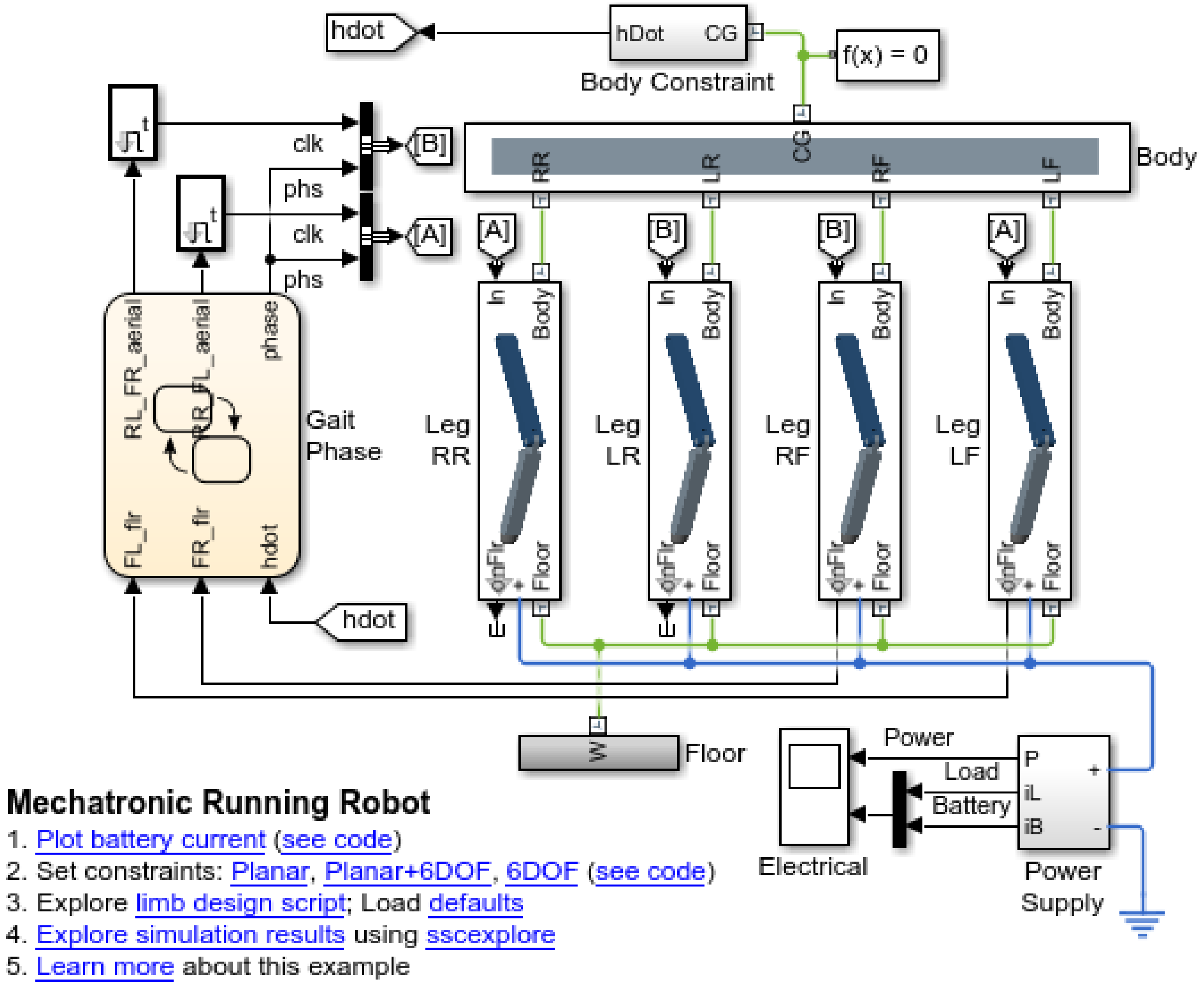
Creating custom Simscape components

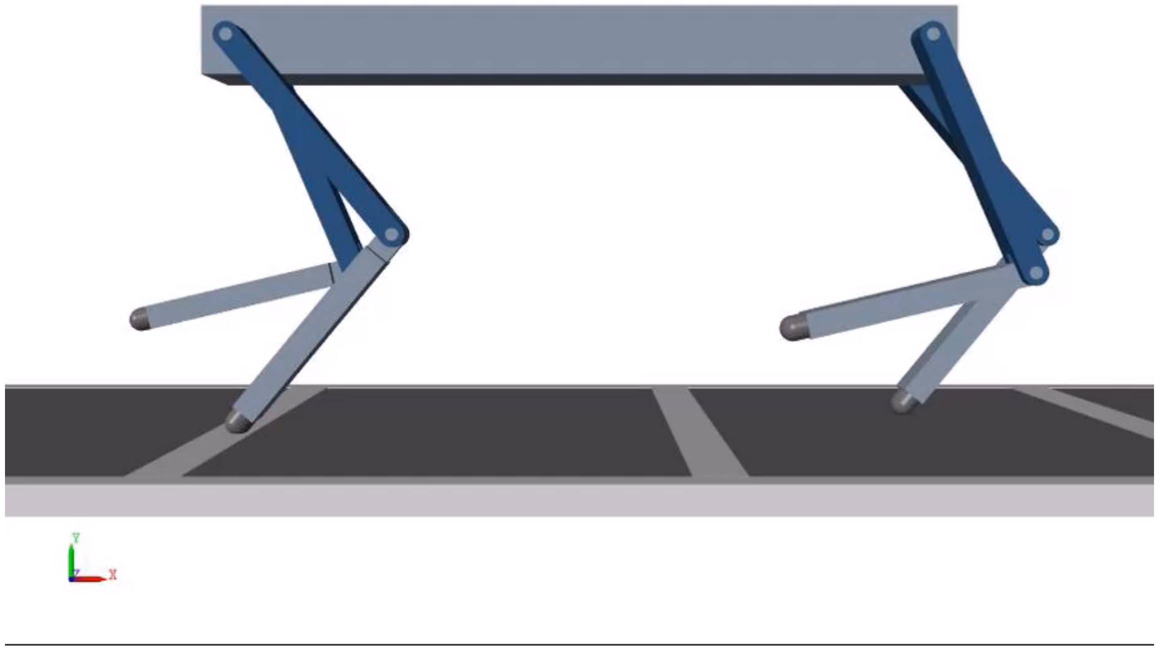
Step 4: Build library and run test model





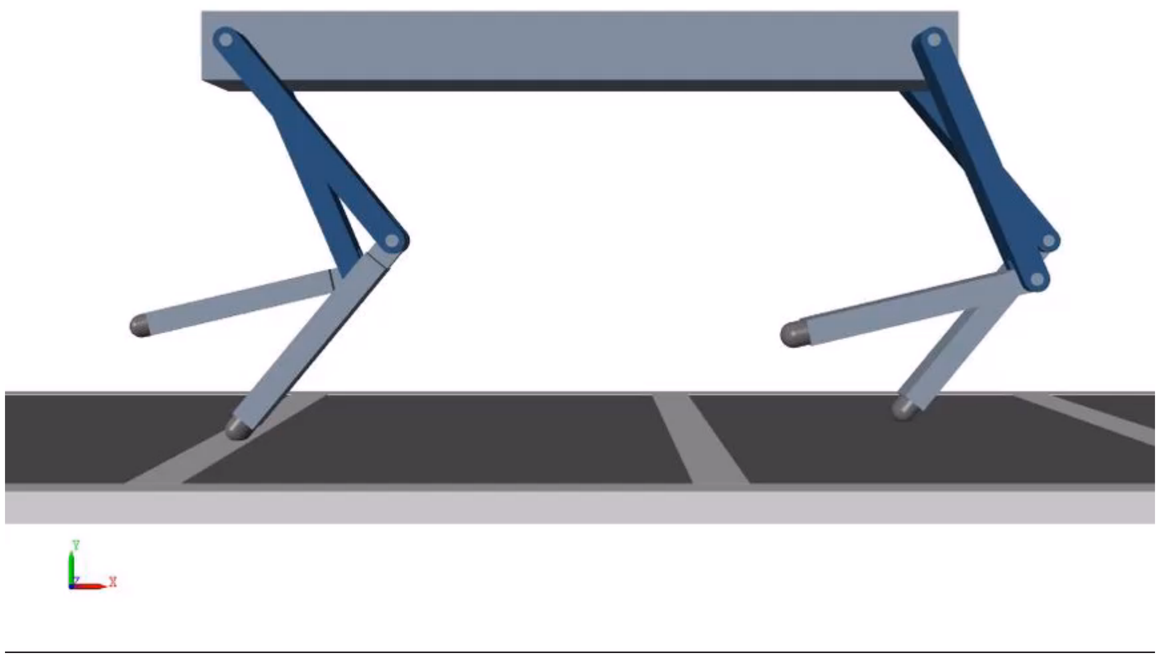
Why use Simscape?

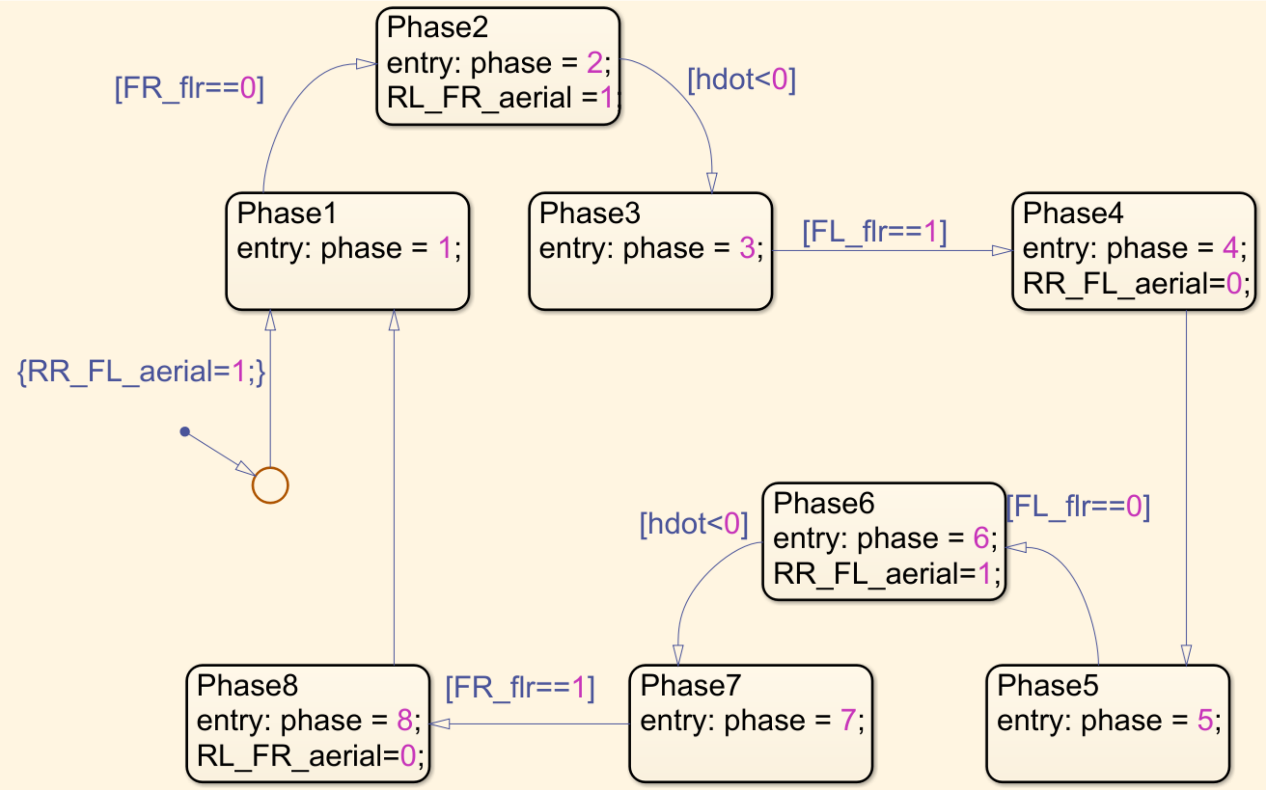




Why use Simscape?

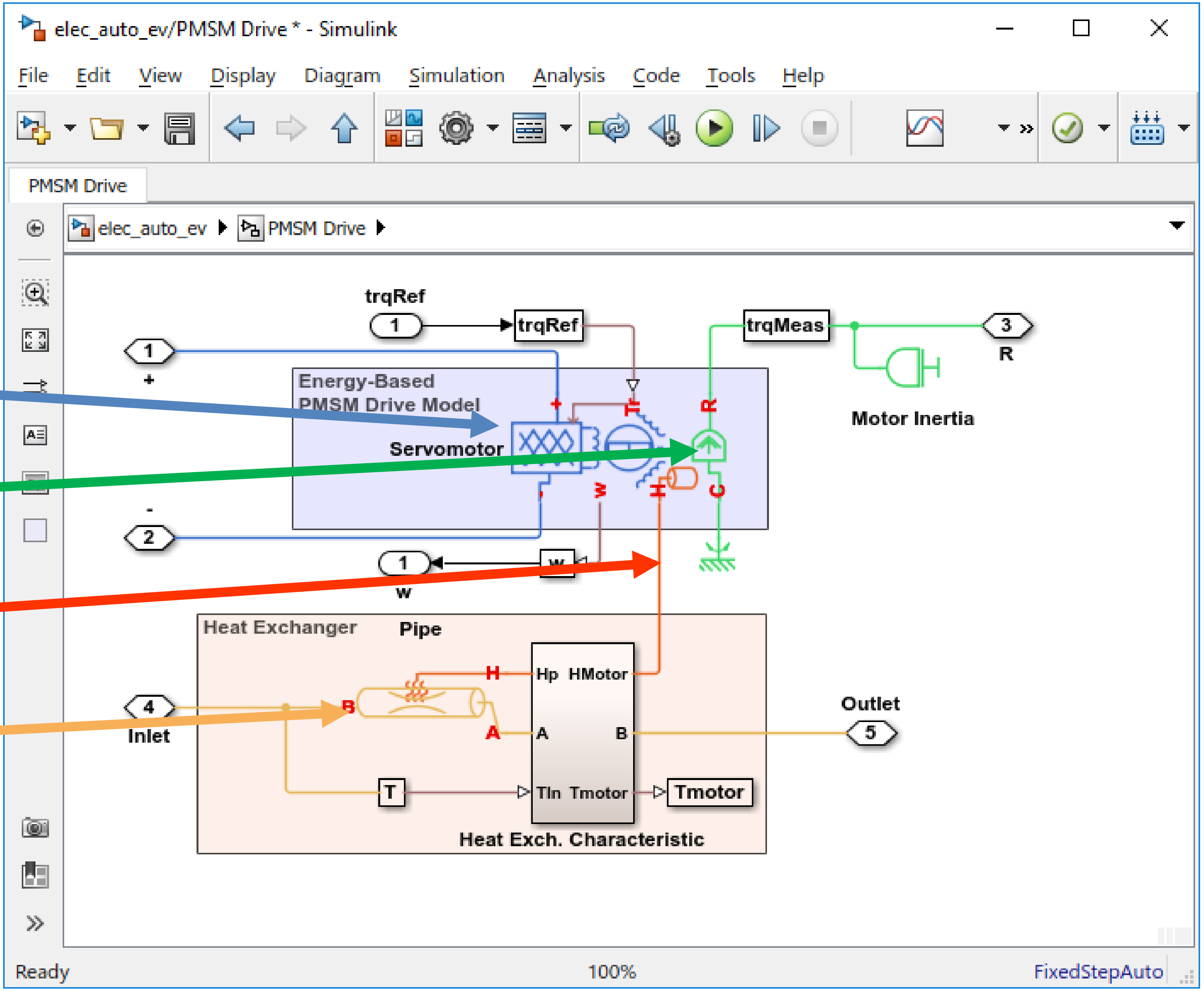
- **Plant and control**





Why use Simscape?

- Plant and control
- **Multidomain**
  - **Electrical**
  - **Mechanical**
  - **Thermal**
  - **Fluid**

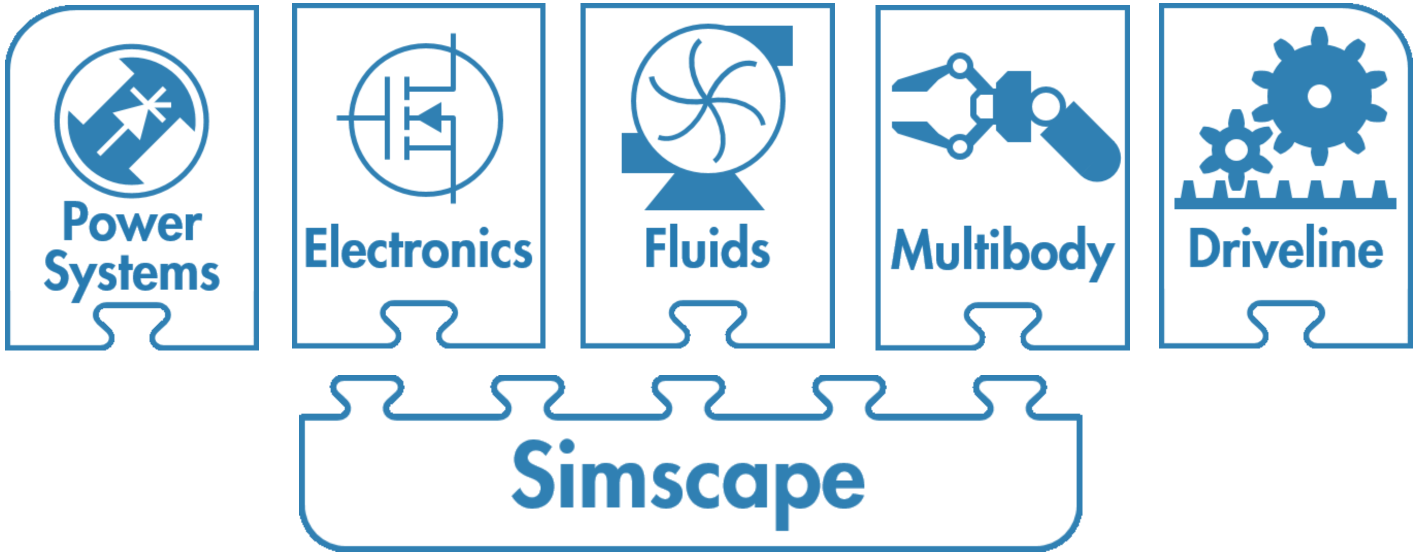


Why use Simscape?

- Plant and control
- Multidomain
- **Code generation and V&V tools**
  - **Test controller on HIL plant**
  - **Deploy to simulator**
  - **Use plant model in real-time controller**

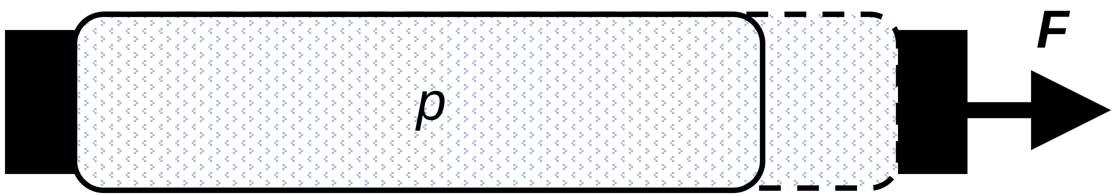
Why use Simscape?

- Plant and control
- Multidomain
- Code generation and V&V tools
- **Libraries, examples, documentation & webinars**



Why use Simscape?

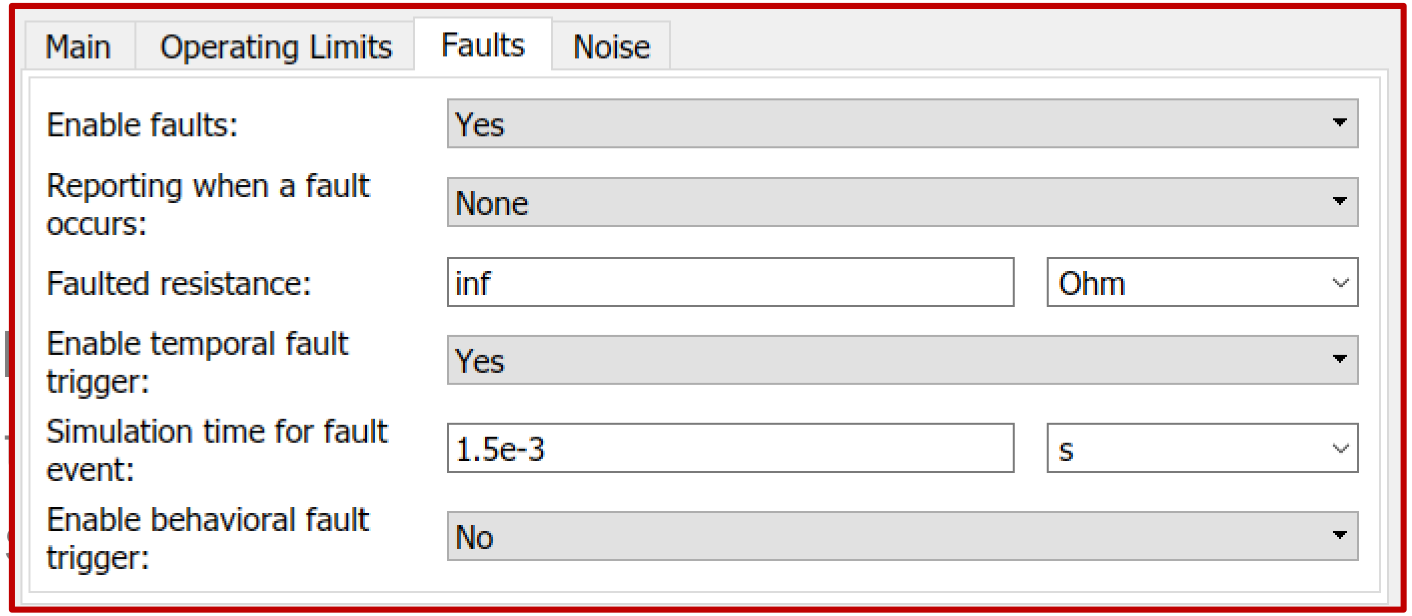
- Plant and control
- Multidomain
- Code generation and V&V tools
- Libraries, examples, documentation & webinars
- **Simscape language – build custom components**




```
parameters
    K = {140,'N/cm'}; % Stiffness
    pVec = {[0 1 2 3 4 5 6] ,'bar'}; % Pressures
    LuVec  = {[30 27.3 25.1 23.5 22.3 21.7 21.5],'cm'};
end
```

Why use Simscape?

- Plant and control
- Multidomain
- Code generation and V&V tool
- Libraries, examples, document
- Simscape language – build cu
- **Workflow**
  - **Tight integration with MathWorks control and optimization tools**
  - **MATLAB for scripting and automation**
  - **Fault-capable components (R, L, C, Servomotor, …)**

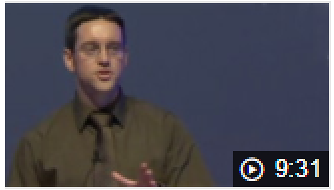


Why use Simscape?

- Plant and control
- Multidomain
- Code generation and V&V tools
- Libraries, examples, documentation & webinars
- Simscape language – build custom components
- Workflow
  - Tight integration with MathWorks control and optimization tools
  - MATLAB for scripting and automation
  - Fault-capable components (R, L, C, Servomotor, …)
- **Support, training, consulting**
- **MATLAB Central**

How to find out more

- MathWorks physical modelling page:
  – https://www.mathworks.com/solutions/physical-modeling.html
- Steve Miller's introduction video
  – https://www.mathworks.com/videos/physical-modeling-introduction-75883.html



- MATLAB Central File Exchange
  – https://www.mathworks.com/matlabcentral/fileexchange/
- Contact us direct