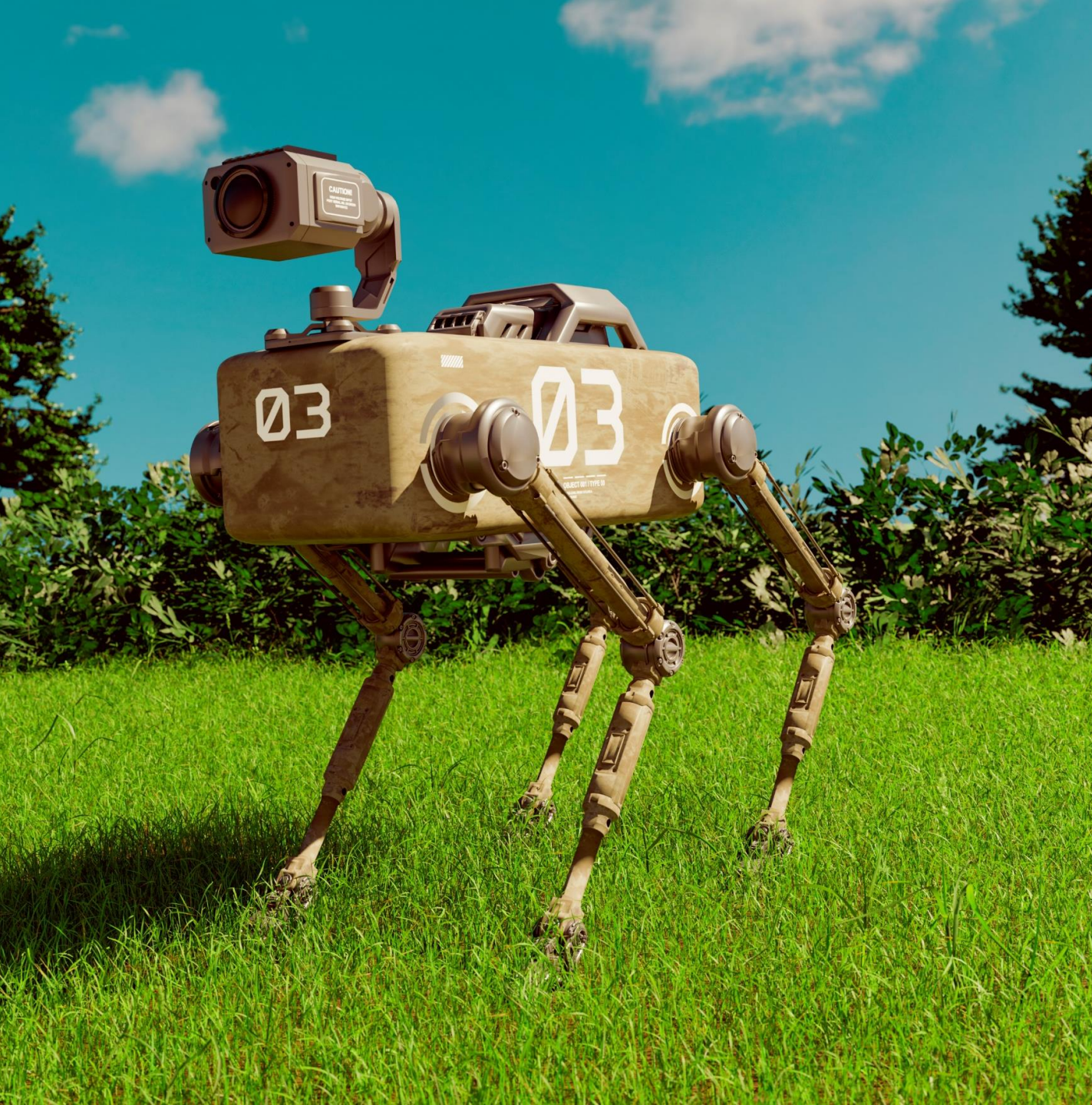


# MATLAB EXPO

Simscape를 이용한 멀티도메인 물리 모델링

*이종일 부장, 매스웍스코리아*





Can you analyze how the system behaves before building any physical prototypes?

How do you model the electromechanical and multibody components?

# How to analyze the system behaves before building any physical prototypes?

Solution :

- Simscape & Simscape Multibody

Benefit :

- Perform multiple systems such as mechanical, electrical system
- Visualize to understand system behavior
- Evaluate performance criteria to decide system requirements

# How to analyze the system behaves before building any physical prototypes?

The image displays two windows from the MATLAB Simulink environment. The left window, titled 'Mechanics Explorer-sm\_robot\_run\_4legs', shows a 3D CAD model of a four-legged robot on a flat surface. The right window, titled 'sm\_robot\_run\_4legs - Simulink', shows the control system block diagram.

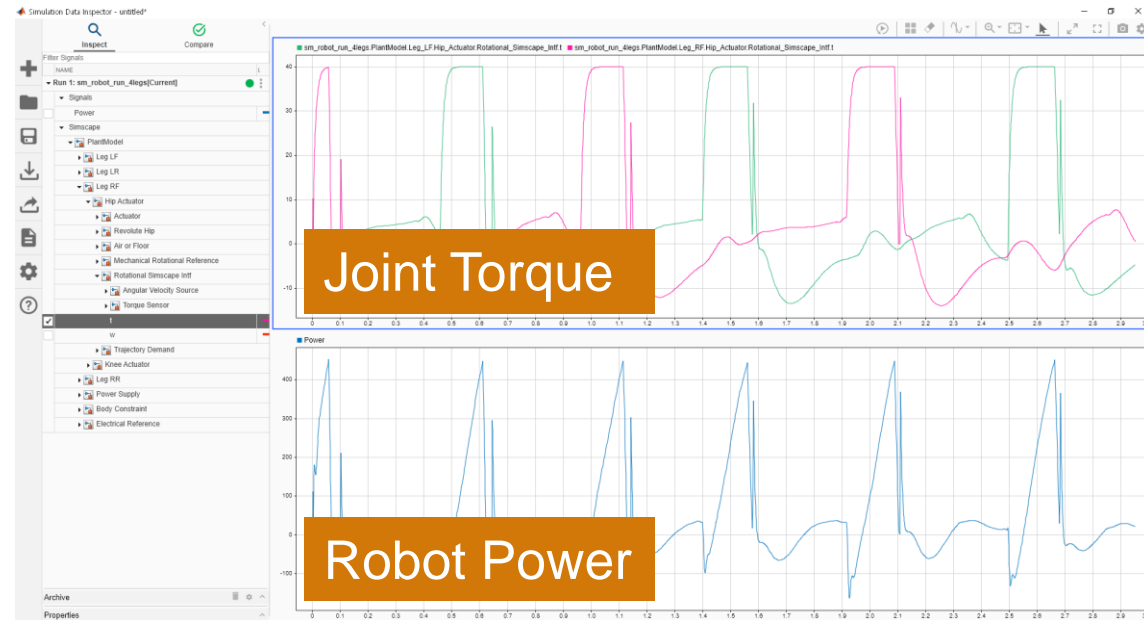
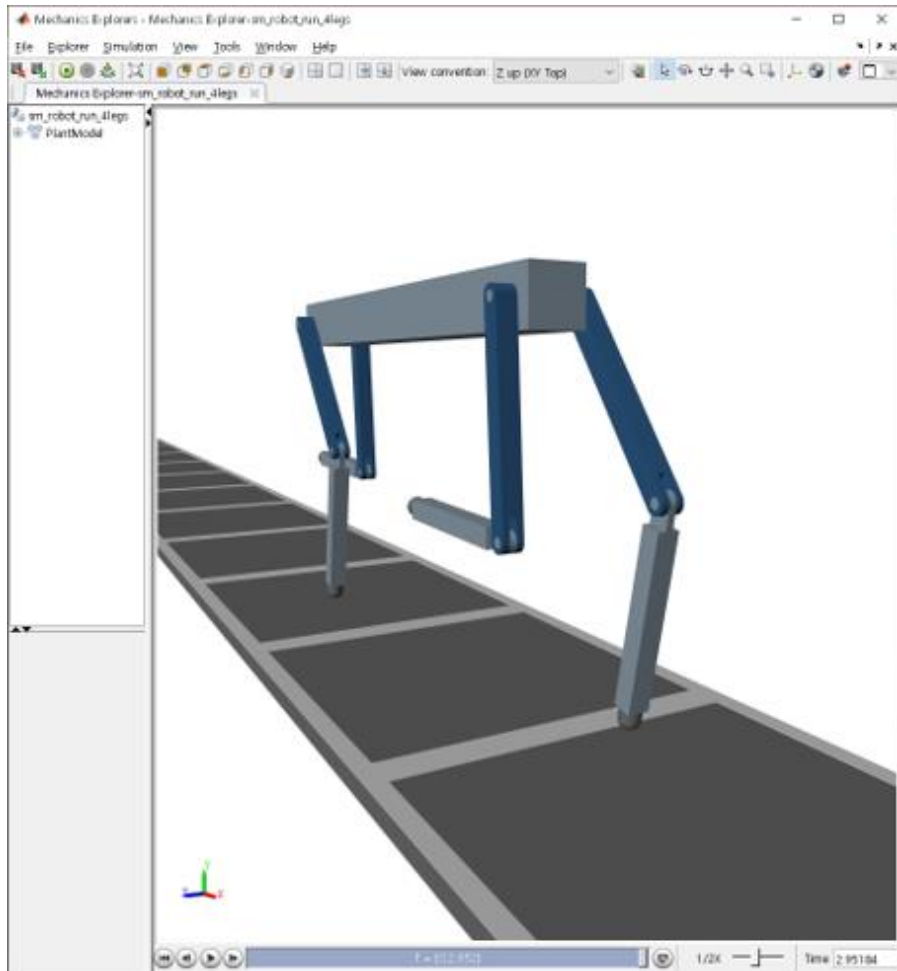
The block diagram includes the following components and connections:

- Body Constraint:** A block labeled 'Body Constraint' with input  $f(x) = 0$  and output  $hDot$  CG.
- Body:** A block representing the robot's body with inputs for phase and position, and outputs for center of gravity (CG) and height  $h$ .
- Gait Phase:** A block that generates phase signals for the legs based on a clock ( $clk$ ) and phase ( $phs$ ) inputs.
- Legs (RR, LR, RF, LF):** Four leg blocks, each receiving phase and position signals and outputting joint angles and forces.
- Electrical:** A power supply block connected to a battery and a load, providing power to the robot's system.

**Mechatronic Running Robot**

1. [Plot battery current](#) (see code)
2. Set constraints: [Planar](#), [Planar+6DOF](#), [6DOF](#) (see code)
3. Explore [limb design script](#); Load [defaults](#)
4. [Explore simulation results](#) using [ssxexplore](#)
5. [Learn more](#) about this example

# How to analyze the system behaves before building any physical prototypes?



- **Confirm system requirements of performance**
- **Check design requirements of motor, gear, battery**
- **Evaluate new control algorithm and calibration data**

---

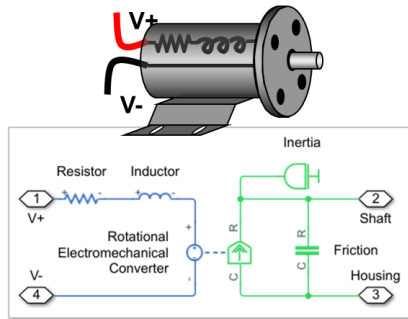
# Agenda

- **Simscape Overview**
- **Demos**
- **Applications**
- **Key Takeaways**

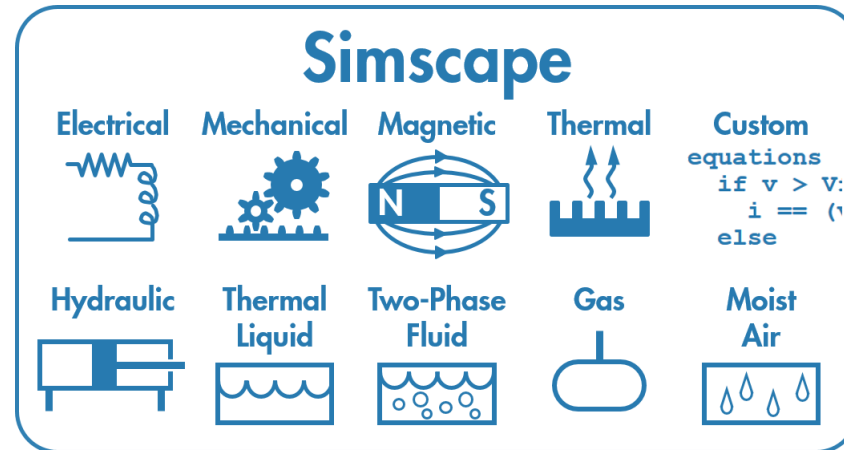
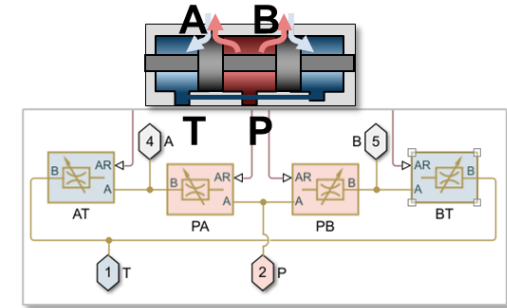
# Simscape Overview

- Multidomain physical systems within the Simulink environment

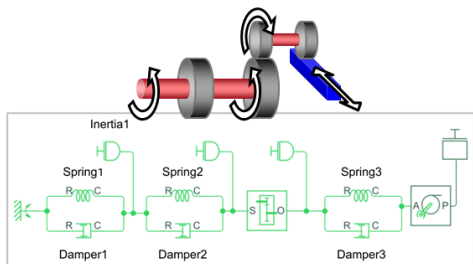
Electro-mechanical



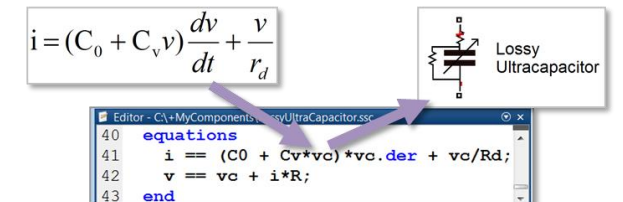
Hydraulic



Mechanical (translation and rotation)



Custom



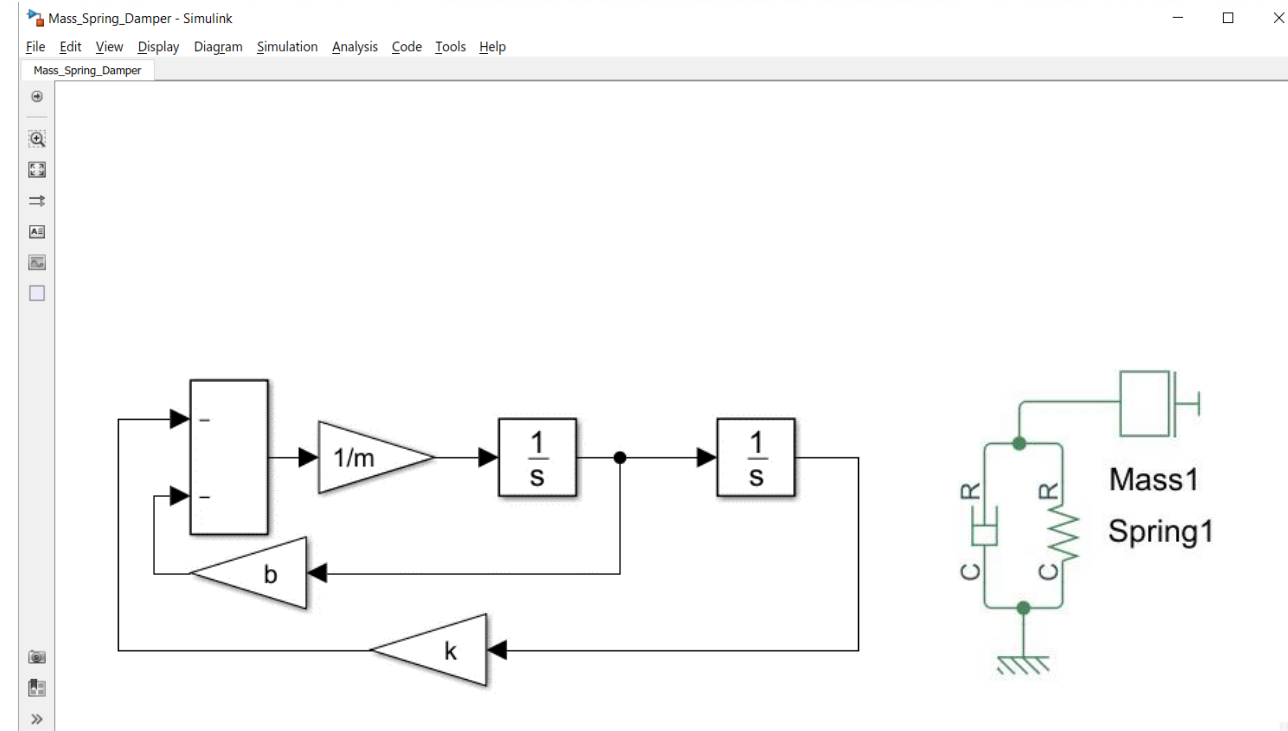
# Simscape Overview

- Equations derived automatically
  
- Assemble schematics that simulate
  
- With Simscape you can:
  - Test without hardware prototypes
  - Optimize system-level performance
  - Design control systems and logic
  - Refine requirements for system

$$F_{\text{Spring}} = k_{\text{Spring}} * (x_{\text{Mass}})$$

$$F_{\text{Damper}} = b_{\text{Damper}} * \left(\frac{dx_{\text{Mass}}}{dt}\right)$$

$$\frac{d^2 x_{\text{Mass}}}{dt^2} = \frac{-F_{\text{Spring}} - F_{\text{Damper}}}{m_{\text{Mass}}}$$

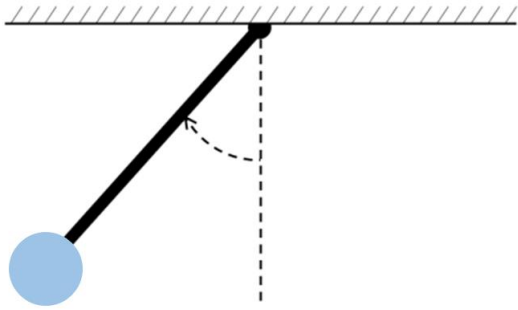


Input/Output Block Diagram

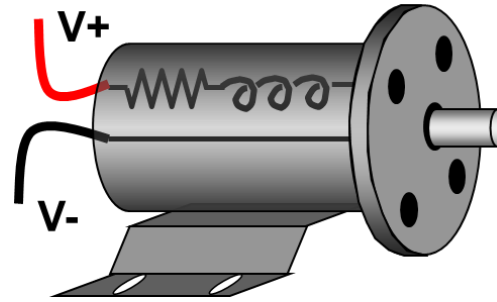
Simscape



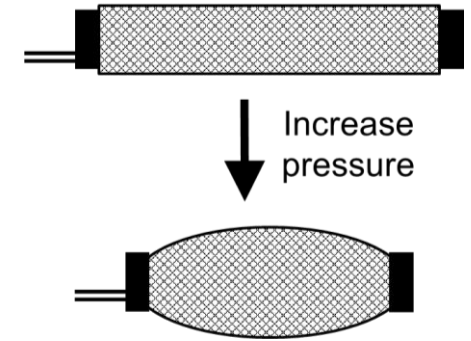
# How to build walking robot?



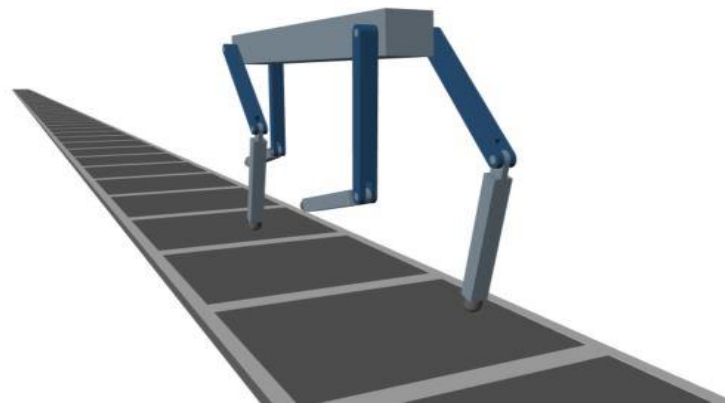
**Pendulum**



**Motor**

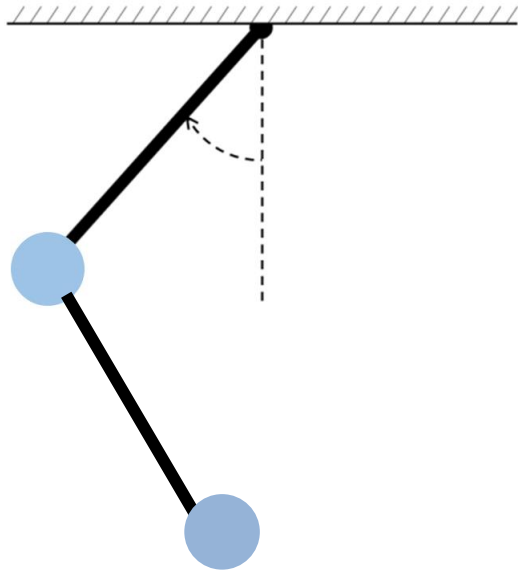


**McKibben air muscle**

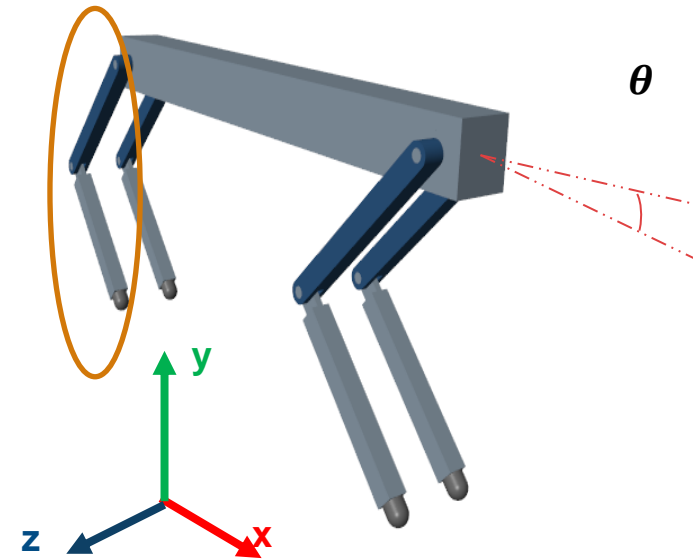


**Walking Robot**

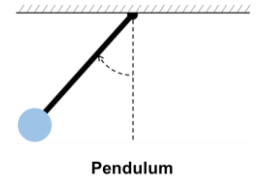
# Demos – Design Inverted Pendulum



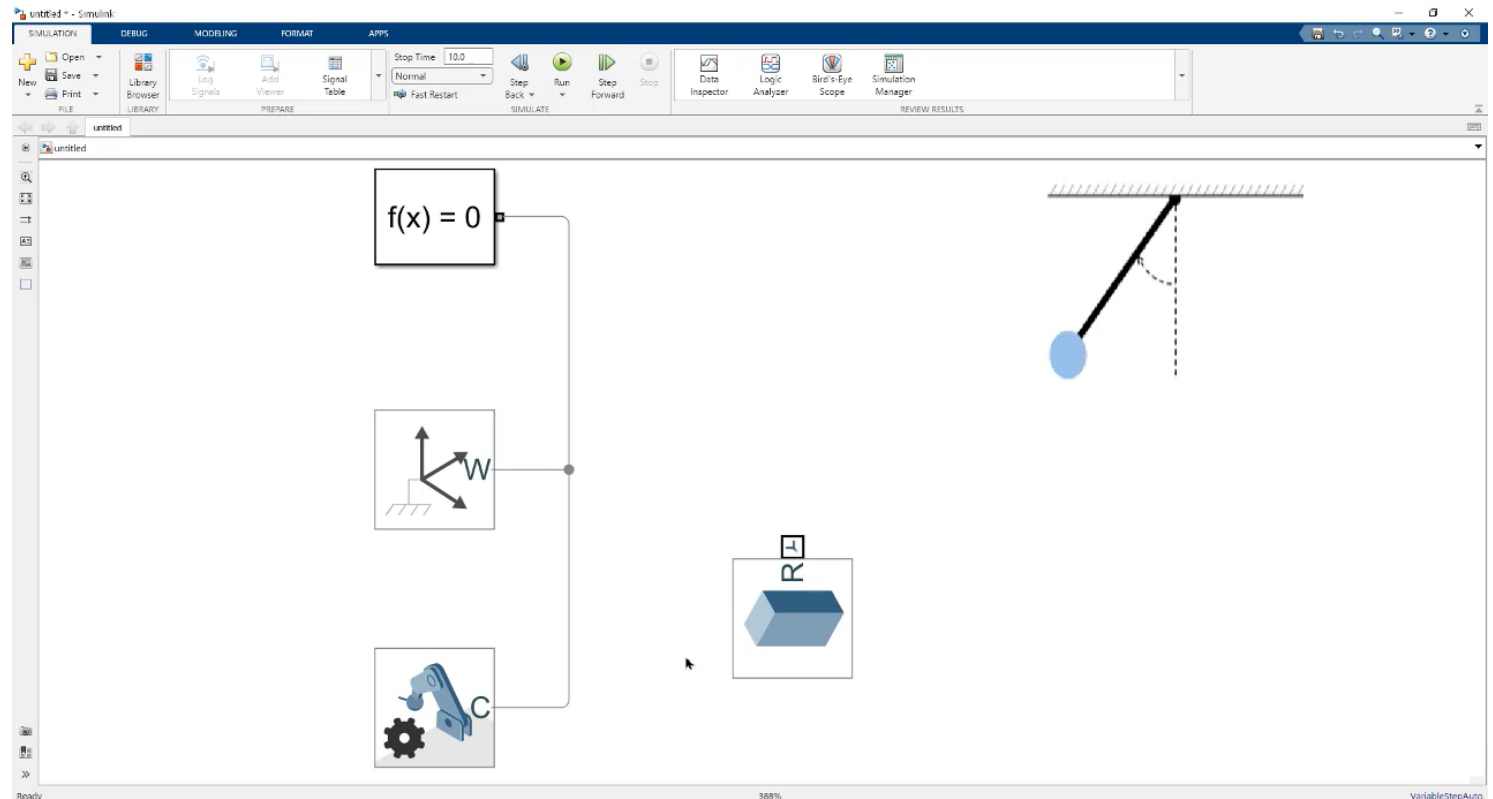
**Double Pendulum**



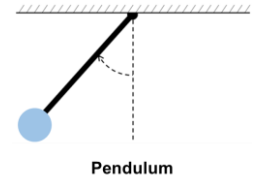
# Demos – Design Inverted Pendulum



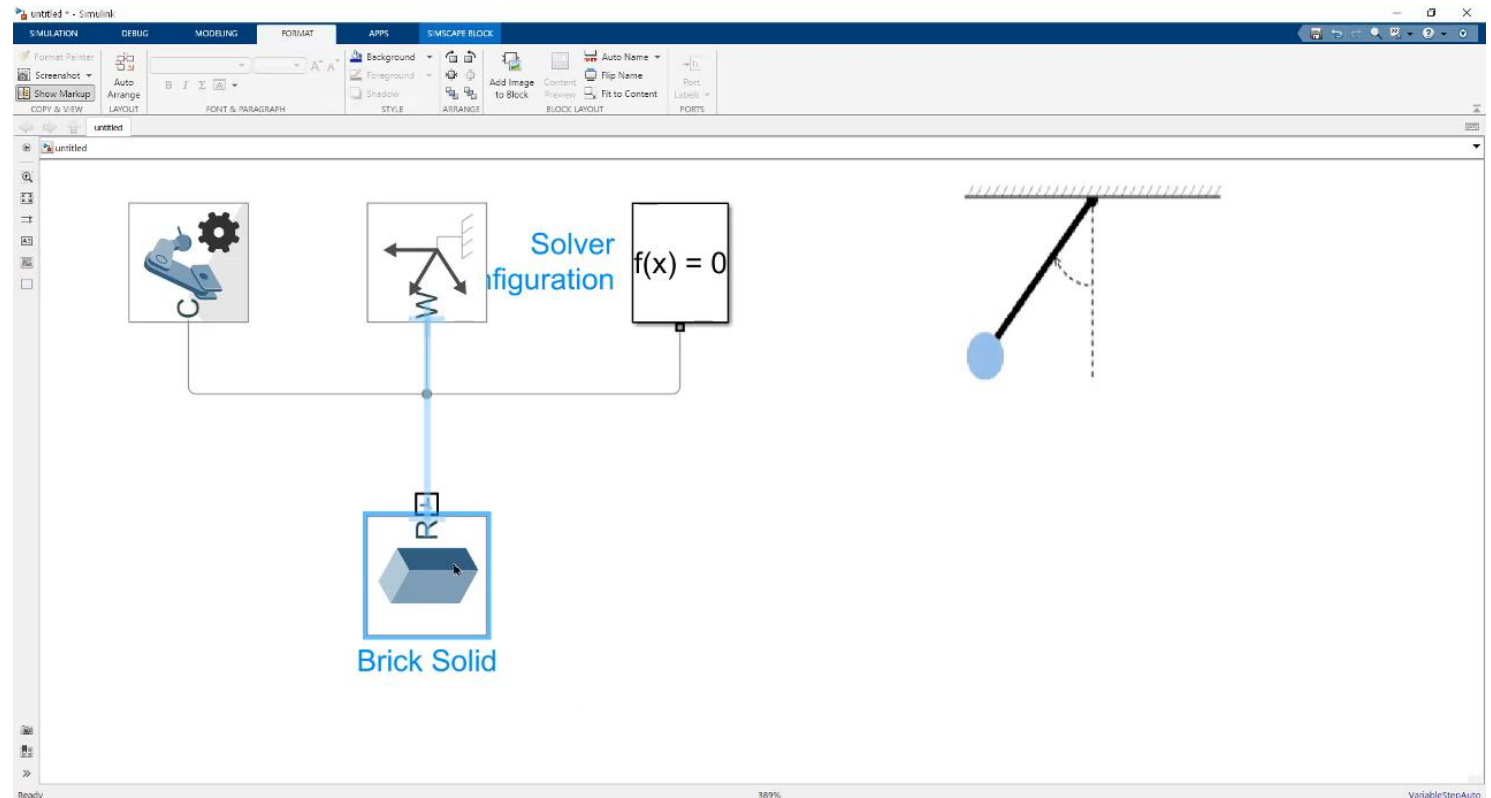
- Start from Simulink
  - Use Simulink in toolstrip
  - Command as “smnew”
  
- Clean up canvas
  - Delete unused block
  - Delete ref. descriptions
  
- Insert block to build model
  - Use ref. image for design
  - Align blocks from format tab



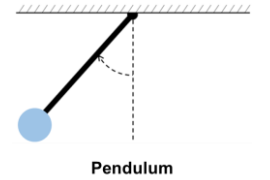
# Demos – Design Inverted Pendulum



- Parameterize blocks
  - Update dimensions  
[1 1 1] → [1 0.1 0.1]
  
- Update models
  - Refresh model data (F5)
  - Change View Convention

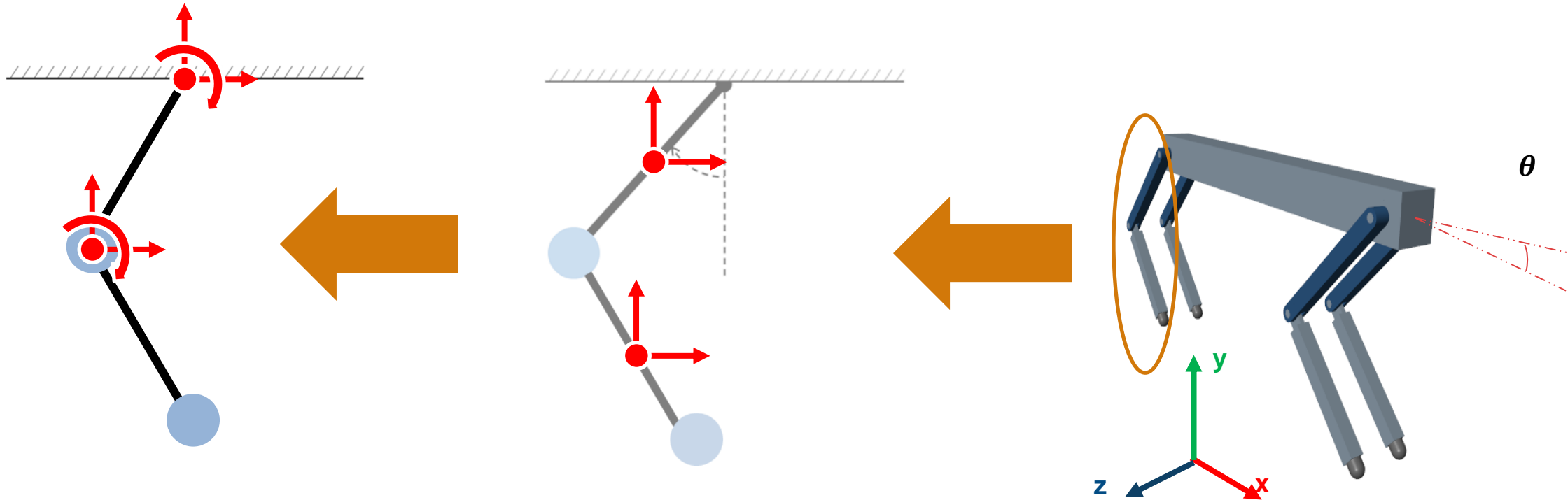


# Demos – Design Inverted Pendulum

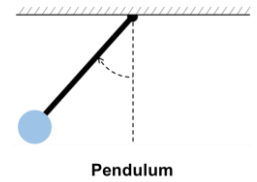


## Define Coordinate Frames

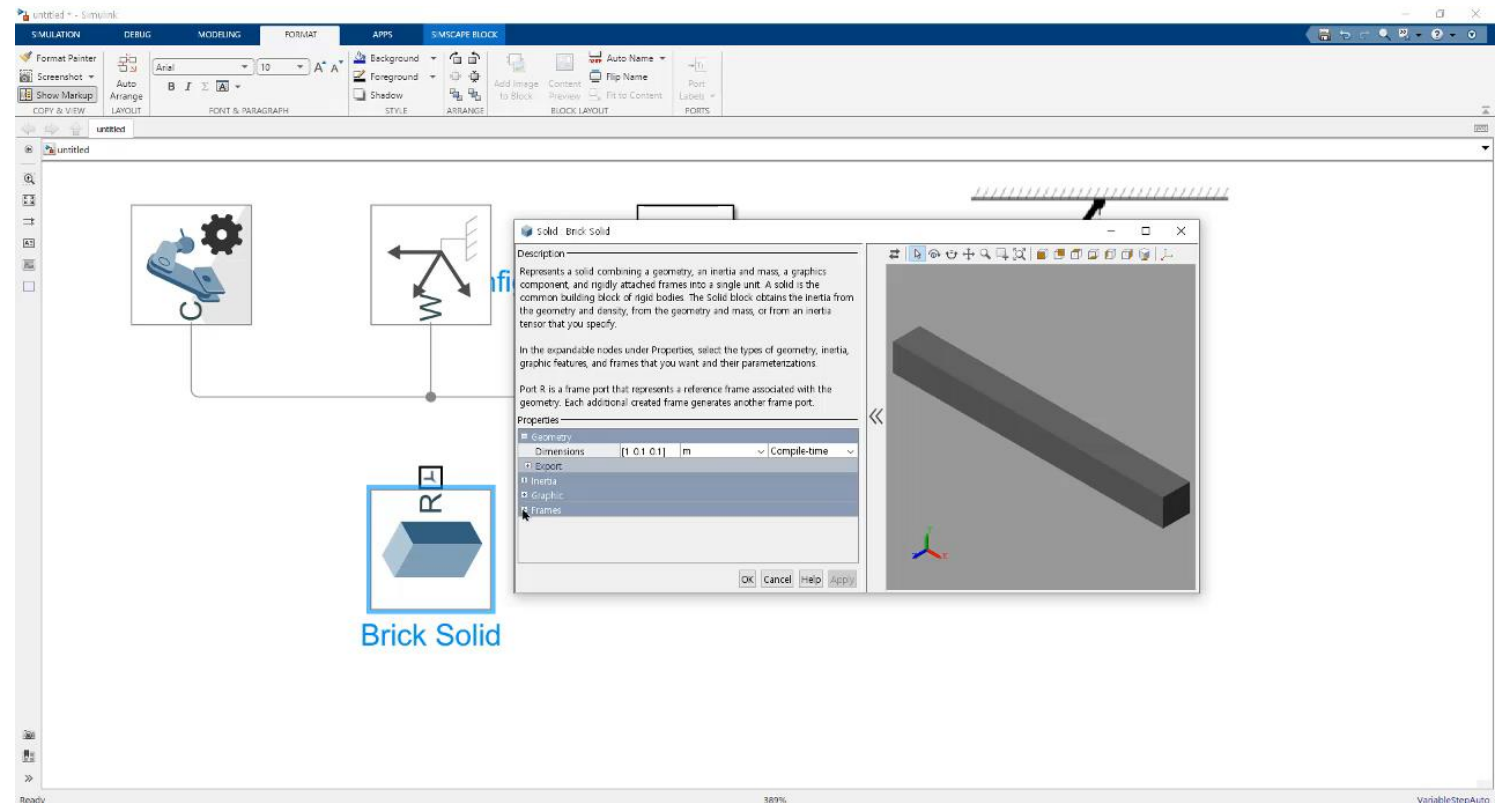
- Add new coordinate frame to create joint at certain features (faces, edges, ETC)



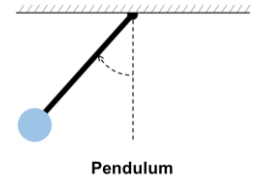
# Demos – Design Inverted Pendulum



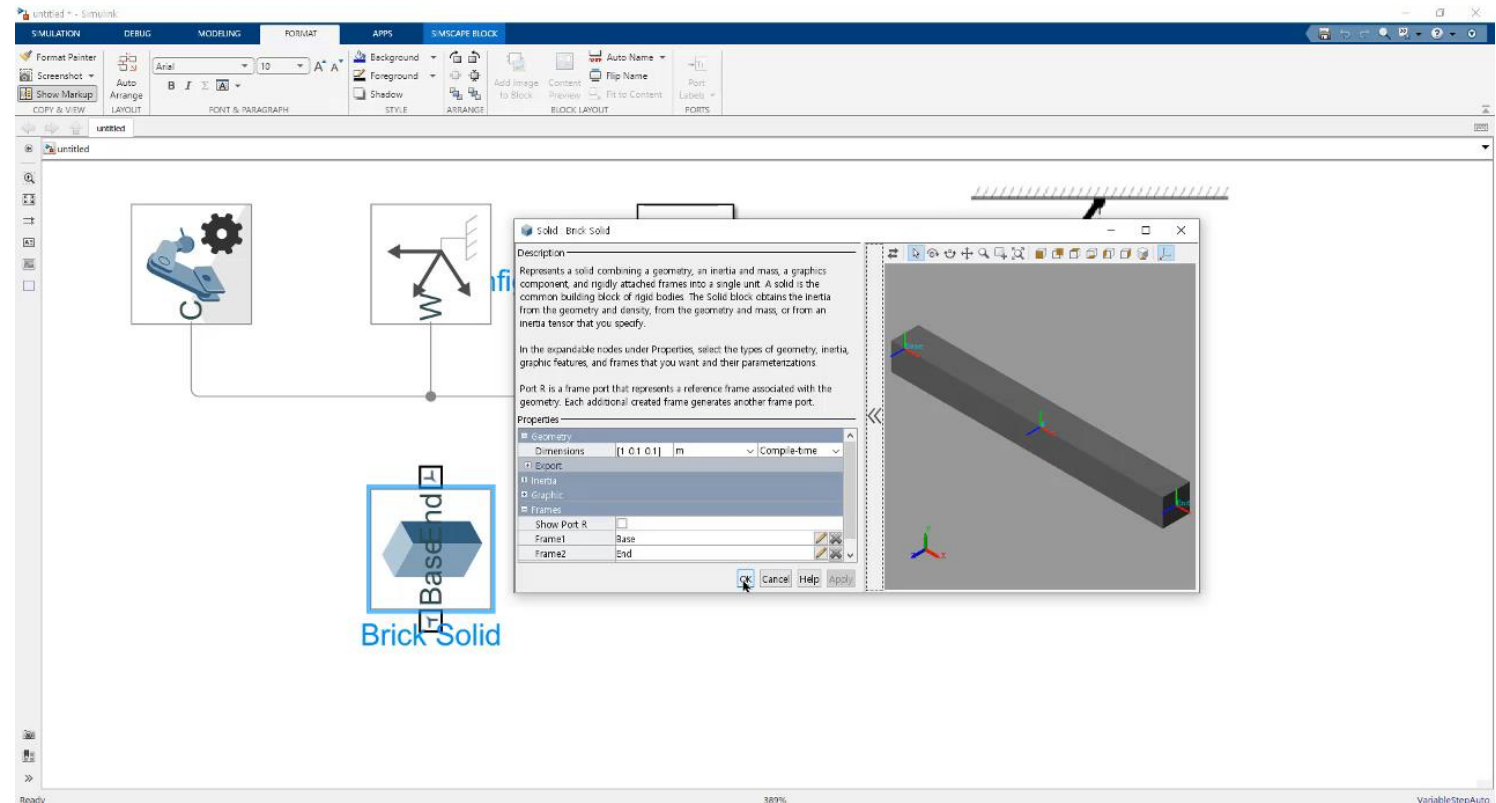
- Add new frame to link joint
  - Select surface to add new frame using “Based on geometric feature”
  - Add two new frame at the end of block
  - Rename of new frame as “Base” and “End”
  
- Update models
  - Refresh model data (F5)



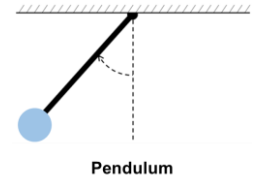
# Demos – Design Inverted Pendulum



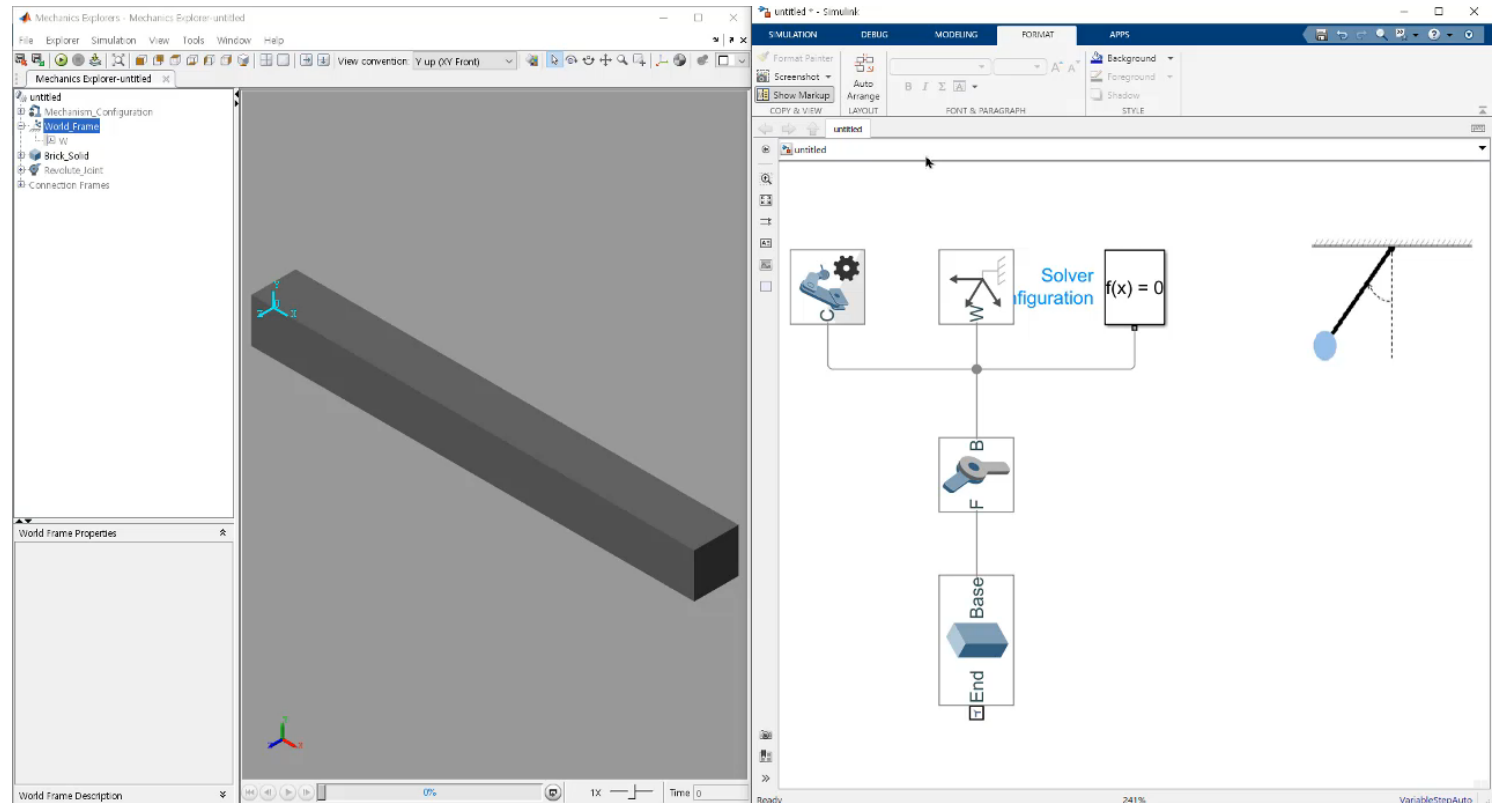
- Connect blocks with joint
  - Resize block to connect model
  - Insert “Revolute Joint” and connect block between Brick Solid block and World Coordinate
  
- Update models
  - Refresh model data (F5)
  - Change view convention



# Demos – Design Inverted Pendulum

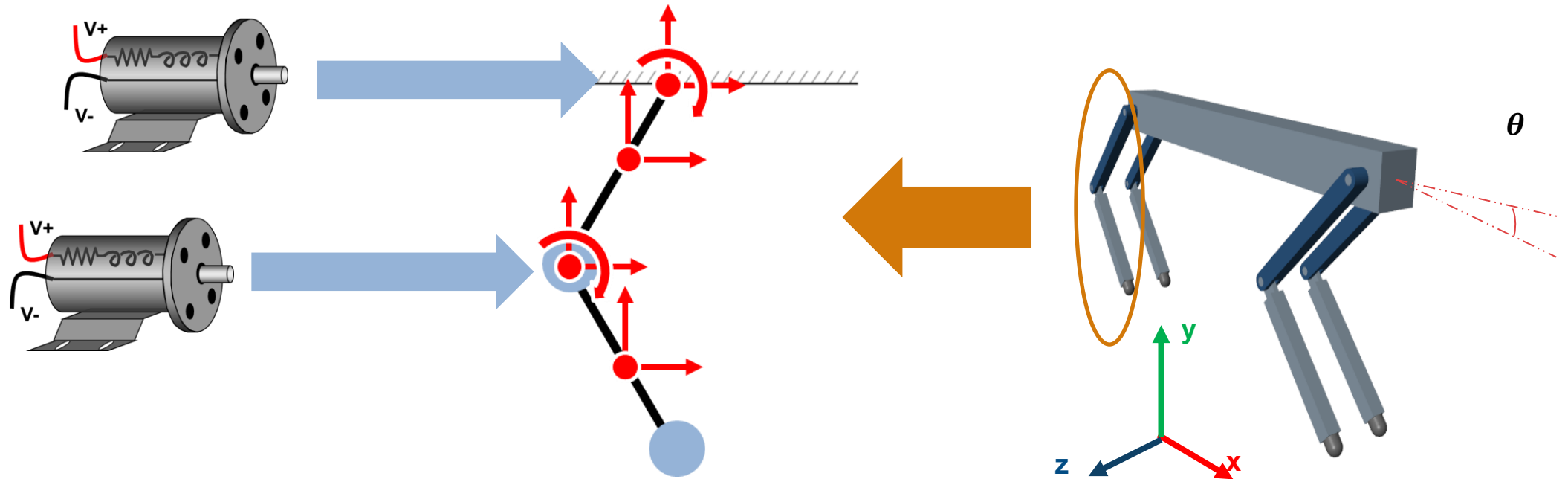


- Run simulation
  - Modify coordination of gravity  
 $[0 \ 0 \ 9.81] \rightarrow [0 \ 9.81 \ 0]$
  
- Extend model complexity
  - Copy & Paste your model to extend application
  - Physical network model can be utilized easily without delivering equation

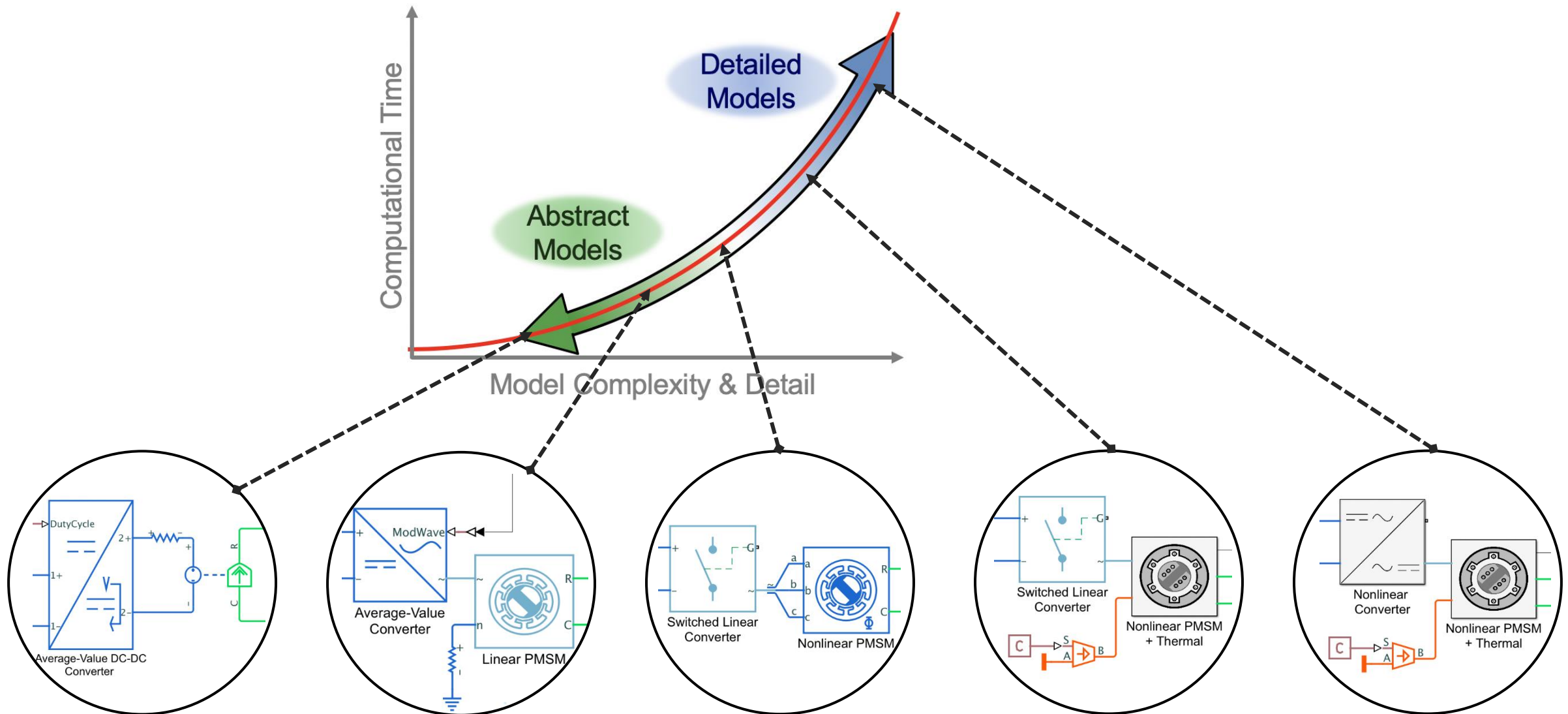




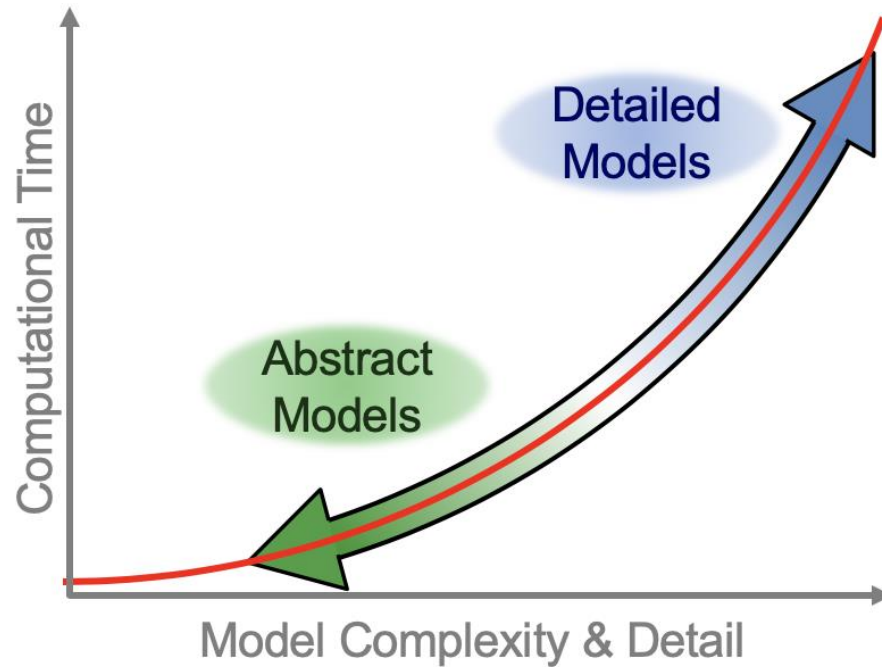
# Demos – Estimate Motor Parameters



# Configure the Model to Balance Model Fidelity and Simulation Speed



# In Summary, Simscape Electrical Lets You Model Electrical Systems with Varying Level of Detail



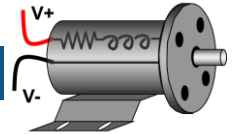
Proven Technology



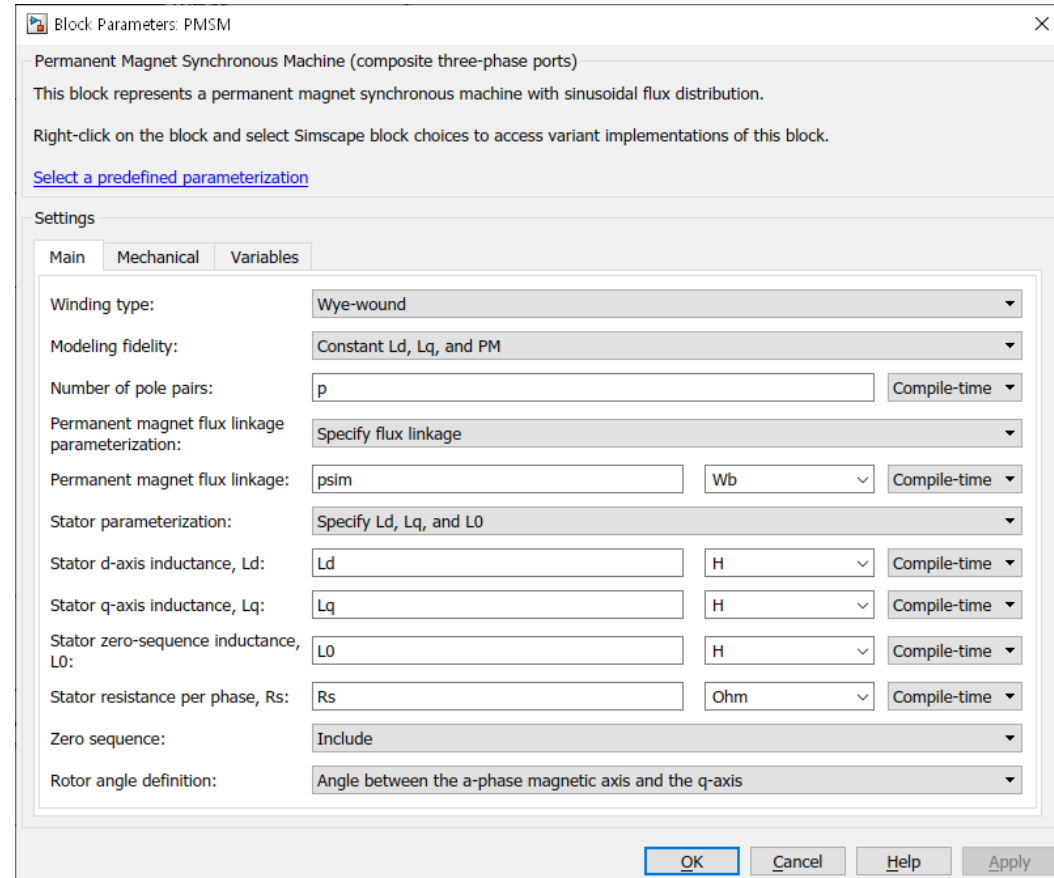
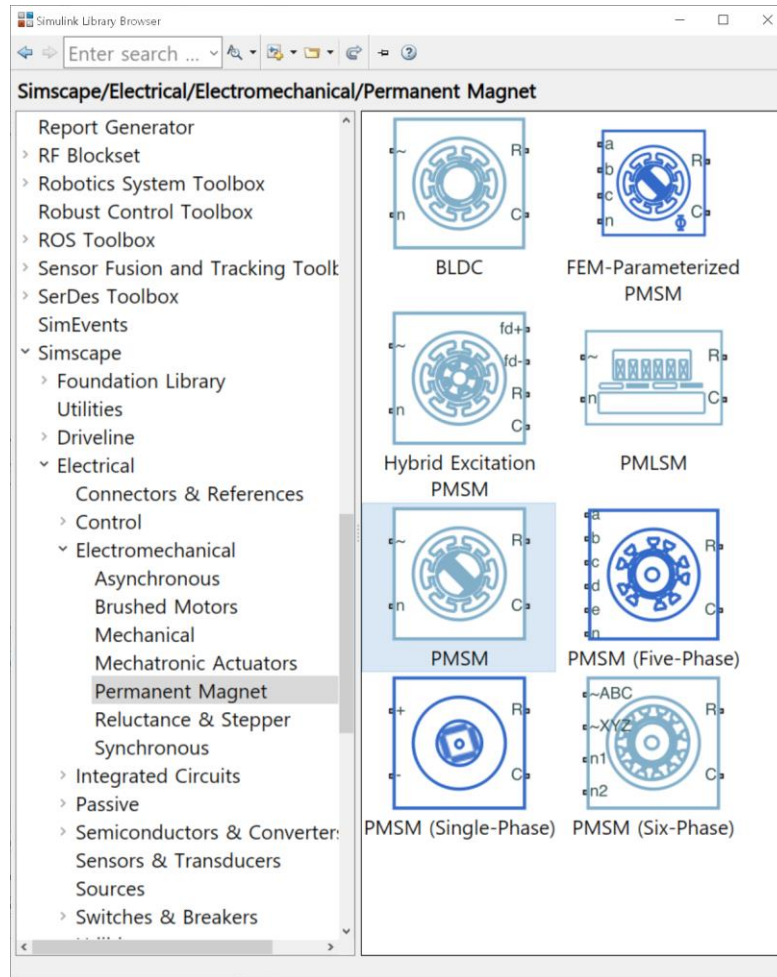
Level of detail

Proof-of-Concept

# Supported Different Type of Motor Model in Simscape Electrical

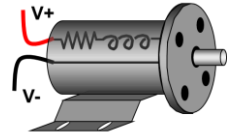


Motor



# Parameterization of Motor Modeling Using Simscape Electrical

Select a predefined parameterization



Motor

Block Parameters: PMSM

Permanent Magnet Synchronous Machine (composite three-phase ports)

This block represents a permanent magnet synchronous machine with sinusoidal flux distribution.

Right-click on the block and select Simscape block choices to access variant implementations of this block.

[Select a predefined parameterization](#)

Settings

Main Mechanical Variables

Winding type: Wye-wound

Modeling fidelity: Constant Ld, Lq, and PM

Number of pole pairs: p Compile-time

Permanent magnet flux linkage parameterization: Specify flux linkage

Permanent magnet flux linkage: psim Wb Compile-time

Stator parameterization: Specify Ld, Lq, and L0

Stator d-axis inductance, Ld: Ld H Compile-time

Stator q-axis inductance, Lq: Lq H Compile-time

Stator zero-sequence inductance, L0: L0 H Compile-time

Stator resistance per phase, Rs: Rs Ohm Compile-time

Zero sequence: Include

Rotor angle definition: Angle between the a-phase magnetic axis and the q-axis

OK Cancel Help Apply

Block Parameterization Manager: PMSM

Select manufacturer: ABB\_BALDOR

Select part: BSM132C\_8200AA

Attribute	Value
Manufacturer	ABB_BALDOR
Part number	BSM132C-8200AA
Part series	
Web link	<a href="https://www.baldor.com/brands/baldor-reliance/products/motors/servo-motors/ac-brushless">https://www.baldor.com/brands/baldor-reliance/products/motors/servo-motors/ac-brushless</a>
Part type	SPMSM, 325Vdc, 23.373kW, 1800rpm, 80A
Parameterization date	18-Jun-2020
Parameterization note	Predefined parameterizations of Simscape components use available data sources for su
Part data file location	Electromechanical\Permanent_Magnet\PMSM\ABB_BALDOR\BSM132C_8200AA.xml

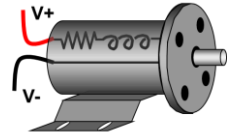
Update block with selected part

Compare block settings with selected part

PMSM model  
ABB\_BALDOR:BSM132C-8200AA

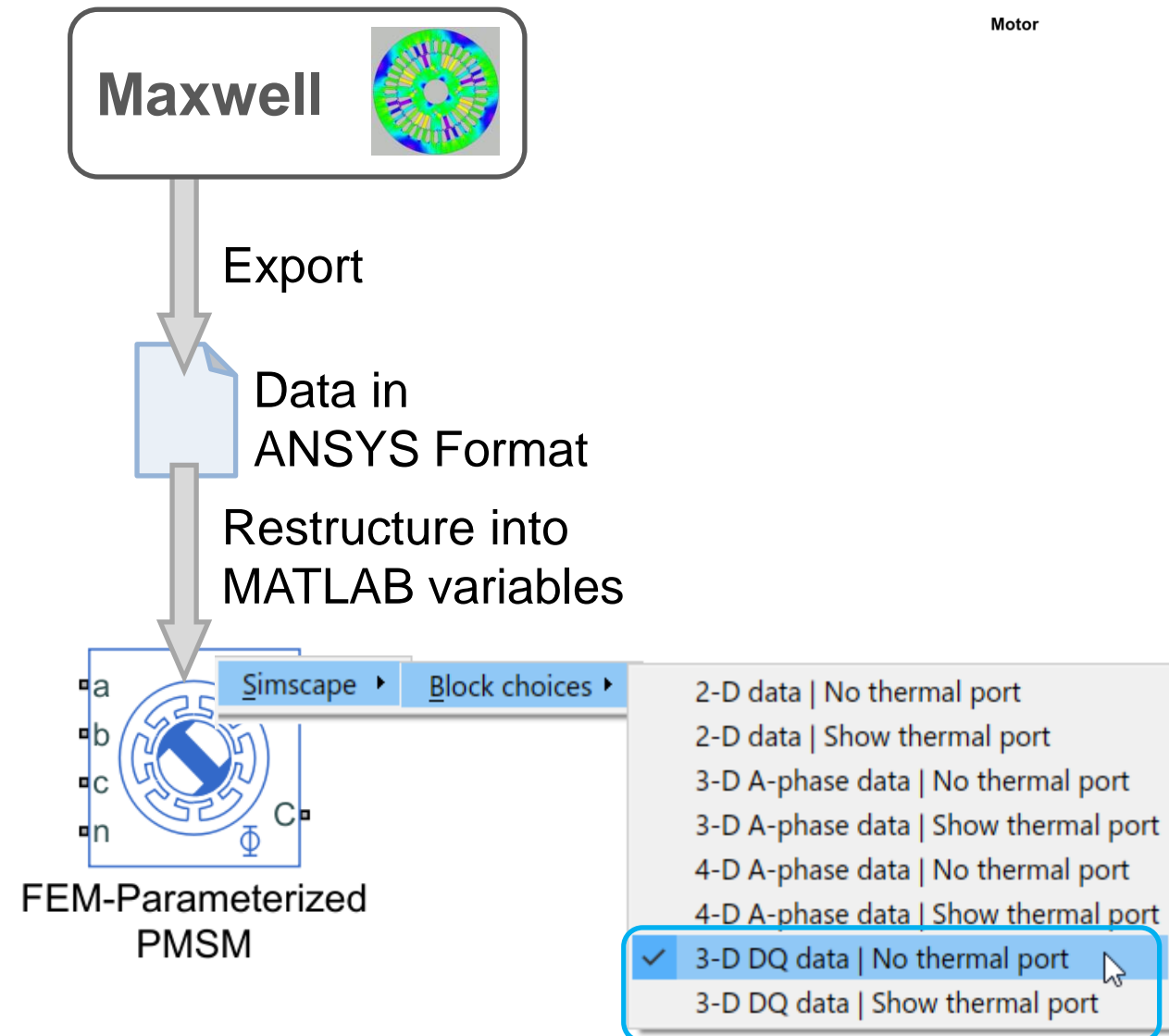
# Parameterization of Motor Modeling Using Simscape Electrical

## Nonlinear PMSM Model

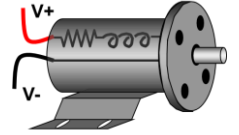


Motor

- Define PMSM behavior using d- and q-axis flux linkage
- Parameterization option is directly compatible with Maxwell, JMAG and Motor-CAD data
  - With a few changes to text file, MATLAB variables that match block parametrization can be generated

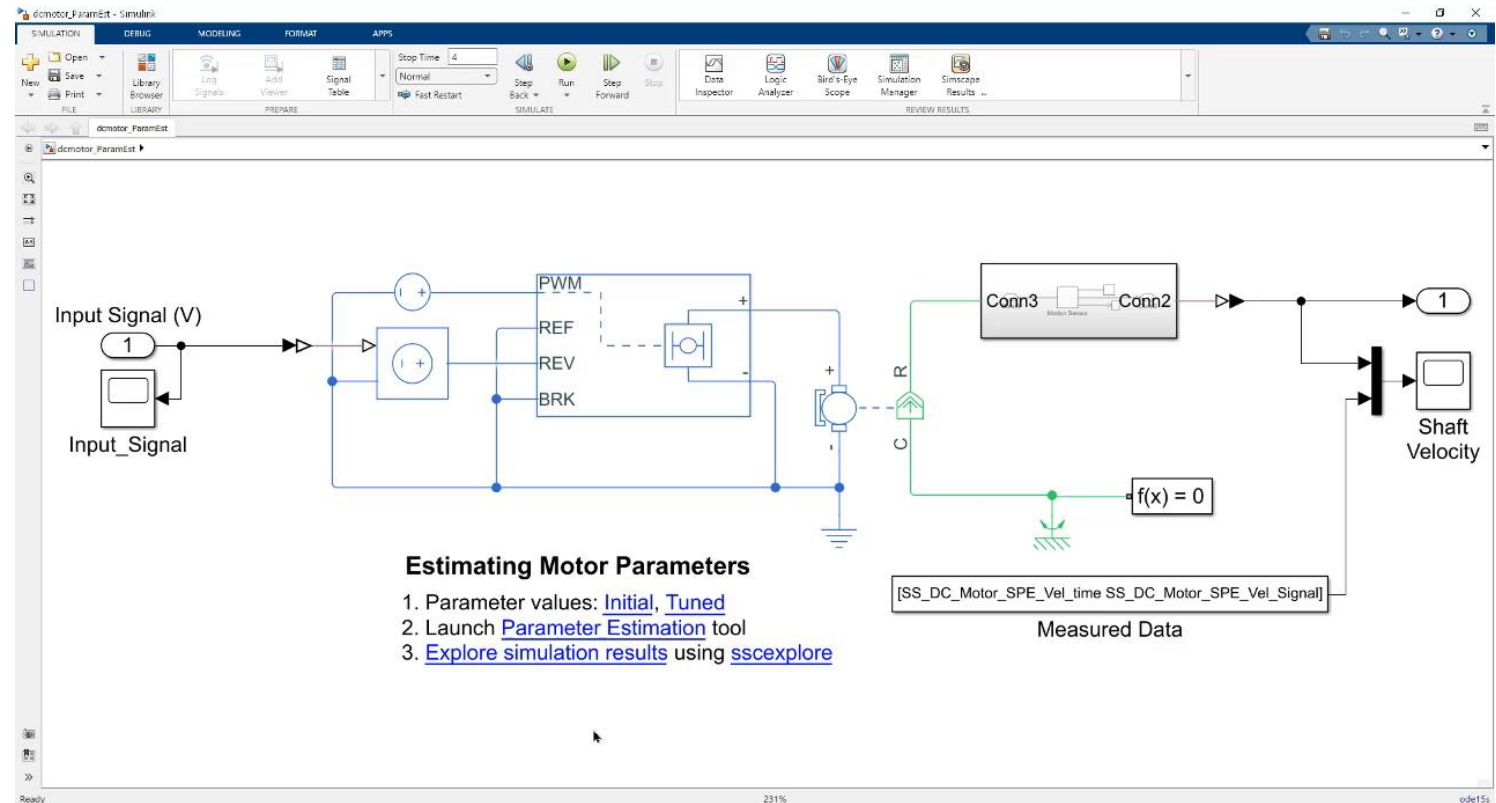


# Demos – Estimate Motor Parameters

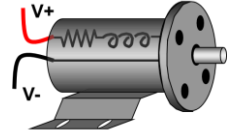


Motor

- Motor modeling
  - Simscape Electrical supports multiple type of motor and driver
  
- Parameterize components
  - Simulink Design Optimization support parameter estimation

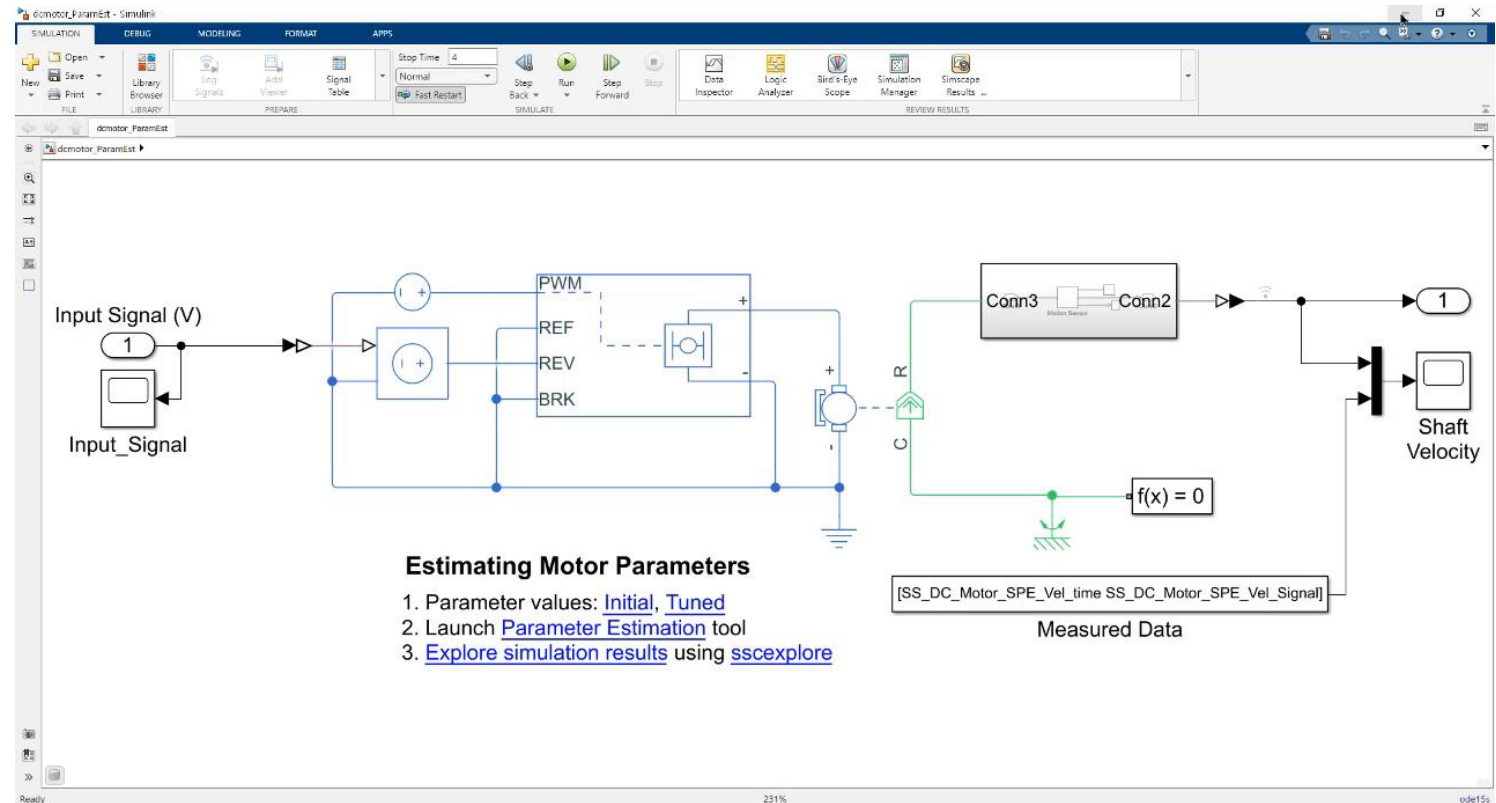


# Demos – Estimate Motor Parameters

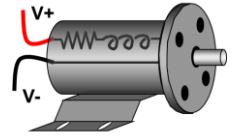


Motor

- Tip for fast simulation
  - Use Fast Restart
  - Utilize Parallel Computing toolbox
  
- Estimate parameters
  - Adapt experiments data to estimate components parameters using Simulink Design Optimization Toolbox







Motor

# Demos – Estimate Motor Parameters

## Select Parameters and Set Ranges

- Select the motor parameters to tune
  - Mechanical Characteristics : DC\_Motor\_B, DC\_Motor\_J
  - Electrical Characteristics : DC\_Motor\_L, DC\_Motor\_R, DC\_Motor\_K
- Set the valid range for each value

Select model variables

Filter by variable name

▼	Variable	Current val...	
<input checked="" type="checkbox"/>	DC_Motor_B	0.5	<a href="#">dcmotor</a>
<input checked="" type="checkbox"/>	DC_Motor_J	0.01	<a href="#">dcmotor</a>
<input checked="" type="checkbox"/>	DC_Motor_K	0.02	<a href="#">...DC Mot</a>
<input checked="" type="checkbox"/>	DC_Motor_L	0.01	<a href="#">dcmotor</a>
<input checked="" type="checkbox"/>	DC_Motor_R	3	<a href="#">dcmotor</a>

► Specify expression indexing if necessary (e.

Edit: Estimated Parameters

Parameters Tuned for all Experiments

[DC\\_Motor\\_B](#)

▼ 0.5

Minimum: 0.01

Maximum: Inf

Scale: 0.5

[DC\\_Motor\\_J](#)

▶ 0.01

[DC\\_Motor\\_K](#)

▶ 0.02

[DC\\_Motor\\_L](#)

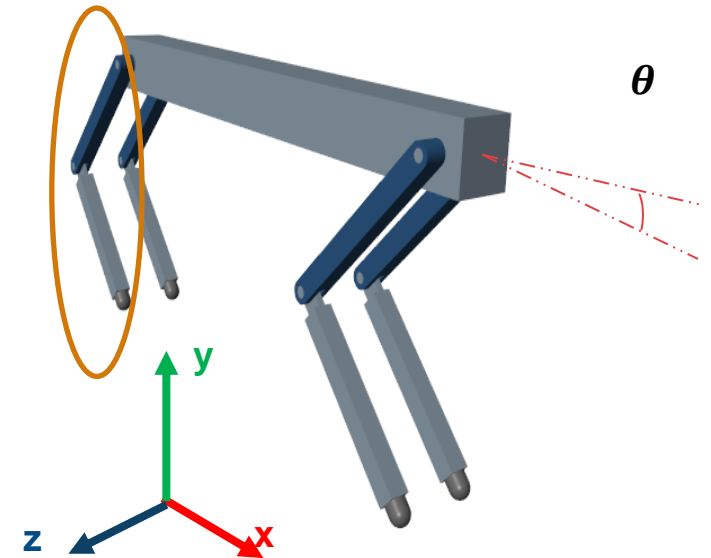
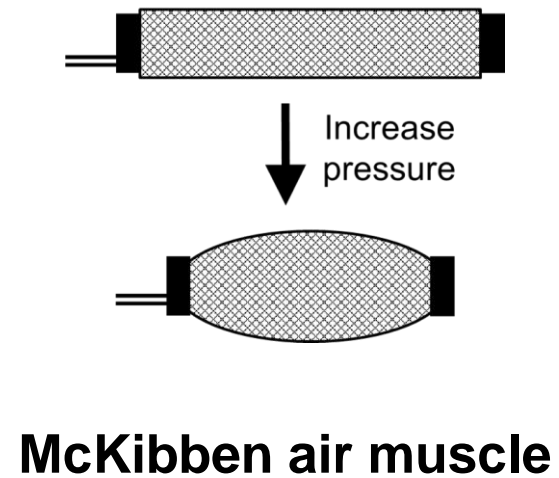
▶ 0.01

[DC\\_Motor\\_R](#)

▶ 3

Select parameters

# Demos – Creating custom Simscape components

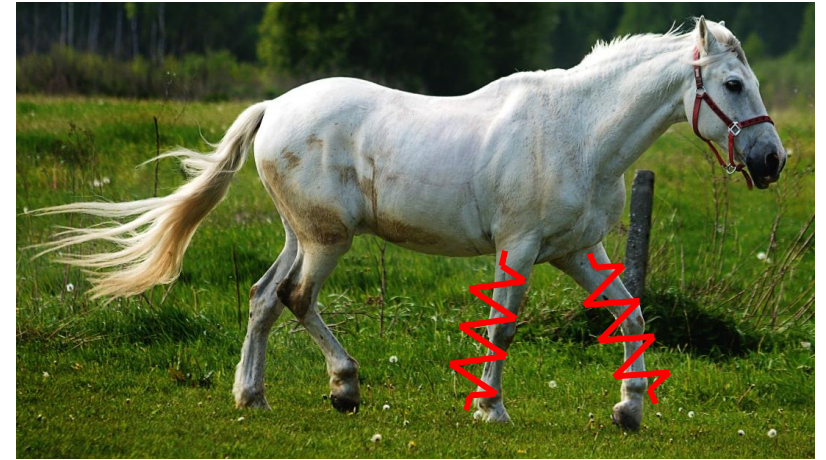


# Creating custom Simscape components

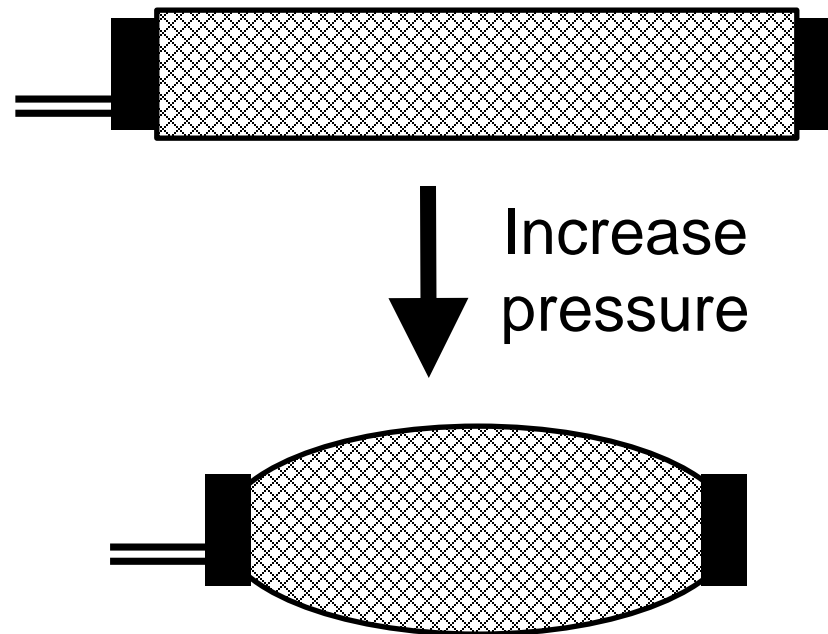
## *Example: McKibben air muscle*

### Steps:

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



McKibben air muscle



# Creating custom Simscape components

## Step 1: Write out equations

$L_u$  = Un-stretched length

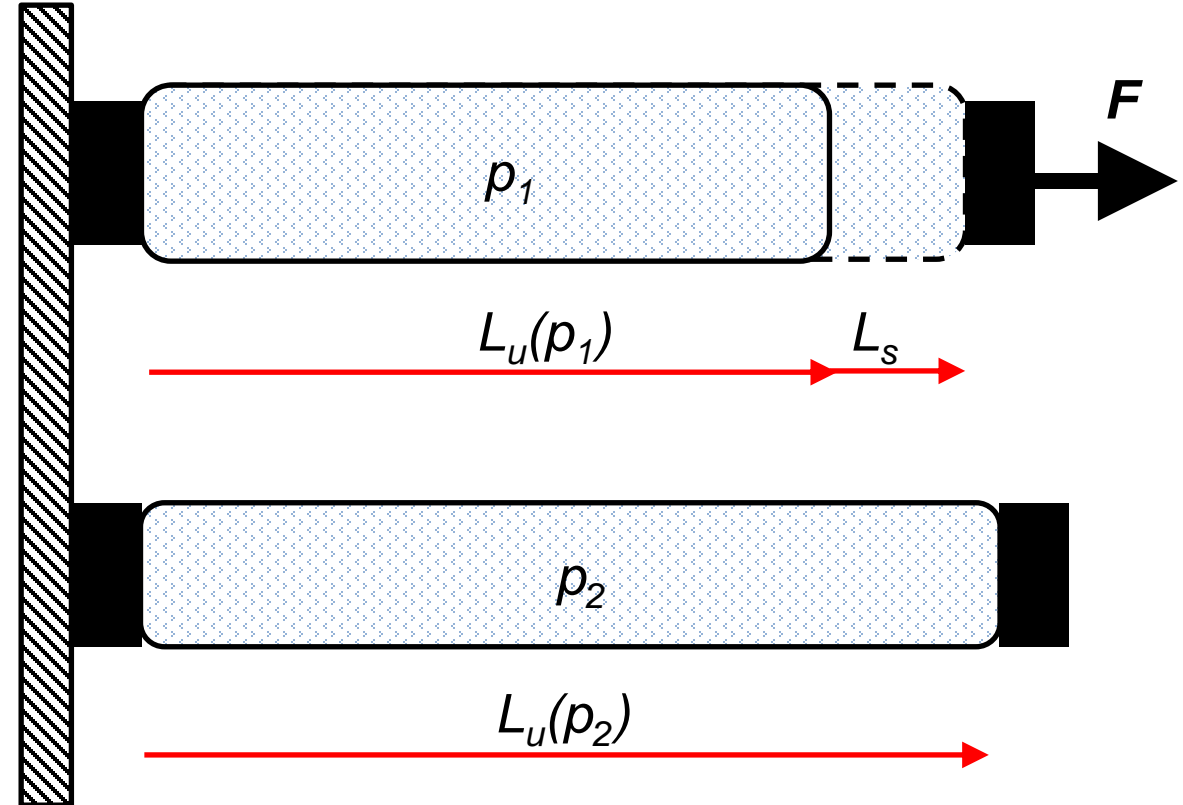
$L_s$  = Additional stretch due to force,  $F$

### Assumptions:

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

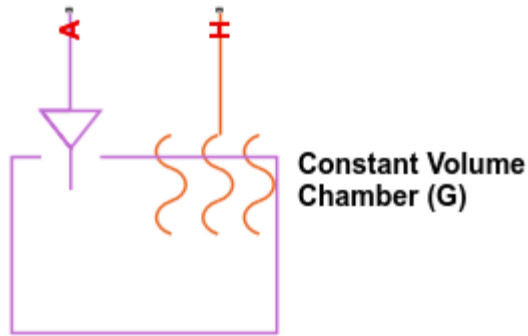
### Equations:

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



# Creating custom Simscape components

## Step 2: Find starting point from foundation library



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

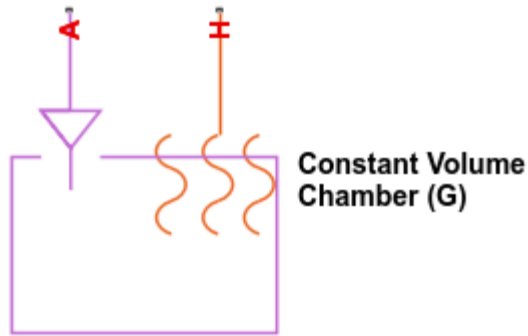
```

C:\Program Files\MATLAB\R2017a2\toolbox\physmod\simscape\library\m\+foundation\+gas\+elements\constant_volume_chamber.ssc
EDITOR VIEW
+ Find Files
+ Compare
+ Go To
+ Comment
+ Indent
+ Breakpoints
FILE NAVIGATE EDIT BREAKPOINTS
1 component constant_volume_chamber < foundation.gas.one_port_vertical
2 % Constant Volume Chamber (G)
3 % This block models mass and energy storage in a gas network. The chamber
4 % contains a constant volume of gas. The pressure and temperature evolve
5 % based on the compressibility and thermal capacity of this gas volume.
6 %
7 % Port A is the gas conserving port associated with the chamber inlet. Port
8 % H is the thermal conserving port associated with the temperature of the
9 % gas inside the chamber.
10
11 % Copyright 2016 The MathWorks, Inc.
12
13 nodes
14     H = foundation.thermal.thermal; % H:top
15 end
16
17 parameters
18     volume = {0.001, 'm^3'}; % Chamber volume
19     area_A = {0.01, 'm^2'}; % Cross-sectional area at port A
20 end
21
Simscape model file Ln 1 Col 1

```

# Creating custom Simscape components

## Step 3: Incrementally add functionality



Add:

- Two mechanical ports

```

1  component air_muscle < foundation.gas.one_port_vertical
2  % Air Muscle (G)
3  % This block models a McKibben air muscle.
4
5  % Copyright 2016-2017 The MathWorks, Inc.
6
7  nodes
8  |   H = foundation.thermal.thermal; % H:top
9  |   R = foundation.mechanical.translational.translational; % R:bottom
10 |   C = foundation.mechanical.translational.translational; % C:top
11 end

```

- Two additional new equations

$$L = L_u(p) + L_s \longrightarrow 152 \quad \mathbf{L} \quad == \quad \mathbf{Ls} + \mathbf{Lu};$$

$$F = k \times L_s \longrightarrow 153 \quad \mathbf{force} == \mathbf{K} * \mathbf{Ls};$$

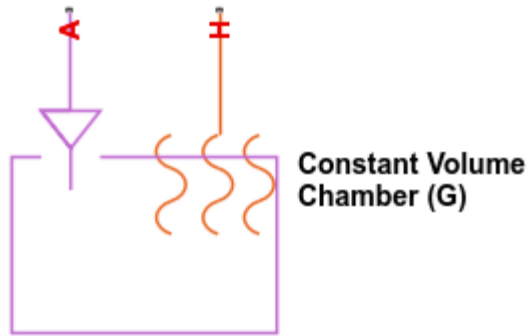
```

149 Lu = tablelookup(pVec, LuVec, p_chamber,

```

# Creating custom Simscape components

## Step 3: Incrementally add functionality



Add definitions for:

- Variables
- Parameters

```

33 variables
34     % Mechanical variables
35     force = {0, 'N'}; % Force
36     Ls = {0, 'm'};    % Stretch
37 end

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

# Creating custom Simscape components

## Step 4: Build library and run test model

Block Parameters: McKibben Air Muscle

Air Muscle (G)

This block models a McKibben air muscle

[Source code](#)

Settings

Parameters Variables

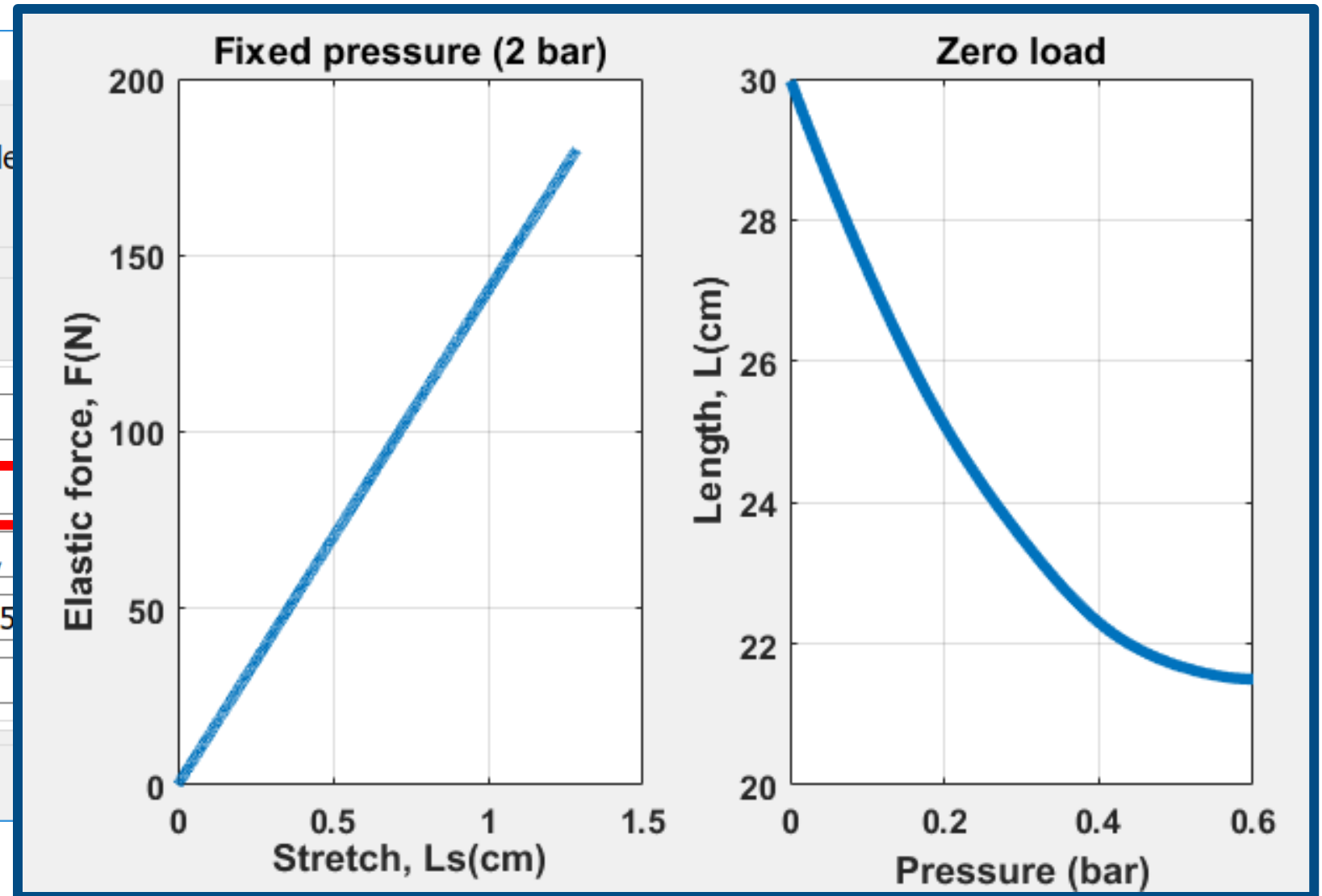
Cross-sectional area at port A: 0.01

**Stiffness: 140**

Pressures: [0.0, 1.0, 2.0, ...]

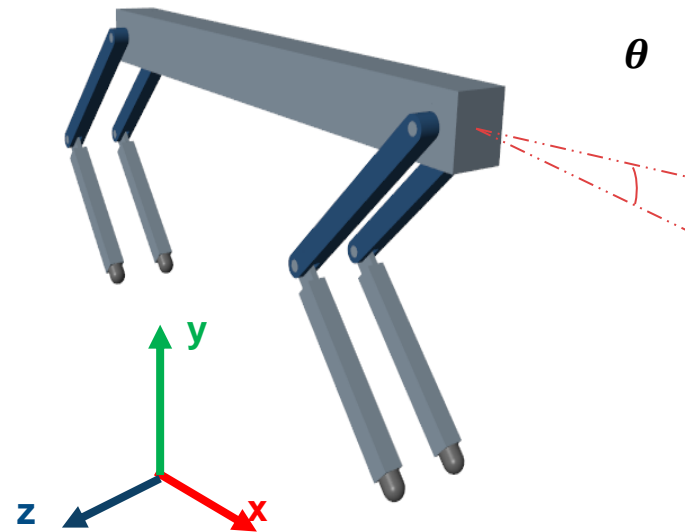
Unstretched lengths: [7.3, 25.1, 23.5, ...]

Volume: 85





# Demos - Quadruped Robot



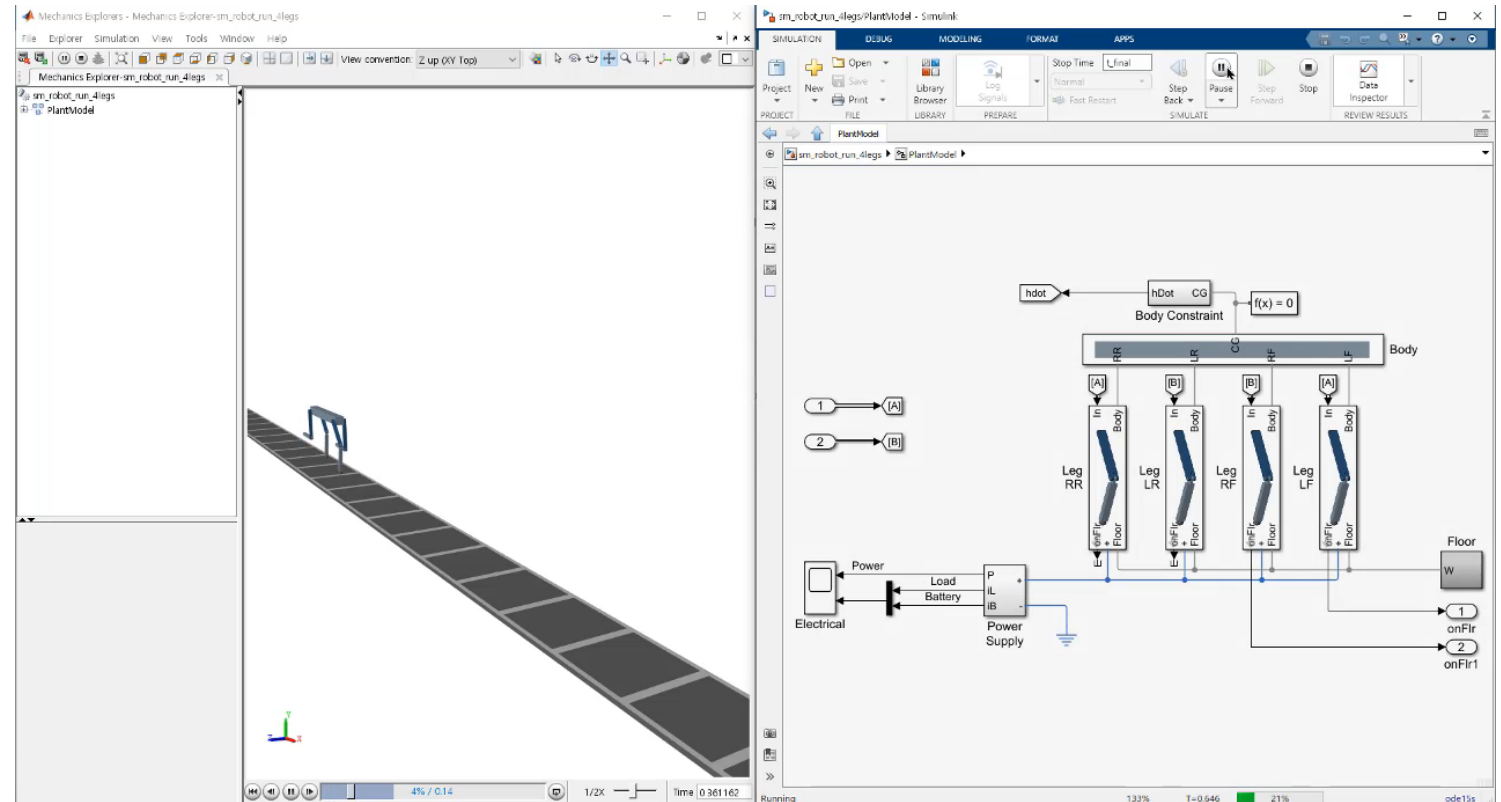
**Walking Robot**



Walking Robot

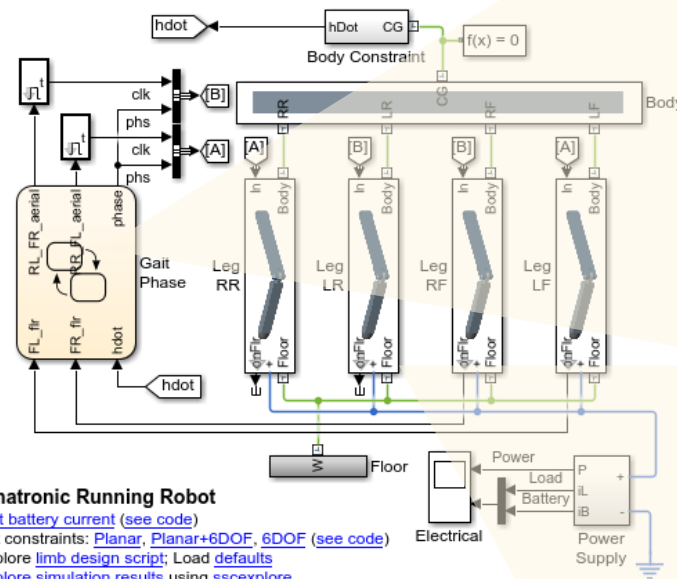
# Demos - Quadruped Robot

- Simulate models
  - Identify system behavior
  - Design controller
  
- Analyze physical system
  - Evaluate overall system requirements
  - Confirm system capacity such as battery power, motor capacity ETC



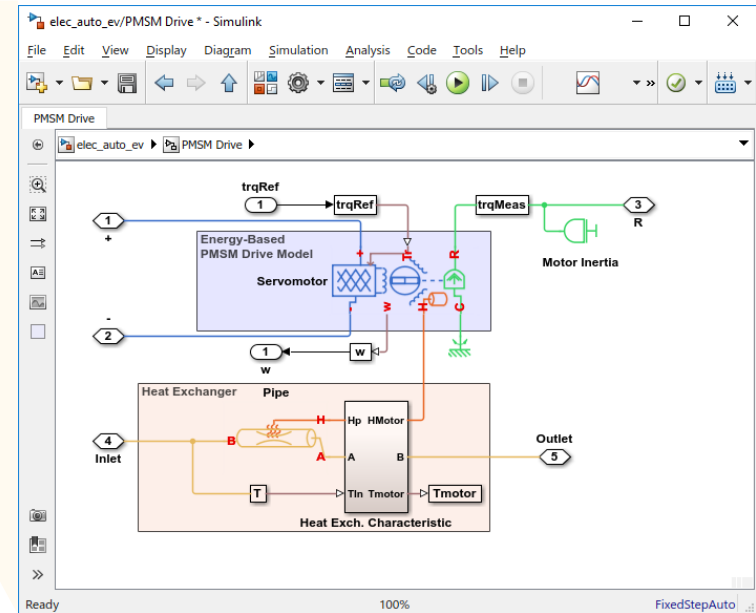
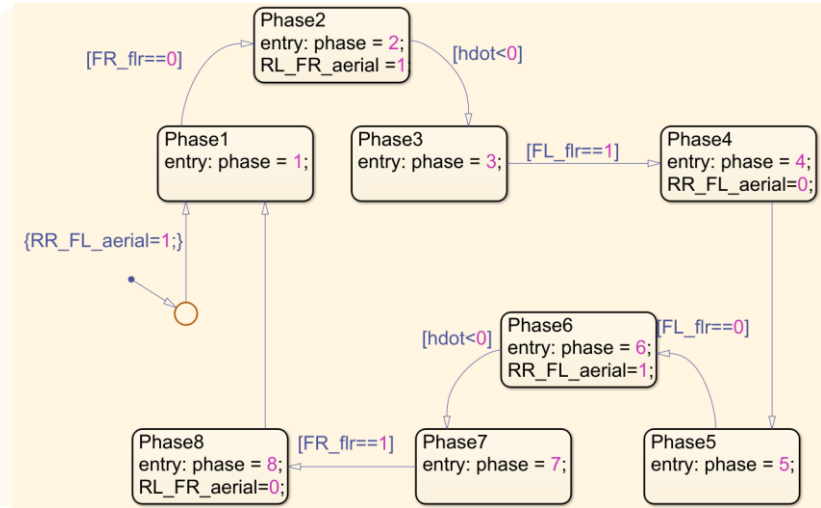
# Simscape works in a variety of applications

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



**Mechatronic Running Robot**

1. [Plot battery current](#) (see code)
2. Set constraints: [Planar](#), [Planar+6DOF](#), [6DOF](#) (see code)
3. Explore [limb design script](#); Load defaults
4. [Explore simulation results](#) using [sscexplore](#)
5. [Learn more](#) about this example

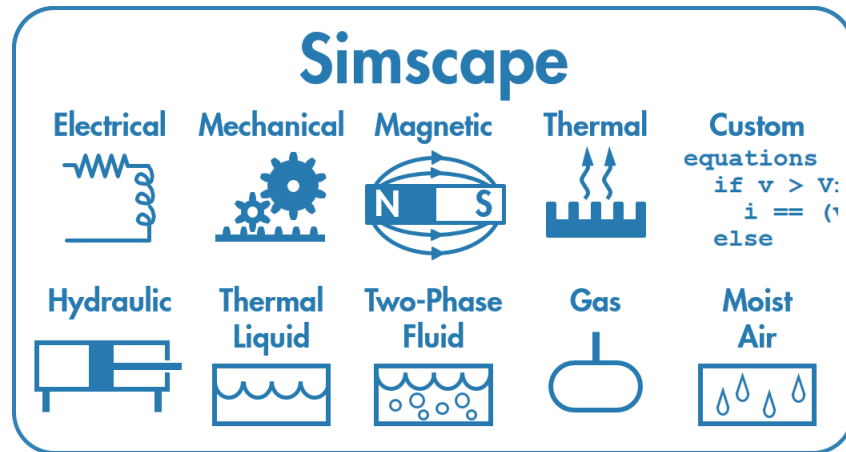




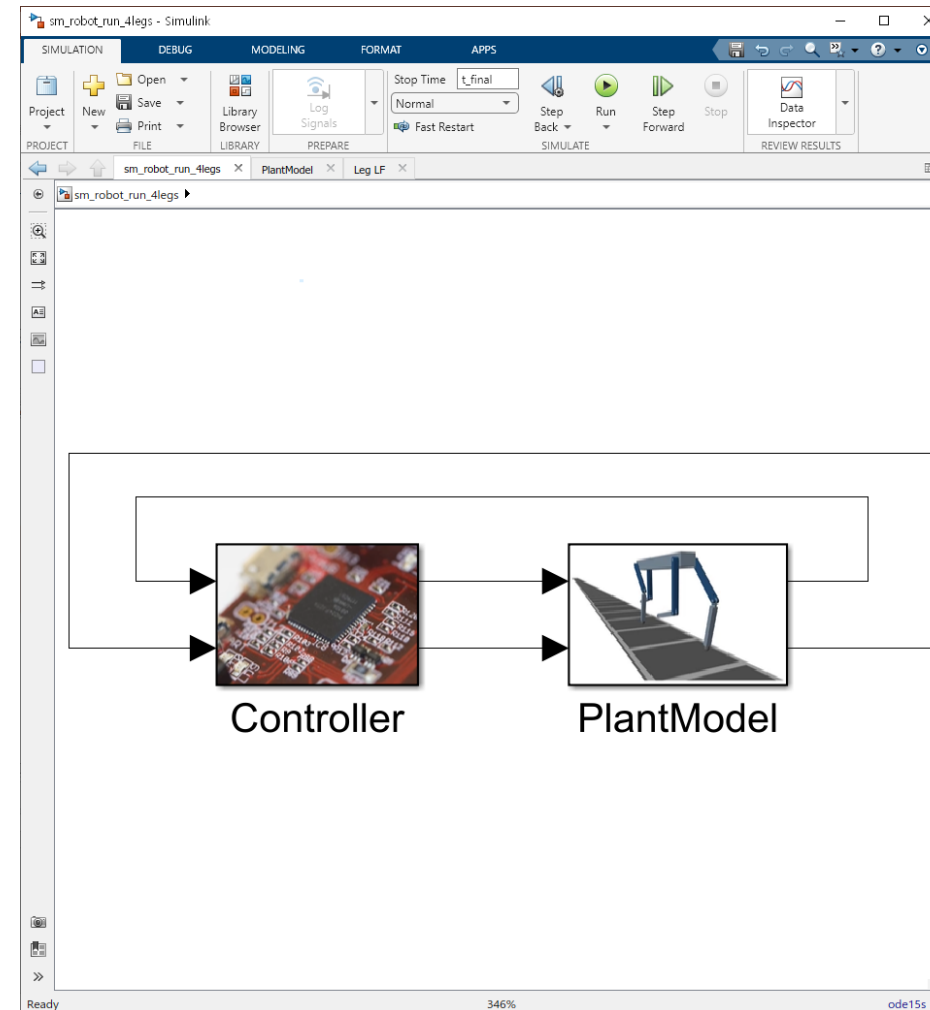
Walking Robot

# Demos - Quadruped Robot

1. Can you analyze how the system behaves before building any physical prototypes?



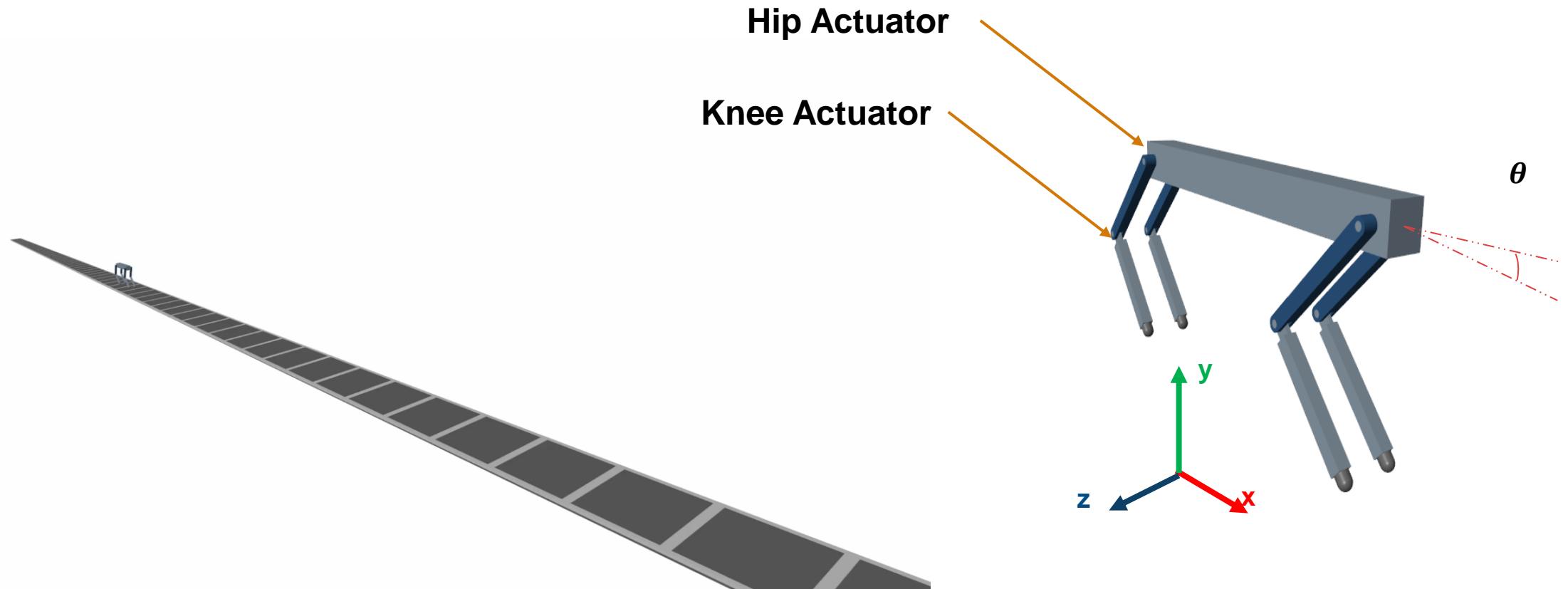
2. How do you model the electromechanical and multibody components?





Walking Robot

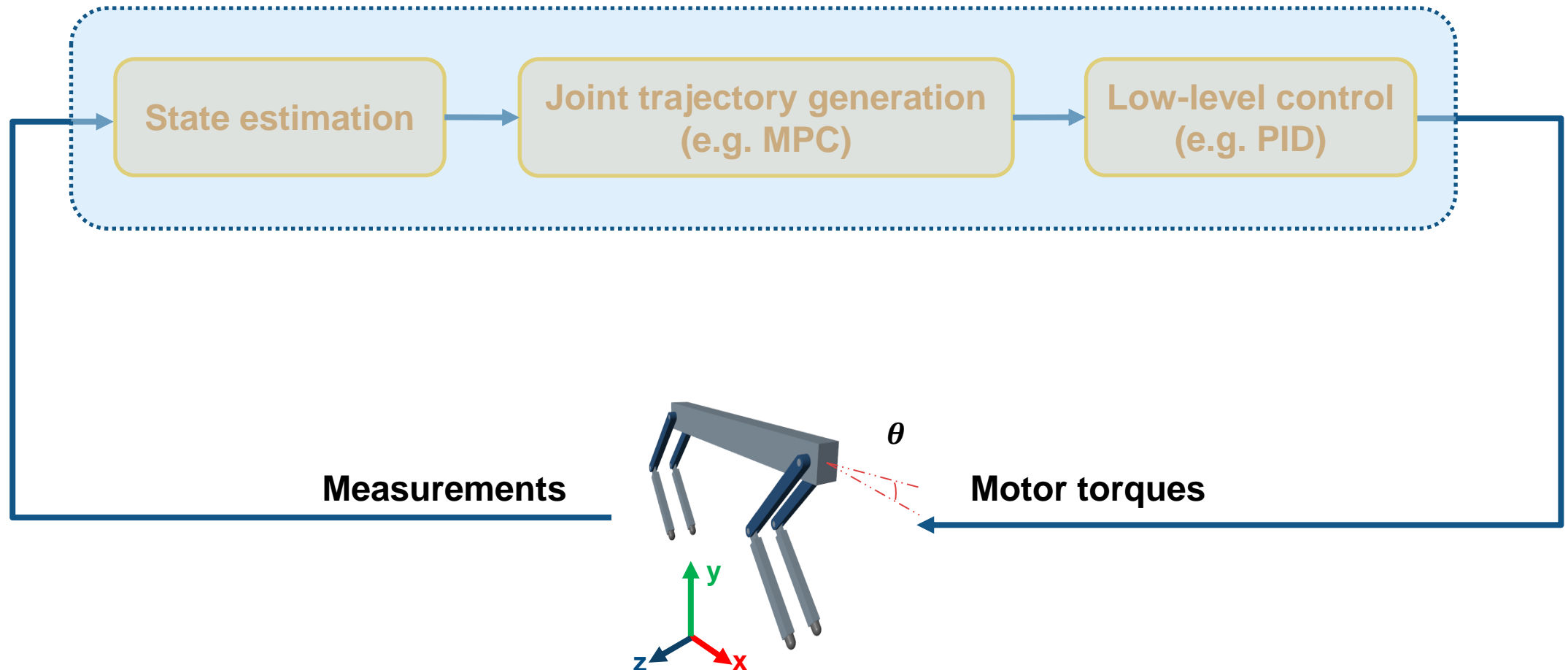
# Why Should You Care About Reinforcement Learning?





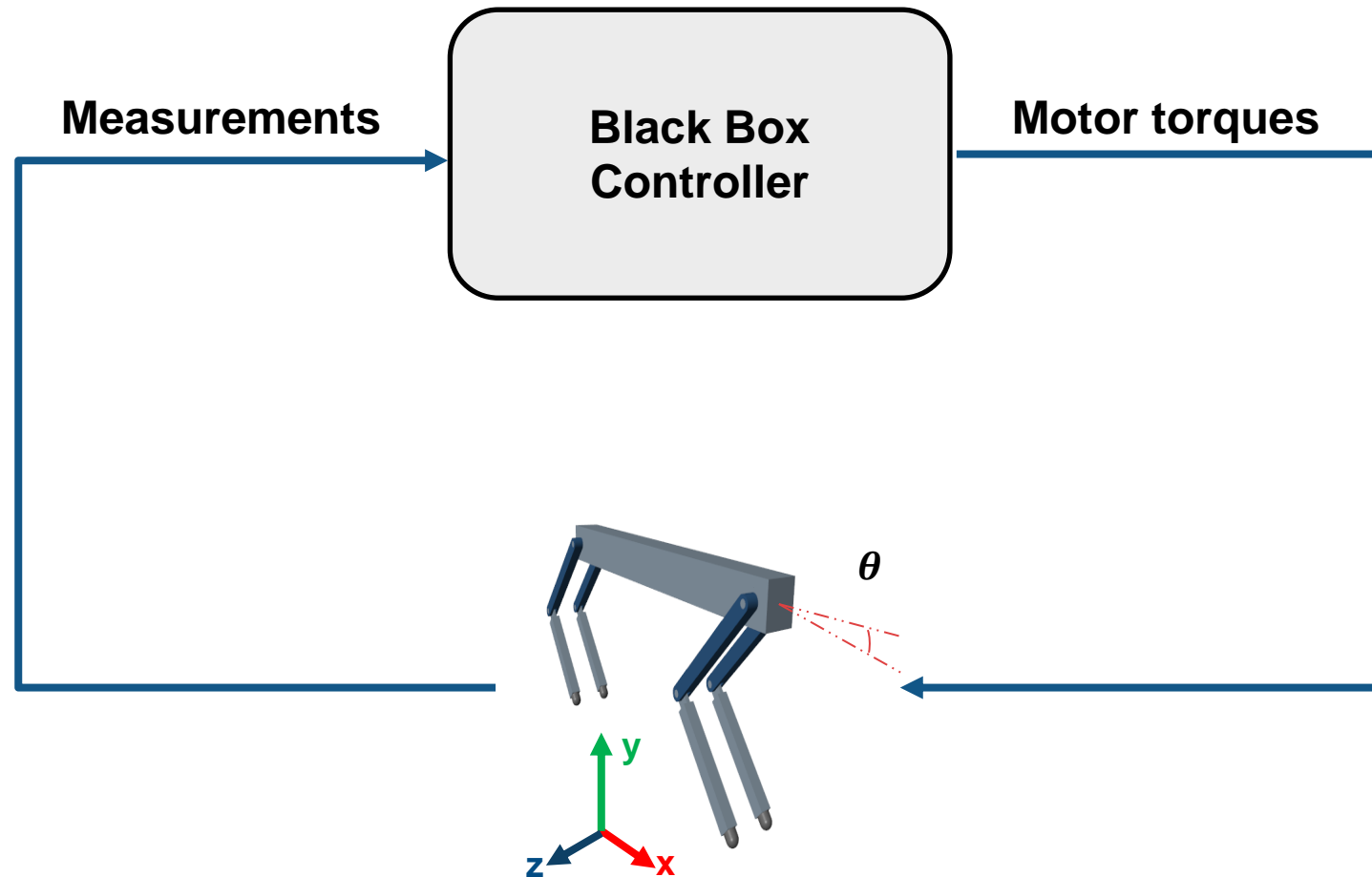
Walking Robot

# One Approach Could Be...





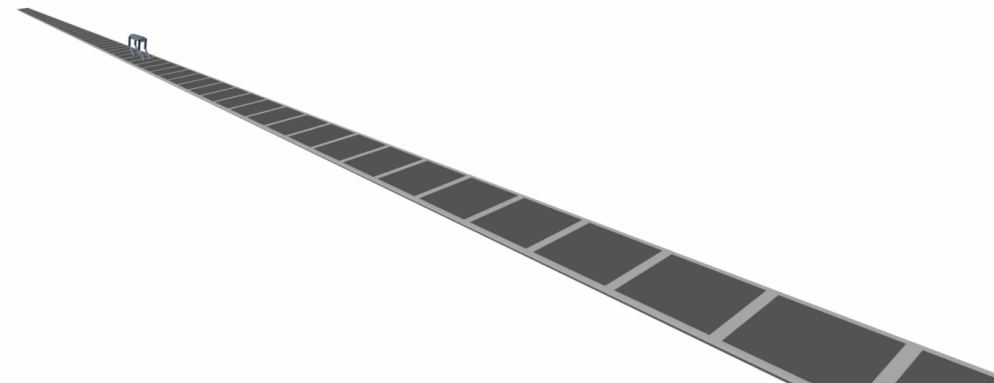
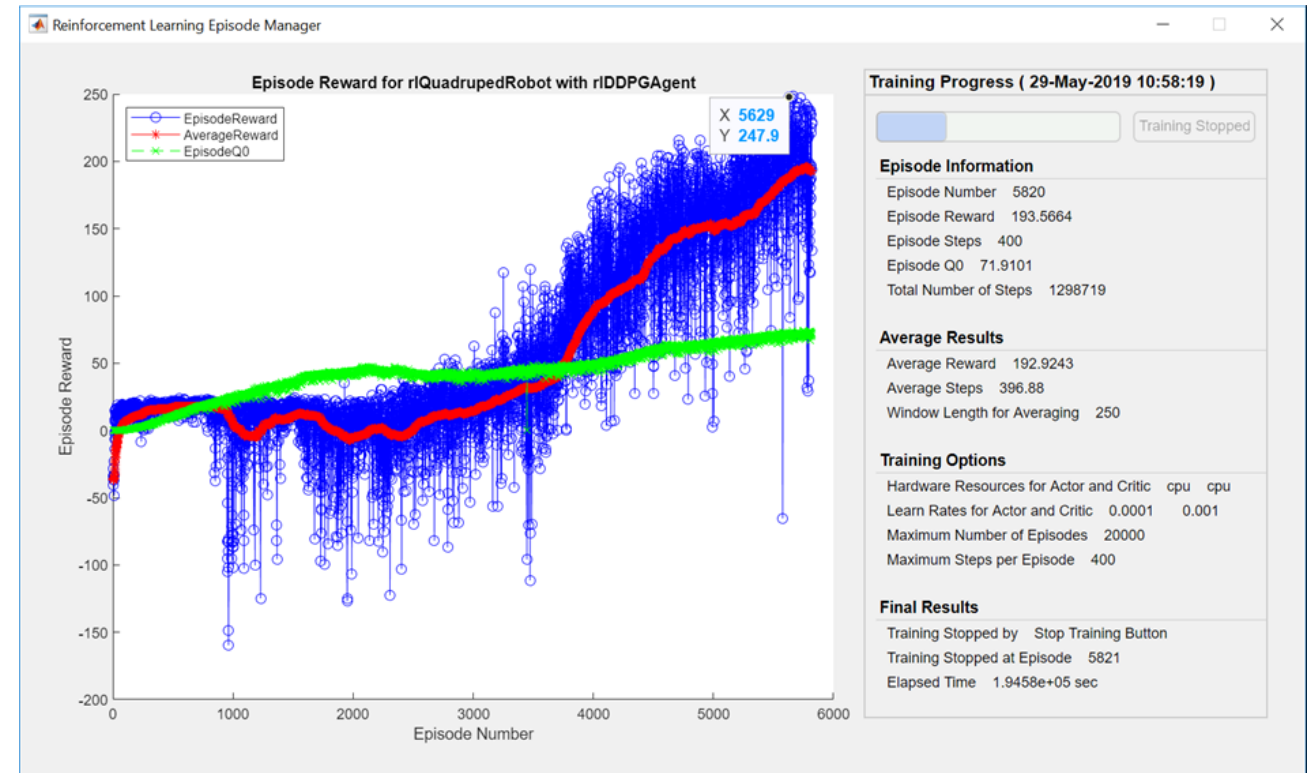
# Any Alternatives?



# Quadruped Robot Locomotion Using Reinforcement Learning

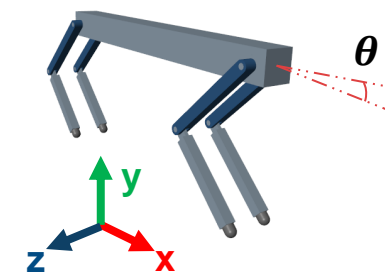
## Train quadruped robot to walk with DDPG agent

- Quadruped robot modeled in Simscape Multibody with contact forces
- Policy takes in 44 observations and outputs 8 torque values





# Reinforcement Learning Model



**44 Observations:**

**Body C.M.:**

- 2 pos (y, z)
- 3 vel (vx, vy, vz)
- 4 orient (quaternion)
- 3 ang vel (wx, wy, wz)

**Each Leg:**

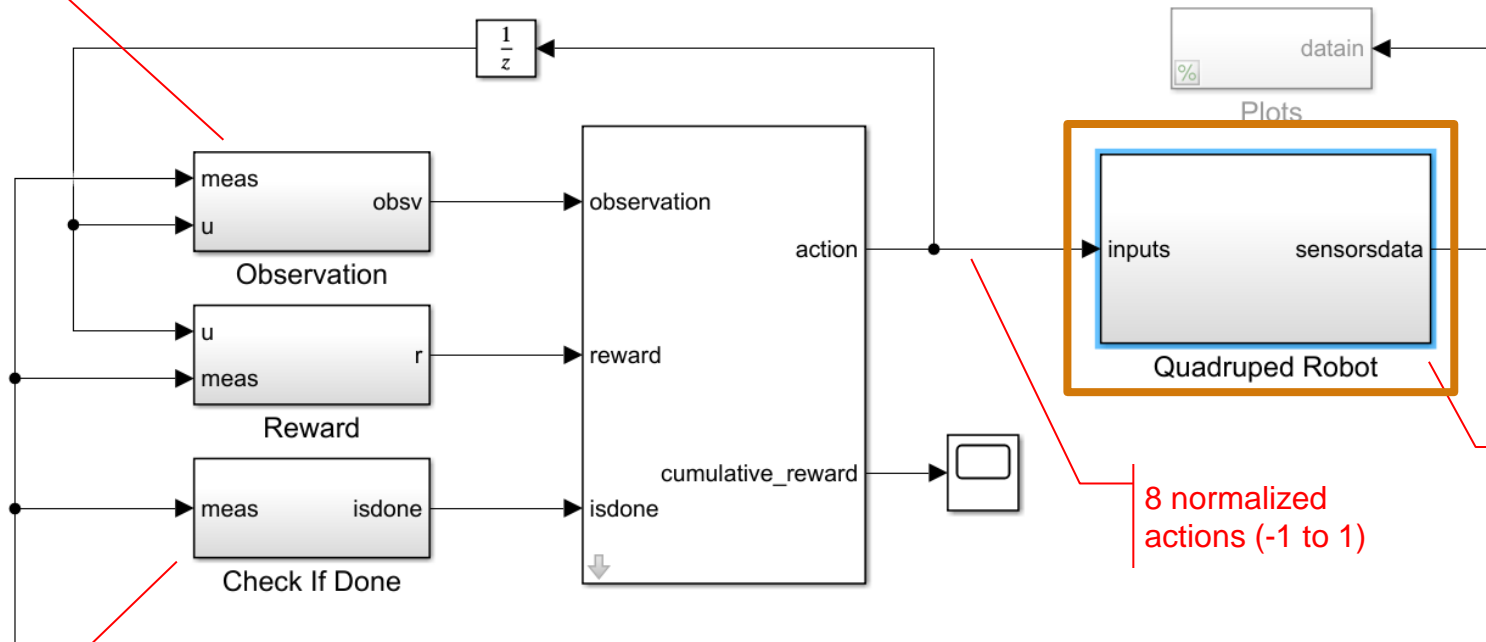
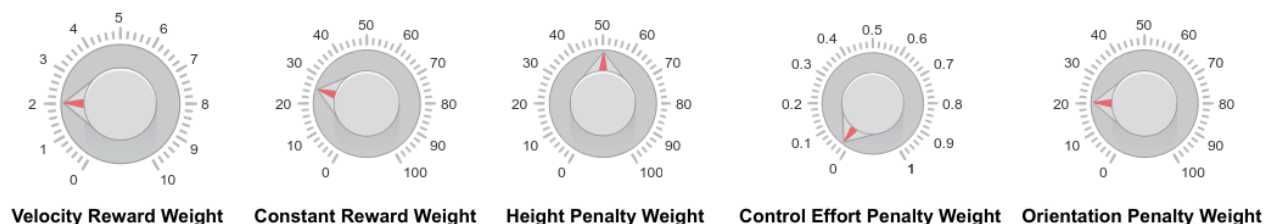
- 2 joint ang (hip, knee)
- 2 joint vel (hip, knee)
- 2 Contact Forces

**Actuation:**

- 8 joint torques

**Episode termination criteria:**

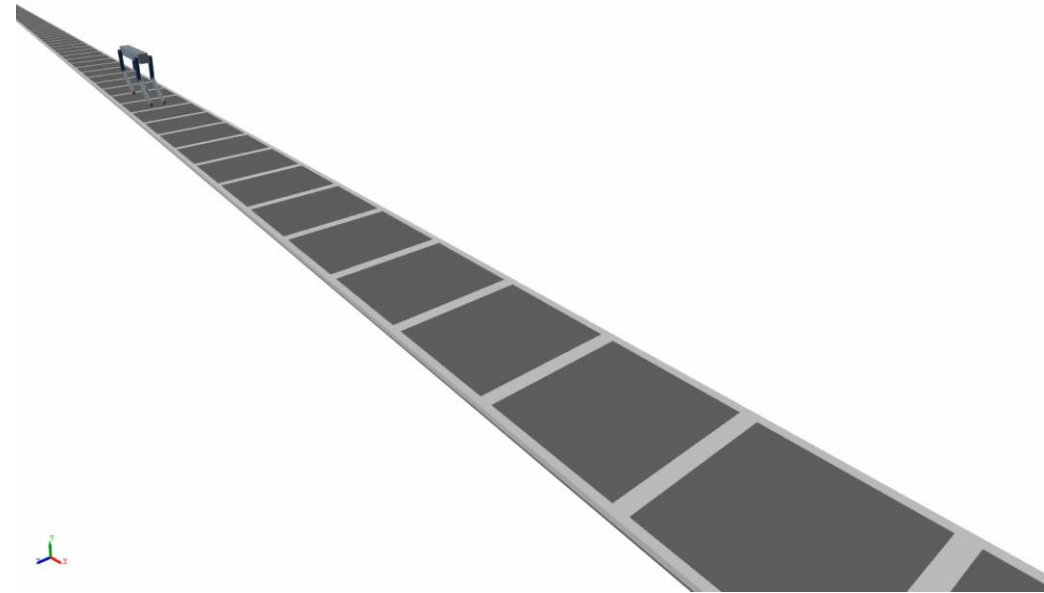
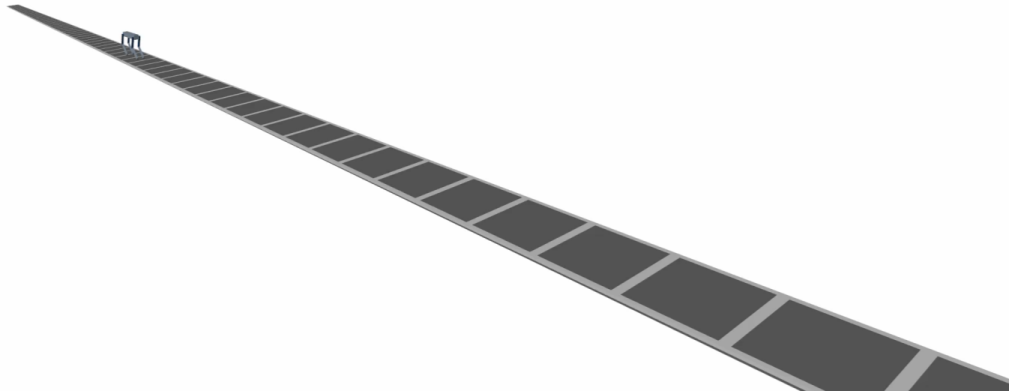
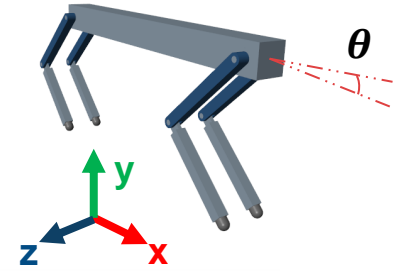
- Body C.M. below min. height
- Head/Tail below ground
- Knee below ground
- Roll, pitch, yaw exceeds bounds



**Quadruped Walking Robot Example**

Copyright 2019 The MathWorks, Inc.

# Training and Reward Function Shaping

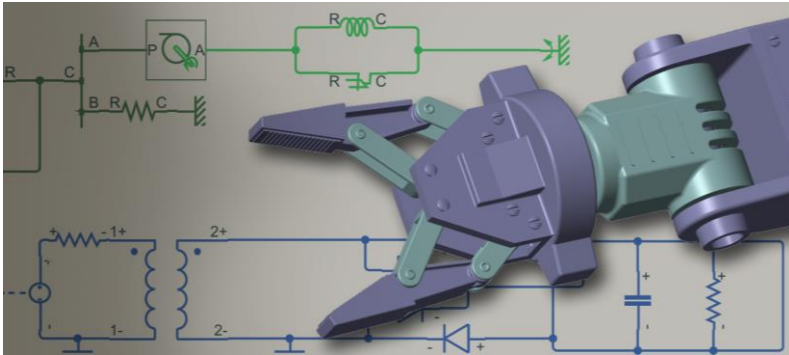


$$r_t = v_x + 25 \frac{T_s}{T_f} - 50\hat{y}^2 - 20\theta^2 - 0.02 \sum_i u_{t-1}^i{}^2$$

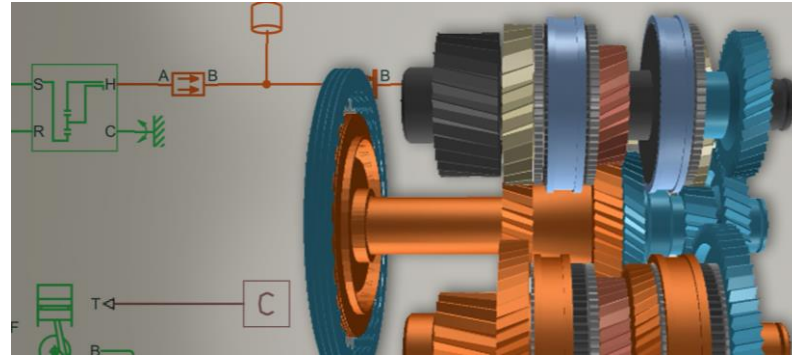
$$r = \mathbf{2}v_x + 25 \frac{T_s}{T_f} - 50\hat{y}^2 - 20\theta^2 - 0.02 \sum_i u_{t-1}^i{}^2$$

# Simscape works in a variety of applications

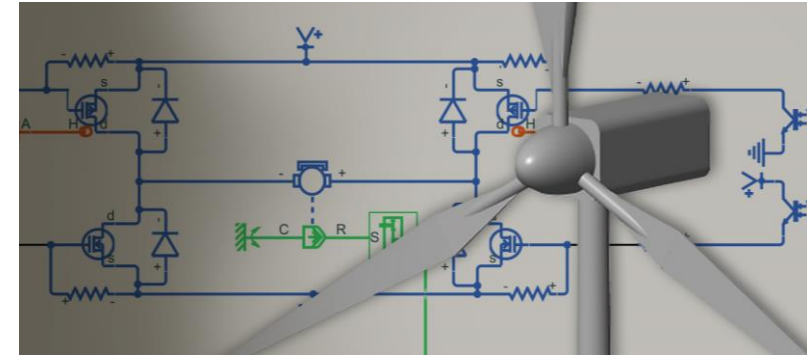
## Robotics



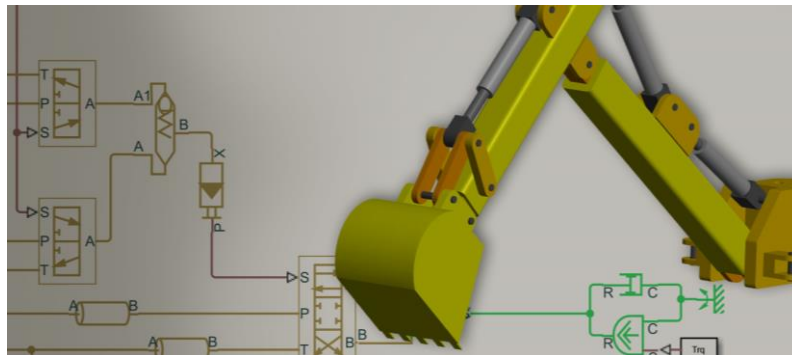
## Drivelines



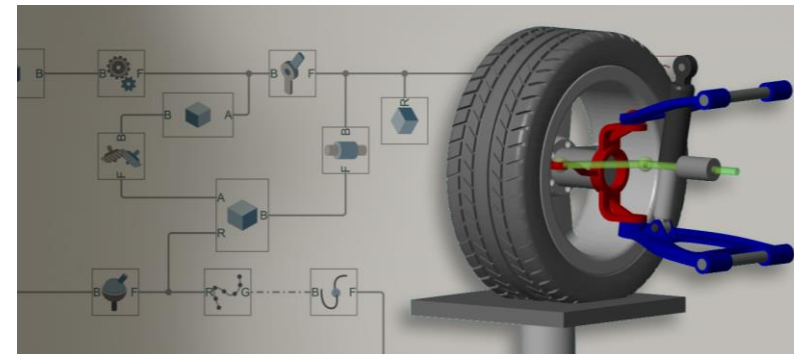
## Renewable Energy



## Actuation



## Hybrid Vehicles



# Volvo Construction Equipment Streamlines Product Development with a Real-Time, Human-in-the-Loop Simulator

## Challenge

Evaluate design concepts and parameter values for construction equipment before building physical prototypes

## Solution

Use Simulink, Simscape, and Simulink Real-Time to model hydraulic, mechanical, and engine systems and perform real-time, operator-in-the-loop simulations

## Results

- Number of prototypes reduced
- Issues in the field resolved faster
- Controller tuned in simulation



Volvo Construction Equipment's real-time, human-in-the-loop simulator.

*"It was technically impossible for us to build a full-scale hydraulic system model to run in real time without Simulink, Simscape, and Simulink Real-Time. Our simulator enables us to test new concepts for construction equipment, tune parameters, reduce lead times, and minimize issues in the field."*

*- Jay Yong Lee, Volvo Construction Equipment*

# Krones Develops Package-Handling Robot Digital Twin

## Challenge

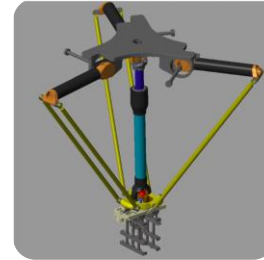
Increase the performance of an automated beverage-packaging system by incorporating a dynamic tripod robot into the design

## Solution

Use Simulink and Simscape Multibody to create an accurate digital twin that supports design optimization, fault testing, and predictive maintenance

## Results

- Robot performance increased
- Product development time shortened
- Testing time significantly reduced



The Krones Robobox T-GM package-handling robot.

*“Simulations of the digital twin in Simulink enabled us to obtain data and insights that would be either impossible to get via hardware tests or simply too costly and time-consuming. Visualizing forces and moments helped us to understand the effects of individual components on a highly dynamic robot.”*  
- Benedikt Böttcher, Krones

## Key Takeaways

- Simscape helps you create models of multidomain systems in Simulink
- Explore design of physical systems, to support controller development, more...
- Simscape is widely used for many applications
- Resources are available to help you get started and succeed

# MATLAB EXPO

Thank you



© 2022 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.