

Wentworth Institute of Technology Workshops

System Modeling, Control Design and Hardware Integration with Simulink

Siddharth Jawahar
MathWorks
January 14, 2026

Agenda

- Understanding engineered systems
- Introduction to Model-Based Design and Simulink
- Simulink 101 – An Industry Standard tool for Model-Based Design
- Resources

To follow along

- Set up a MathWorks account if you don't have one
 - please use **Google Chrome** browser 
 - go to <https://www.mathworks.com/mwaccount/>
 - Then open up MATLAB Online (<https://matlab.mathworks.com/>)
- Copy the exercise materials via GitHub
 - <https://github.com/SidJawMW/WIT---Simulink-MBD-workshop>



Agenda

- **Understanding engineered systems**
- Introduction to Model-Based Design and Simulink
- Simulink 101 – An Industry Standard tool for Model-Based Design
- Resources

Engineered systems require system-level, multi-domain integration



Aerospace and Defense



Automotive



Biological Sciences



Biotech and Pharmaceutical



Communications



Electronics



Energy Production



Financial Services



Industrial Machinery



Medical Devices



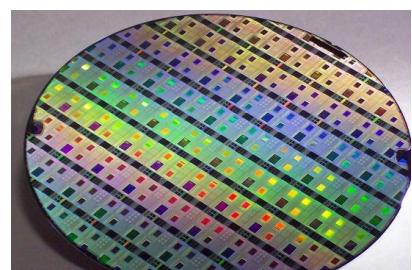
Metals, Materials, Mining



Neuroscience



Railway Systems



Semiconductors



Software and Internet

Engineered systems require system-level, multi-domain integration

Renewable
Energy



Aerial
Robotics



Wireless
Communication

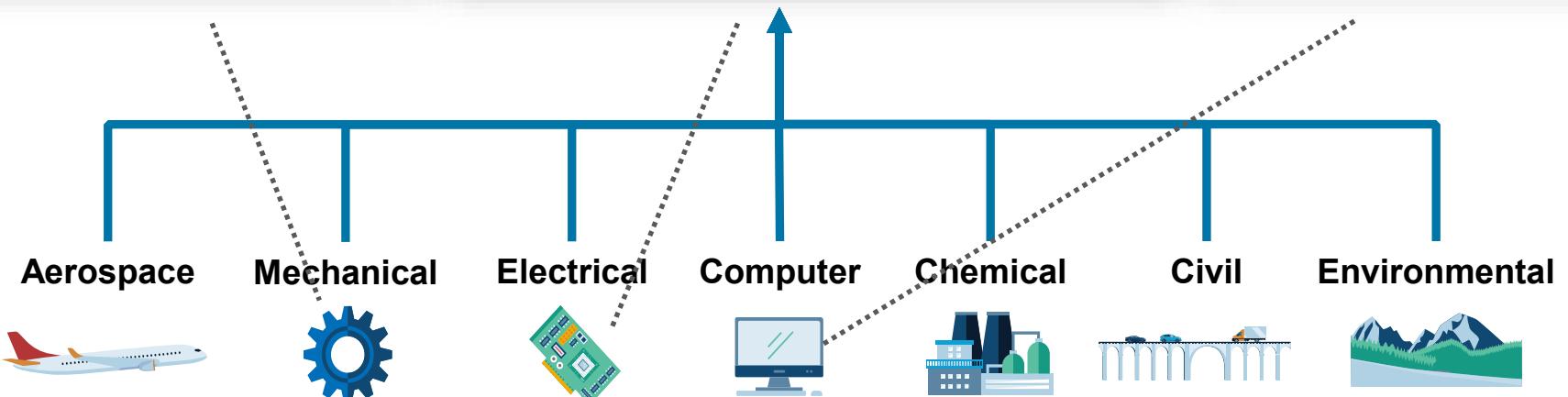
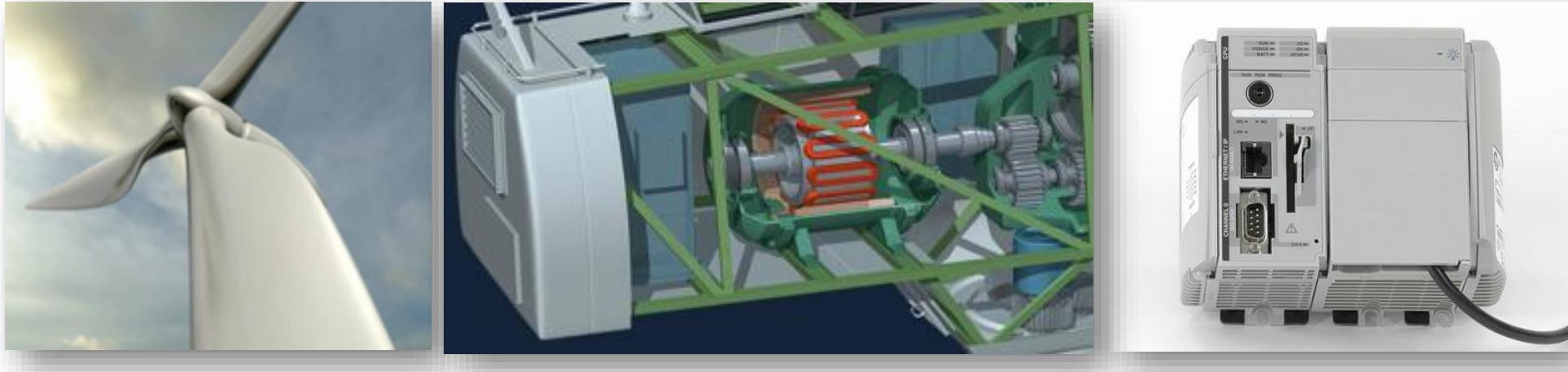


Electric
Vehicles



Engineered systems require system-level, multi-domain integration

Renewable Energy



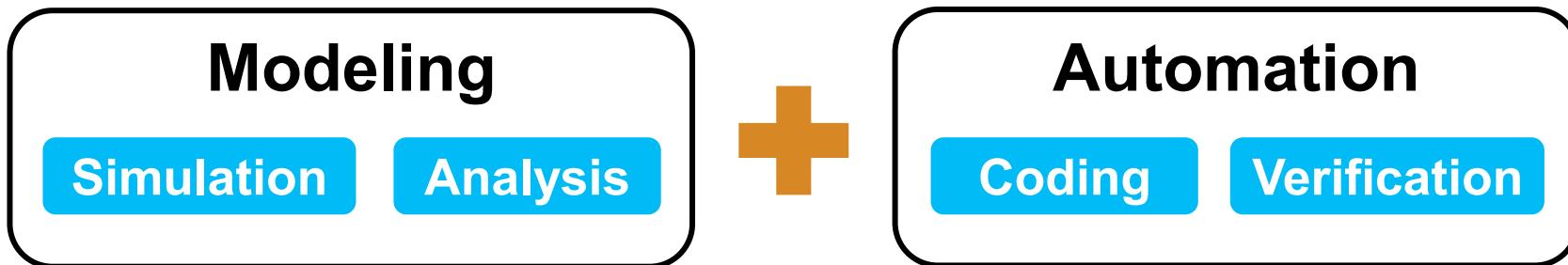
Model-Based Design can help!

Agenda

- Understanding engineered systems
- **Introduction to Model-Based Design and Simulink**
- Simulink 101 – An Industry Standard tool for Model-Based Design
- Resources

Model-Based Design

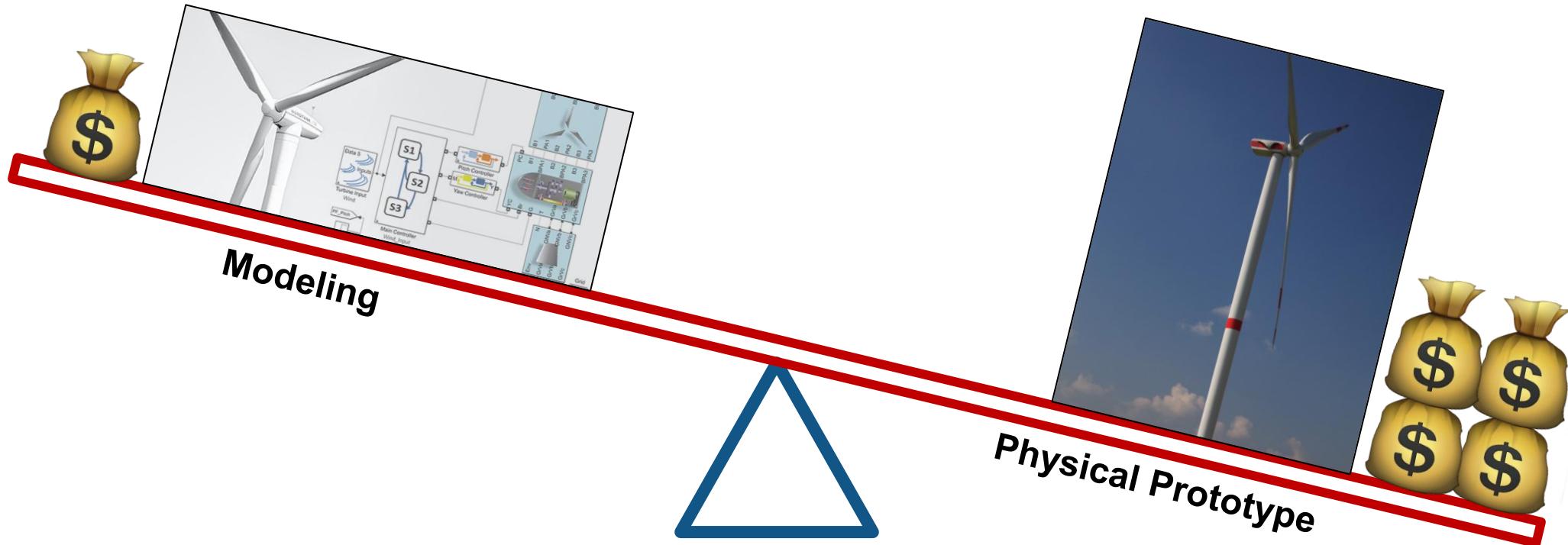
Systematic use of models throughout the development process



Try out new ideas
Fast repeatable tests

Eliminate manual steps
and human error

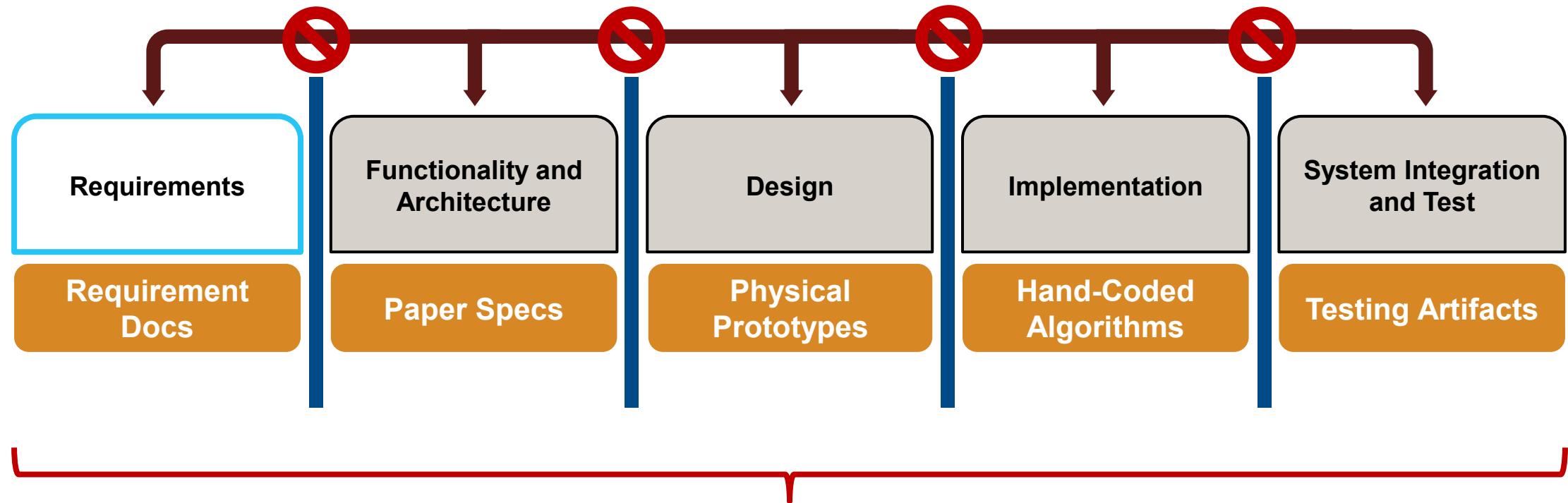
Physical prototypes are costly and iterate slowly



Design space **exploration**
Continuous design **improvement**

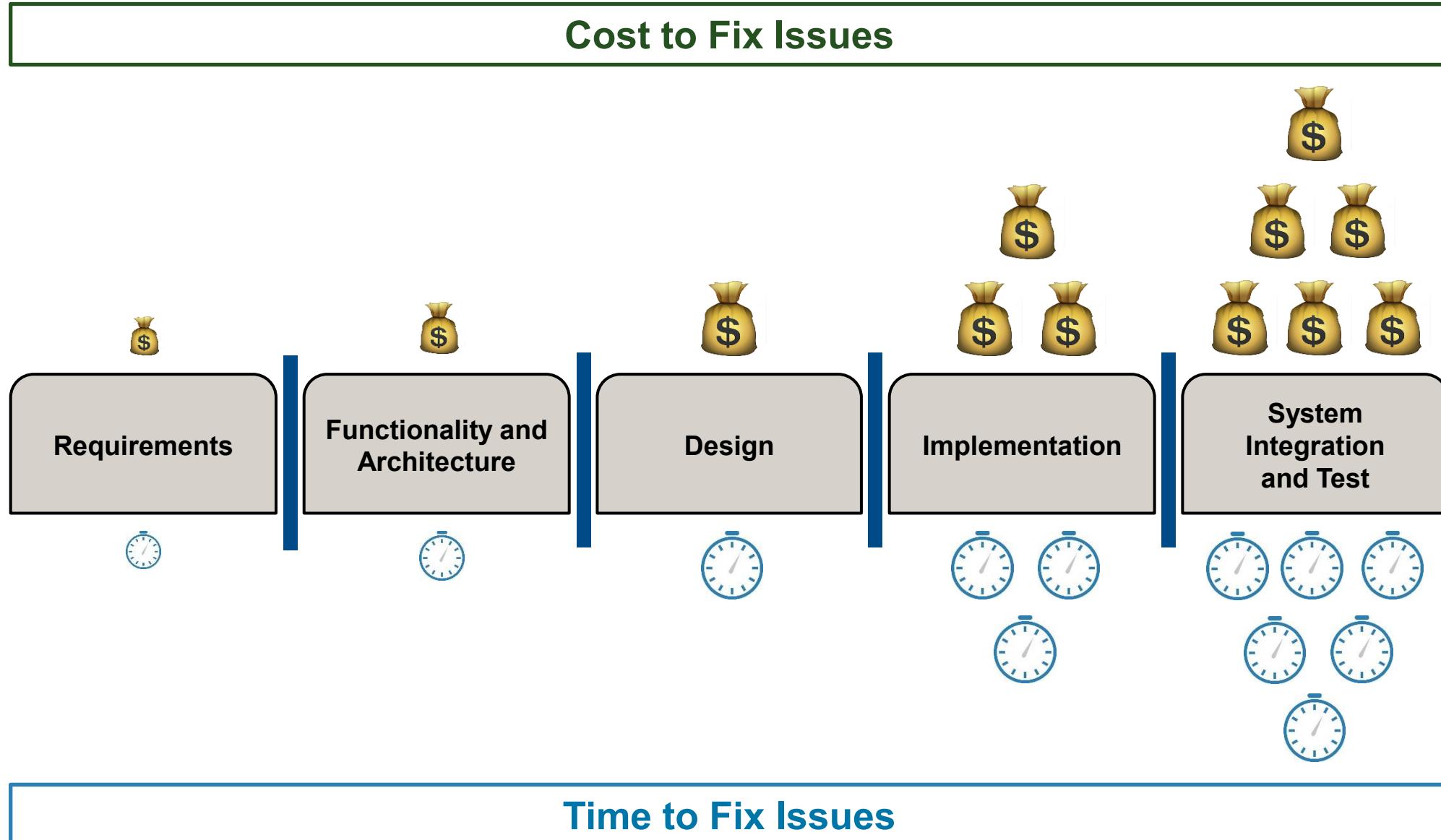
Costly and **time-consuming** to build
Hinders rapid iterations

Requirements and artifacts are **hard to manage, change, and trace**



Manual steps introduce errors and slow down the development process

Issues found late in the process are more **costly** and **time-consuming** to fix



Why Model-Based Design?

Systematic use of **models** throughout the development process

Modeling



Automation

Why Model-Based Design?

Systematic use of **models** throughout the development process

Modeling

Simulation

Analysis



Automation



Why Model-Based Design?

Systematic use of **models** throughout the development process

Modeling

Simulation

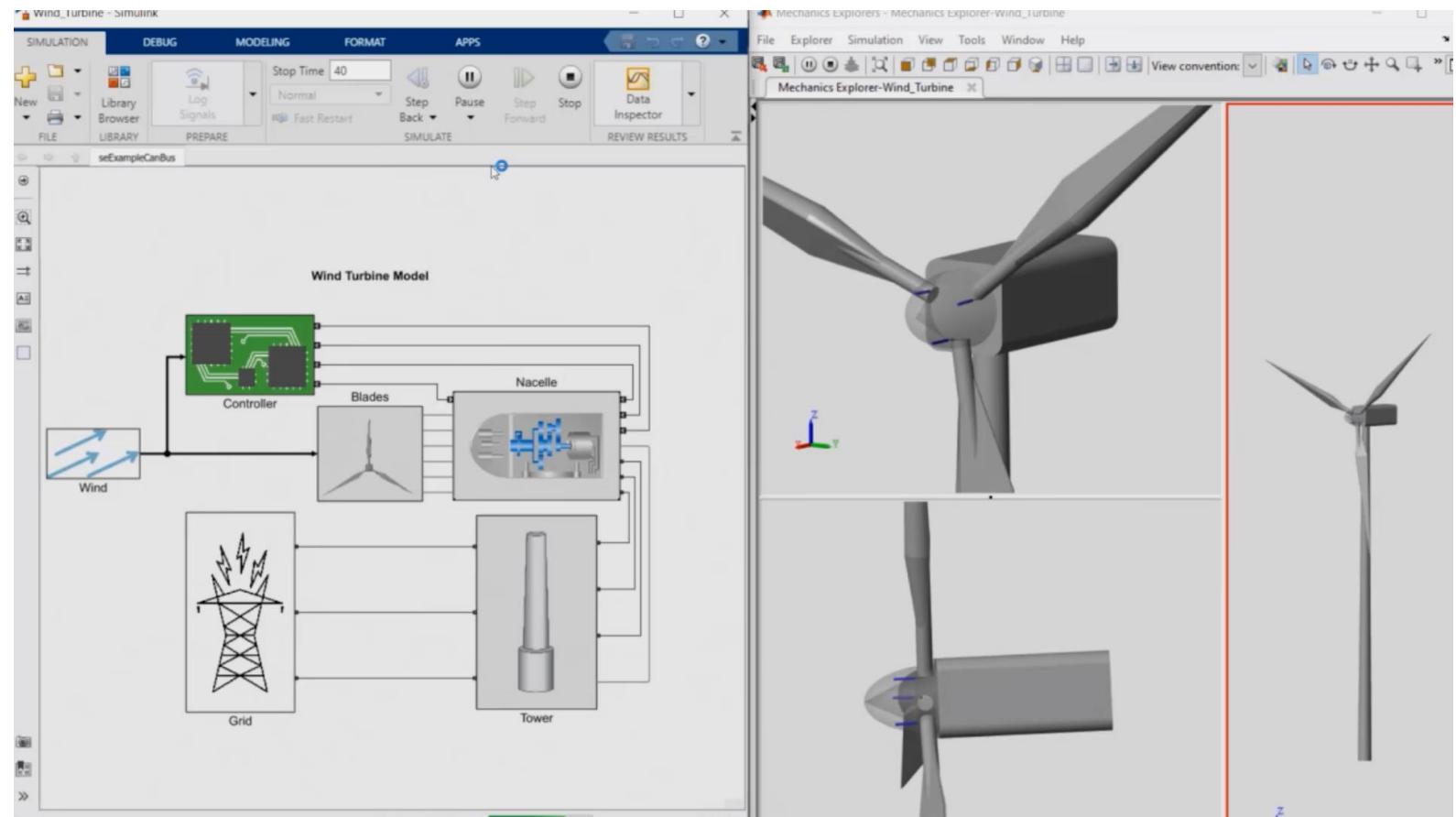
Analysis



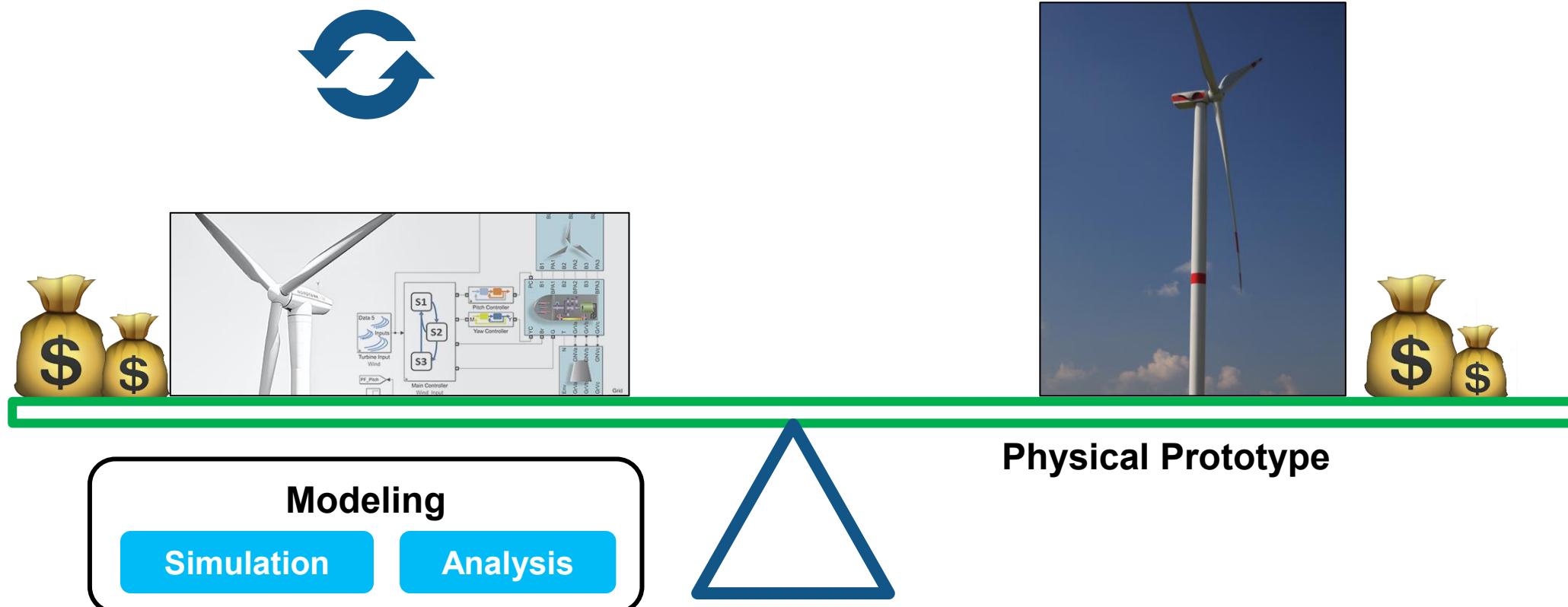
Automation

Coding

Verification



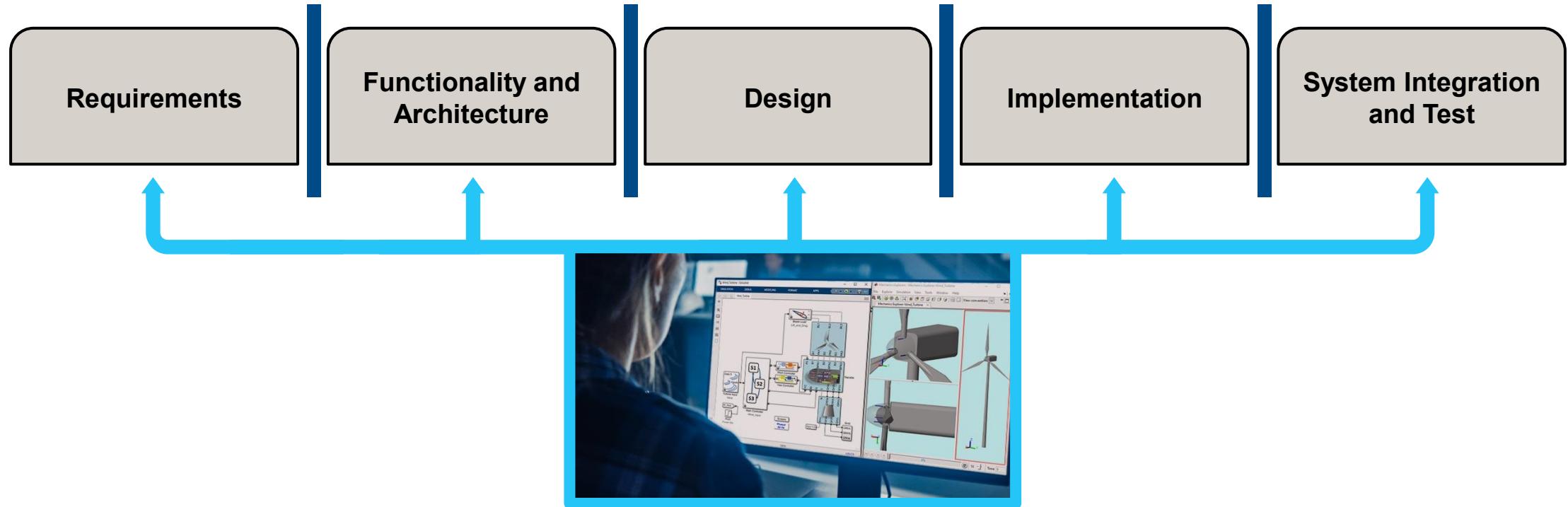
Modeling and simulation help cut costs and speed up design iterations



Rapid prototyping and testing

Requirements **capture** and artifact traceability throughout the process

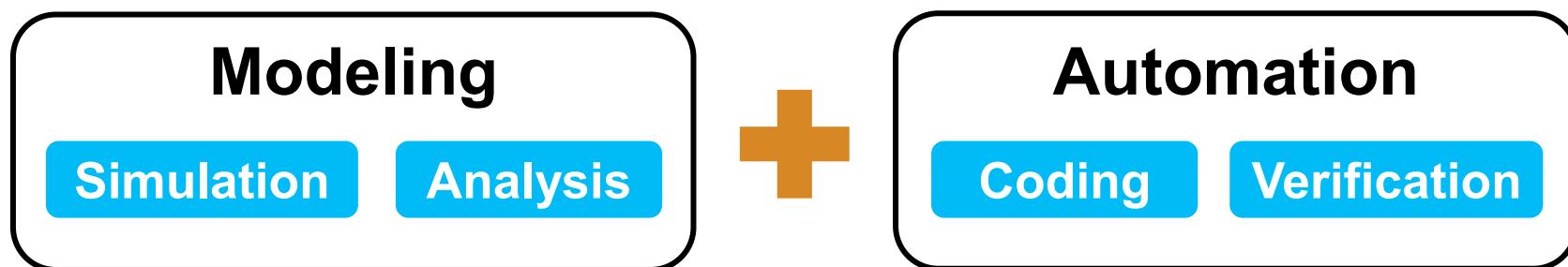
Model-Based Design



Models are at the **center** of your development process
Create a **digital thread**

Model-Based Design

Systematic use of models throughout the development process



Short **agile**
iteration cycles



Saved time and
cost

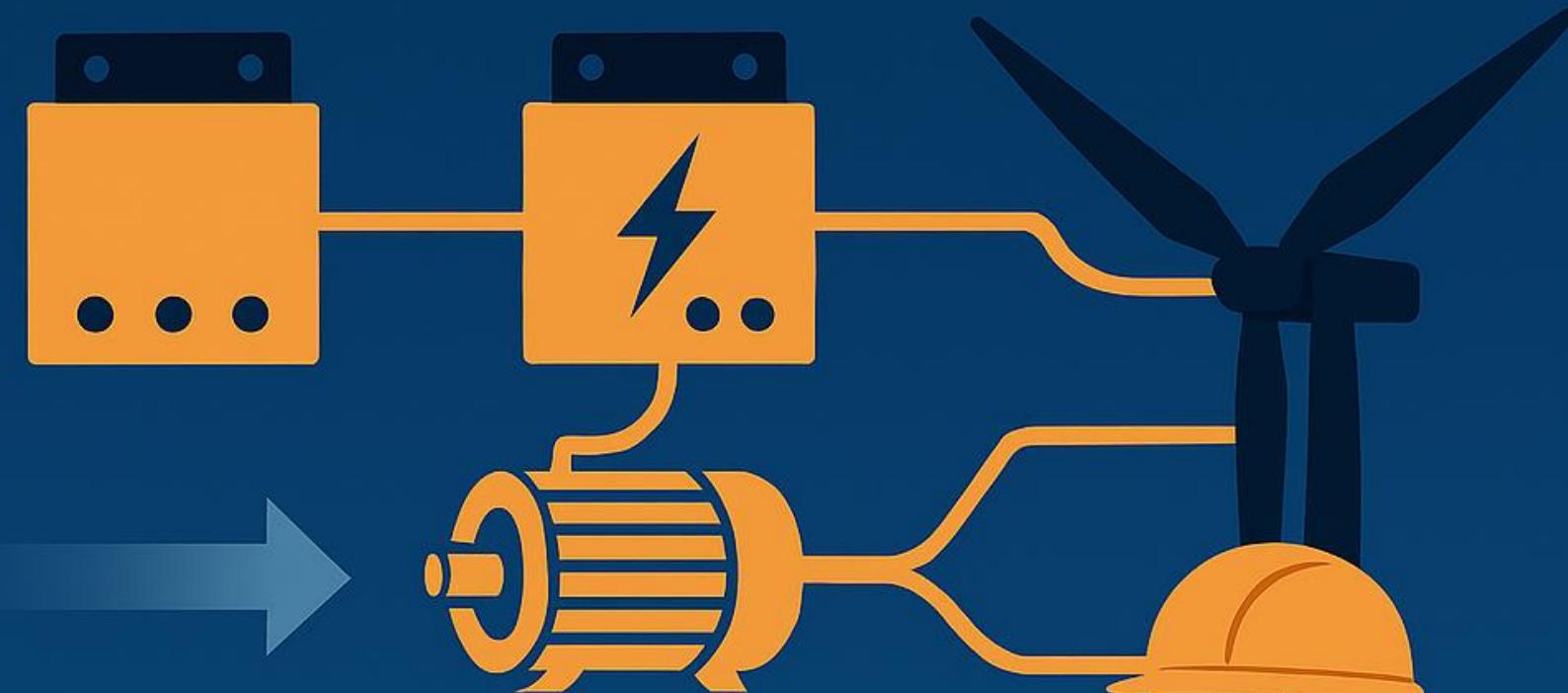


Minimal defects
and **high quality**

End to end advanced air mobility development at Supernal

*“Complete flight simulation
of autonomous vehicle
operation using
photorealistic environment”*





User story: ETH Zurich SAE competition team uses Model-Based Design to develop motor controllers

AMZ Racing – Zurich, Switzerland
Formula SAE Electric

AMZ Racing Designed the Motor Controller to Achieve 0 to 100 km/h in 0.956 Seconds

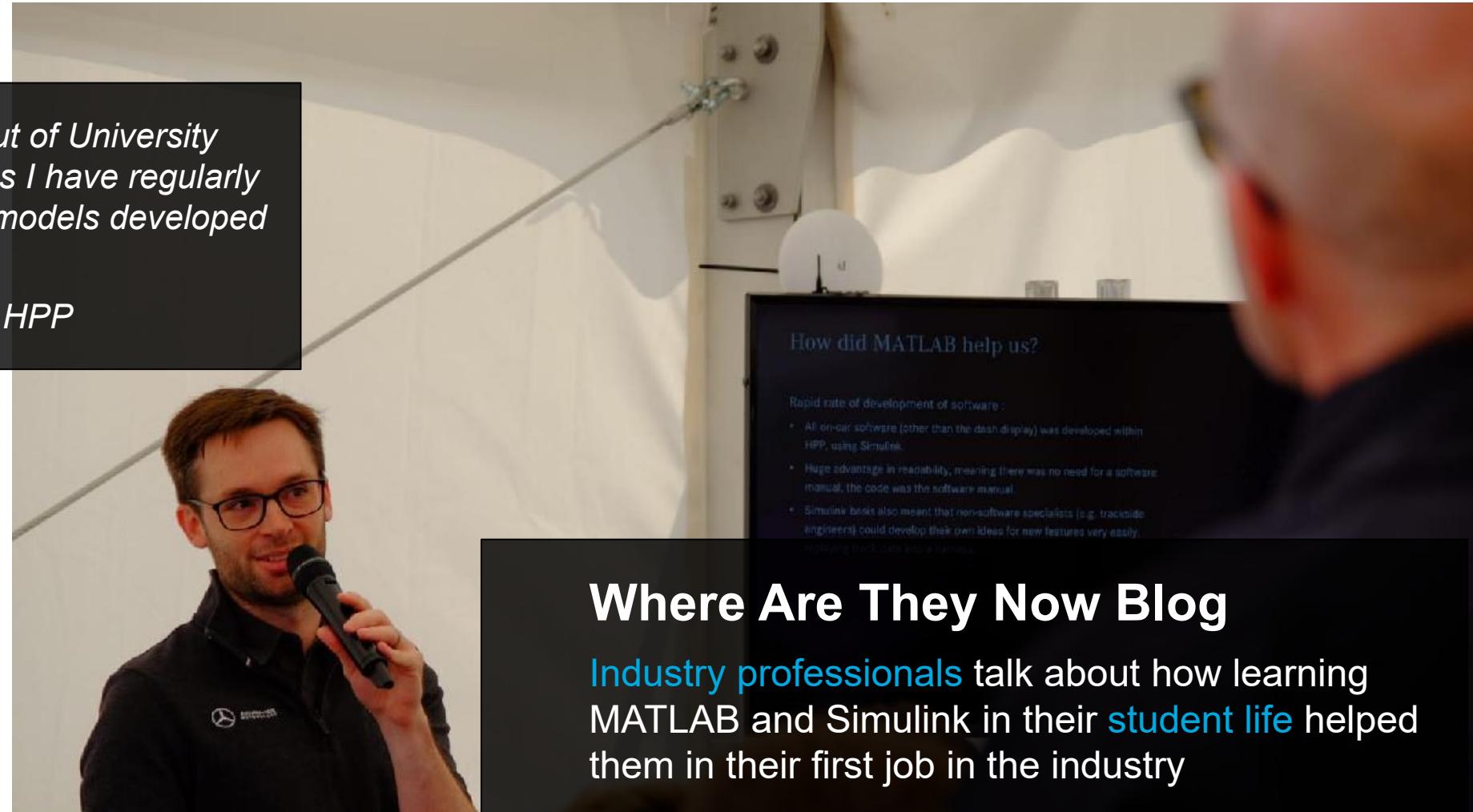
Use Simulink for system simulations including, FOC Motor control model, physical model and **closed loop controller** to optimize parameters. **Generated HDL code to run motor controllers on FPGAs.**



Such project-based learning outcomes using Model-Based Design prepare students for industry

"Simulink knowledge coming out of University has also been very beneficial as I have regularly interacted with large Software models developed using this platform."

- Graham Iles, Mercedes AMG HPP



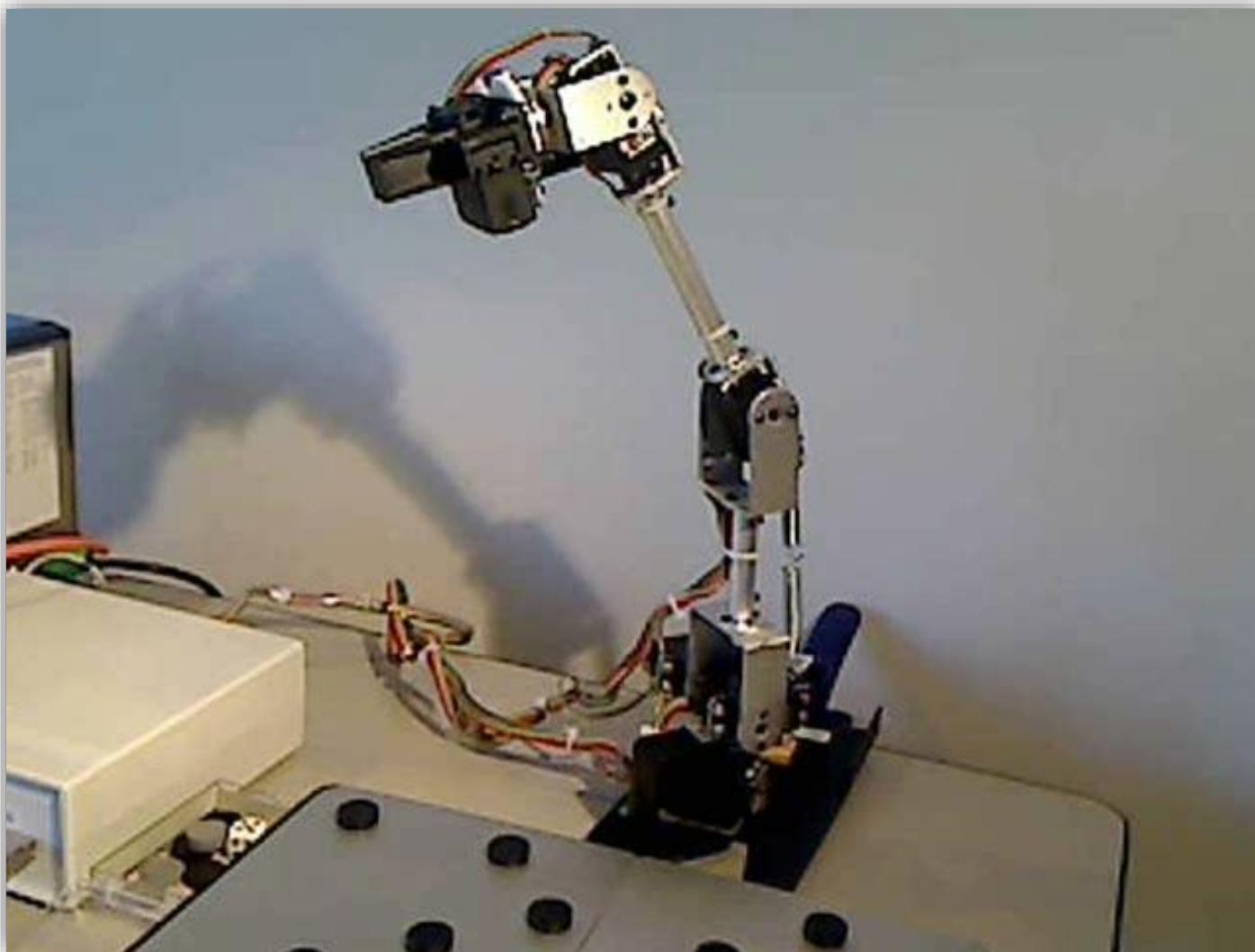
Where Are They Now Blog

Industry professionals talk about how learning MATLAB and Simulink in their student life helped them in their first job in the industry

Agenda

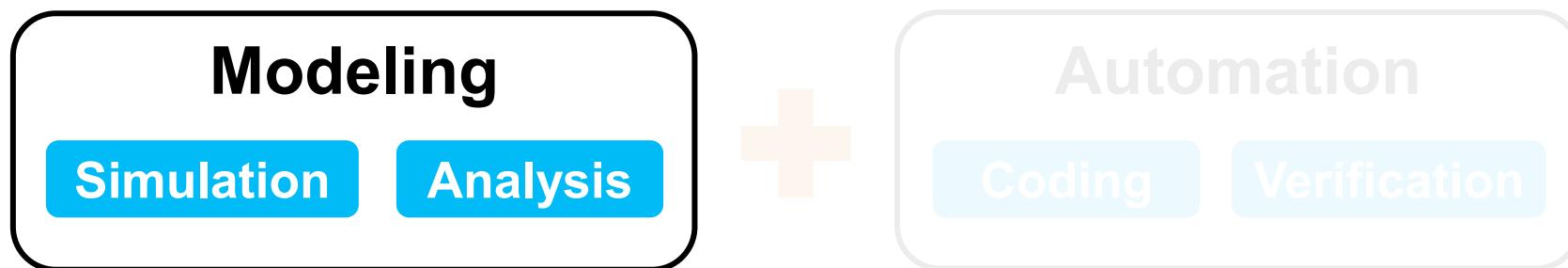
- Understanding engineered systems
- Introduction to Model-Based Design and Simulink
- **Simulink 101 – An Industry Standard tool for Model-Based Design**
- Resources

Let's look at a simple pick and place robot...

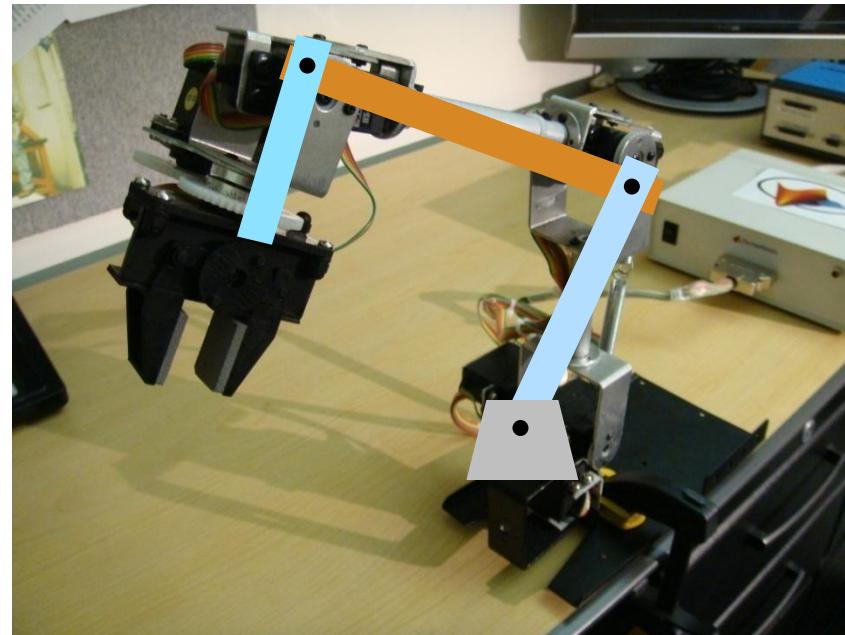


Model-Based Design

Systematic use of models throughout the development process



To model the system, let's break it down into its subcomponents

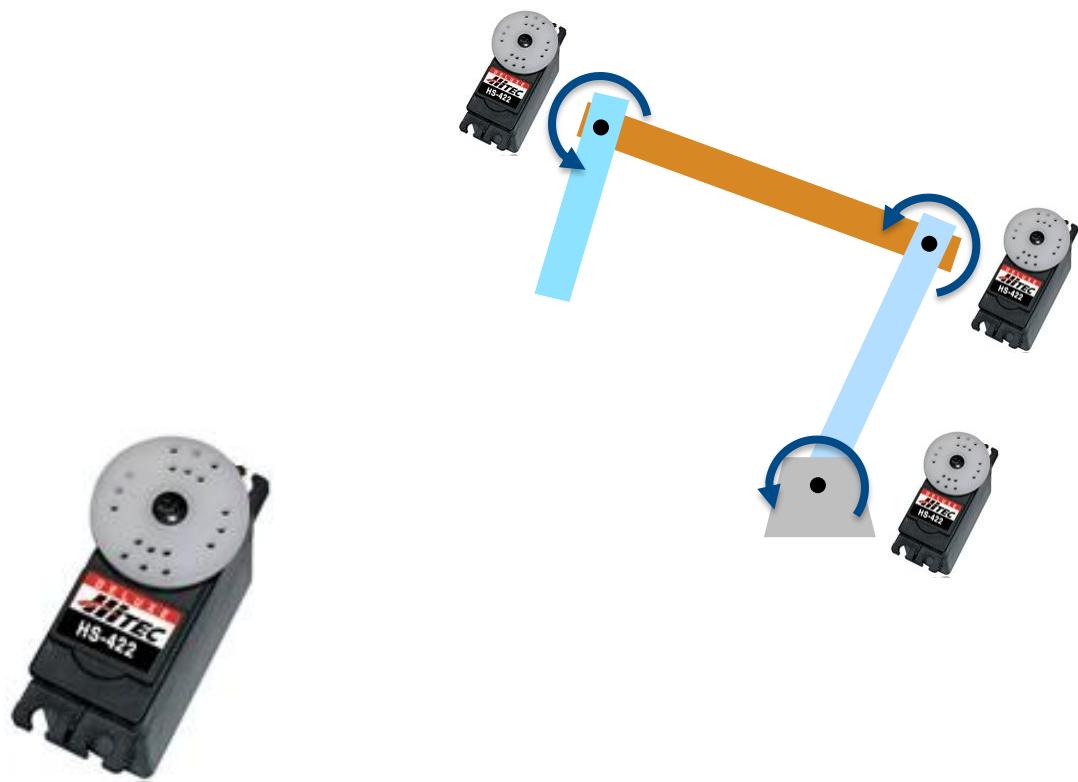


Modeling

Simulation

Analysis

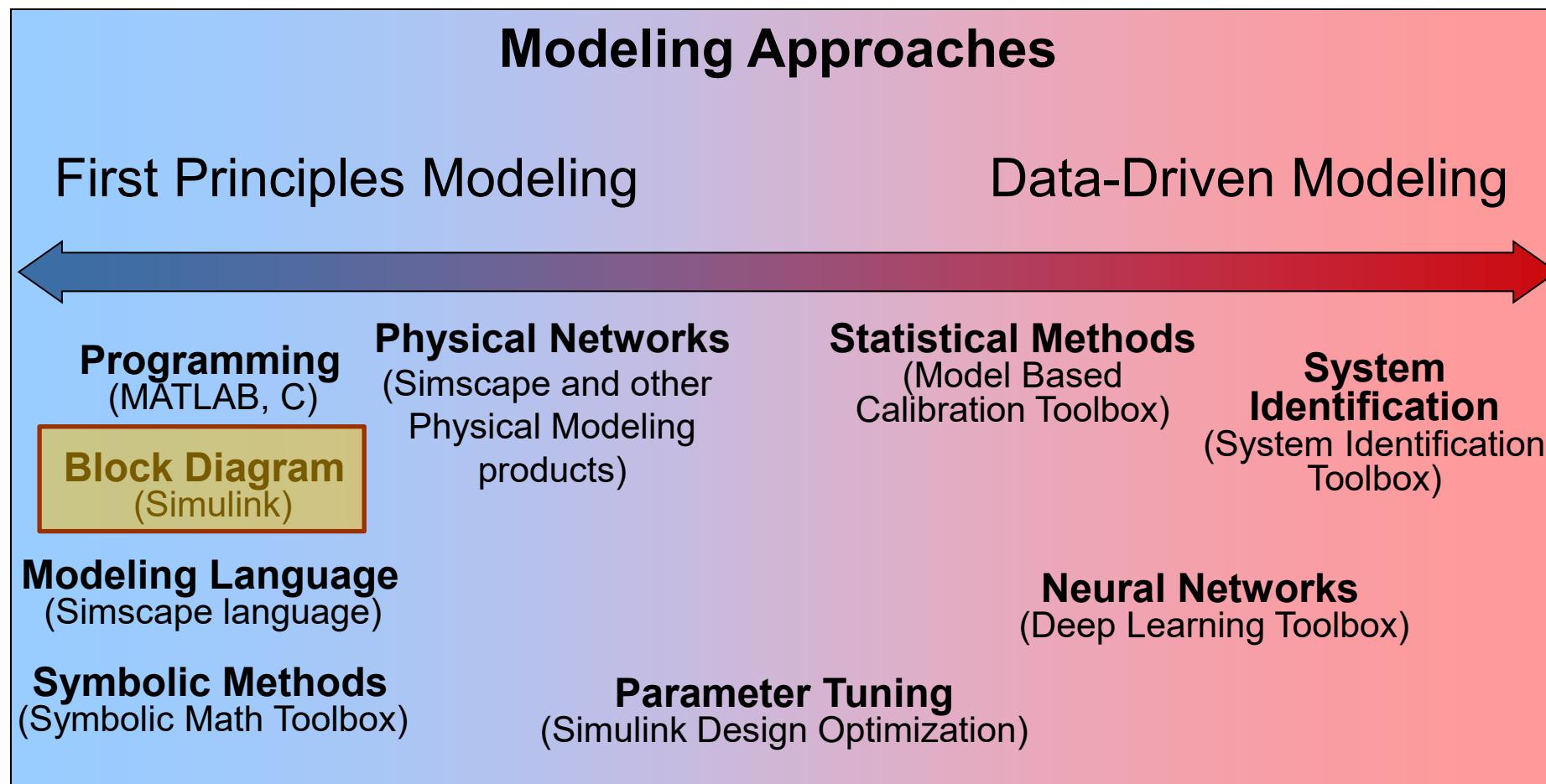
To model the system, let's break it down into its subcomponents



Let's model the DC motor!



There are several approach to modeling dynamic systems



Modeling

Simulation

Analysis

We know the governing equations for a DC motor



$$V = K\omega + iR + L \frac{di}{dt}$$

$$\frac{di}{dt} = \frac{1}{L} (V - K\omega - iR)$$

$$i = \int \frac{1}{L} (V - K\omega - iR) dt$$

Electrical

$$J \frac{d\omega}{dt} = Ki - b\omega - T_{Load}$$

$$\frac{d\omega}{dt} = \frac{1}{J} (Ki - b\omega - T_{Load})$$

$$\omega = \int \frac{1}{J} (Ki - b\omega - T_{Load}) dt$$

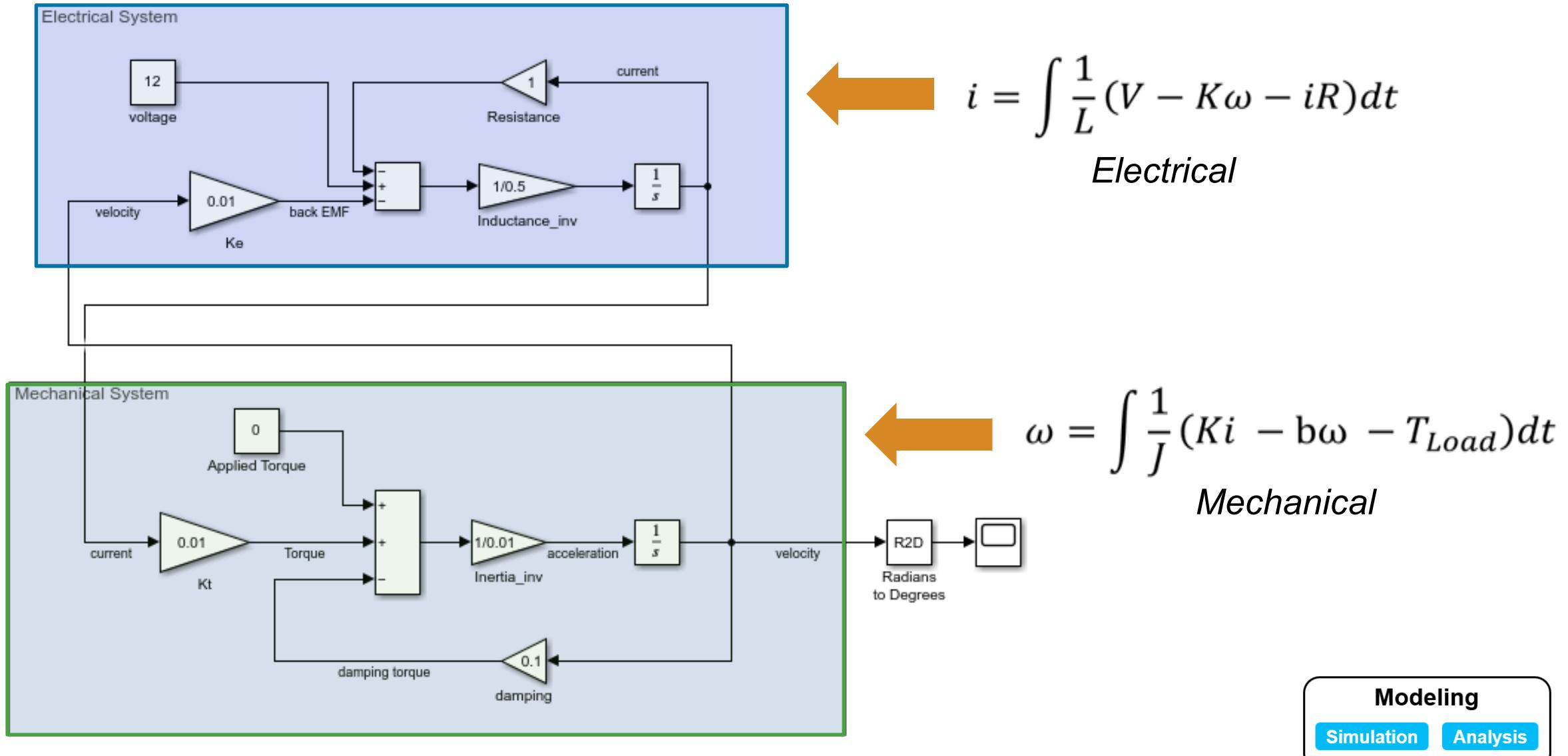
Mechanical

Modeling

Simulation

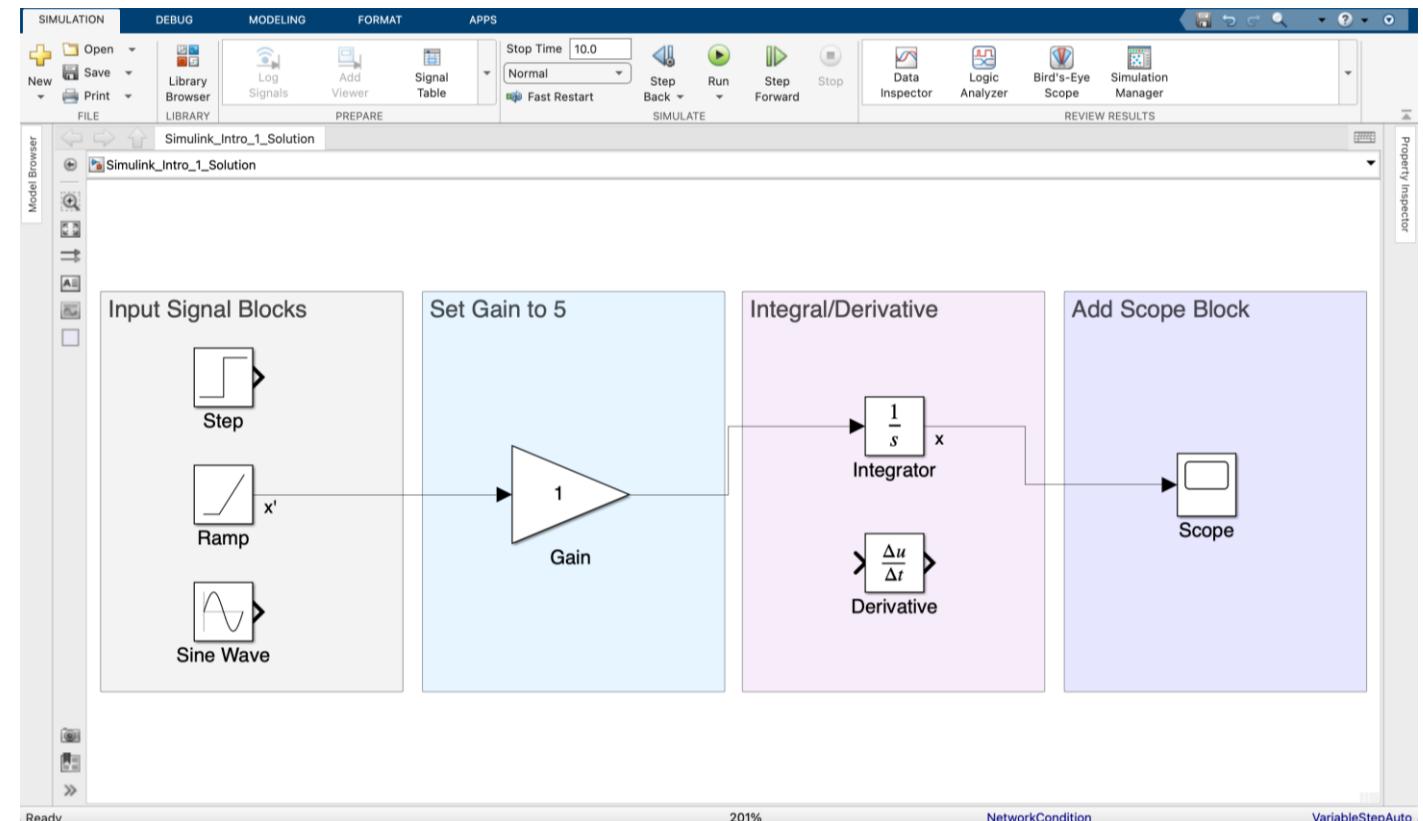
Analysis

Let's model these equations in Simulink



Exercise 1: Introduction to Simulink

- Open:
Ex1_Simulink_Intro.slx
- Connect Input blocks to Gain and Integral/Derivative blocks
- Add a Scope block and connect signals to view them
- Adjust Stop Time to change simulation duration



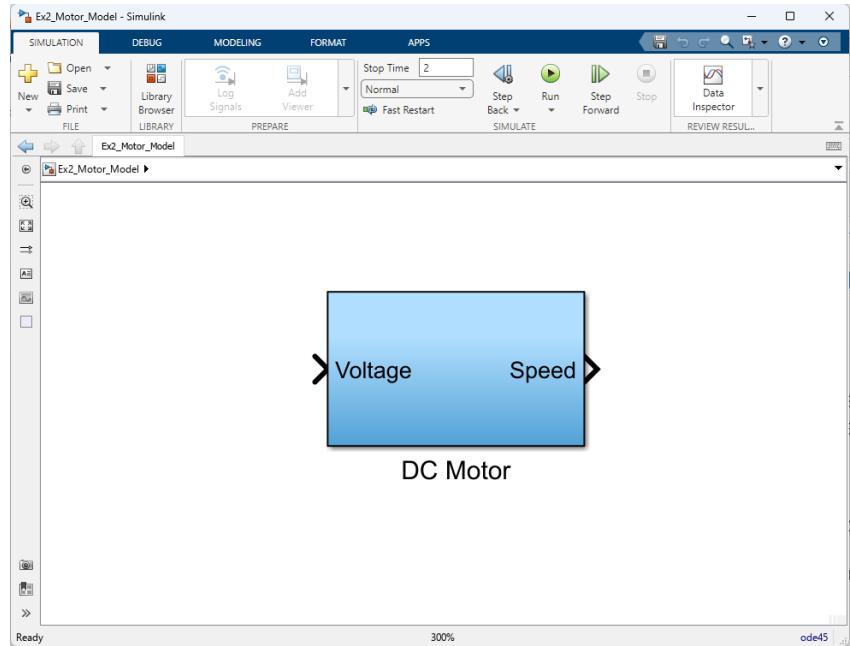
Modeling

Simulation

Analysis

Exercise 2: Motor Control Design with Simulink

- Open: **Ex2_Motor_Model.slx**
- Simulate the model with a reference step input
- Construct a PID controller with base Simulink blocks
- Add tuning knobs from dashboard library to tune gains
- Use scope blocks to see motor output, reference input, controller output
- Tune the controller for these specifications
 - **Overshoot <15%**
 - **Rise time < 0.25 sec**
 - **Steady state error is zero**
 - **Settling time <1.5 sec**
- How about including real world entities: delays, sensor noise, controller saturation...



CL RESPONSE	RISE TIME	OVERSHOOT	SETTLING TIME	S-S ERROR
Kp	Decrease	Increase	Small Change	Decrease
Ki	Decrease	Increase	Increase	Decrease
Kd	Small Change	Decrease	Decrease	No Change

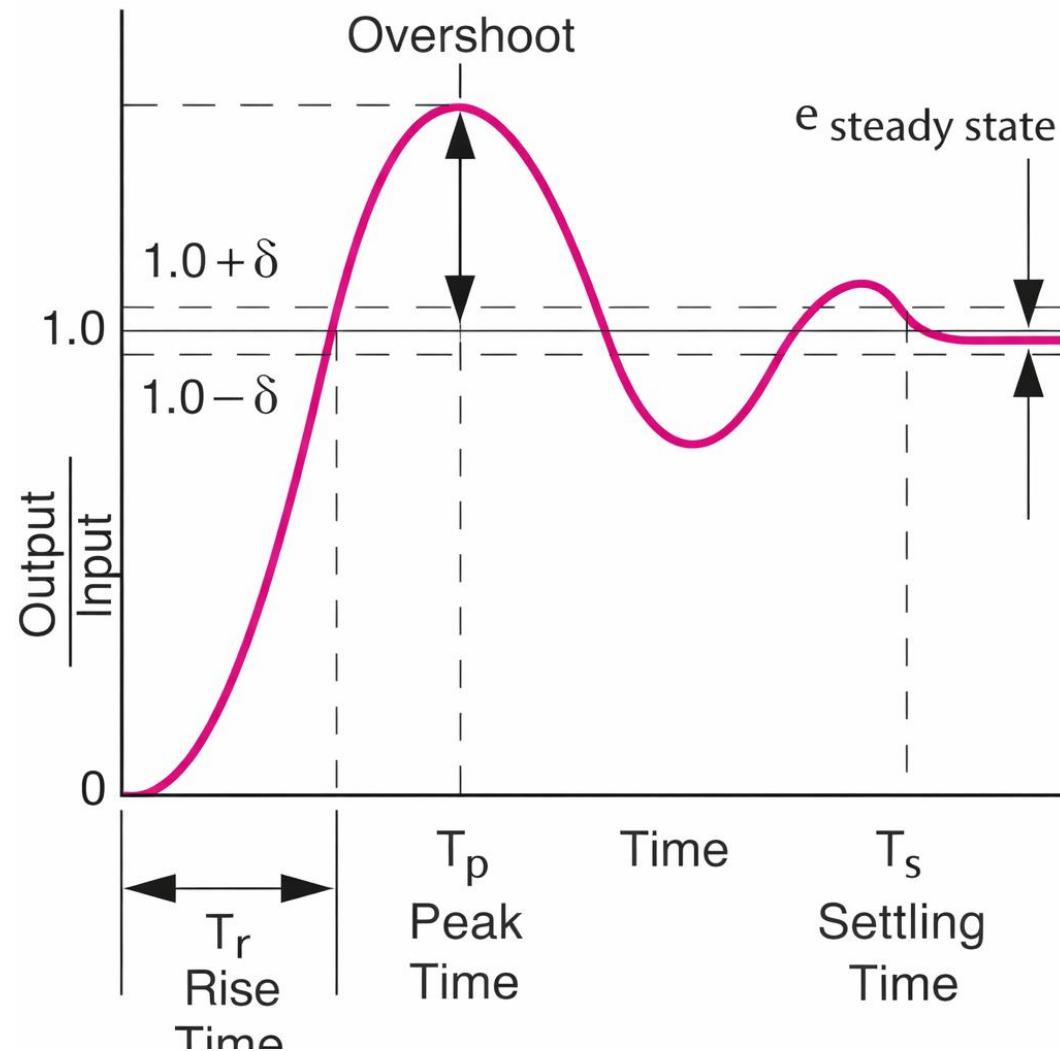
PID tuning cheat sheet

Modeling

Simulation

Analysis

Step response characteristics



Controls tutorials with MATLAB and Simulink

The screenshot shows the homepage of the Control Tutorials for MATLAB & Simulink (CTMS). The top navigation bar includes links for MICHIGAN, Carnegie Mellon, DETROIT MERCY, TIPS, ABOUT, BASICS, HARDWARE, INDEX, and NEXT ▶. Below the navigation is a banner with icons for various control systems: INTRODUCTION, CRUISE CONTROL, MOTOR SPEED, MOTOR POSITION, SUSPENSION, INVERTED PENDULUM, AIRCRAFT PITCH, and BALL & BEAM.

The main content area features a grid of images representing different control applications, such as a robotic arm, a car speedometer, a car driving on a road, gears, a suspension system, an inverted pendulum, an aircraft, and a ball on a beam. To the left of the grid, there are vertical columns of text labels corresponding to the images:

- SYSTEM**: INTRODUCTION, CRUISE CONTROL, MOTOR SPEED, MOTOR POSITION, SUSPENSION, INVERTED PENDULUM, AIRCRAFT PITCH, BALL & BEAM.
- MODELING**: MODELING
- ANALYSIS**: ANALYSIS
- CONTROL**: CONTROL
- PID**: PID
- ROOT LOCUS**: ROOT LOCUS
- FREQUENCY**: FREQUENCY
- STATE-SPACE**: STATE-SPACE
- DIGITAL**: DIGITAL
- SIMULINK**: SIMULINK
- MODELING**: MODELING
- CONTROL**: CONTROL
- SIMSCAPE**: SIMSCAPE

A central text block welcomes users to the CTMS, stating: "Welcome to the Control Tutorials for MATLAB and Simulink (CTMS): They are designed to help you learn how to use MATLAB and Simulink for the analysis and design of automatic control systems. They cover the basics of MATLAB and Simulink and introduce the most common classical and modern control design techniques."

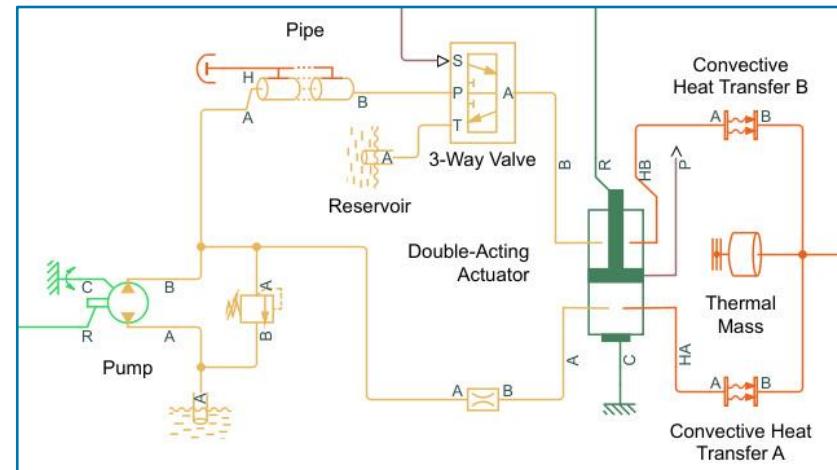
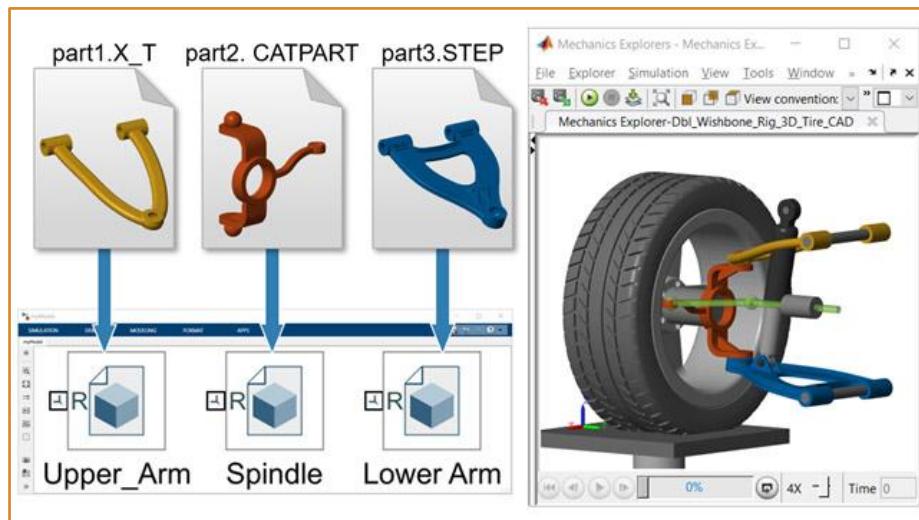
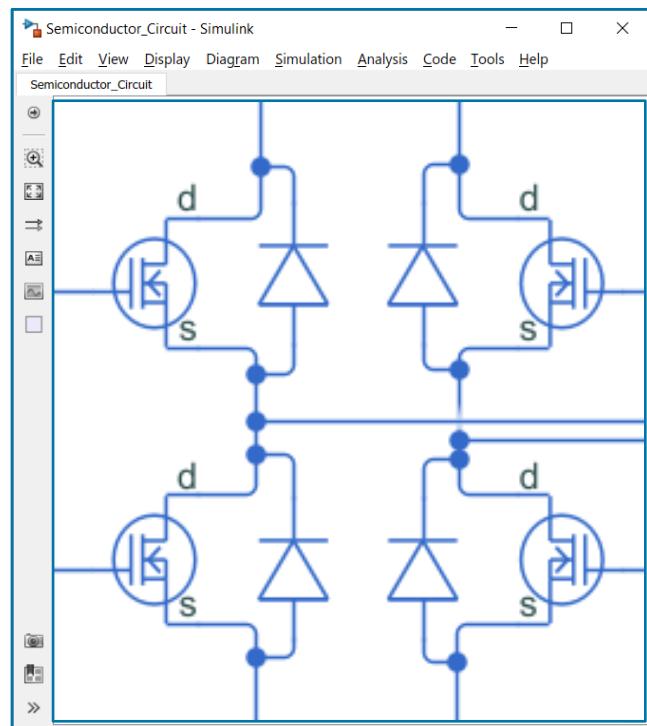
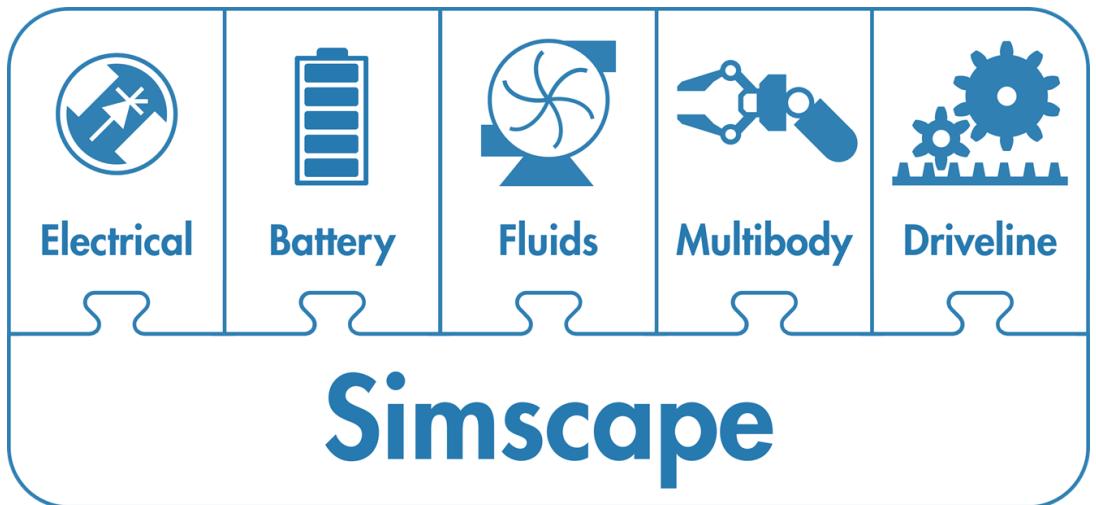
Below this, another text block about the authors notes: "About the Authors: These tutorials were originally developed by Prof. Dawn Tilbury at the University of Michigan and Prof. Bill Messner at Carnegie Mellon with funding from NSF. With further support from the MathWorks in 2011 and 2017, Prof. Messner, Prof. Rick Hill (Detroit Mercy), and PhD Student JD Taylor (CMU), expanded the tutorials, completely redesigned the web interface, and

At the bottom, a footer states: "All contents licensed under a Creative Commons Attribution-ShareAlike 4.0 International License." and includes three small icons.

Simscape

Overview

- Simscape enables you to rapidly create models of physical systems within the Simulink environment



MATLAB, Simulink, and Simscape for Physical Modeling

Below is a simple script using MATLAB code to simulate the dynamics of a spring-mass-damper system.

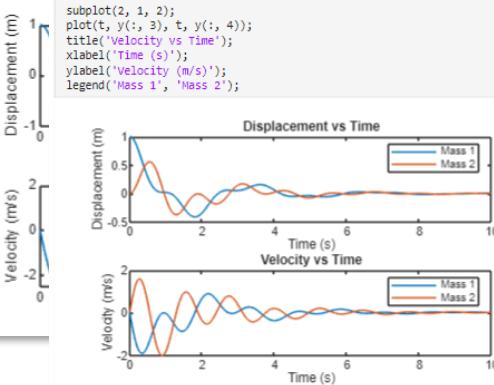
```
% Parameters
m = 1; % mass /kg
k = 10;
b = 1;

% Initial c
x0 = 1;
v0 = 0;

% Time span
tspan = [0
```

```
% ODE funct
odefun = @(t, y)
    % ODE function
    dydt = zeros(2, 1);
    dydt(1) = y(2);
    dydt(2) = (-b1 * y(3) - k1 * (y(1) - y(2))) / m1;
    dydt(3) = y(4);
    dydt(4) = (-b2 * y(4) + k1 * (y(1) - y(2)) - k2 * y(2)) / m2;
    % solve ODE
[t, y] = ode45(odefun, tspan, [x1_0; x2_0; v1_0; v2_0]);
% Plot results
figure;
subplot(2, 1, 1);
plot(t, y(:, 1));
title('Displacement vs Time');
xlabel('Time (s)');
ylabel('Displacement (m)');
legend('Mass 1', 'Mass 2');

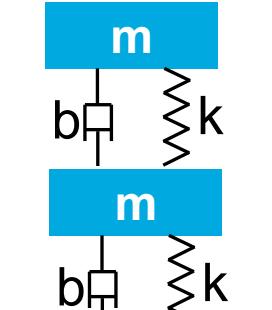
subplot(2, 1, 2);
plot(t, y(:, 2));
title('Velocity vs Time');
xlabel('Time (s)');
ylabel('Velocity (m/s)');
legend('Mass 1', 'Mass 2');
```



MATLAB

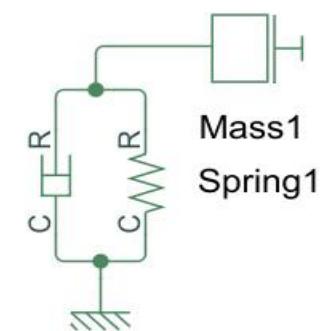
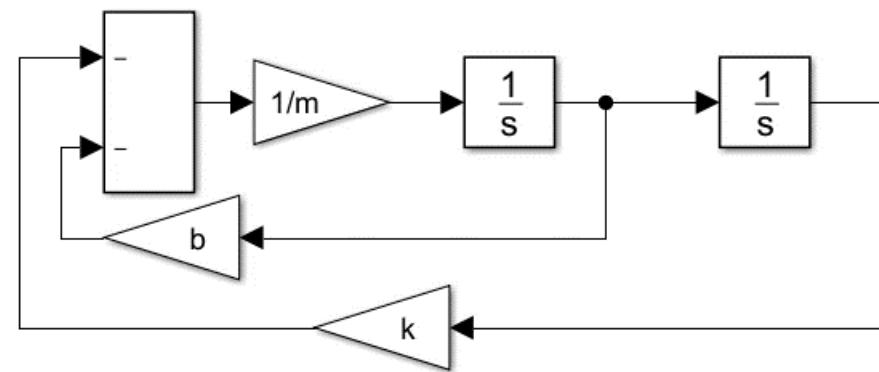
$$\begin{aligned} F_{\text{Spring}} &= k_{\text{Spring}} * (x_{\text{Mass}}) \\ F_{\text{Damper}} &= b_{\text{Damper}} * \left(\frac{dx_{\text{Mass}}}{dt} \right) \\ \frac{d^2x_{\text{Mass}}}{dt^2} &= \frac{-F_{\text{Spring}} - F_{\text{Damper}}}{m_{\text{Mass}}} \end{aligned}$$

$$\begin{aligned} F_{\text{Spring2}} &= k_{\text{Spring}} * (x_{\text{Mass}}) \\ \dots \end{aligned}$$



Simscape Blocks

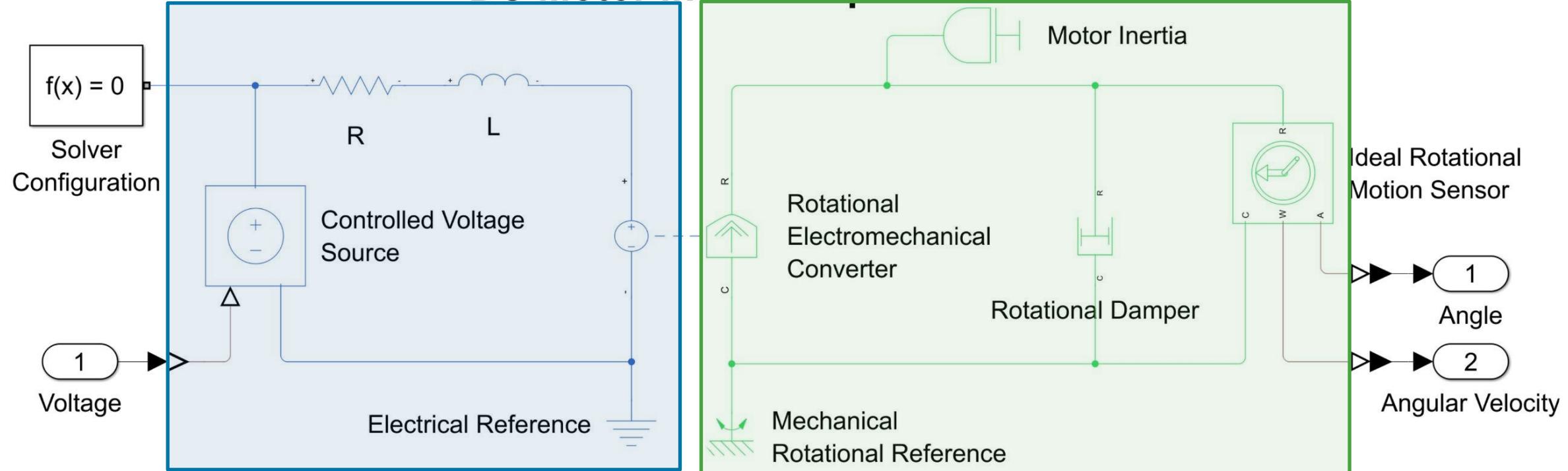
Input/Output Block Diagram



Simulink

Physical Network Model of a DC Motor using Simscape

DC Motor in Simscape



Electrical

Mechanical

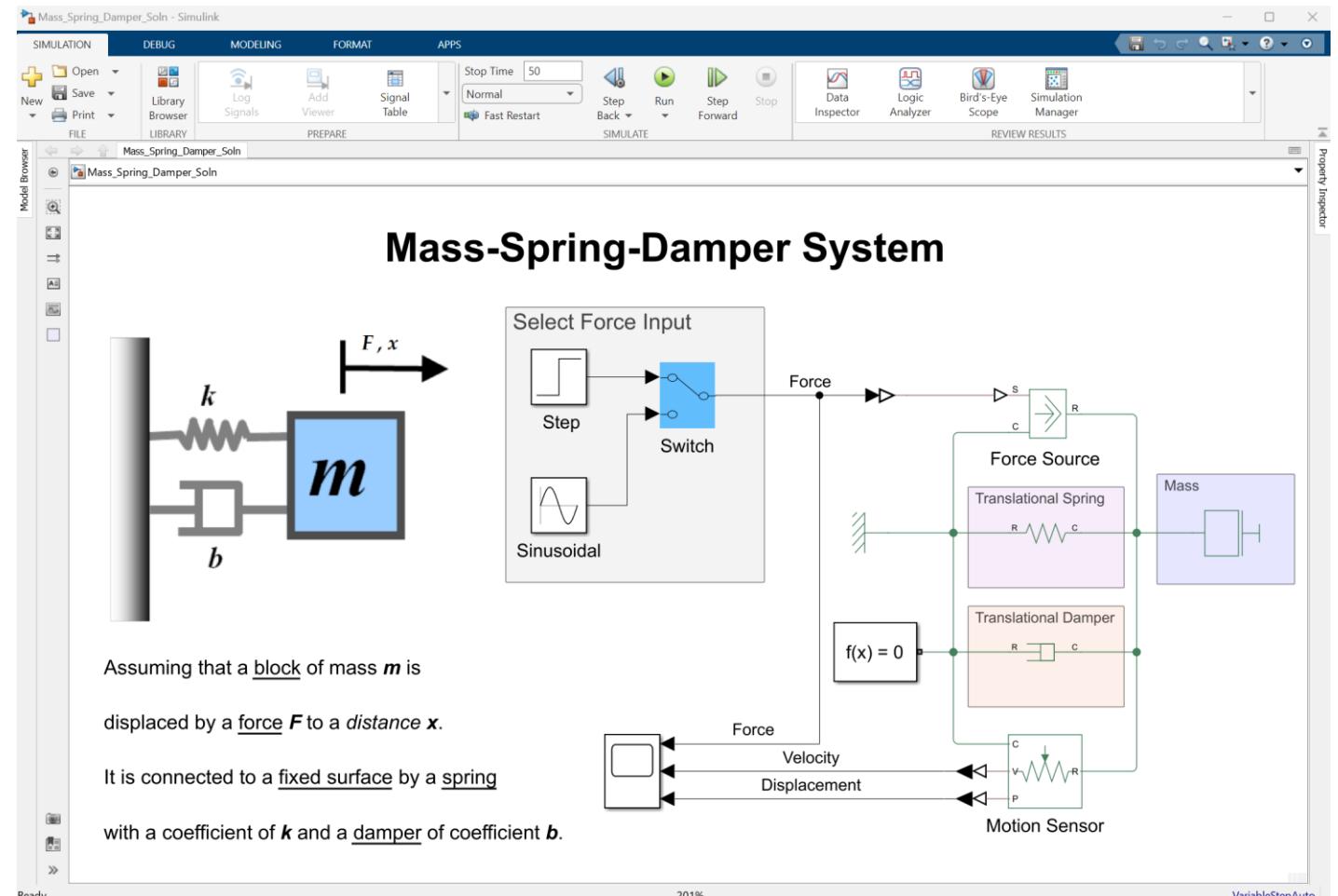
Modeling

Simulation

Analysis

Exercise 3: Mass-Spring Damper

- Open:
Ex3_Mass_Spring_Damper.slx
- Add Translational Spring,
Translational Damper, and Mass
blocks
- Edit block parameters and view
results in Scope block



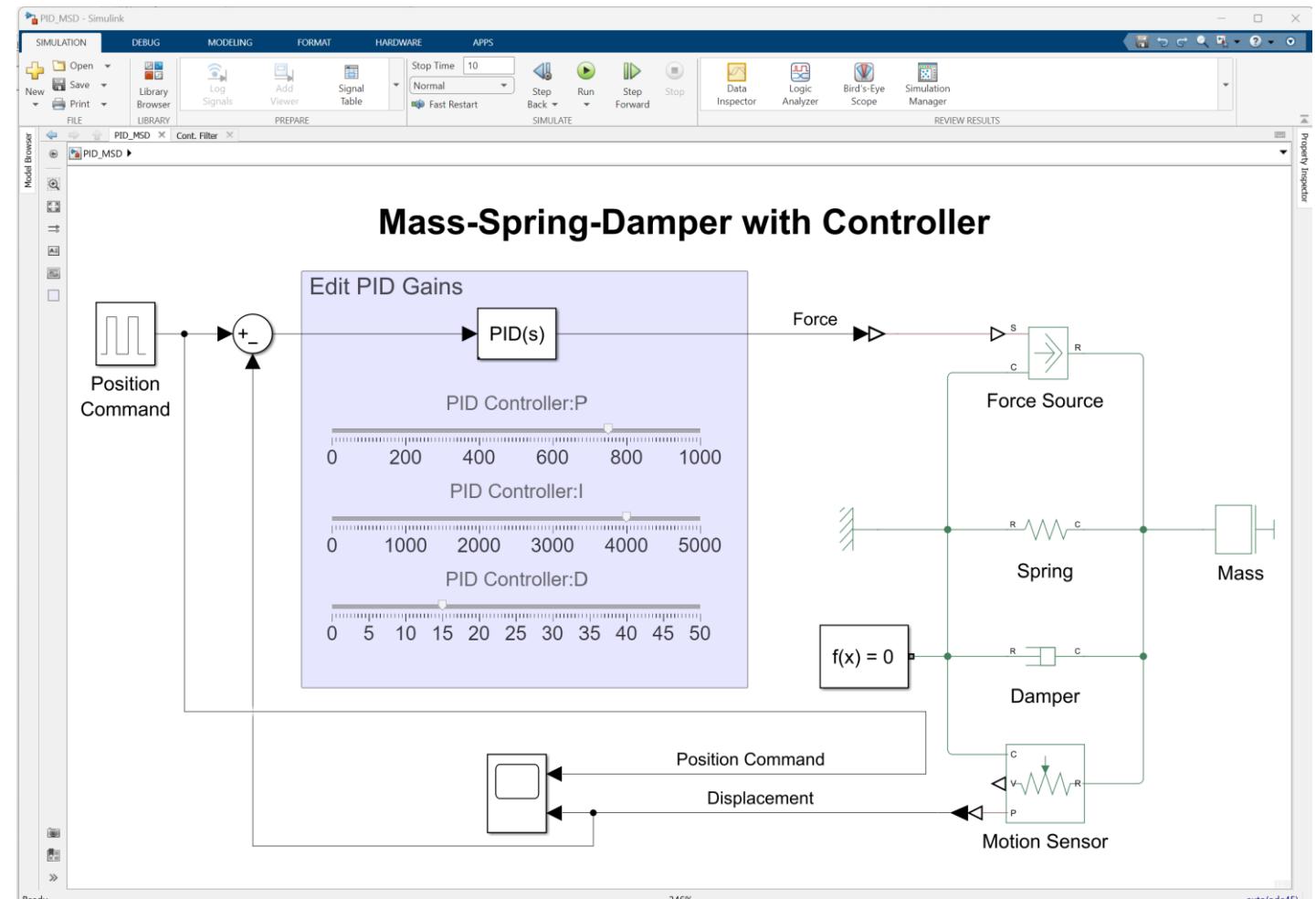
Modeling

Simulation

Analysis

Exercise 4: Mass Spring Damper with Controller

- Open: **Ex4_PID_MSD.slx**
- Edit desired position commands or change input block
- Use sliders to adjust PID gains
- View results in Scope block



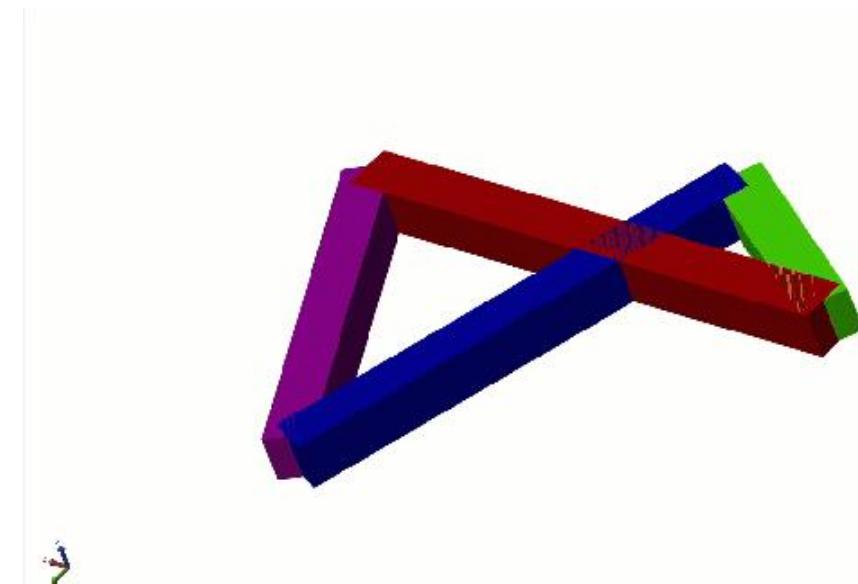
Modeling

Simulation

Analysis

Exercise 5: Modeling a linkage with Simscape Multibody

- Open a Simscape Multibody model in Simulink
- Set up 4 interconnected linkages (fixed, input, coupler, output) with the following measurements
 - Fixed – [12 1 1] cm
 - Input – [5 1 1] cm
 - Coupler– [8 1 1] cm
 - Output– [10 1 1] cm
- Connect them using revolute joints
- Set velocity target of 120 deg/sec for the input linkage.
- Simulate the model!

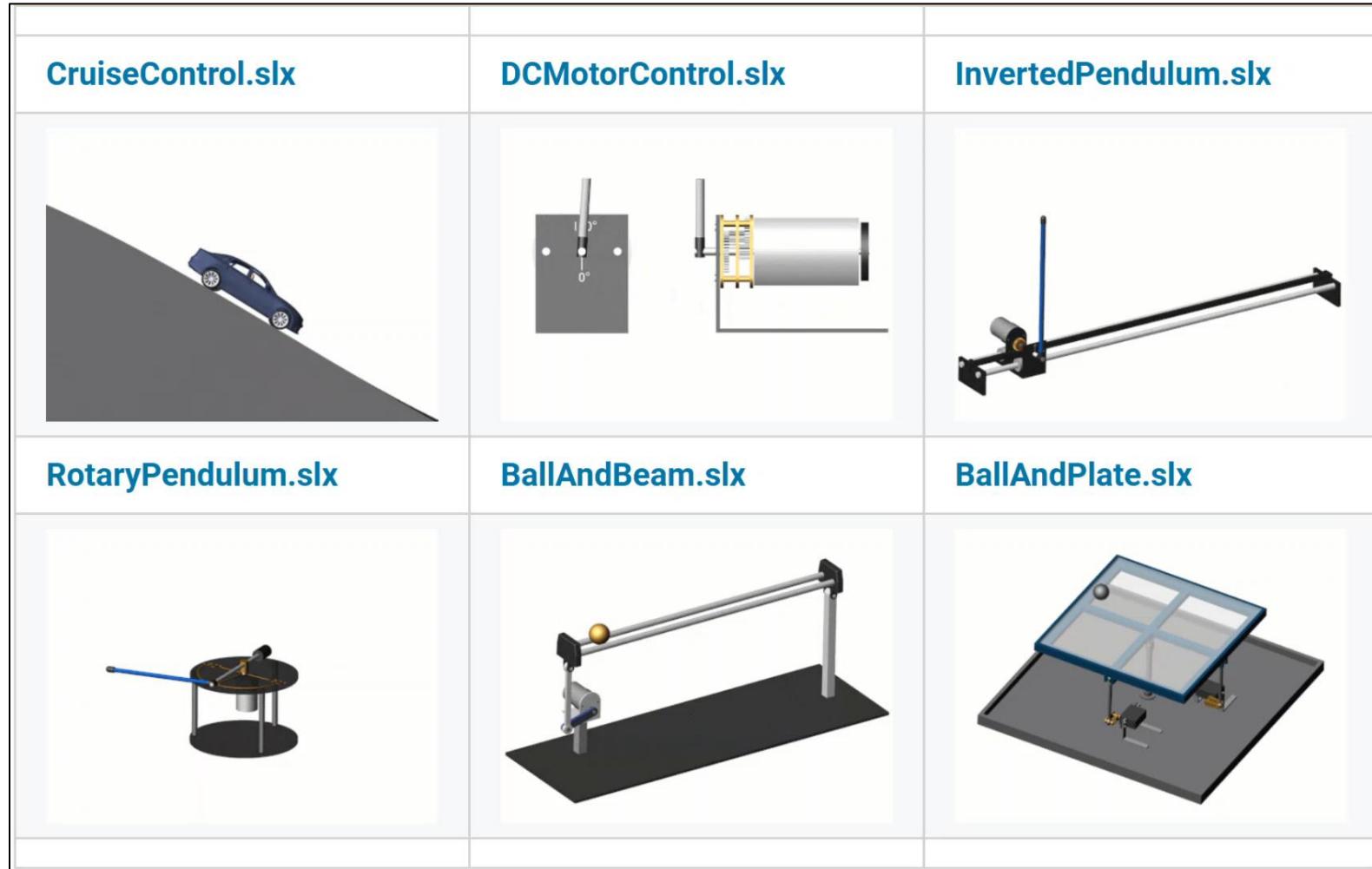


Modeling

Simulation

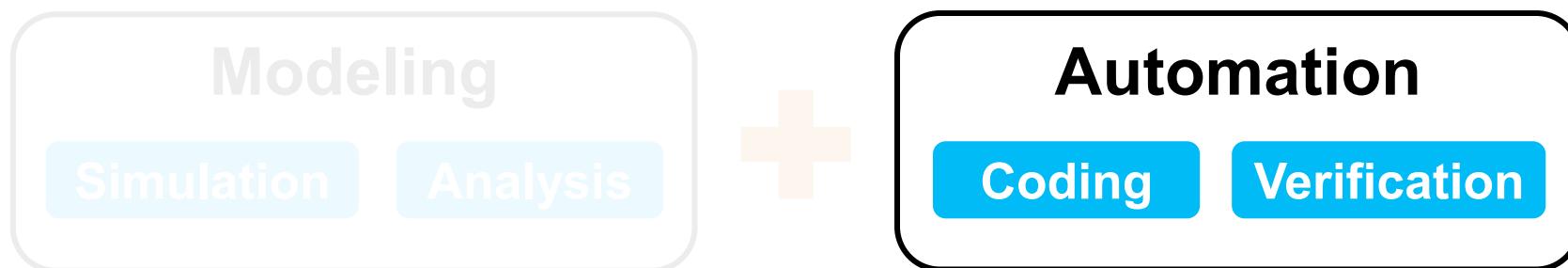
Analysis

Virtual hardware labs for control design



Model-Based Design

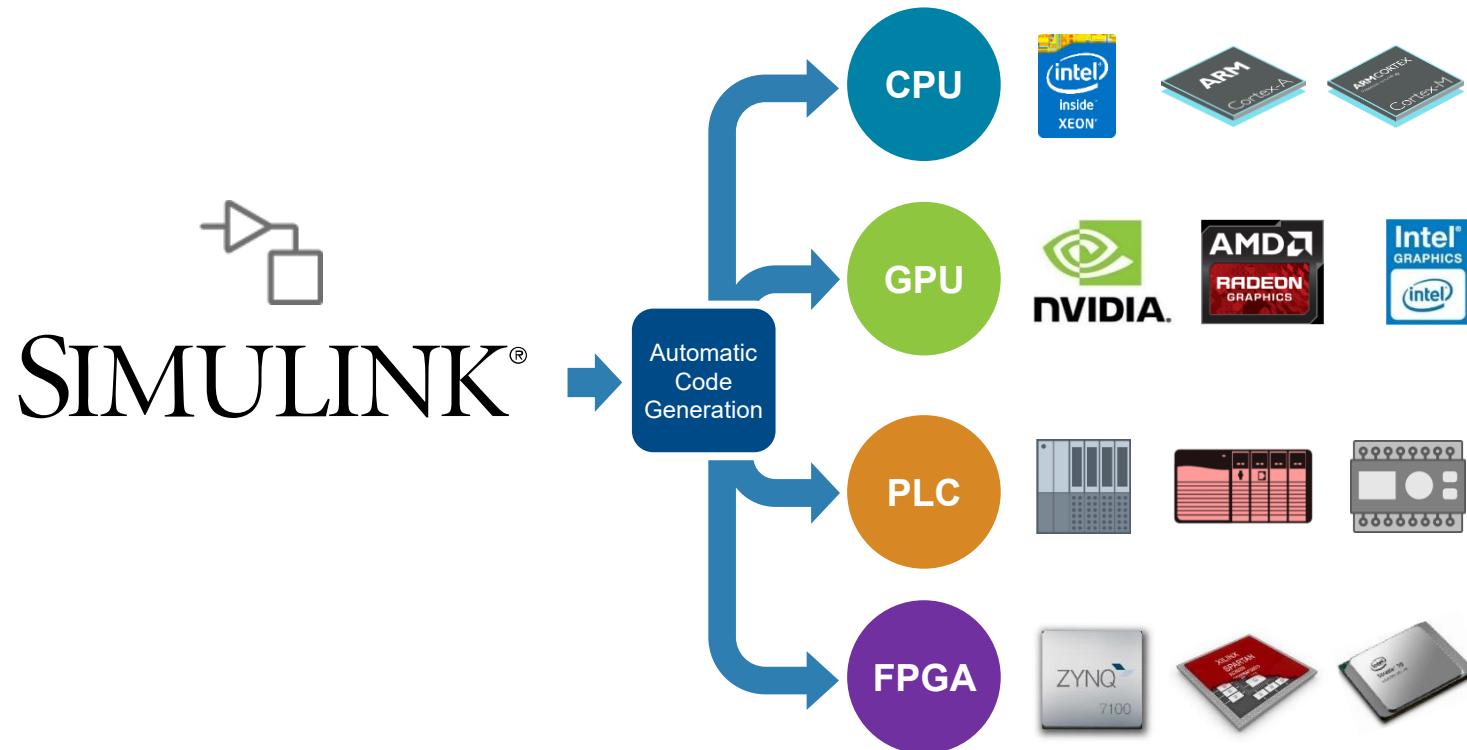
Systematic use of models throughout the development process



Automatically generate code for prototyping and production

Automation

Coding Verification



Reliable and **high performance**, with **flexible** choice of targets

MathWorks supports a variety of low-cost hardware



Arduino



Lego EV3



Raspberry Pi



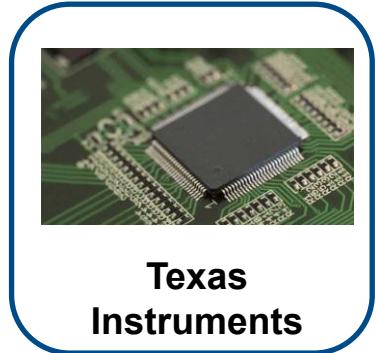
Android/iOS
Devices



Kinect for
Windows



BeagleBone
Black



Texas
Instruments



STM
Electronics

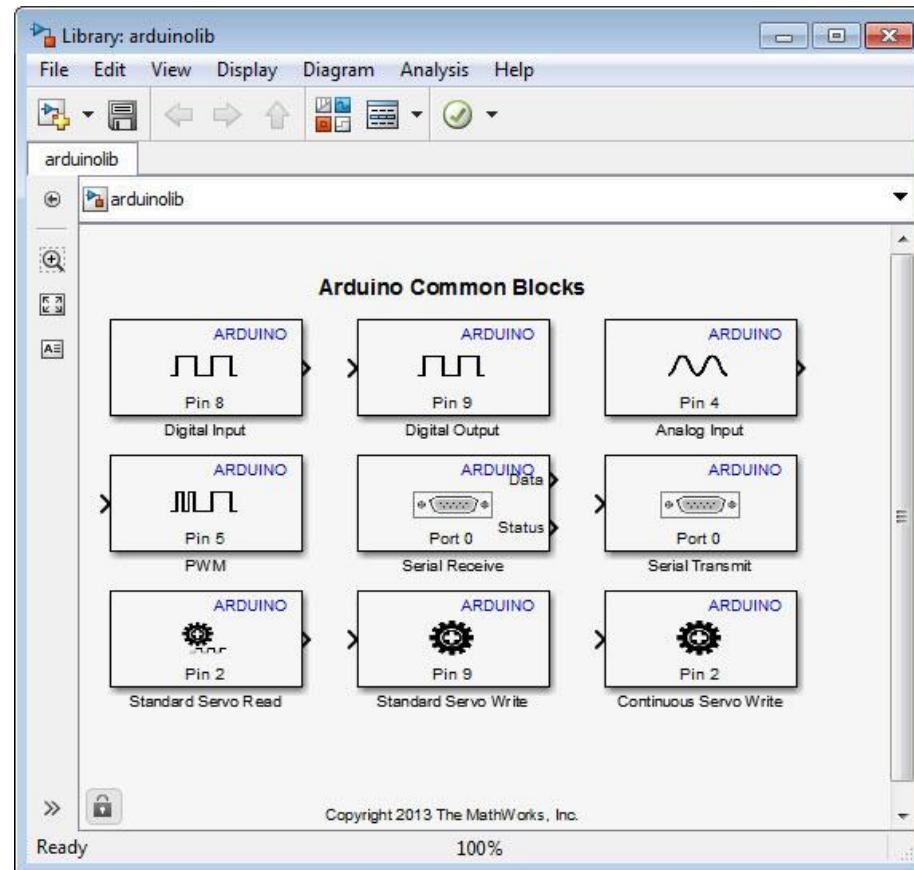


Freescale

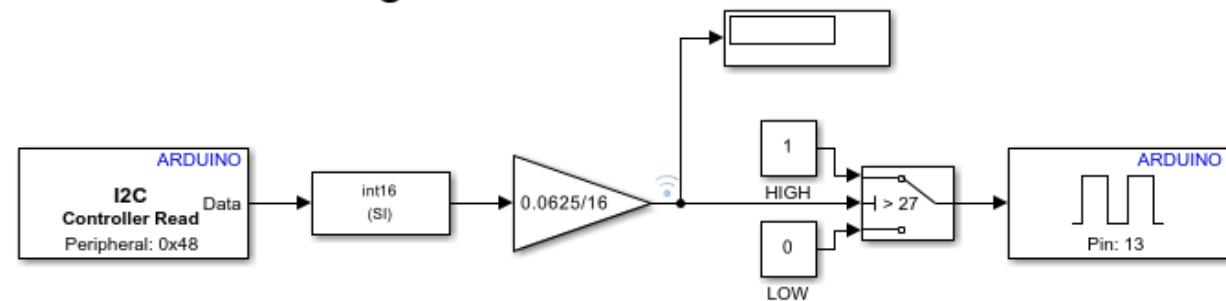


Zynq SDR

Interact with hardware using Simulink/Simulink Online



**Read temperature from an I²C based sensor
using Arduino Hardware**



To run this model on hardware, goto the "HARDWARE" tab and under "RUN ON HARDWARE" section, click the "Monitor & Tune" button. This would allow you to tune parameters and monitor signals in the mode while the application is running on hardware.

Copyright 2015-2021 The MathWorks, Inc.

Control an LED from Simulink!

The screenshot displays the MATLAB R2025b interface with several windows open:

- Command Window:** Shows the message "New to MATLAB? See resources for Getting Started" and a list of files in the current directory.
- Simulink Model:** The model is named "R4TestModel". It consists of two main blocks: "Read from Analog Input" (Arduino Pin A2) and "Write to Digital Output" (Arduino Pin 8). The "Read from Analog Input" block has a scope connected to its output. The "Write to Digital Output" block has a pulse generator connected to its input.
- Diagnostic Viewer:** Shows the build log for "R4TestModel_ert_rtw" and a summary of the build process.
- Camera View:** Shows a photograph of an Arduino Uno connected to a breadboard via a USB cable. Wires from the breadboard are connected to the Arduino pins.

Controls tutorials with MATLAB and Simulink

The screenshot shows the homepage of the Control Tutorials for MATLAB & Simulink (CTMS). The top navigation bar includes links for MICHIGAN, Carnegie Mellon, DETROIT MERCY, TIPS, ABOUT, BASICS, HARDWARE, INDEX, and NEXT ▶. Below the navigation is a banner with icons for various control systems: INTRODUCTION, CRUISE CONTROL, MOTOR SPEED, MOTOR POSITION, SUSPENSION, INVERTED PENDULUM, AIRCRAFT PITCH, and BALL & BEAM.

The main content area features a grid of images representing different control applications, such as a robotic arm, a speedometer, a car, gears, a suspension system, an inverted pendulum, an aircraft, and a ball on a beam. To the left of the grid, there are vertical columns of text labels corresponding to the images:

- SYSTEM**: INTRODUCTION, CRUISE CONTROL, MOTOR SPEED, MOTOR POSITION, SUSPENSION, INVERTED PENDULUM, AIRCRAFT PITCH, BALL & BEAM.
- MODELING**: MODELING
- ANALYSIS**: ANALYSIS
- CONTROL**: CONTROL
- PID**: PID
- ROOT LOCUS**: ROOT LOCUS
- FREQUENCY**: FREQUENCY
- STATE-SPACE**: STATE-SPACE
- DIGITAL**: DIGITAL
- SIMULINK**: SIMULINK
- MODELING**: MODELING
- CONTROL**: CONTROL
- SIMSCAPE**: SIMSCAPE

A central text block welcomes users to the CTMS, stating: "Welcome to the Control Tutorials for MATLAB and Simulink (CTMS): They are designed to help you learn how to use MATLAB and Simulink for the analysis and design of automatic control systems. They cover the basics of MATLAB and Simulink and introduce the most common classical and modern control design techniques."

Below this, another text block about the authors notes: "About the Authors: These tutorials were originally developed by Prof. Dawn Tilbury at the University of Michigan and Prof. Bill Messner at Carnegie Mellon with funding from NSF. With further support from the MathWorks in 2011 and 2017, Prof. Messner, Prof. Rick Hill (Detroit Mercy), and PhD Student JD Taylor (CMU), expanded the tutorials, completely redesigned the web interface, and

At the bottom, a footer states: "All contents licensed under a Creative Commons Attribution-ShareAlike 4.0 International License." and includes a Creative Commons license logo.

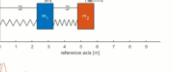
Modular courseware and virtual labs

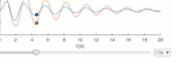
Mass-Spring-Damper Systems
Version 1.0.1.0 (7.05 MB) by Emma Smith Zbarsky STAFF
<https://github.com/MathWorks-Teaching-Resources/Mass-Spring-Damper-Systems>
Follow

Mass-Spring-Damper Systems

[File Exchange](#) or [Open in MATLAB Online](#)

Curriculum Module
Created with R2020b. Compatible with R2020b and later releases.





Mass-Spring-Damper Systems

CruiseControl.mlx
In this script, students will...
Academic disciplines

- Compare and contrast open-loop and feedback control.
- Implement a simple open-loop controller in Simulink.
- Analyze the performance of an open-loop controller.
- Implement a proportional controller in Simulink.
- Analyze the performance of a proportional controller.

VehicleModel.mlx
In this script, students will...
Academic disciplines

- Derive the transfer function of a first order system.
- Compare the transfer function and virtual vehicle responses.
- Identify the model parameters for the virtual vehicle.

PositionControl.mlx
In this script, students will...
Academic disciplines

- Implement a PID controller.
- Identify rise time, settling time, overshoot, and peak time.
- Explain how changes to PID parameters affect the time-domain response.

Virtual Hardware and Labs for Controls

Introduction to Power Electronics
Instructor Prof. Kim Electrical Engineering | Power
14 Weeks All Levels 28 Lessons 14 Quizzes 1403 Students

[Overview](#) [Curriculum](#) [Instructor](#)

Description
Introductory course for power electronics covering switching components, ac-dc rectifiers, dc-dc converters, dc-ac inverters, and basic control.

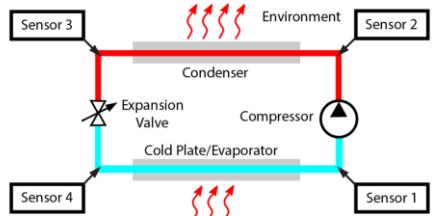
Reference Texts (Optional)
The following textbooks are helpful for a deeper understanding of the material, but the general concepts can be learned without the textbook.

- Philip T. Krein, "Elements of Power Electronics," 2nd Edition, Oxford University Press, 2014.

Introduction to Power Electronics

Model_Fridge.mlx [Open in MATLAB Online](#)

Apply their knowledge of thermodynamics to a Simulink model of a real refrigerator.



In this script, students will...

- Apply understanding about thermodynamic cycles to model a refrigerator

Thermodynamics

Model Predictive Control (MPC) virtual lab
Version 1.0.2 (2.31 MB) by Eric Hillsberg STAFF
<https://github.com/MathWorks-Teaching-Resources/Model-Predictive-Control>
Follow

MPC Virtual Lab

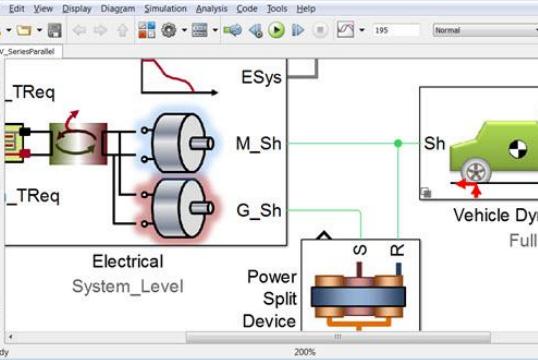
[File Exchange](#) or [Open in MATLAB Online](#)

This virtual lab contains interactive exercises to study the design of linear and adaptive model predictive controllers (MPCs). The lab solutions are available upon instructor request. If you would like to request solutions, find an issue, or have a suggestion, please open an issue.

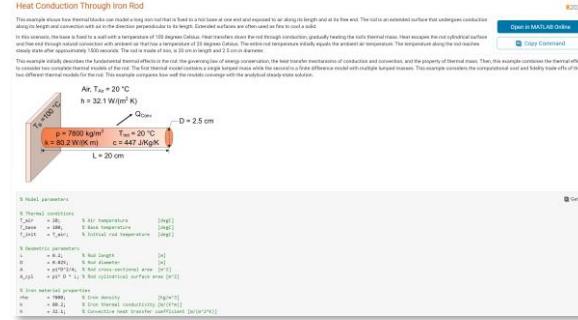


Model Predictive Control

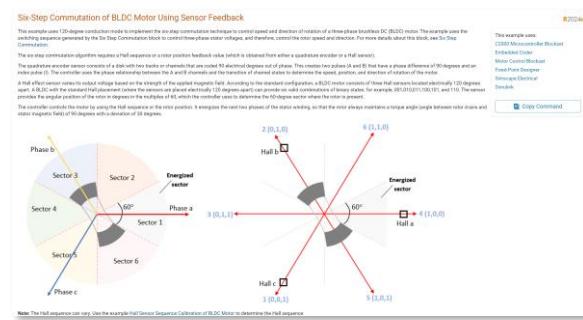
Hybrid-Electric Vehicle Model in Simulink & Simscape



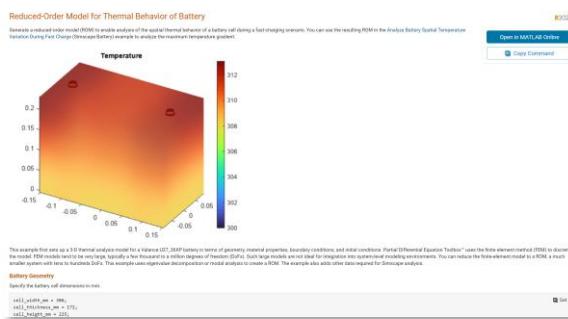
Reference examples in documentation and file exchange



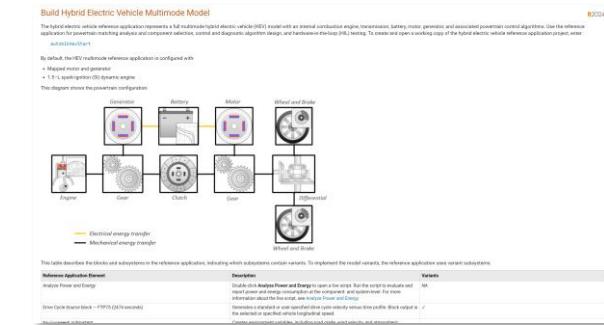
[Heat Conduction Through Iron Rod](#)



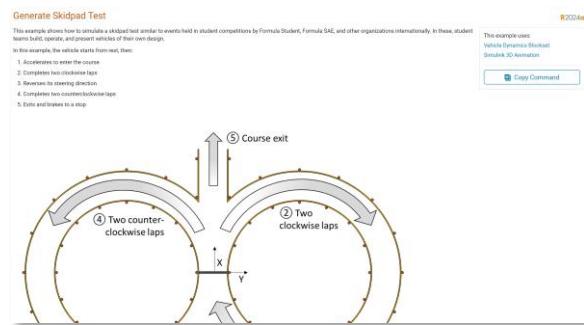
[Six-Step Commutation of BLDC Motor Using Sensor Feedback](#)



[Reduced-Order Model for Thermal Behavior of Battery](#)



[Build Hybrid Electric Vehicle Multimode Model](#)



[Generate Skidpad Test](#)

Examples with Extended Descriptions and Educational Materials

Self-Paced Online Courses

- MATLAB and Simulink online training courses at 44 Short Courses
 - 12 Learning Paths
- Offers learning flexibility for:
 - upskilling/reskilling
 - exploring new topics and ideas
 - meeting prerequisites for instructor-led courses or consulting engagements
- Choose to take individual lessons, short courses, or full learning paths
- Wide range of available topics—new courses and topics added every few months

The screenshot shows a grid of three course cards. The first card is titled 'Build MATLAB Proficiency' with a learning path of 8 courses, described as gaining a comprehensive foundation in MATLAB. The second card is 'Simulink Fundamentals' with 8 hours of content, described as learning how to use Simulink for modeling dynamic physical systems. The third card is 'Find and Extract Subsets of Data' with 1.5 hours of content, described as using logical indexing to filter data and count elements.

Short courses (1-3 hours)

- MATLAB Onramp
- Simulink Onramp
- Circuit Simulation Onramp
- Machine Learning Onramp
- Power Systems Simulation Onramp
- Image Processing Onramp
- System Composer Onramp
- Object Oriented Programming Onramp
- App Building Onramp
- Simscape Onramp
- Stateflow Onramp

Long courses (7-10 hours)

- Simulink Fundamentals
- Image Processing with MATLAB
- Deep Learning with MATLAB
- Machine Learning with MATLAB

MathWorks Education Application Engineer *sjawahar@mathworks.com*

consult with faculty and researchers to support them with their STEM initiatives,
including integrating computational or systems thinking into their curriculum and research



Take a minute to submit a survey!

