From Classroom to Industry: Exploring Model-Based Design with Simulink

Brandon Trombley
Technical Account Manager

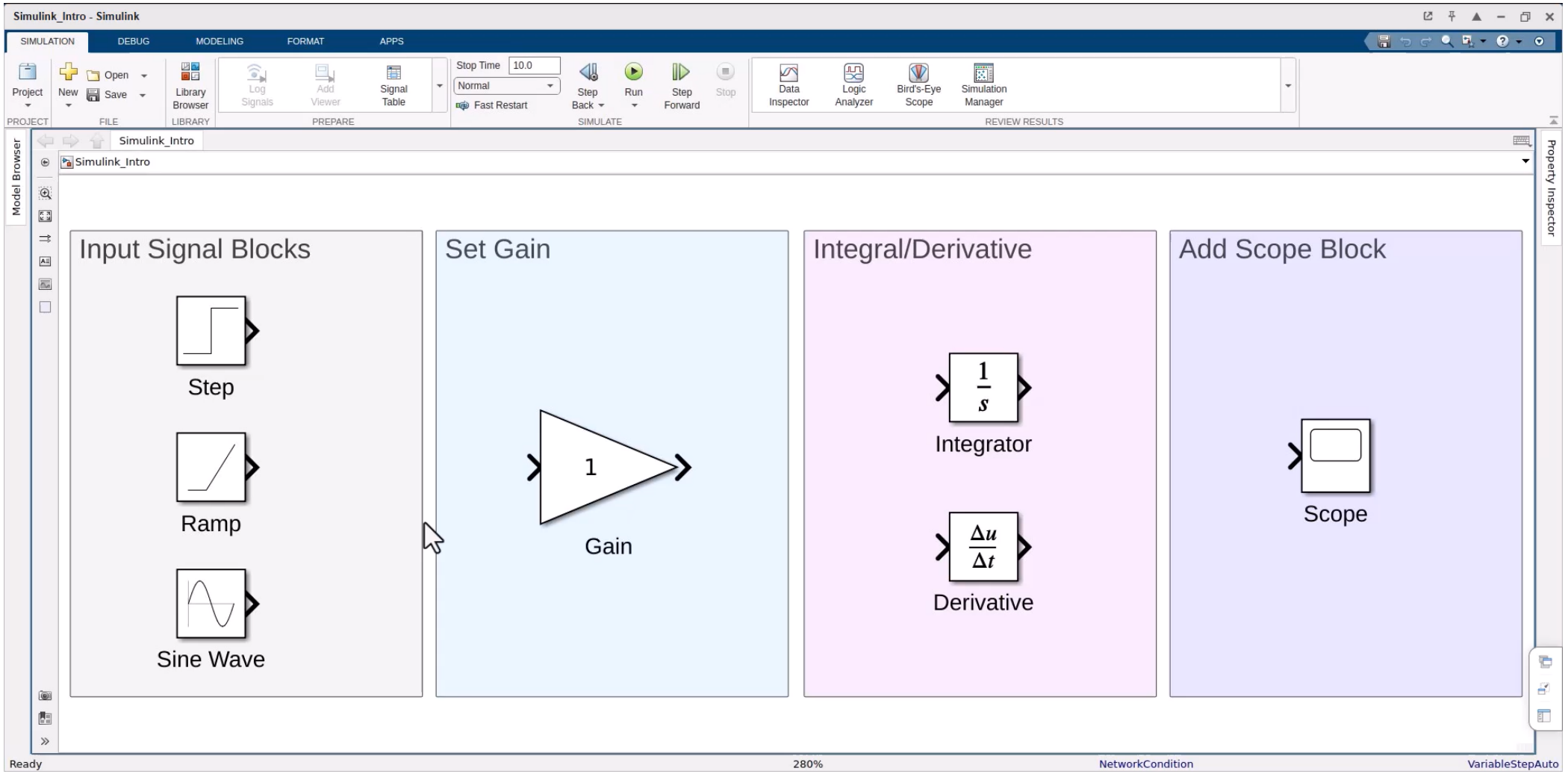
Noah Roberts
Customer Success Engineer

Siddharth Jawahar
Customer Success Engineer

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 materials via GitHub
  - https://github.com/SidJawMW/simulink-intro

What are we going to see today?



Agenda

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

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

Renewable Energy

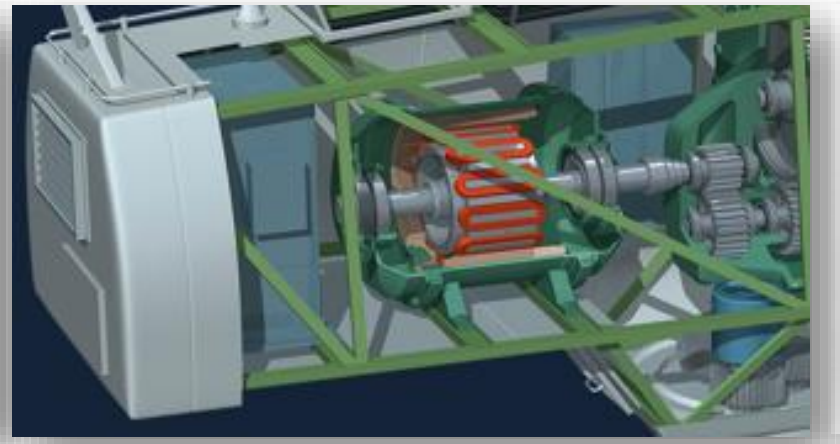
Wireless Communication

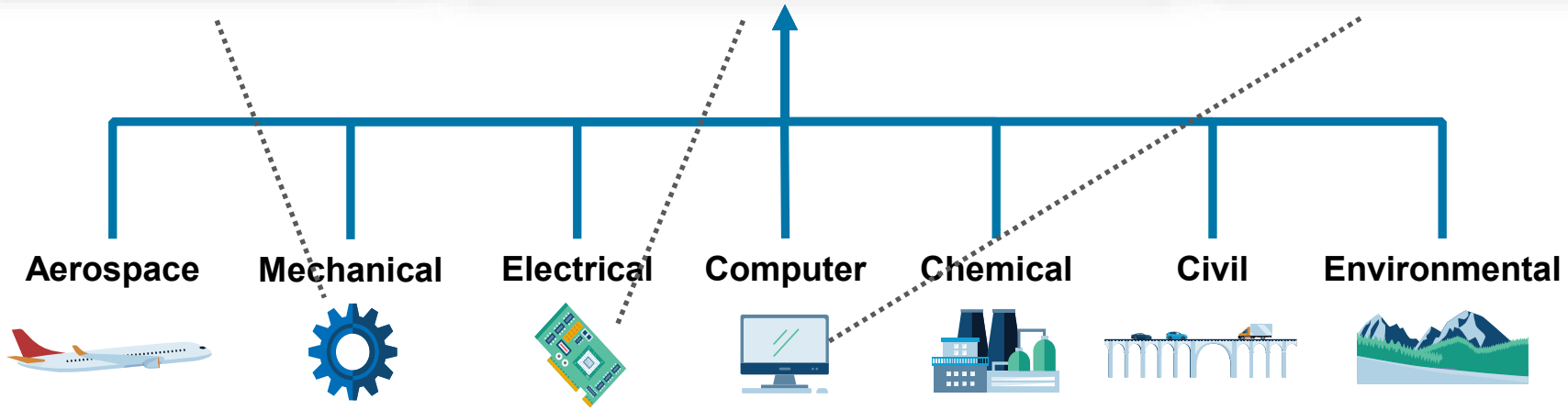
Aerial Robotics

Electric Vehicles

Engineered systems require system-level, multi-domain integration

Renewable Energy





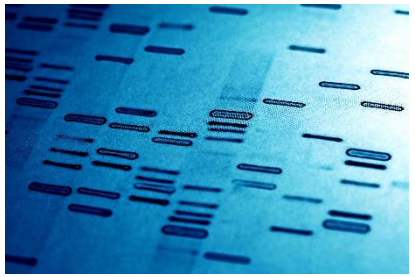
Model-Based Design can help!

Model-based design applied to a wide range of domains

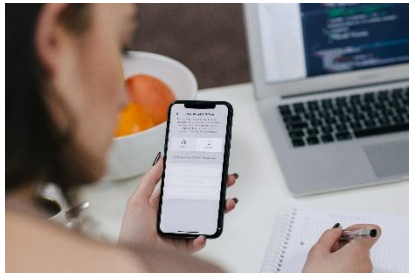
Aerospace and Defense


Automotive


Biological Sciences


Biotech and Pharmaceutical


Communications


Electronics


Energy Production


Financial Services

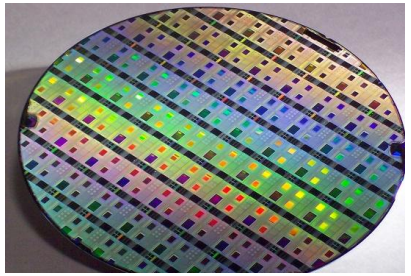

Industrial Machinery


Medical Devices


Metals, Materials, Mining


Neuroscience

Railway Systems


Semiconductors

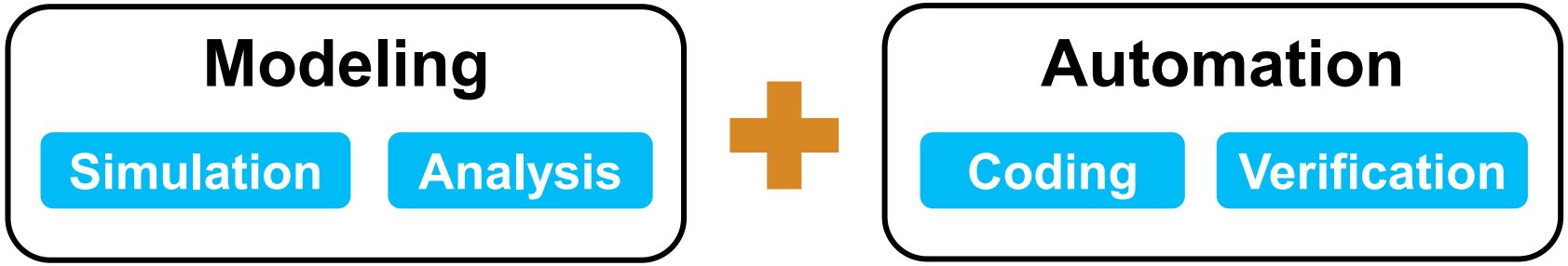
Software and Internet

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

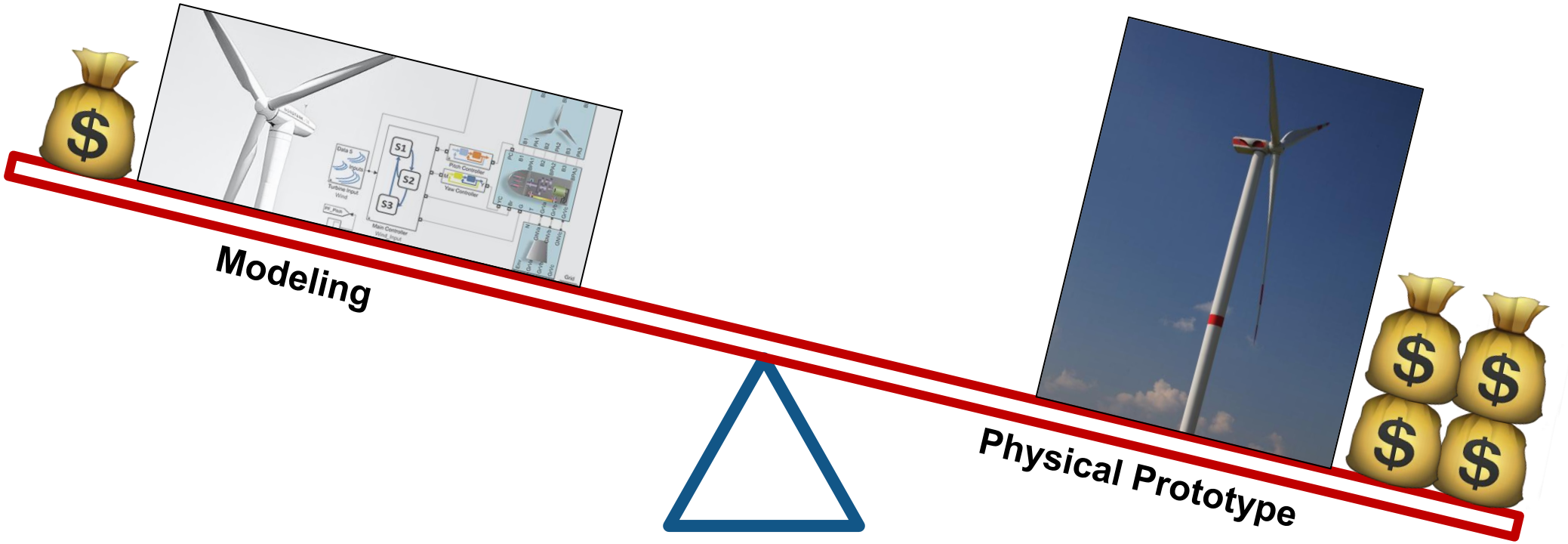
Systematic use of **models** throughout the development process



Try out new ideas
Fast repeatable tests

Eliminate manual steps
and **human error**

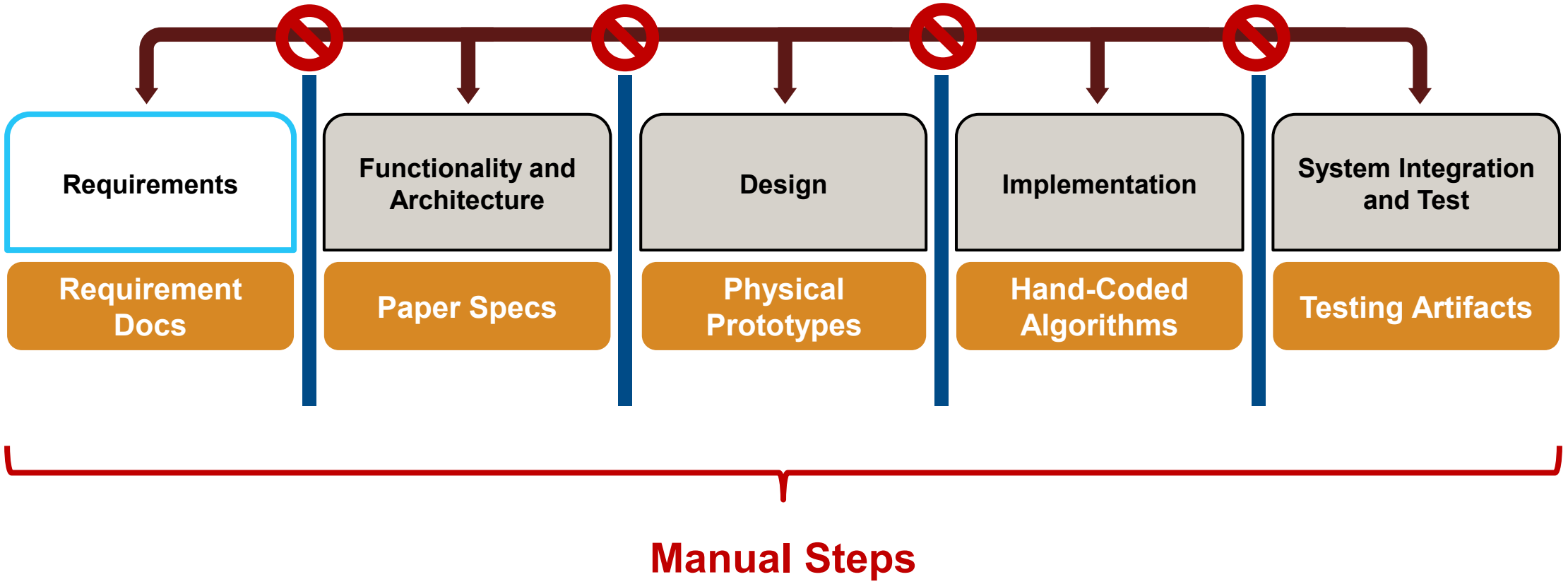
Physical prototypes are **costly** and iterate **slowly**



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

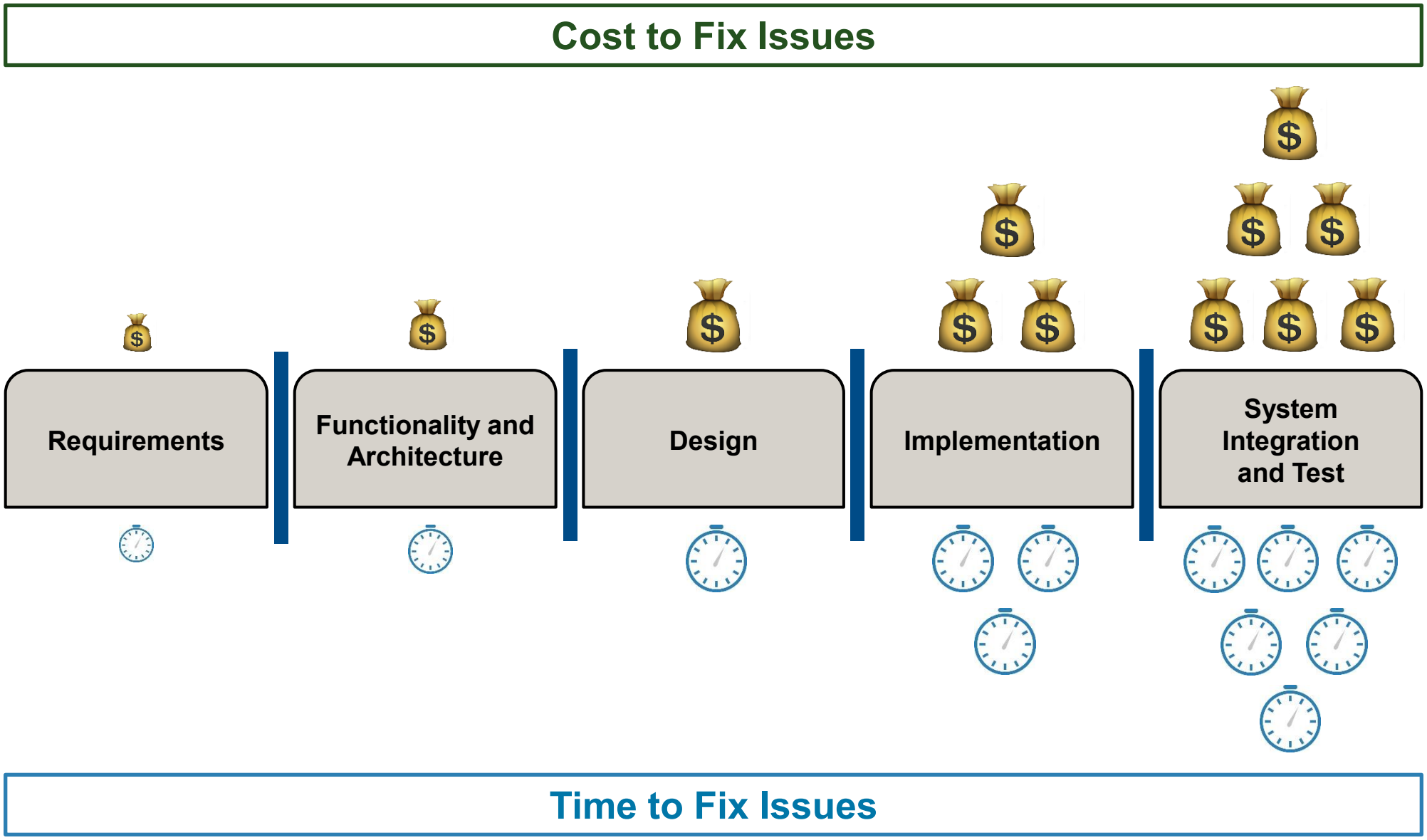
Costly and **time-consuming** to build
Hinders rapid iterations

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

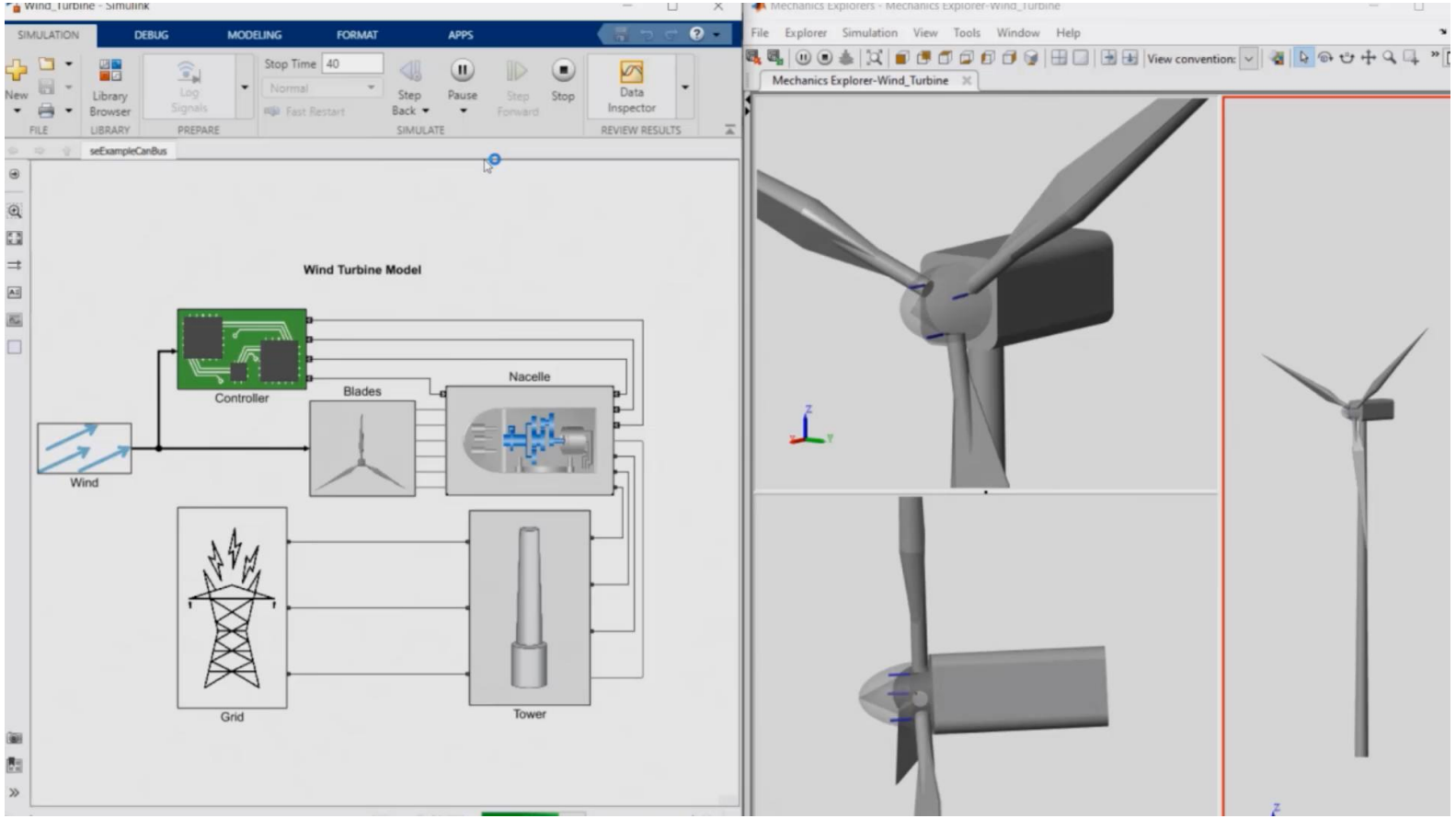
Modeling

Simulation | Analysis

+

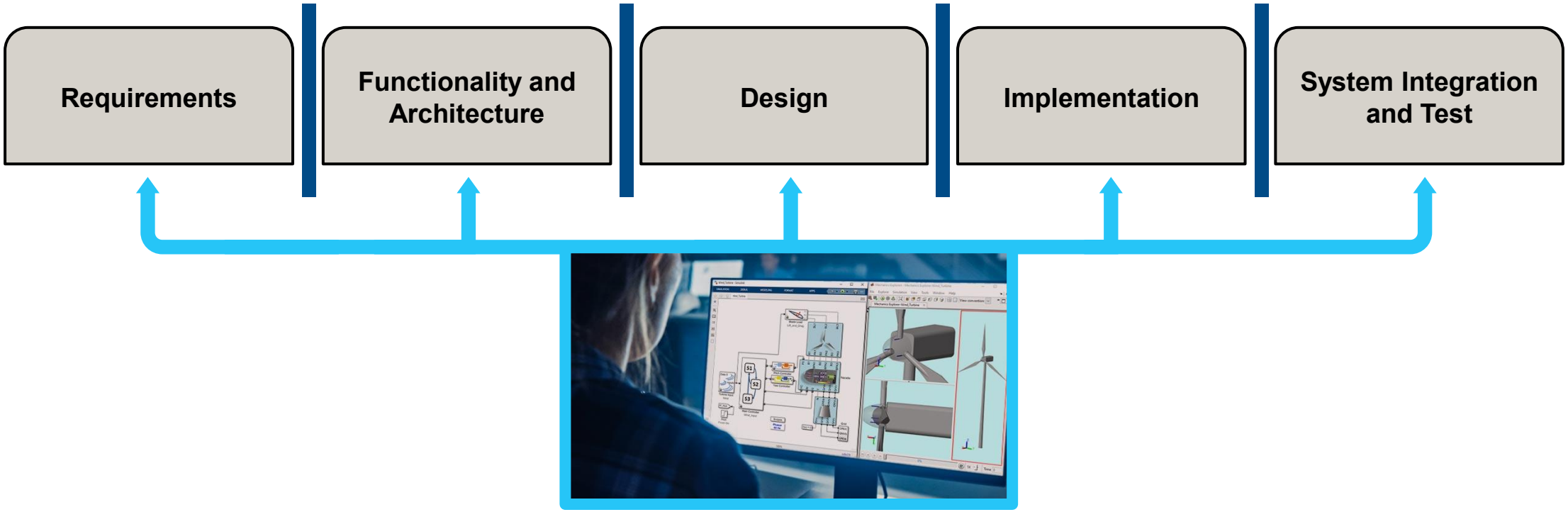
Automation

Coding | Verification



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

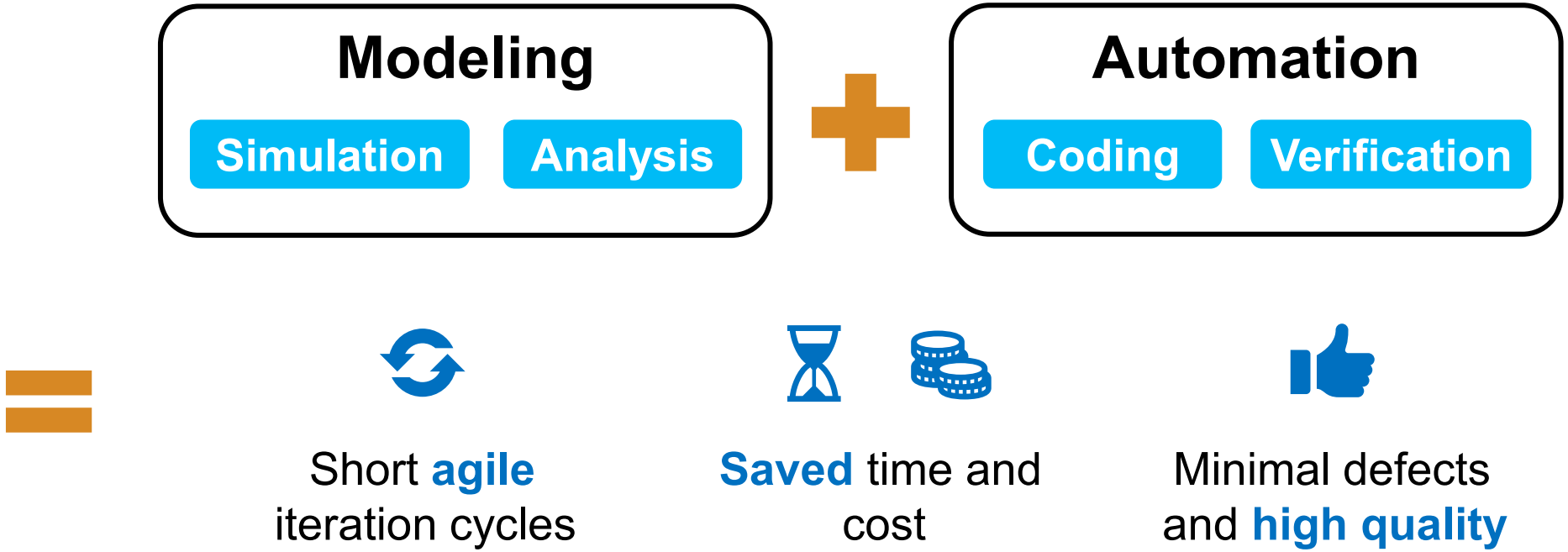
Model-Based Design



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

Model-Based Design

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

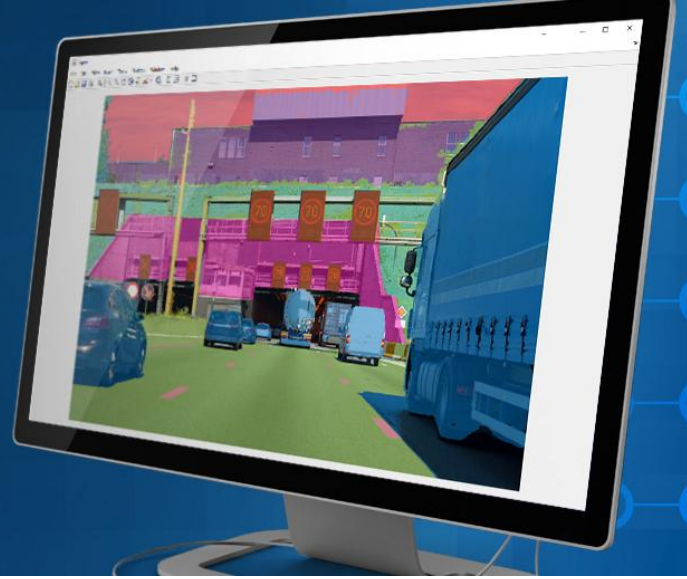


End to end advanced air mobility development at Supernal

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

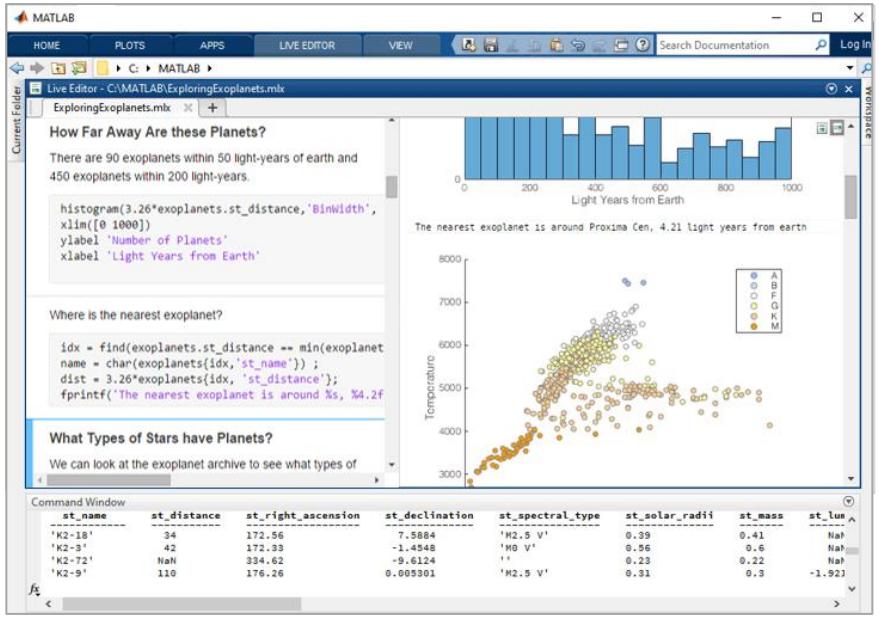
Our Products

MATLAB® & SIMULINK®



- **MATLAB** - Programming environment for algorithm development, data analysis, visualization, and numeric computation.
- **Simulink** - Block diagram environment for simulation and Model-Based Design of multidomain and embedded engineering systems.
- **130+ add-on products** for specialized tasks.

Computer Vision Toolbox



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

Visit https://blogs.mathworks.com/racing-lounge/category/where-are-they-now/

TAM Perspective: Industry Roles and Skill Areas

What is a TAM (Technical Account Manager)?

- Technical expert in MathWorks tools
- Technical / business expert in Industry

How do we work with Industry customers?

- Identifying customer corporate goals
- Developing strategies to support those goals using our tools and methods

What kinds of groups do we interact with / use our tools?

- Engineering for Product Development
- Research & Development
- Data Analytics
- Finance

Industry Insider View

Hiring Manager Perspective

1 in 10 engineers had the right skills in MBD

Find industry contacts and use them!

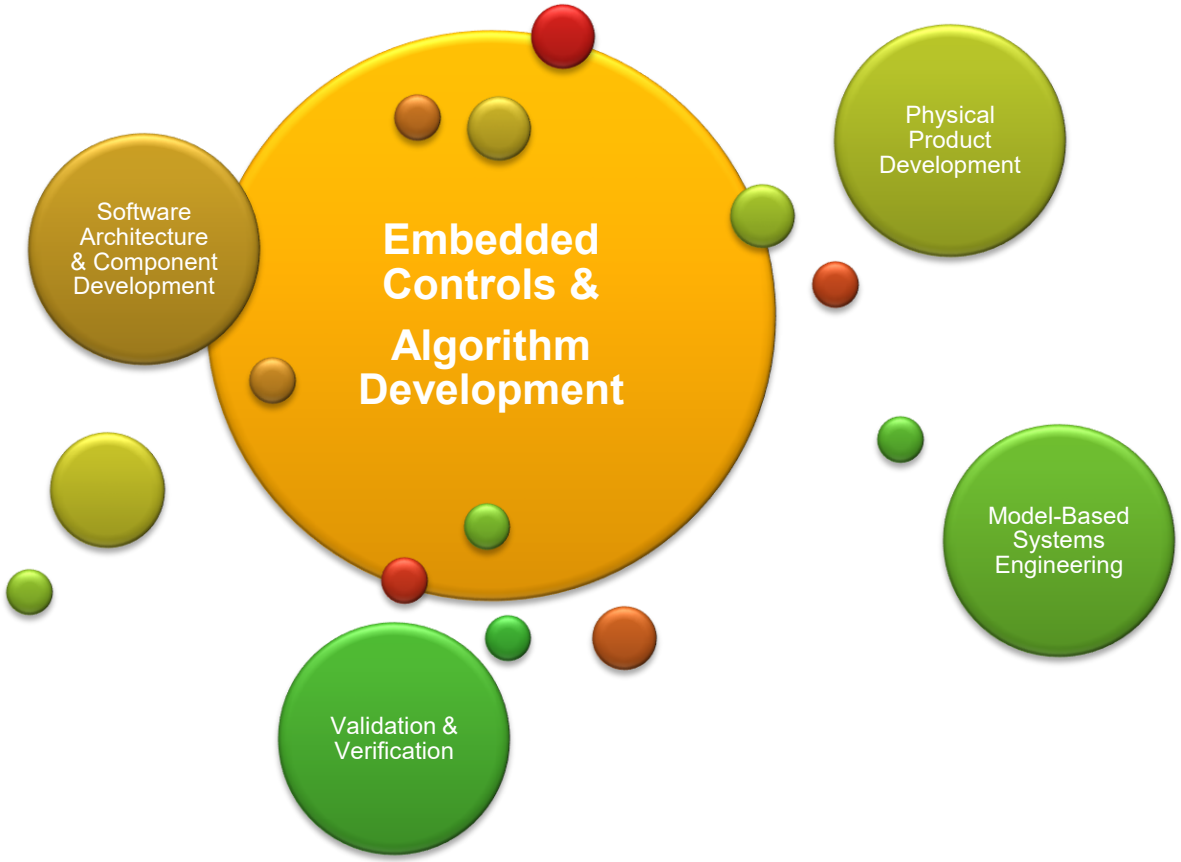
Training is available for campus-wide licenses

Internships

Show direct industry experience directly to hiring managers

Search with keywords: “Model-Based Design”, “Simulink”, and “Code Generation”

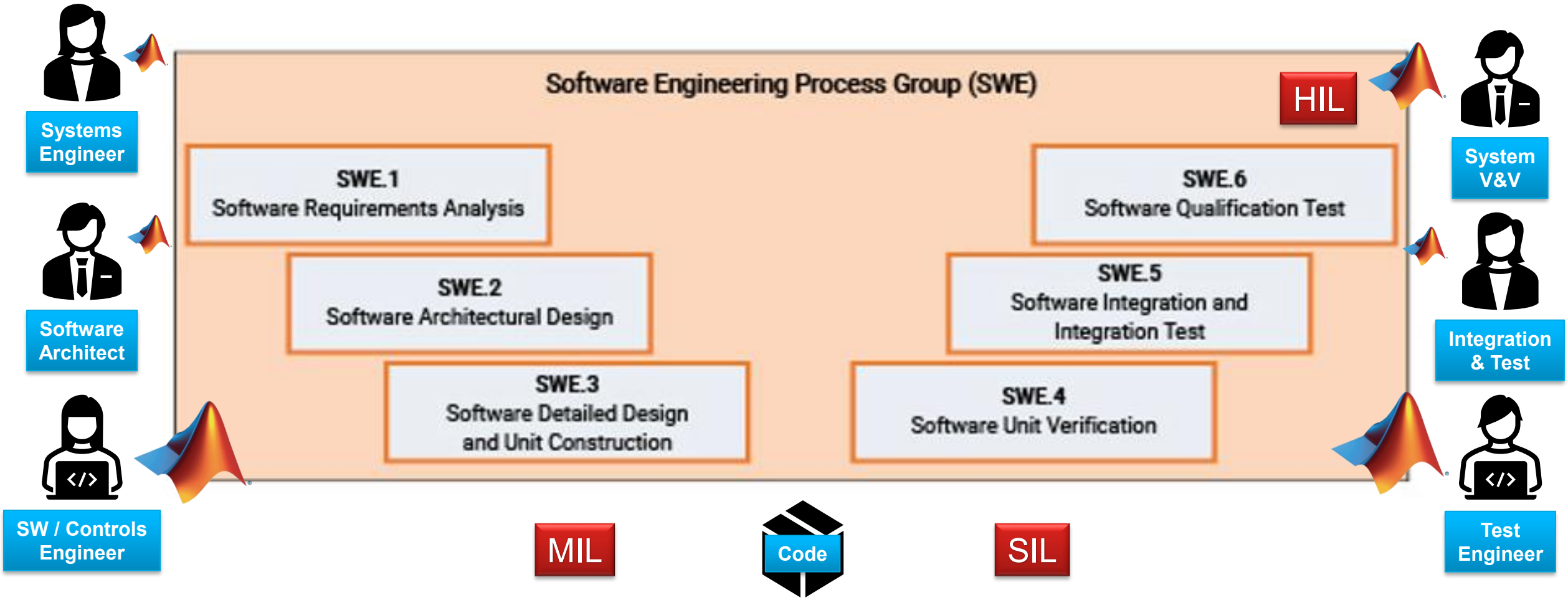
Industry Roles for Model-Based Design



- Embedded Controls Development
  - Controller Modeling
  - Software Component Modeling
  - Requirements-Based Model-In-the-Loop (MIL) Testing
  - Production Code Generation
  - Software-In-the-Loop (SIL) Testing
  - Integration Testing
  - Hardware-In-the-Loop (HIL) Testing
- How many engineering positions?
  - Large companies: ~1000s

MATLAB and Simulink in Product Development Workflows

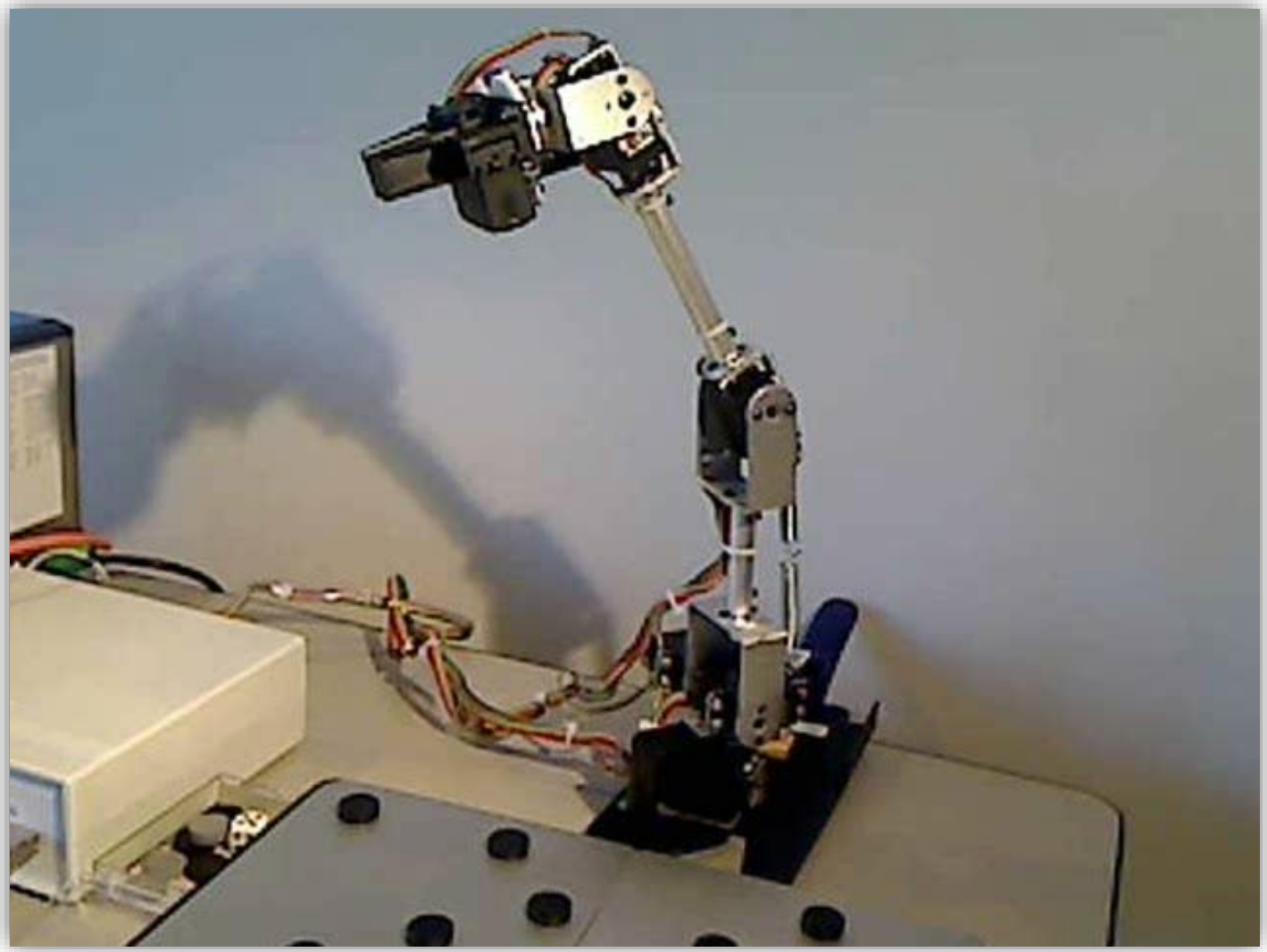
Example Process for Safety-Critical Software



Agenda

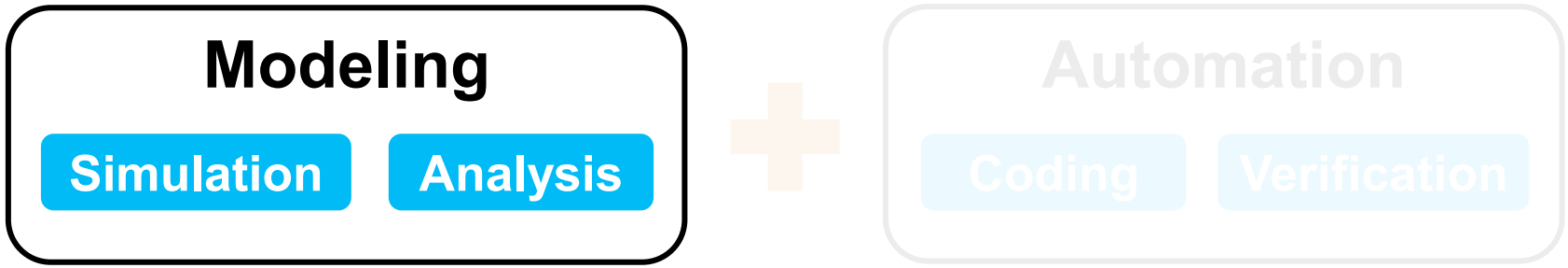
- Understanding engineered systems
- Introduction to Model-Based Design and Simulink
- **Simulink 101 – An Industry Standard tool for Model-Based Design**
- Resources

Let’s look at a simple pick and place robot…

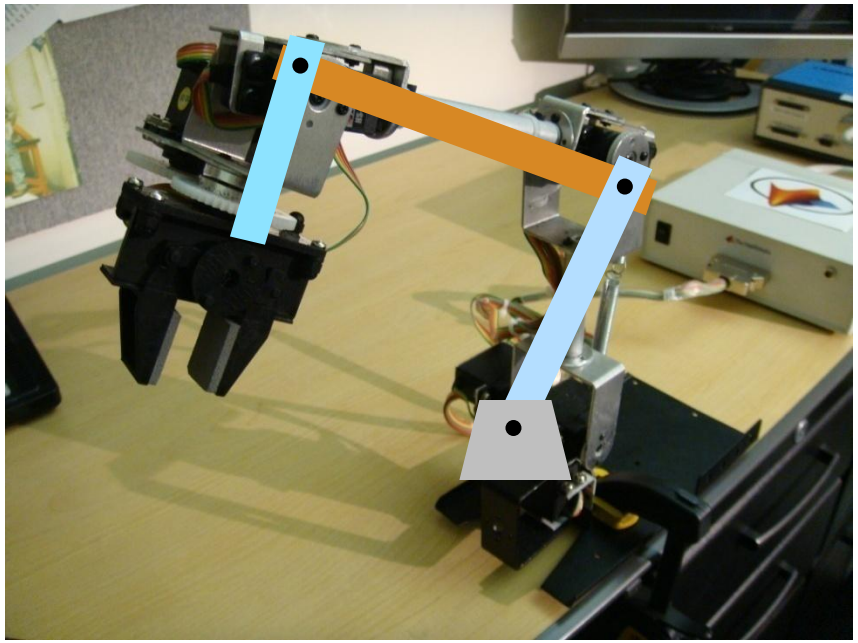


Model-Based Design

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



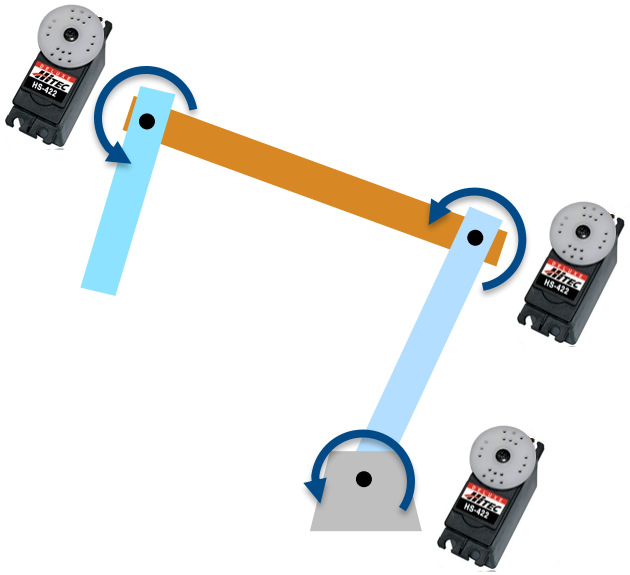
To model the system, let's break it down into its subcomponents



Modeling

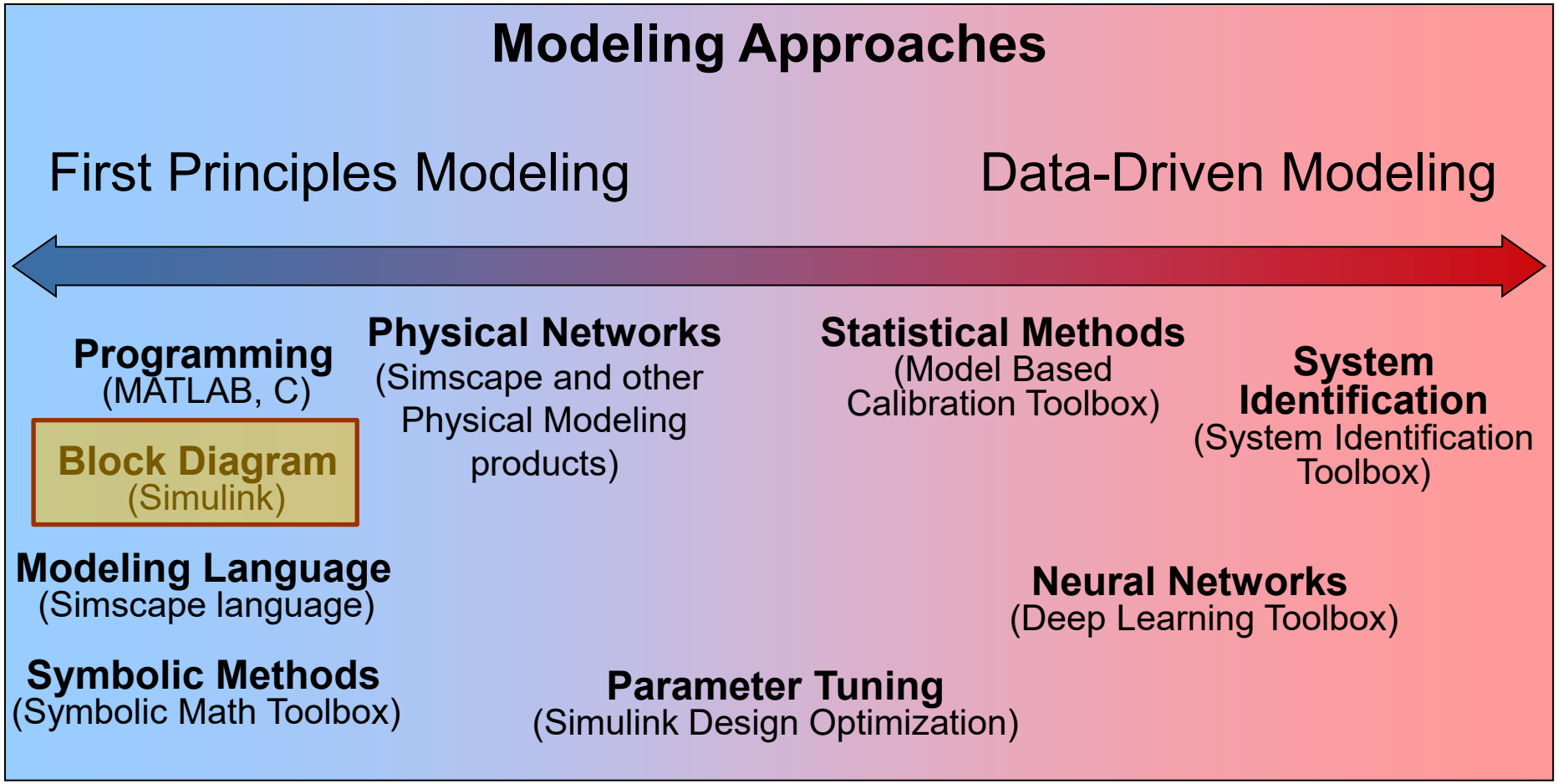
Simulation Analysis

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

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

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

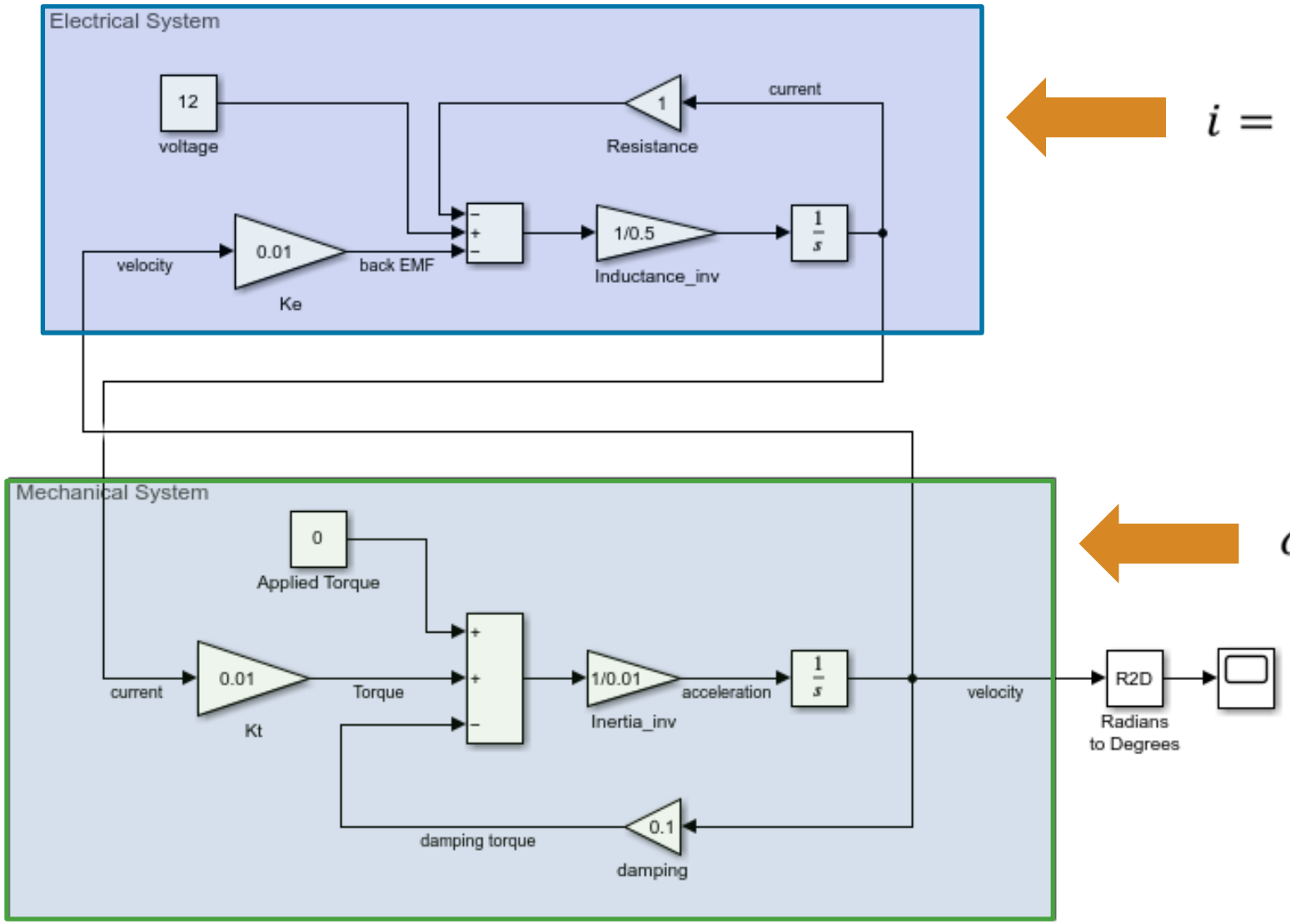
$$\omega = \int \frac{1}{J}(Ki - b\omega - T_{Load})dt$$

Mechanical

Modeling

Simulation | Analysis

Let's model these equations in Simulink



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

Electrical

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

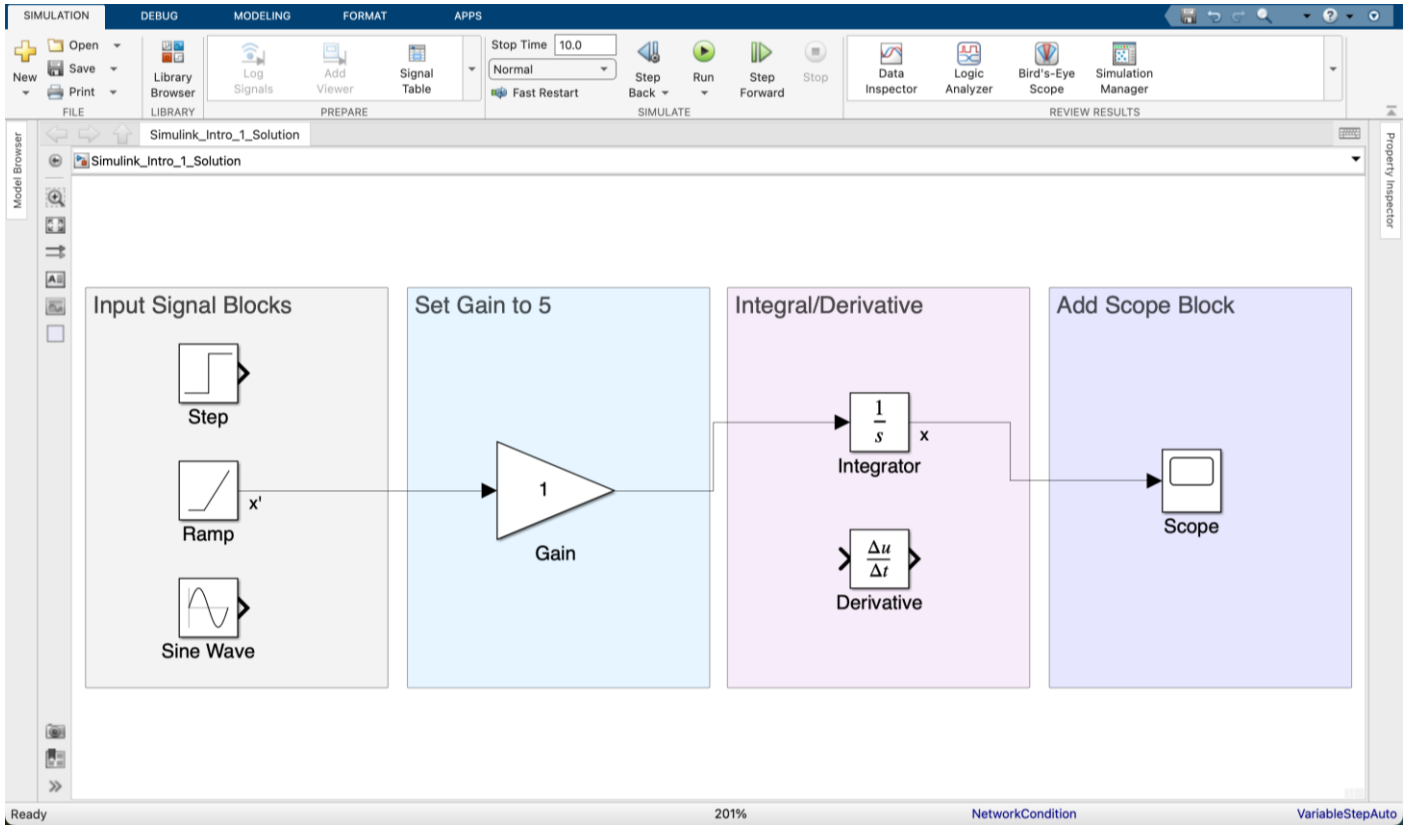
Mechanical

Modeling

Simulation **Analysis**

Exercise 1: Introduction to Simulink

- Open: **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

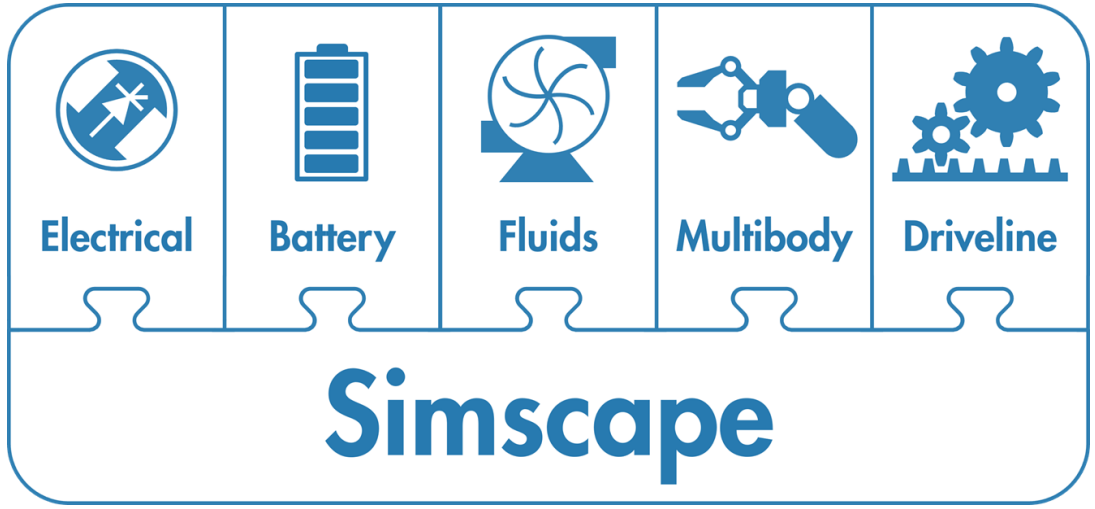


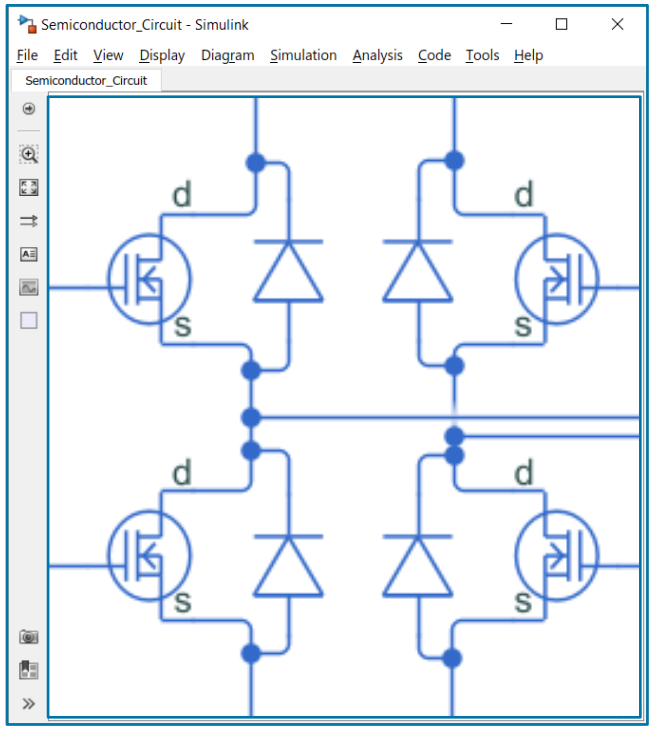
Modeling

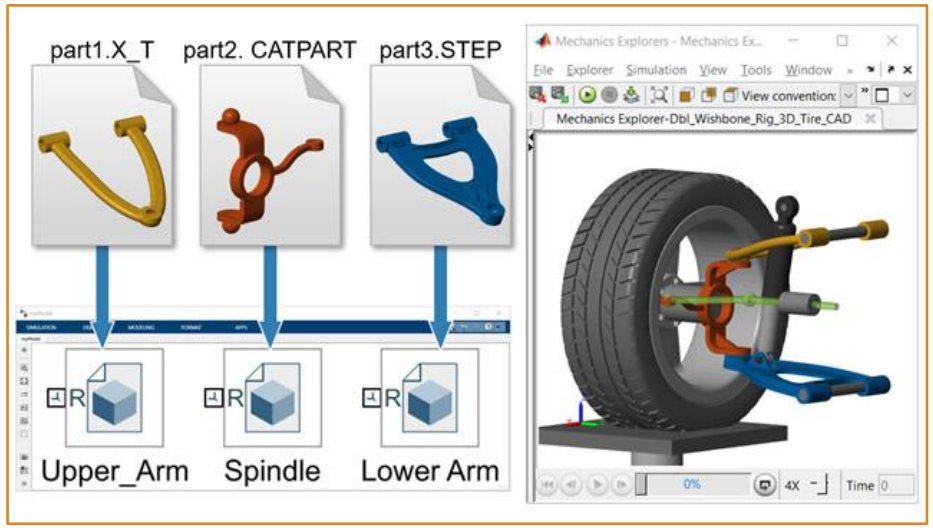
Simulation | Analysis

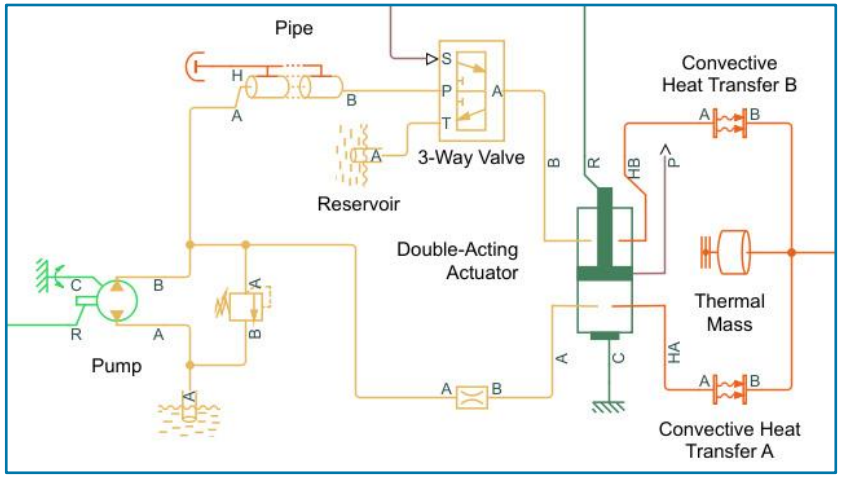
Simscape
Overview

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









MATLAB, Simulink, and Simscape for Physical Modeling

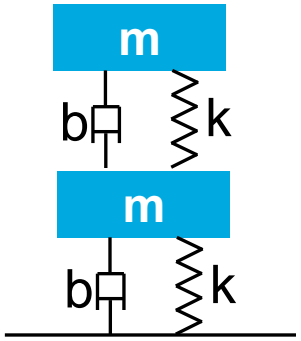
$$F_{Spring} = k_{Spring} * (x_{Mass})$$

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

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

$$F_{Spring2} = k_{Spring} * (x_{Mass})$$

....

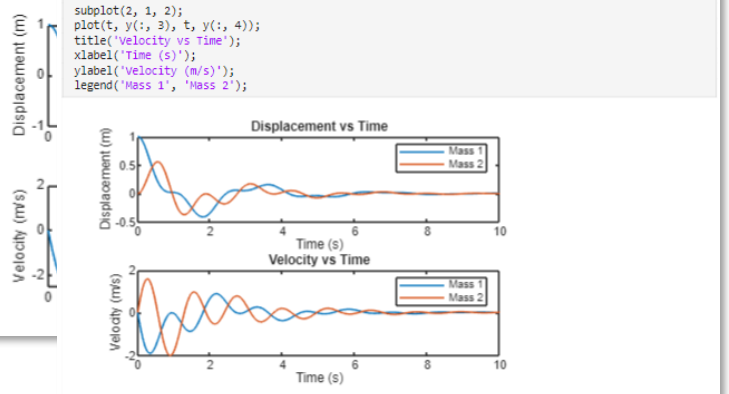


Below is a simple script using MATLAB code to simulate the dynamics of a spring-mass-damper system.

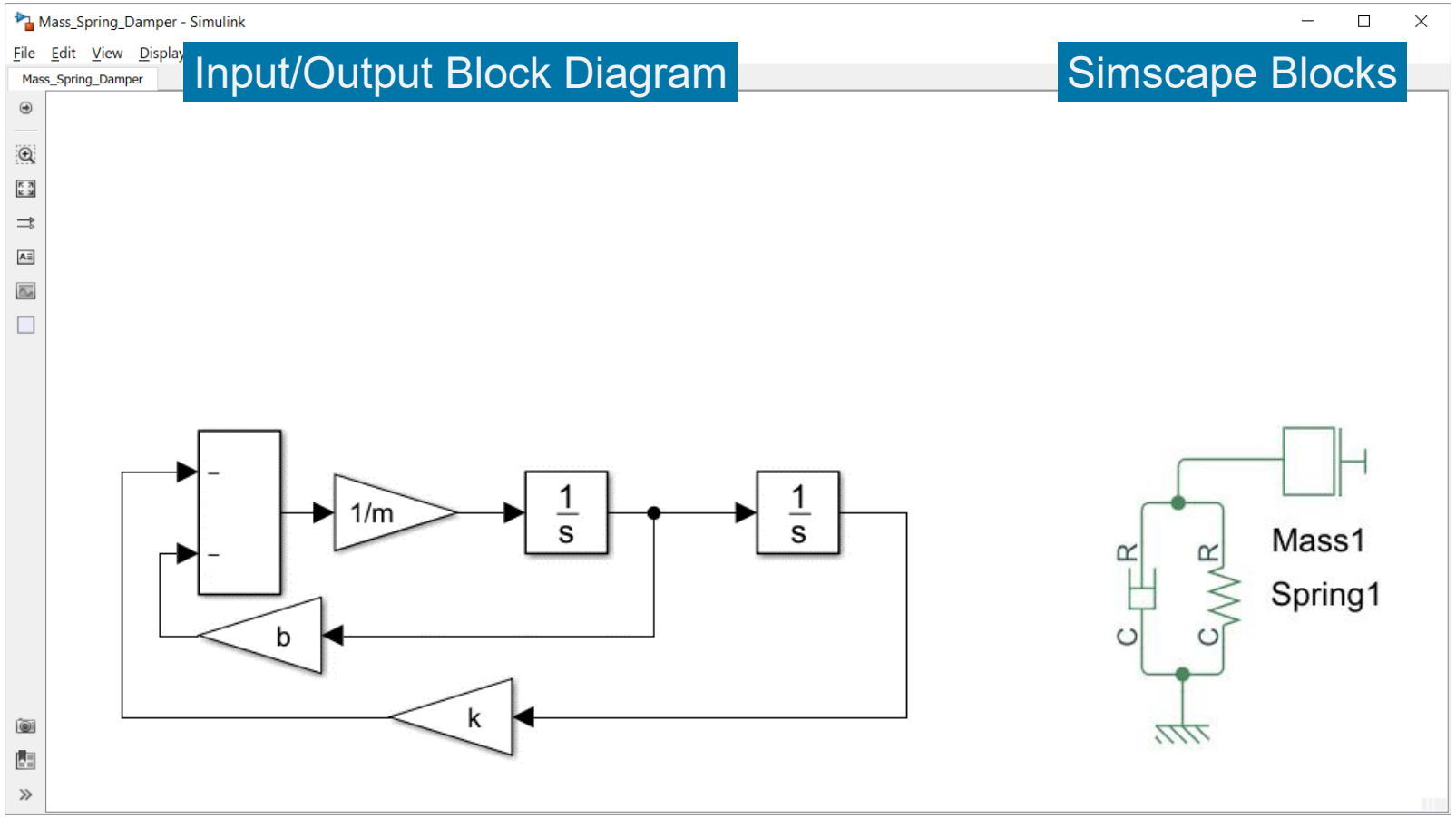
```
% Parameters
m = 1;
k = 10;
b = 1;

% Initial c
x0 = 1;
v0 = 0;

% Time span
tspan = [0

% ODE funct
odefun = @(

% Solve ODE
[t, y] = od

% Plot resu
figure;
subplot(2,
plot(t, y(:
title('Disp
xlabel('Tim
ylabel('Dis

subplot(2,
plot(t, y(:
title('Velo
xlabel('Tim
ylabel('Vel
```

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

```
% Parameters
m1 = 1;      % mass 1 (kg)
m2 = 1;      % mass 2 (kg)
k1 = 10;     % spring constant 1 (N/m)
k2 = 10;     % spring constant 2 (N/m)
b1 = 1;      % damping coefficient 1 (Ns/m)
b2 = 1;      % damping coefficient 2 (Ns/m)

% Initial conditions
x1_0 = 1;    % initial displacement of mass 1 (m)
v1_0 = 0;    % initial velocity of mass 1 (m/s)
x2_0 = 0;    % initial displacement of mass 2 (m)
v2_0 = 0;    % initial velocity of mass 2 (m/s)

% Time span
tspan = [0 10]; % simulate from 0 to 10 seconds

% ODE function
odefun = @(t, y) [y(3); y(4);
                  (-b1 * y(3) - k1 * (y(1) - y(2))) / m1;
                  (-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), t, y(:, 2));
title('Displacement vs Time');
xlabel('Time (s)');
ylabel('Displacement (m)');
legend('Mass 1', 'Mass 2');

subplot(2, 1, 2);
plot(t, y(:, 3), t, y(:, 4));
title('Velocity vs Time');
xlabel('Time (s)');
ylabel('Velocity (m/s)');
legend('Mass 1', 'Mass 2');
```

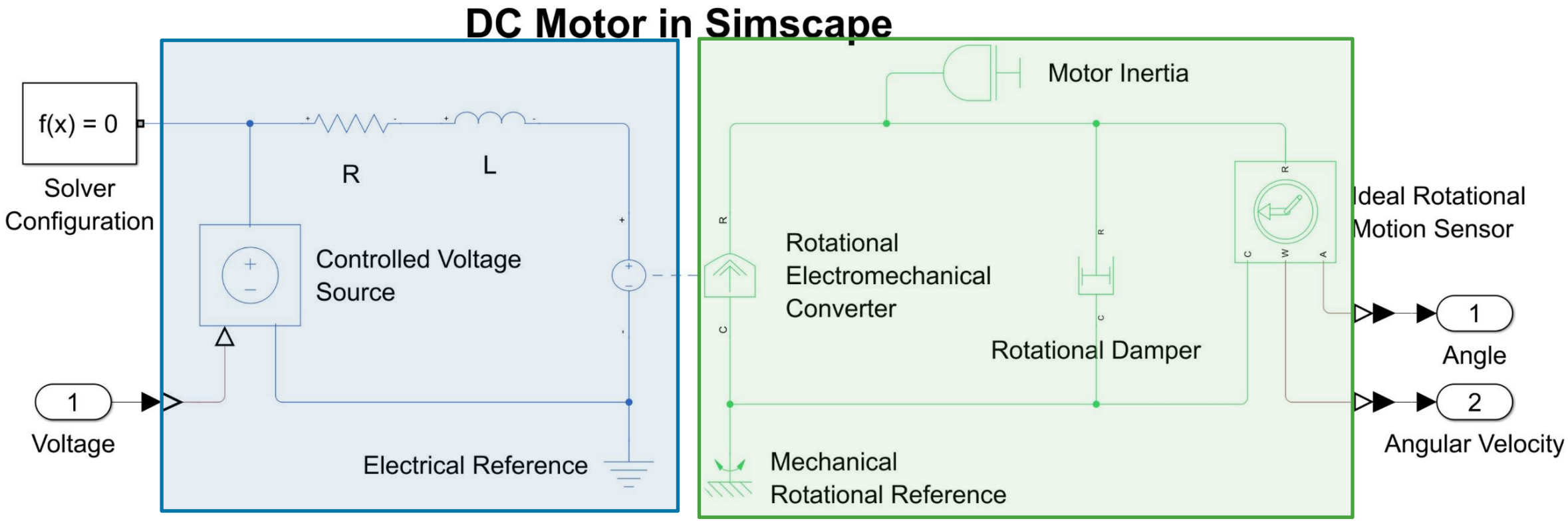


MATLAB



Simulink

Physical Network Model of a DC Motor using Simscape



Electrical

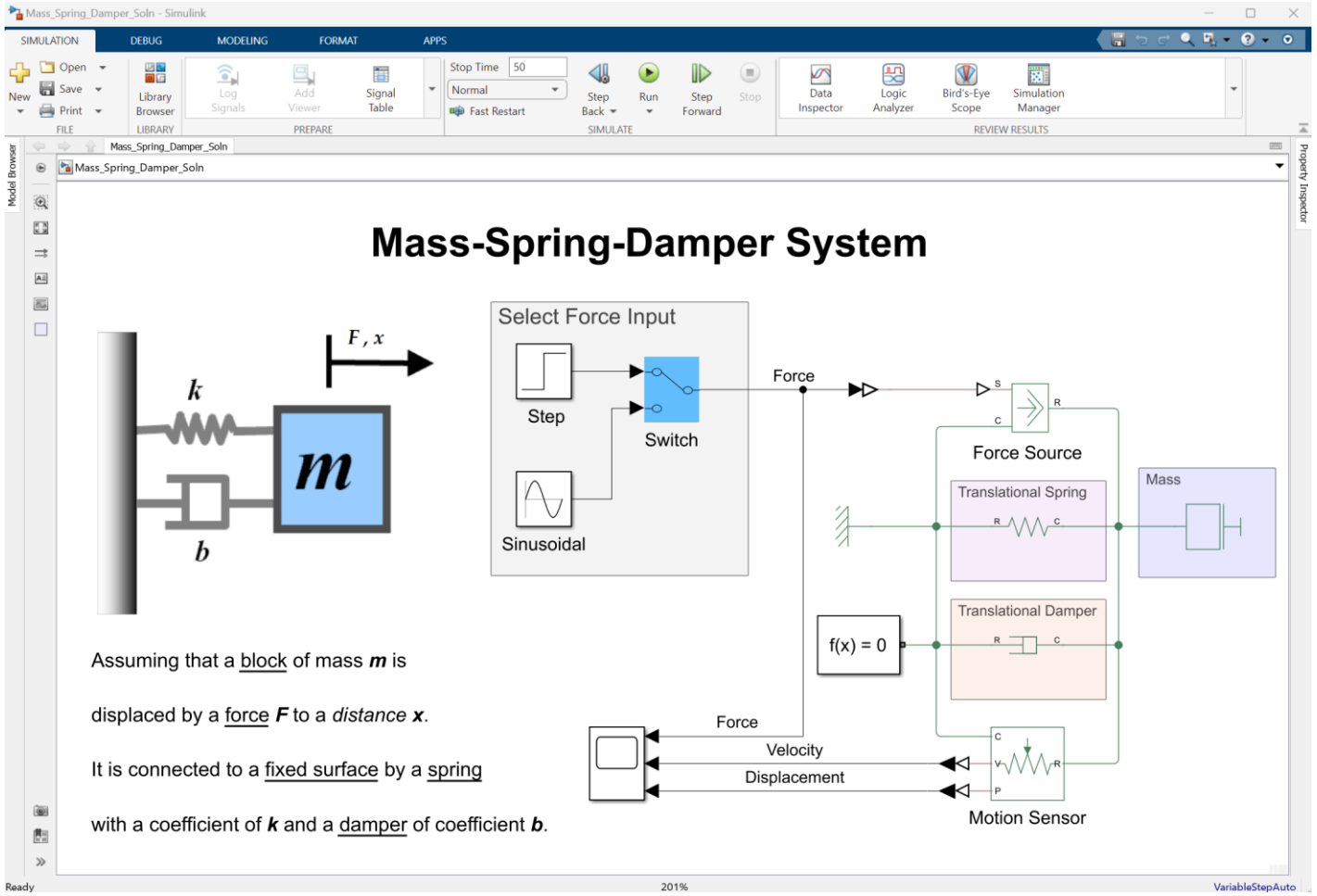
Mechanical

Modeling

Simulation | Analysis

Exercise 2: Mass-Spring Damper

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

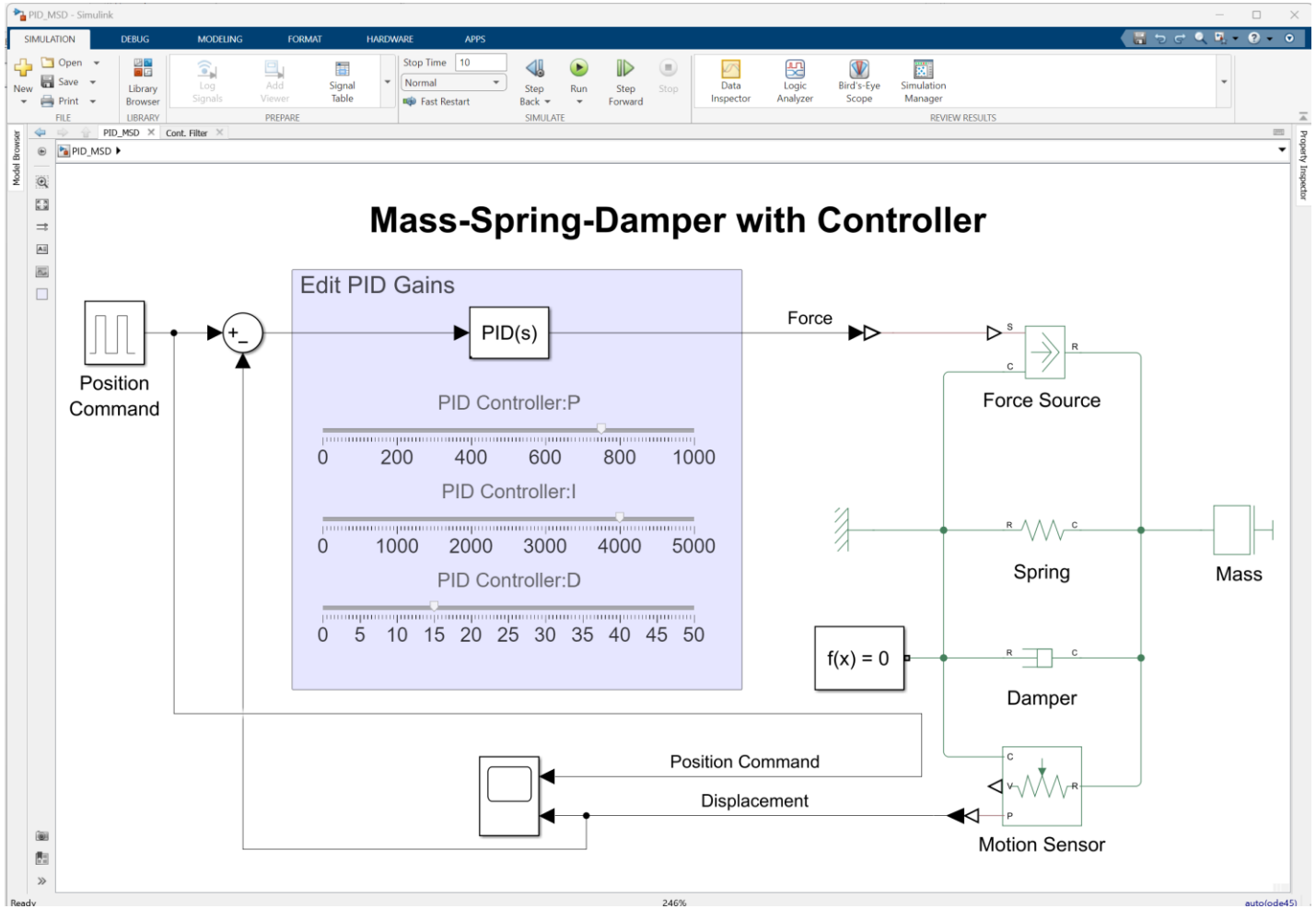


Modeling

Simulation **Analysis**

Exercise 3: Mass Spring Damper with Controller

- Open: ***PID_MSD.slx***
- Edit desired position commands or change input block
- Use sliders to adjust PID gains
- View results in Scope block
- Use **PID Tuner** App to achieve better performance

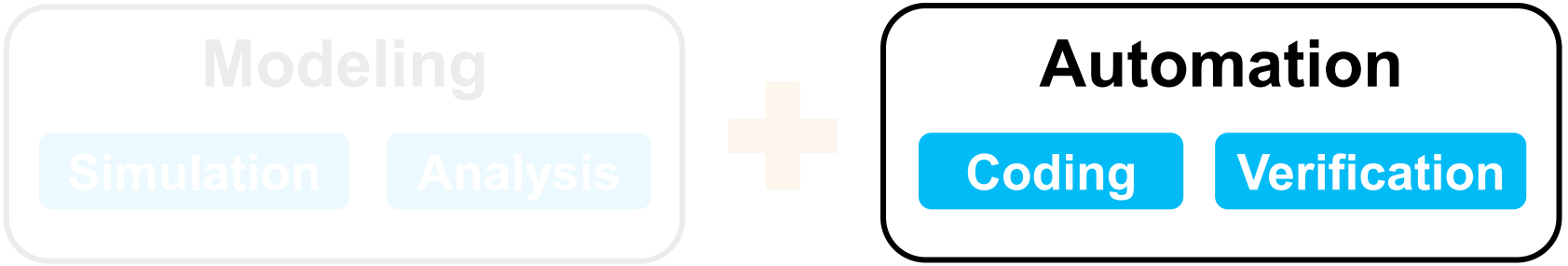


Modeling

Simulation **Analysis**

Model-Based Design

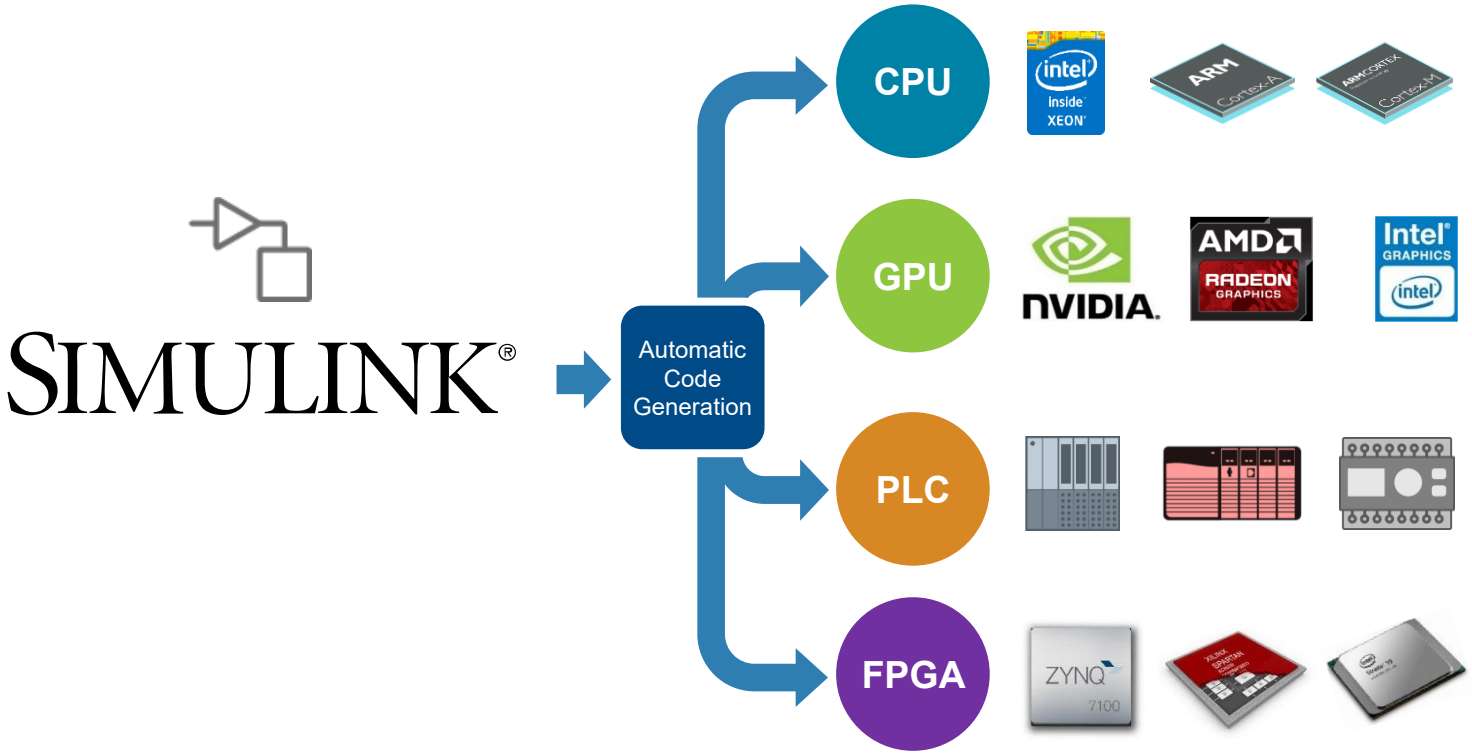
Systematic use of **models** throughout the development process



Automatically generate code for prototyping and production

Automation

Coding | Verification



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

MathWorks supports deployment to a variety of low-cost hardware

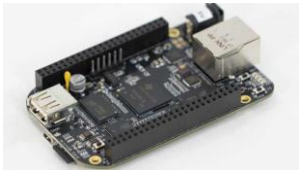
Arduino

Lego EV3

Raspberry Pi

Android/iOS Devices

Kinect for Windows



BeagleBone Black

Texas Instruments

STM Electronics

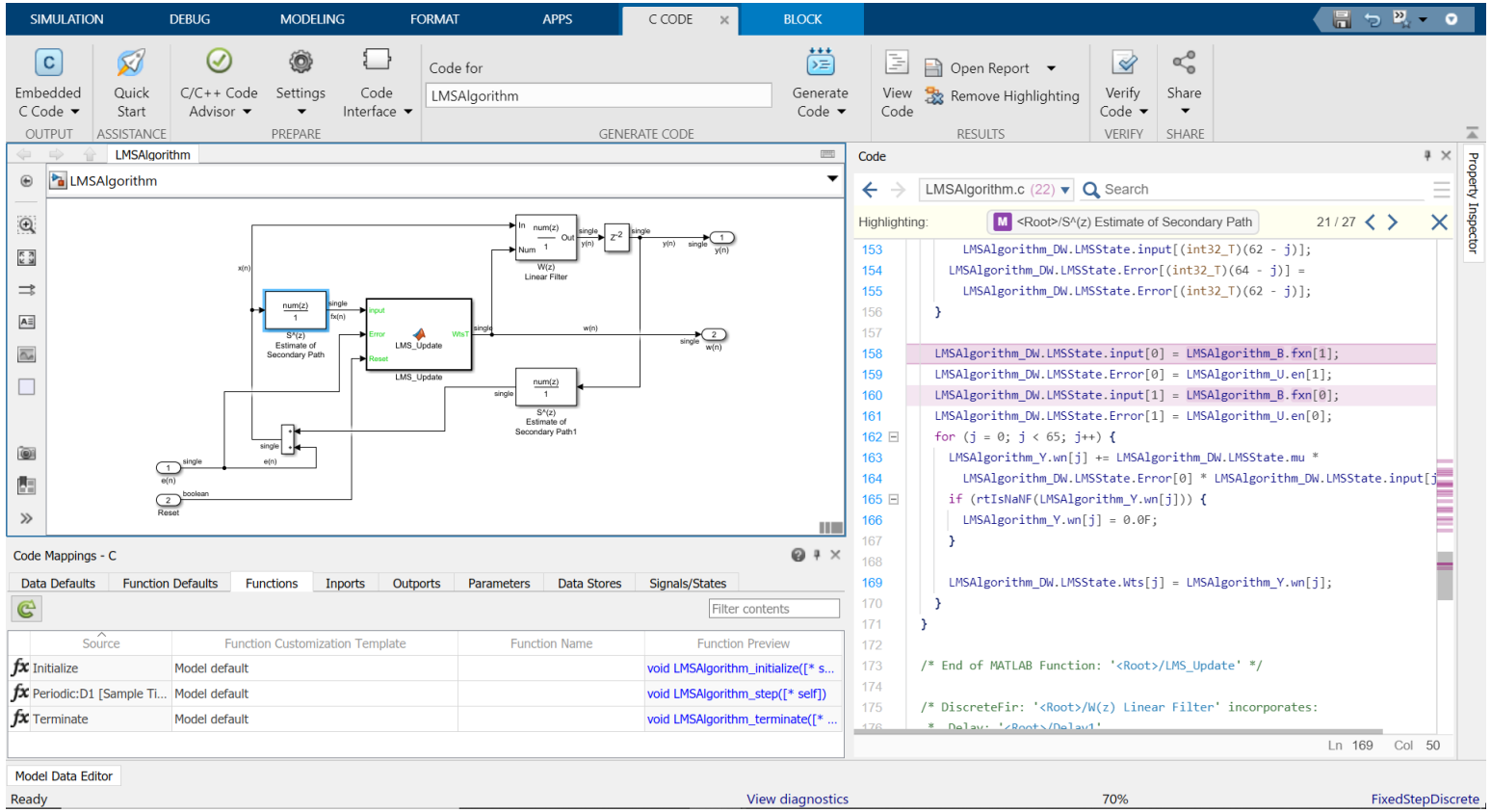
Freescale

Zynq SDR

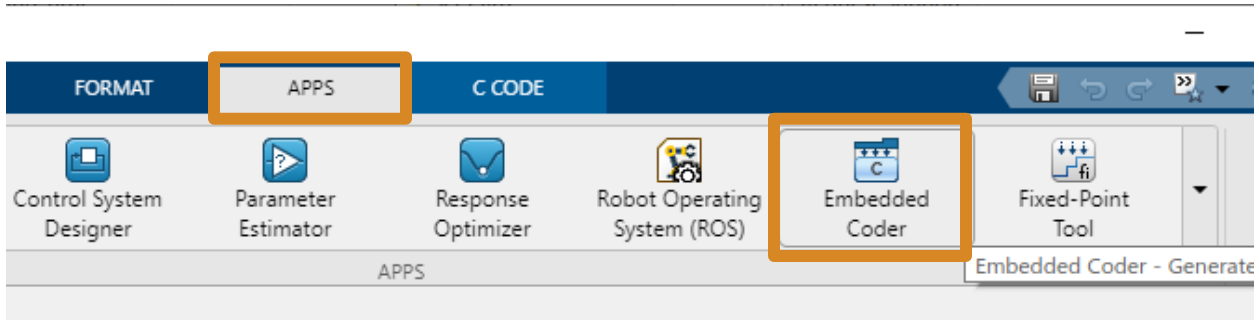
How do we go from model to code?

Embedded Coder – Generate C/C++ code optimized for embedded systems

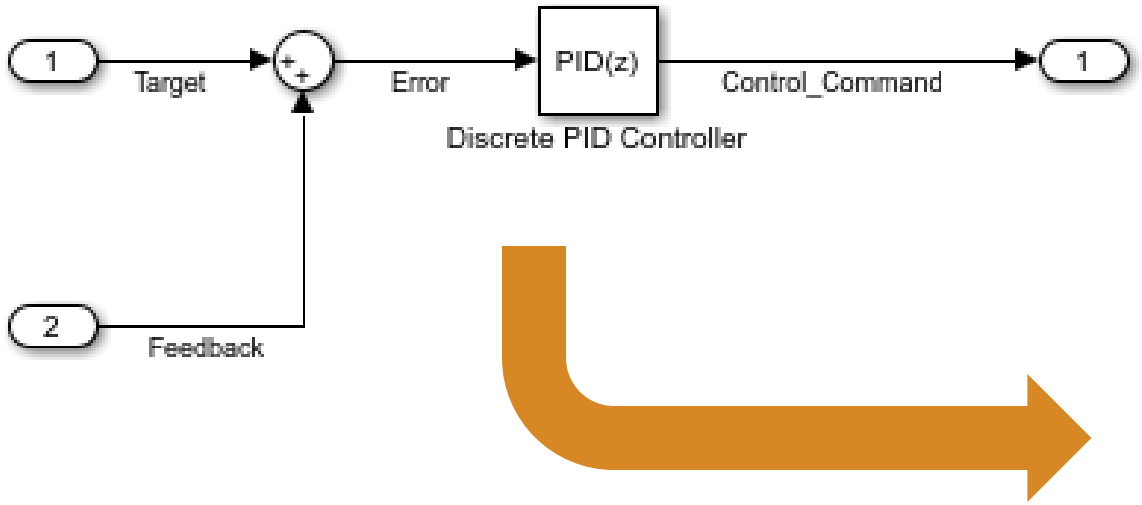
- **Improve** efficiency in design iterations
- **Trace** among requirements, model, code, and tests
- **Streamline** code integration with customizable code interfaces and appearances
- **Leverage** hardware optimized code including DSP library for ARM processors



Embedded Coder



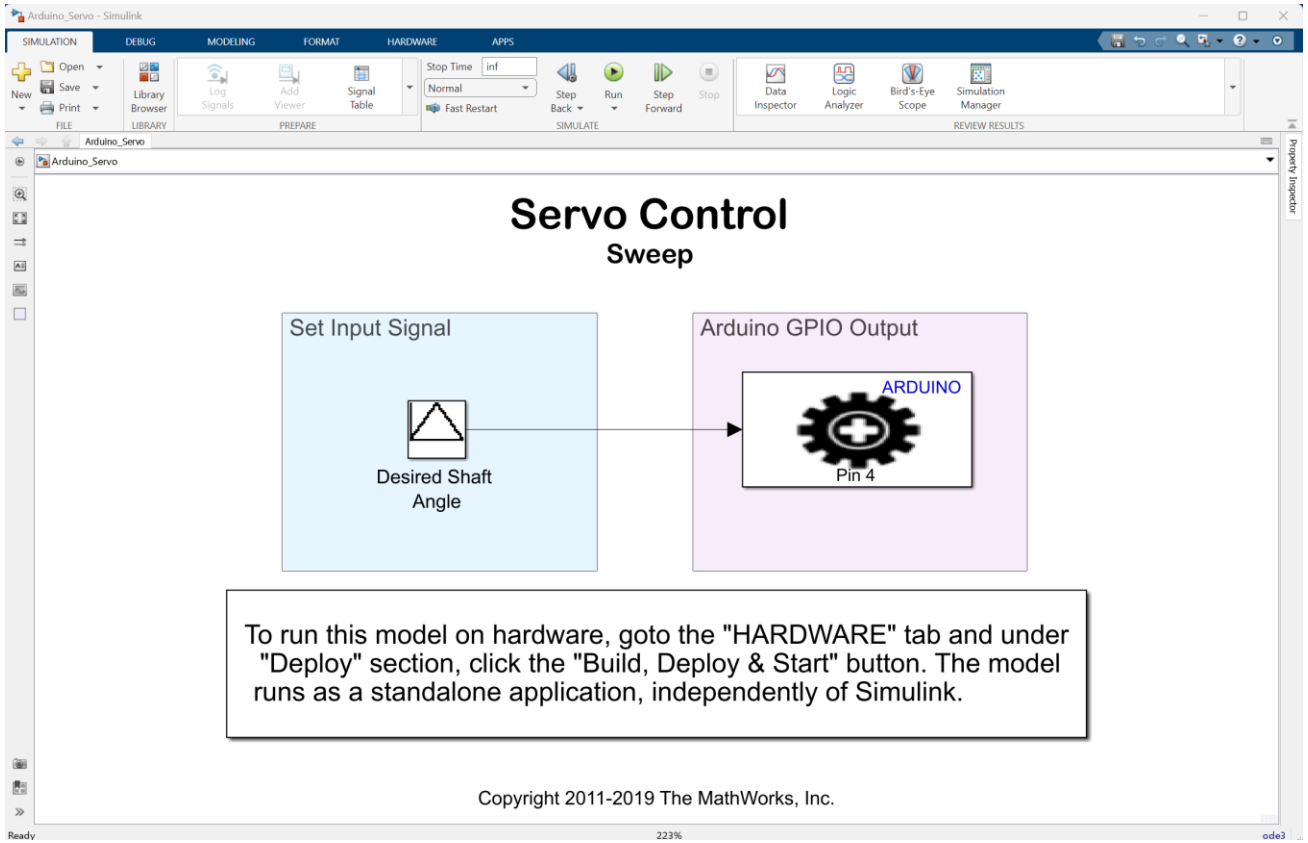
Access code generation tools from the Embedded Coder App



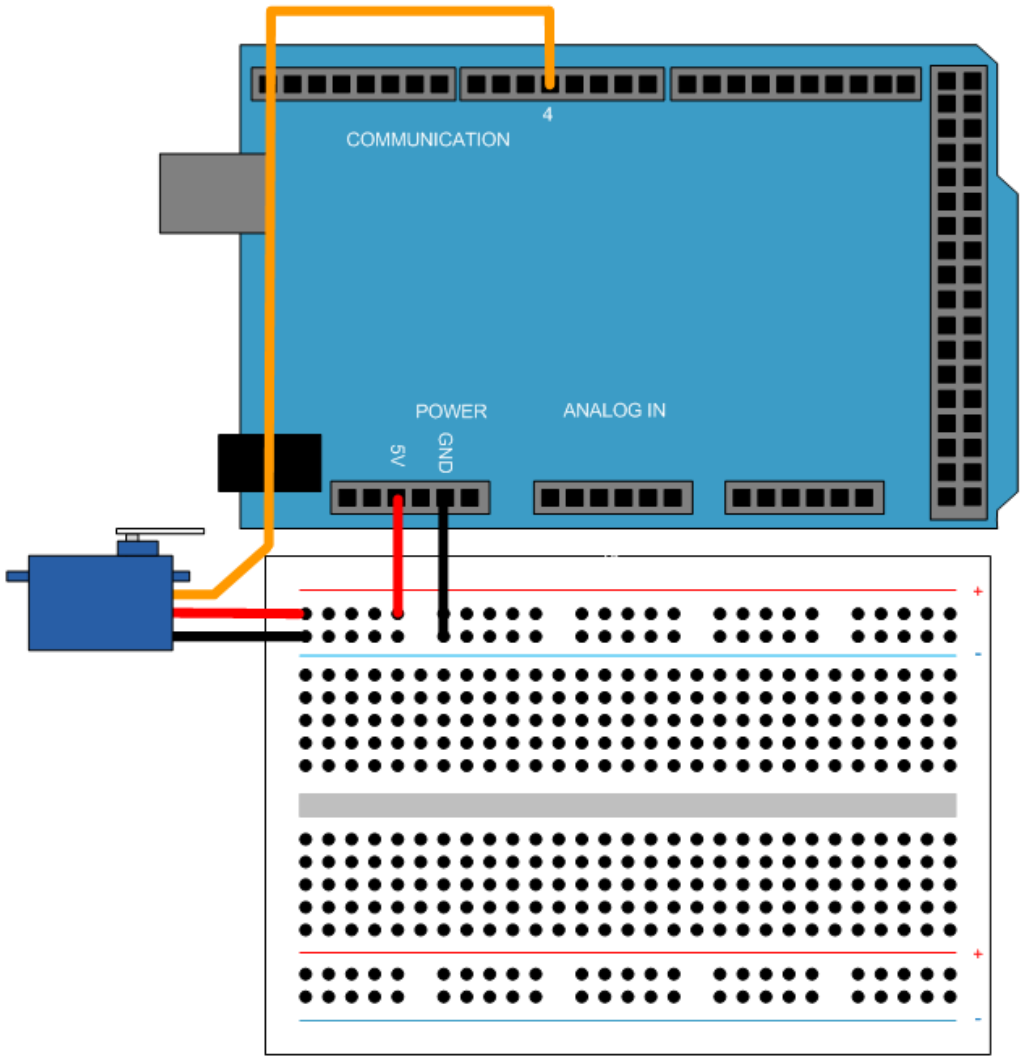

```
32
33  /* Model step function */
34  void codegen_example_step(void)
35  {
36    real_T rtb_Error;
37    real_T rtb_Tsamp;
38
39    /* Sum: '<Root>/Sum' incorporates:
40     *  Inport: '<Root>/Feedback'
41     *  Inport: '<Root>/Target'
42     */
43    rtb_Error = rtU.Target + rtU.Feedback;
44
45    /* SampleTimeMath: '<S30>/Tsamp' incorporates:
46     *  Gain: '<S27>/Derivative Gain'
47     *
48     * About '<S30>/Tsamp':
49     *  y = u * K where K = 1 / ( w * Ts )
50     */
51    rtb_Tsamp = rtP.DiscretePIDController_D * rtb_Error * rtP.Tsamp_WtEt;
52
53    /* Outport: '<Root>/Control_Command' incorporates:
```

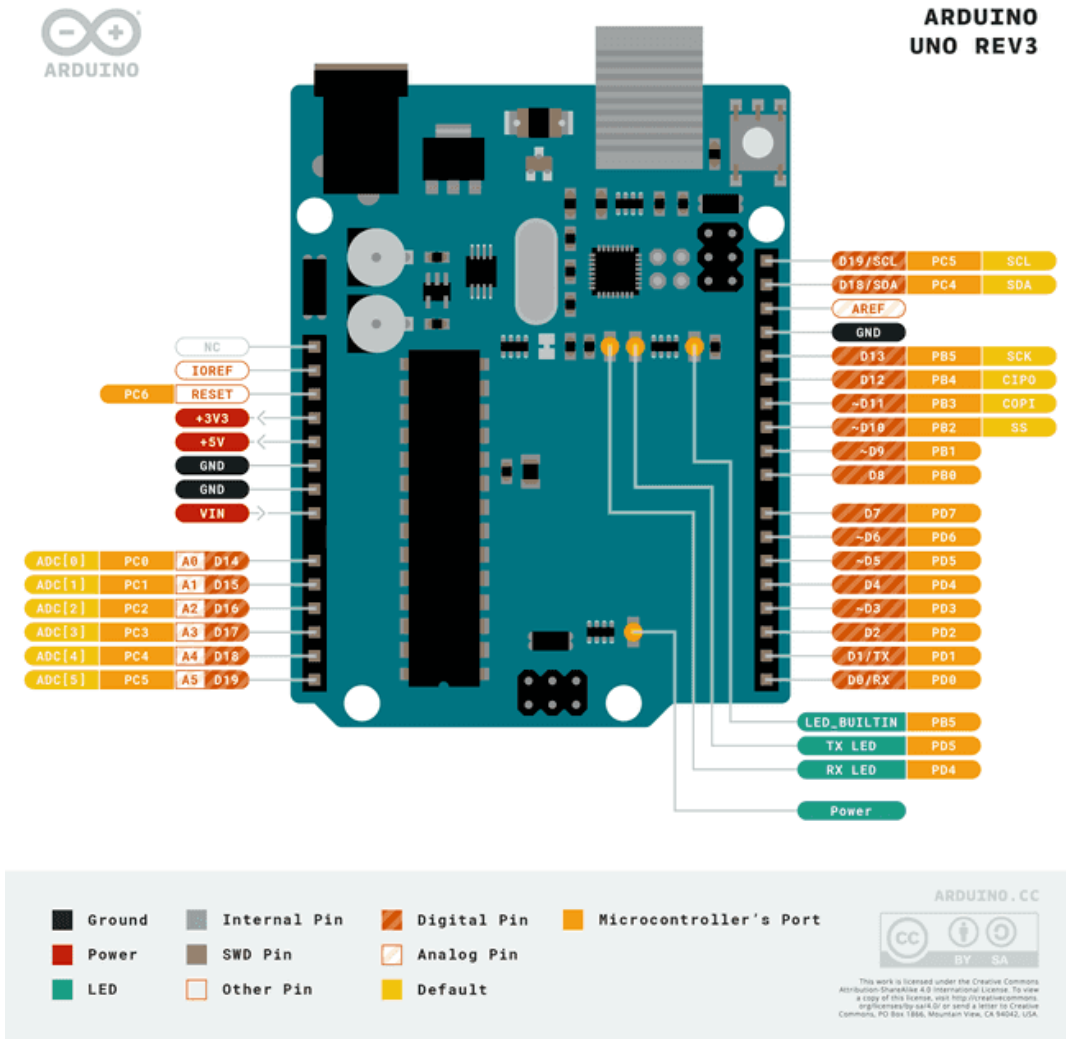
Exercise 4: Control a Servo with Arduino and Simulink

- Open: ***Arduino_Servo.slx***
- Add desired block as input
- Select output GPIO pin
- Run model with various settings:
  - ***Connected IO***
  - ***Monitor & Tune***
  - ***Build, Deploy, & Start***



Exercise 4: Control a Servo with Arduino and Simulink

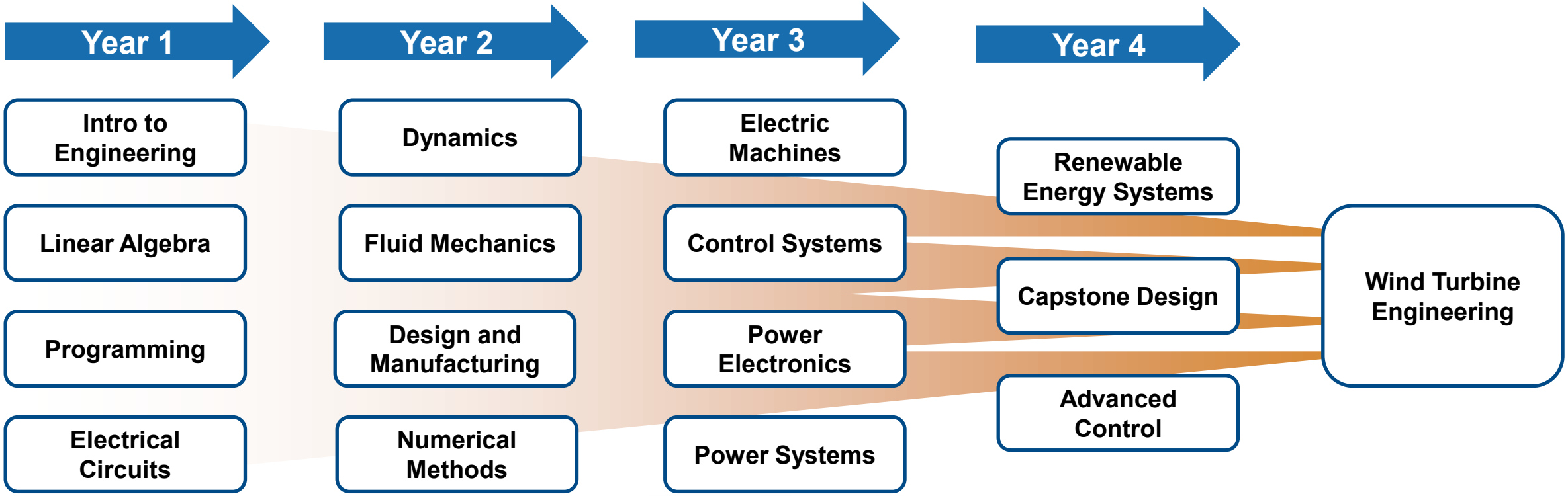




Agenda

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

Introduce Simulink as a modeling tool from foundational courses

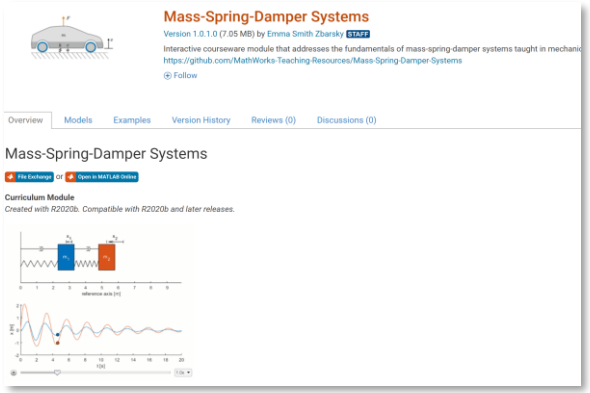


Fundamental Understandings & Skills

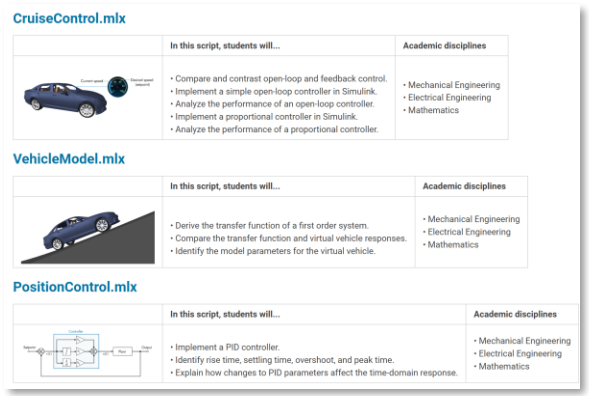
Core Engineering Topics & Subsystem Design

System Design & Implementation

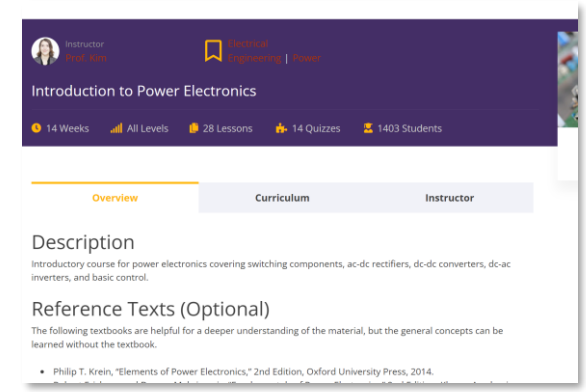
Modular courseware and virtual labs



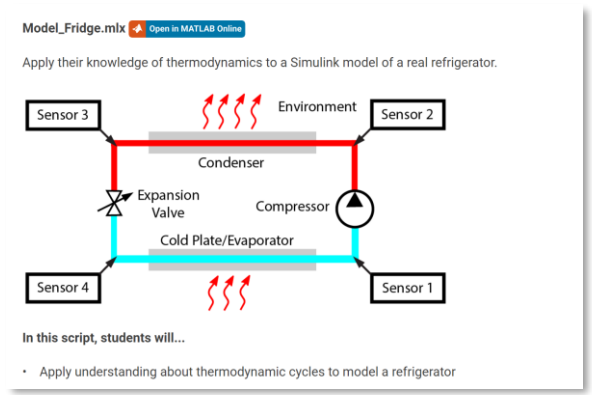
[Mass-Spring-Damper Systems]



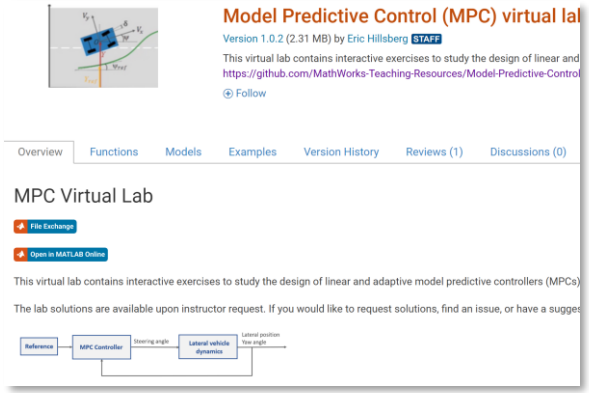
[Virtual Hardware and Labs for Controls]



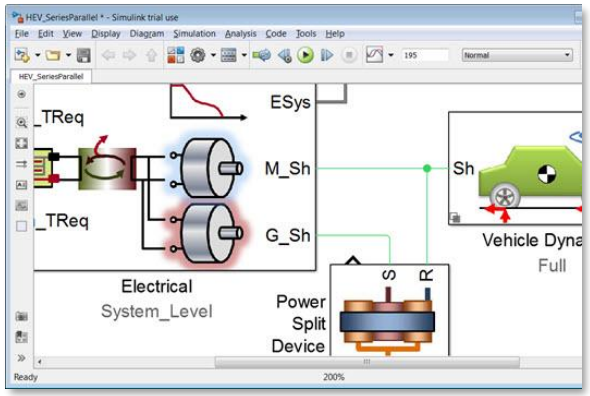
[Introduction to Power Electronics]



[Thermodynamics]

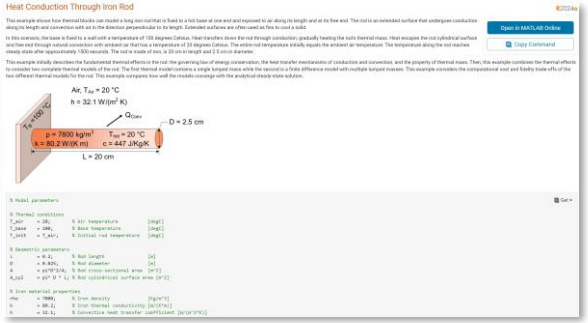


[Model Predictive Control]

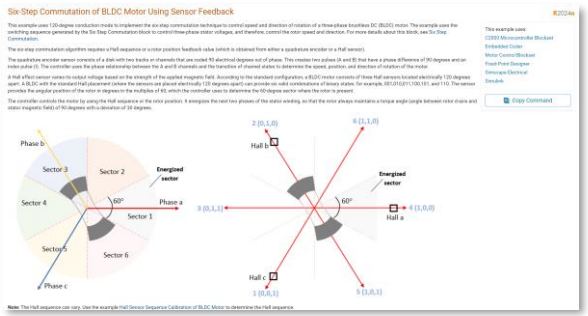


[Hybrid-Electric Vehicle Model in Simulink & Simscape]

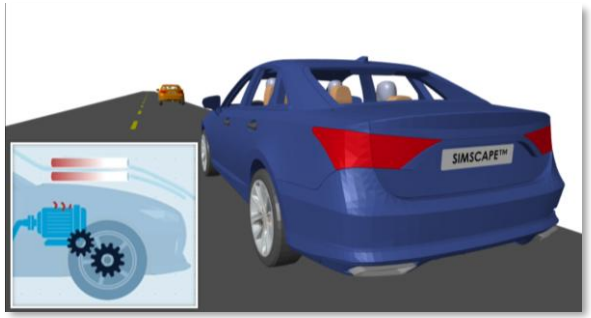
Reference examples in documentation and file exchange



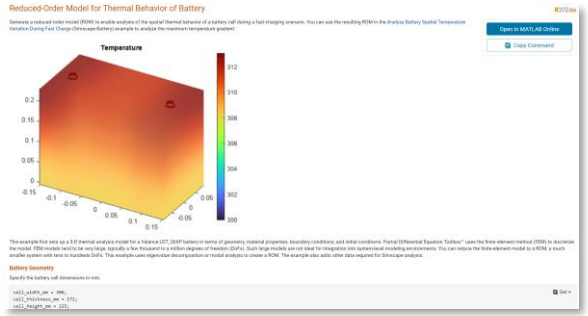
Heat Conduction Through Iron Rod



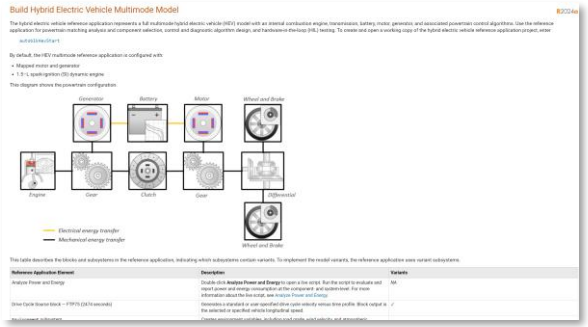
Six-Step Commutation of BLDC Motor Using Sensor Feedback



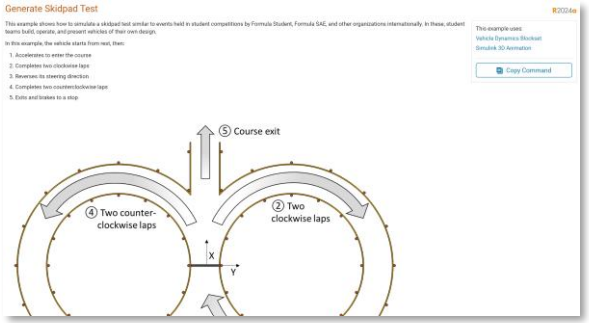
Battery Electric Vehicle with Motor Cooling in Simscape



Reduced-Order Model for Thermal Behavior of Battery



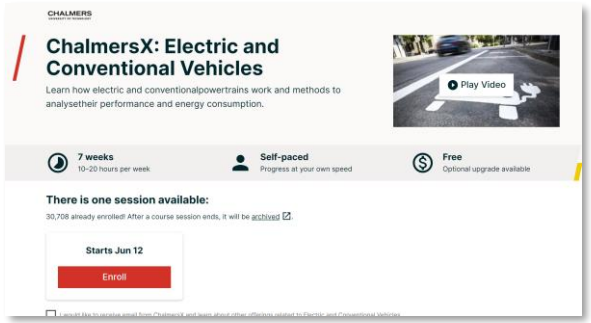
Build Hybrid Electric Vehicle Multimode Model



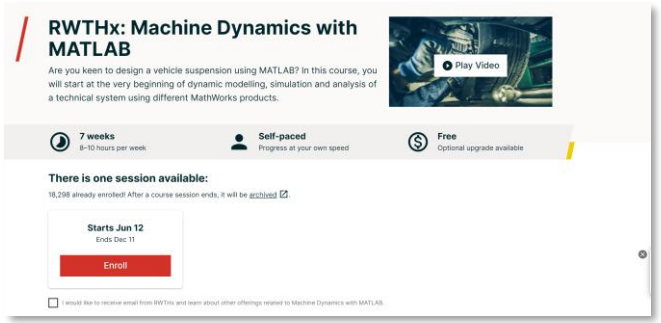
Generate Skidpad Test

Examples with Extended Descriptions and Educational Materials

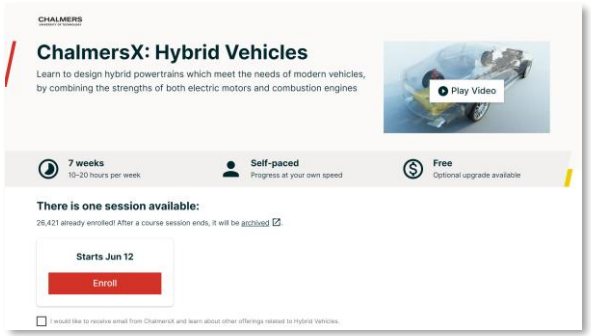
Textbooks and MOOCs



[ChalmersX: Electric and Conventional Vehicles]

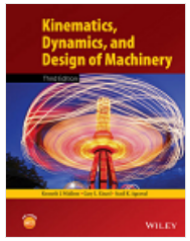


[RWTHx: Machine Dynamics with MATLAB]

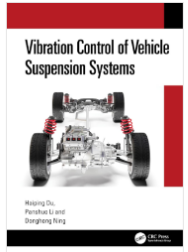


[ChalmersX: Hybrid Vehicles]

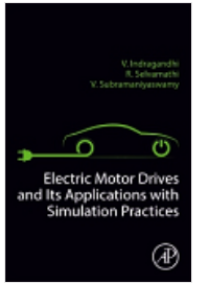
[Averaged-Switch Modeling and Simulation]



Kinematics, Dynamics, and Design of Machinery, 3rd edition
..., **automotive** and production engineering.
Author: Kenneth J. Waldron, Gary L. Kinzel, Sunil K. Agrawal
Copyright: 2016
Language: English
Publisher: John Wiley & Sons, Inc.
Companion Software Available



Vibration Control of Vehicle Suspension Systems
...researchers and graduate students in **automotive** engineering, vehicle vibration, mechatronics, control systems, applied mechanics, and vehicle dynamics....
Author: Haiping Du, Panshuo Li, Donghong Ning
Copyright: 2024
Language: English
Publisher: CRC Press, Inc.

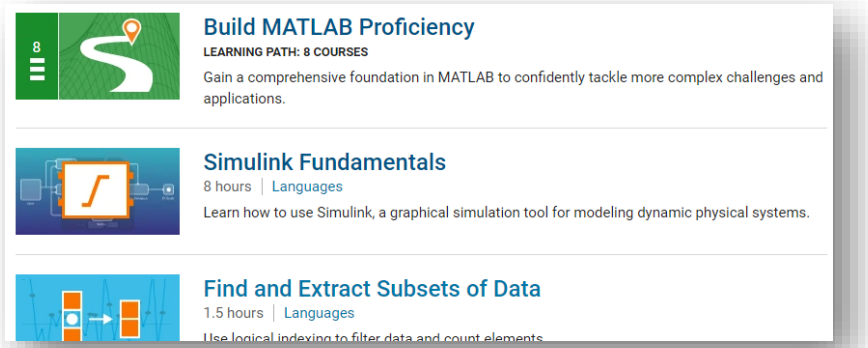


Electric Motor Drives and their Applications with Simulation Practices
Electric **Motor** Drives and Its Applications with Simulation Practices provides comprehensive coverage of the concepts of electric **motor** drives and their applications, along with their simulation using...
Author: Dr. V. Indra Gandhi, R. Selvamathi, V. Subramaniyaswamy
Copyright: 2022
Language: English
Publisher: Elsevier Science

Companion Codes and Models to Enhance the Learning Objective

Self-Paced Online Courses

http://matlabacademy.mathworks.com



- 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

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

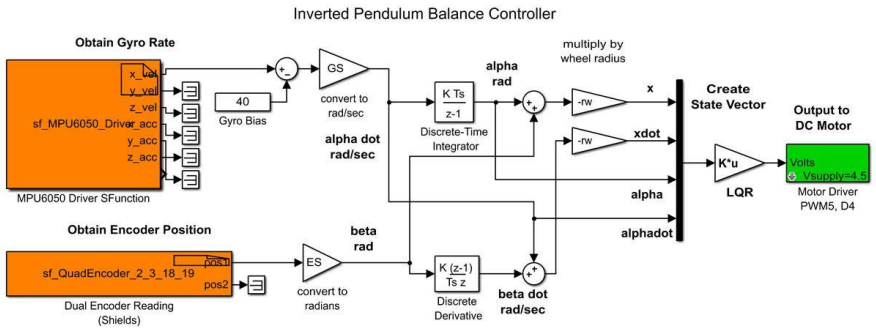
Long courses (7-10 hours)

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

Teaching Mechatronics at Rensselaer Polytechnic Institute

- **Challenge**
  - Need to **bridge multiple domains within a single course**
  - Introduce meaningful **hands-on experience with real-world hardware** effects **without** requiring **too much low-level coding** or time spent just getting things to work.
- **Solution**
  - Instructors build on students' foundation in electrical and mechanical domains from sophomore and junior year using MATLAB & Simulink
  - **MATLAB & Simulink** provide multi-domain modeling, simulation, and control design of the MinSeg
  - **Simulink Support Package for Arduino** enables direct deployment to hardware
- **Key Outcomes**
  - Students **focus on system-level thinking** instead of syntax and wiring
  - Iterative and more engaging labs
  - **Seamless transition** from **simulation to hardware**
  - **Reduced instructor burden** and more scalable labs

MinSeg inverted pendulum robot



Balance controller

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 min to submit a survey!