

# Systems Modeling and Control with Simulink and Simscape

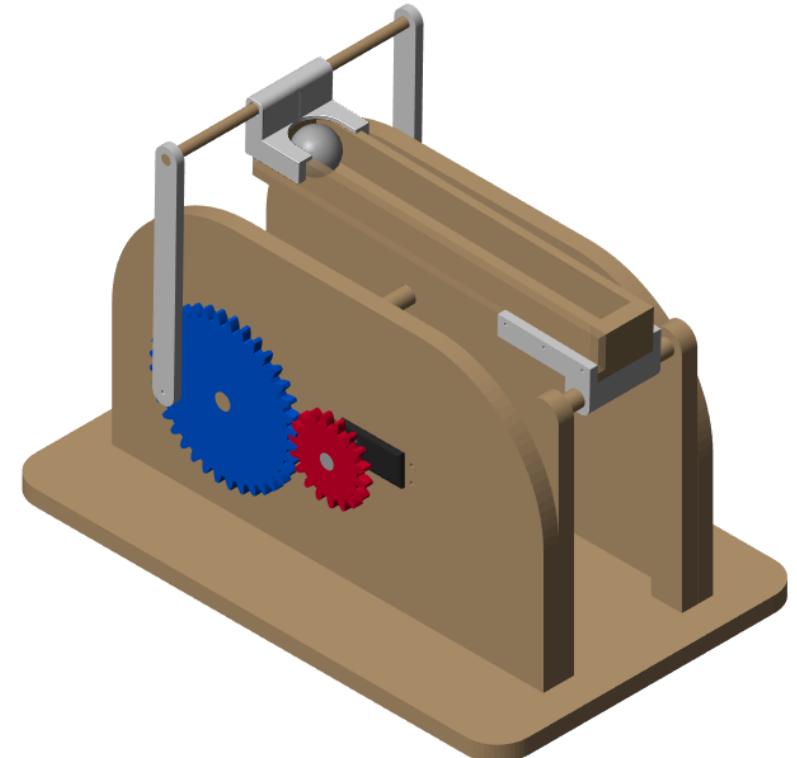
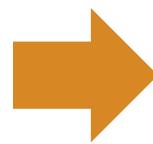
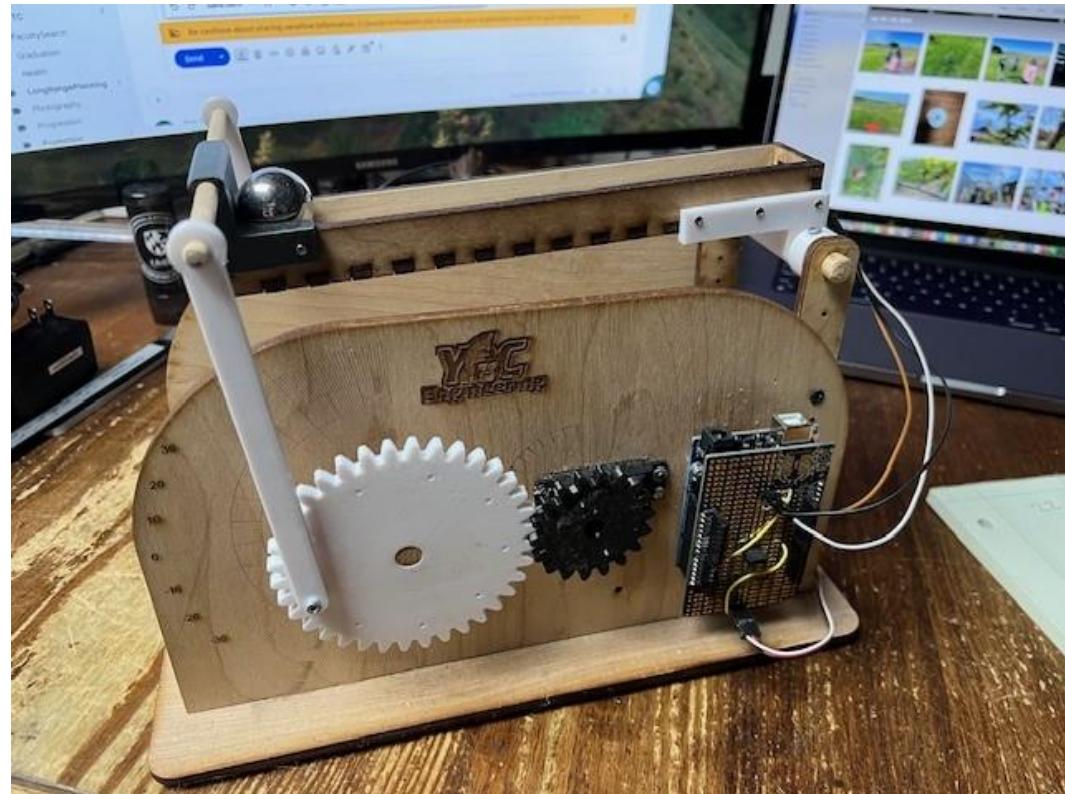
**Noah Roberts**

*Education Application Engineer*

# Agenda

- Simulink Overview
- Introduction to Model-Based Design
- Various ways of modeling a DC Motor
  - Mathematical Modeling with Simulink
  - Physical Modeling with Simscape
- Control System design
- Automatic Code Generation
- Resources and Support

# Great Demo



# Agenda

- **Simulink Overview**
- Introduction to Model-Based Design
- Various ways of modeling a DC Motor
  - Mathematical Modeling with Simulink
  - Physical Modeling with Simscape
- Control System design
- Automatic Code Generation
- Resources and Support

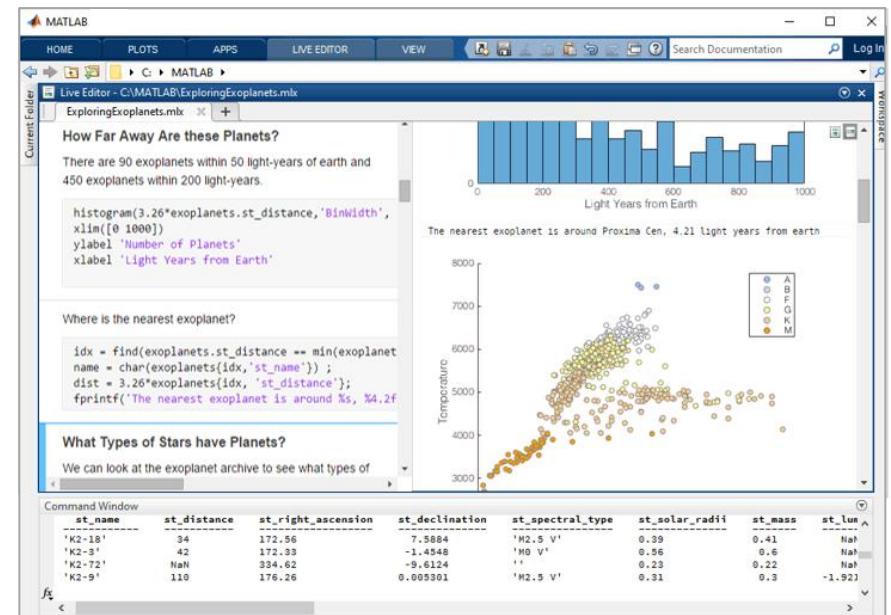
# Our Products

# 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 ~~SIMULINK~~ Toolbox



# SIMULINK®

## Simulation and Model-Based Design

### Model and simulate your system

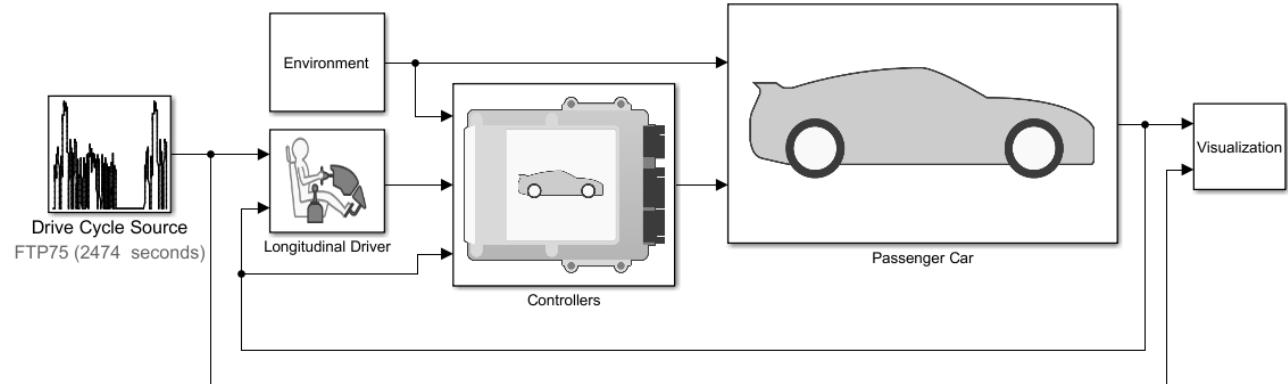
- Use one multi-domain environment
- Model the system under test and the plant
- Simulate closed-loop system behavior

### Test early and often

- Test your system under all conditions
- Validate your design with real-time testing
- Trace from requirements to design to code

### Automatically generate code

- Generate production-quality C and HDL code
- Deploy directly to embedded processors or FPGA's/ASIC's



# Where are Simulink and Simscape used?



Aerospace and Defense



Automotive



Biological Sciences



Biotech and Pharmaceutical



Communications



Electronics



Energy Production



Financial Services



Industrial Machinery



Medical Devices



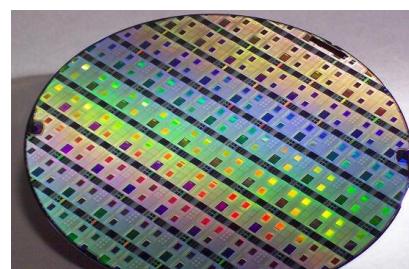
Metals, Materials, Mining



Neuroscience



Railway Systems



Semiconductors



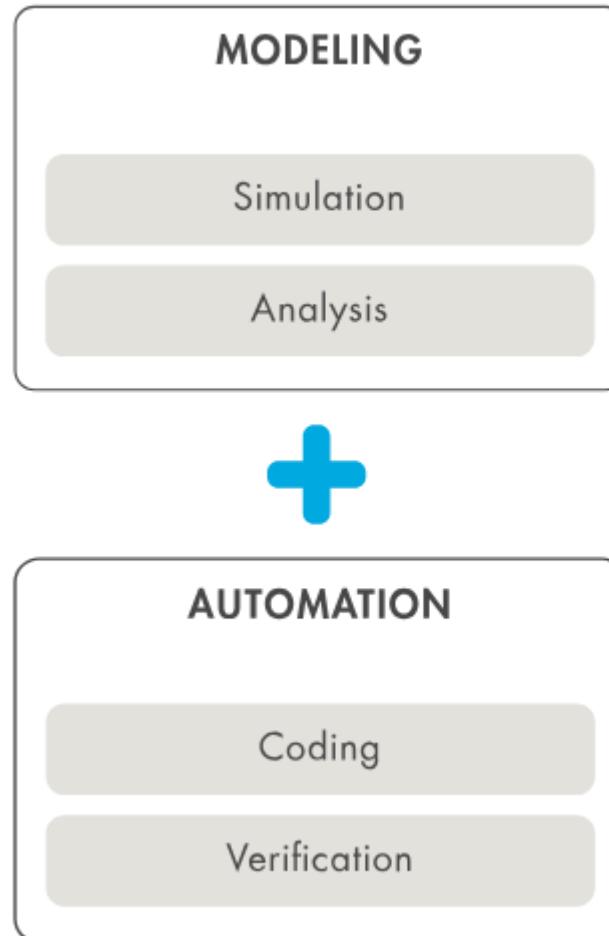
Software and Internet

# Agenda

- Simulink Overview
- **Introduction to Model-Based Design**
- Various ways of modeling a DC Motor
  - Mathematical Modeling with Simulink
  - Physical Modeling with Simscape
- Control System design
- Automatic Code Generation
- Resources and Support

# What is Model-Based Design?

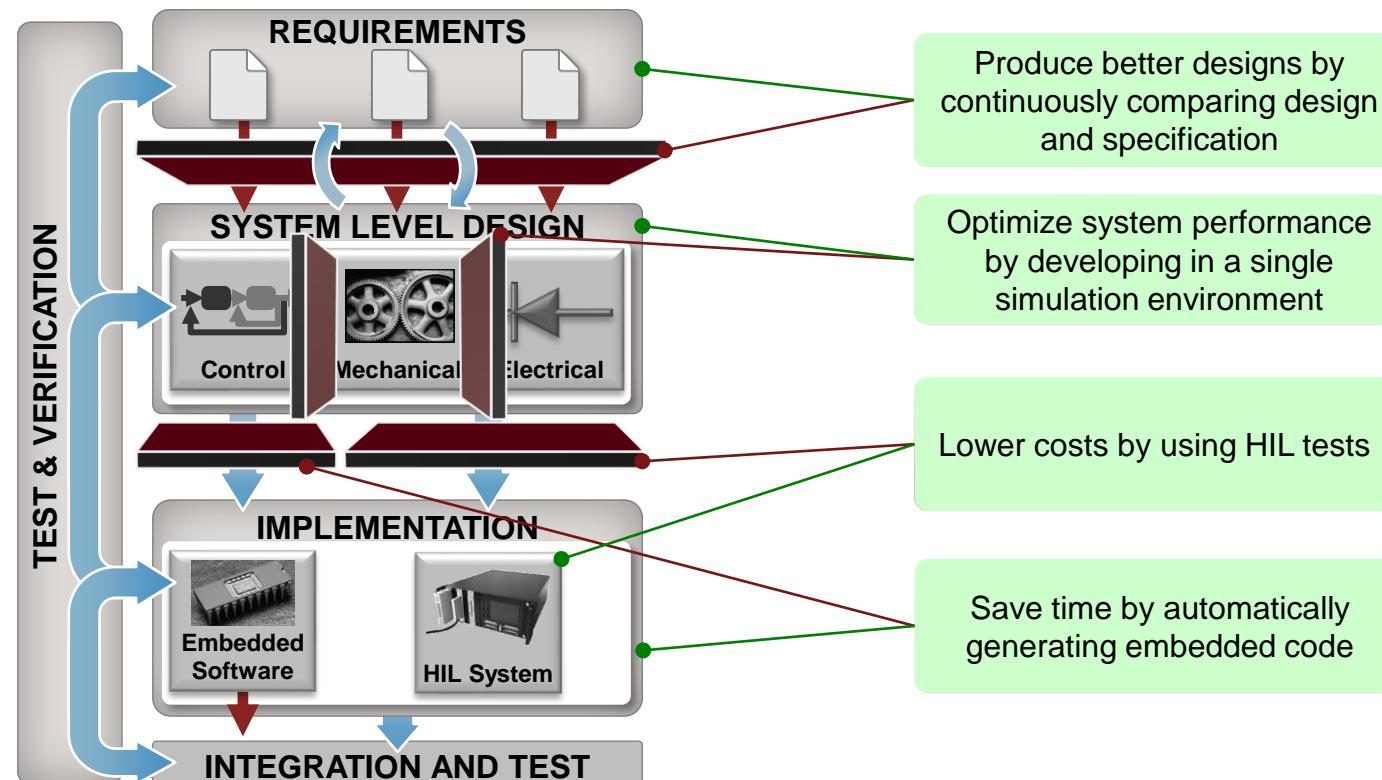
- Creating a virtual representation of a real-world system
  - **Model** new ideas and perform repeatable tests with **simulation**
  - Adjust **design** based on these simulations and **iterate**



Try out **new** ideas.  
Fast **repeatable** tests.

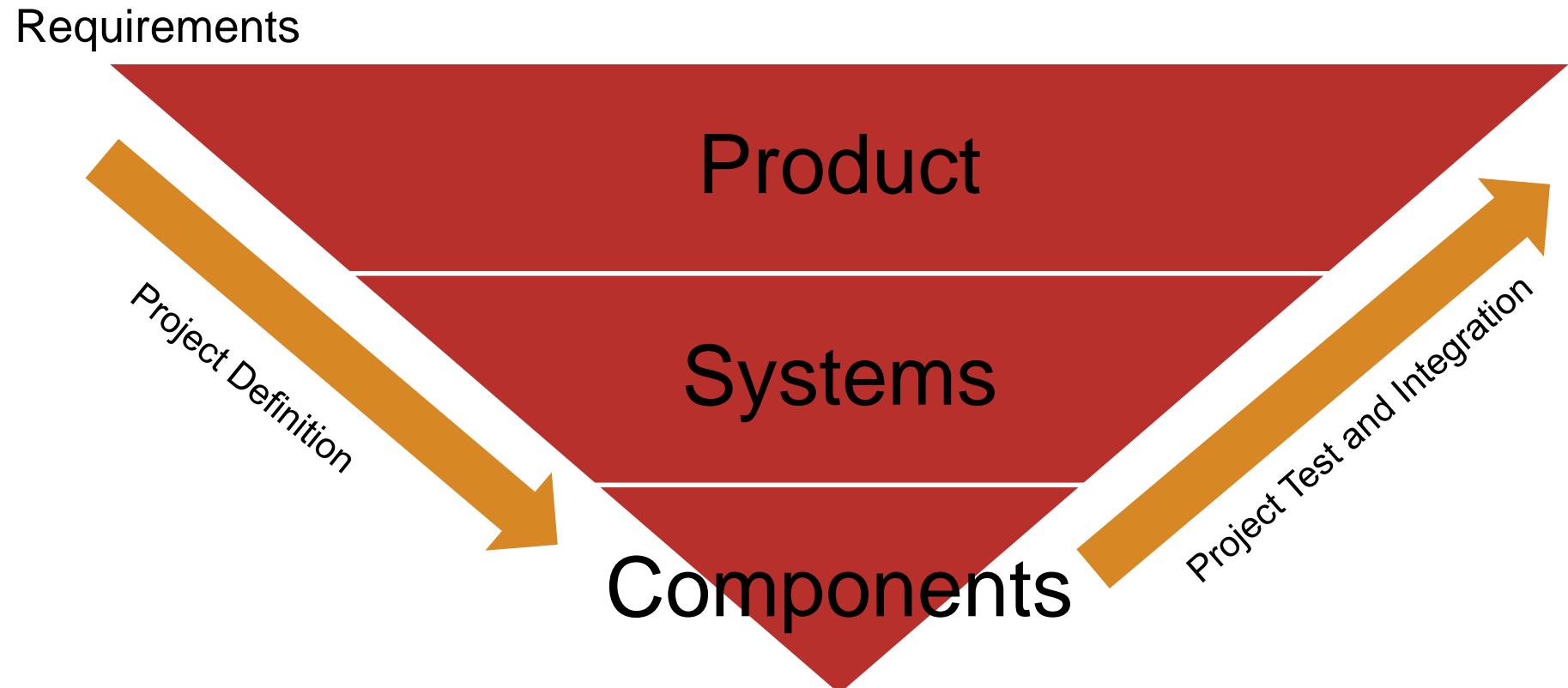
Eliminate **manual steps**  
and reduce **human error**.

# Model-Based Design Process

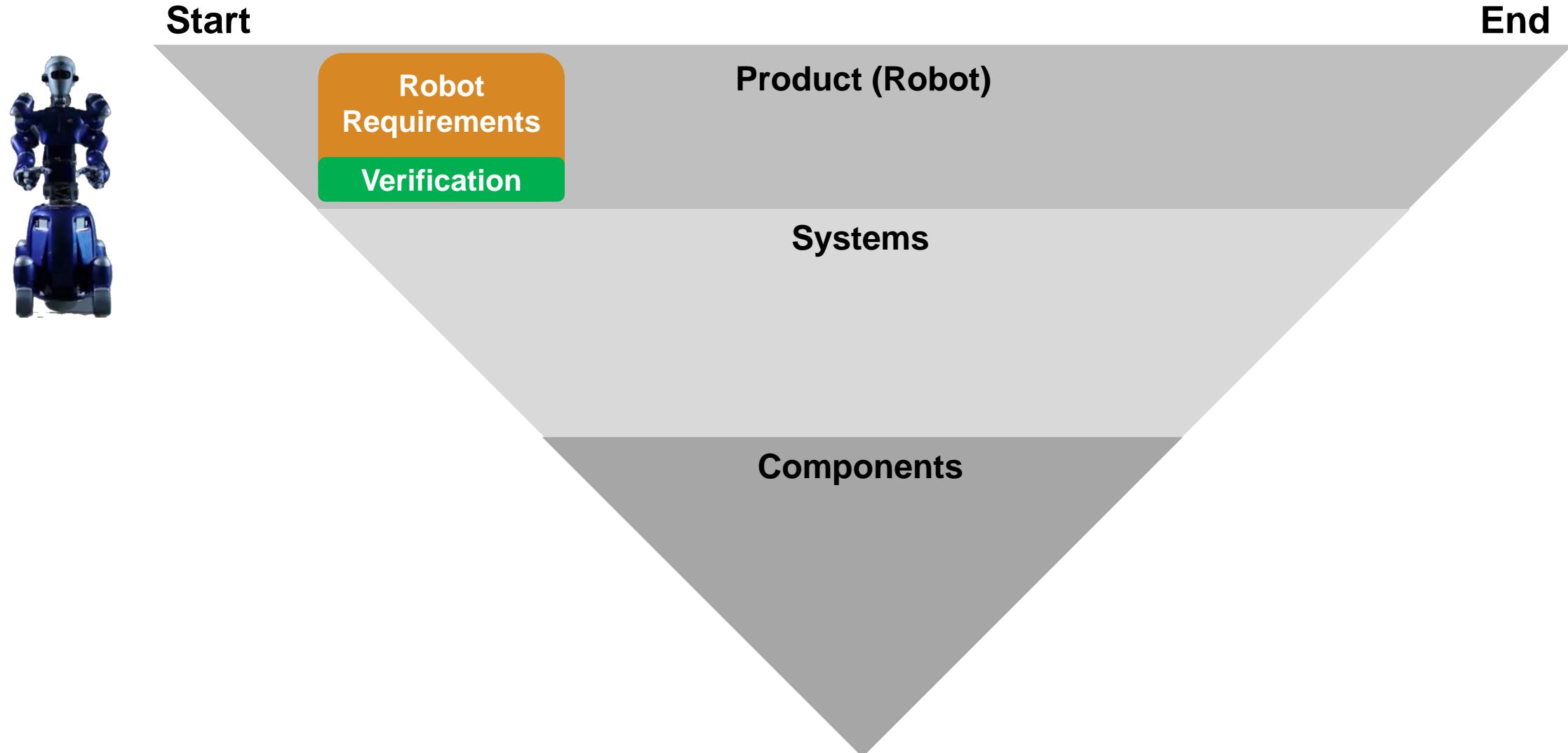




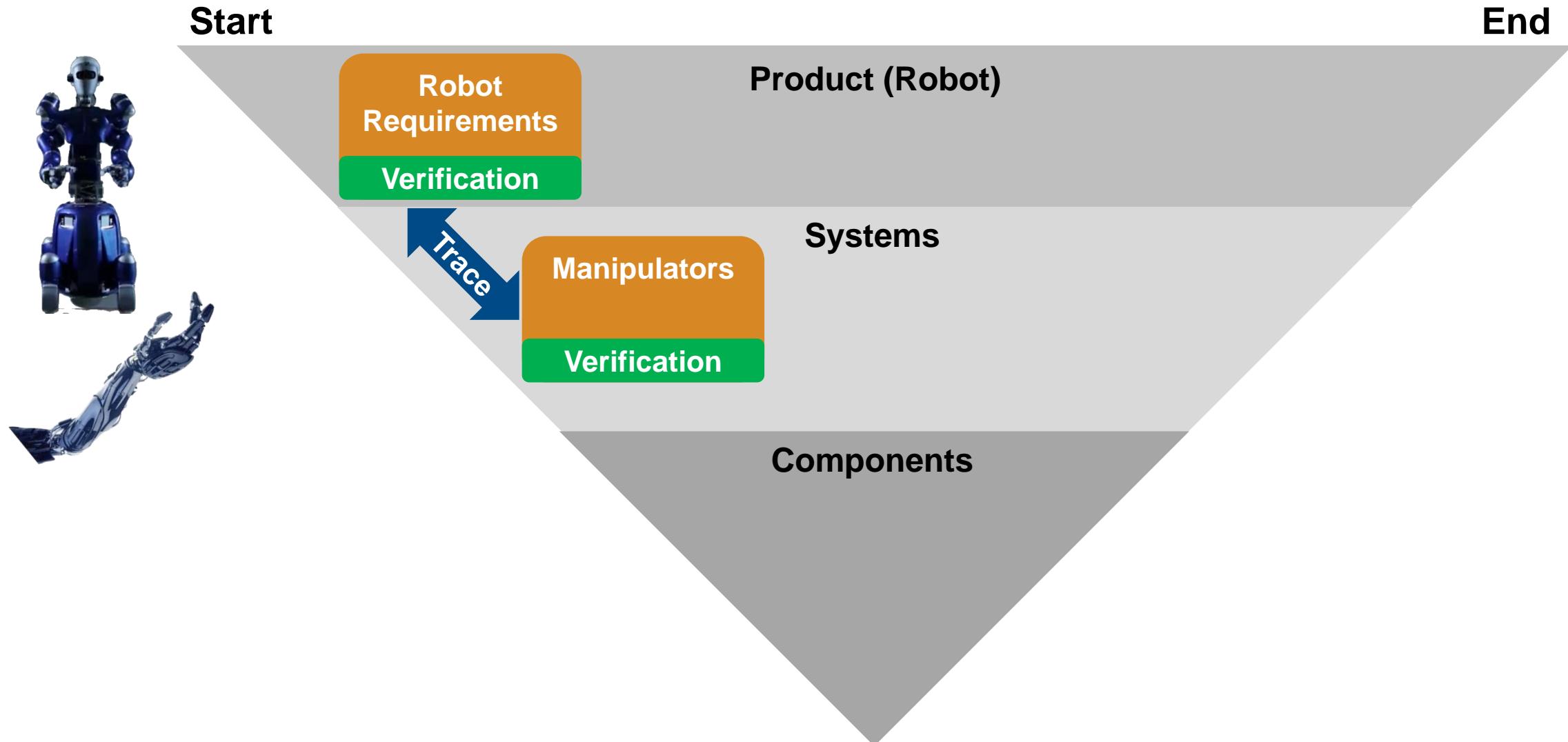
# V Diagram



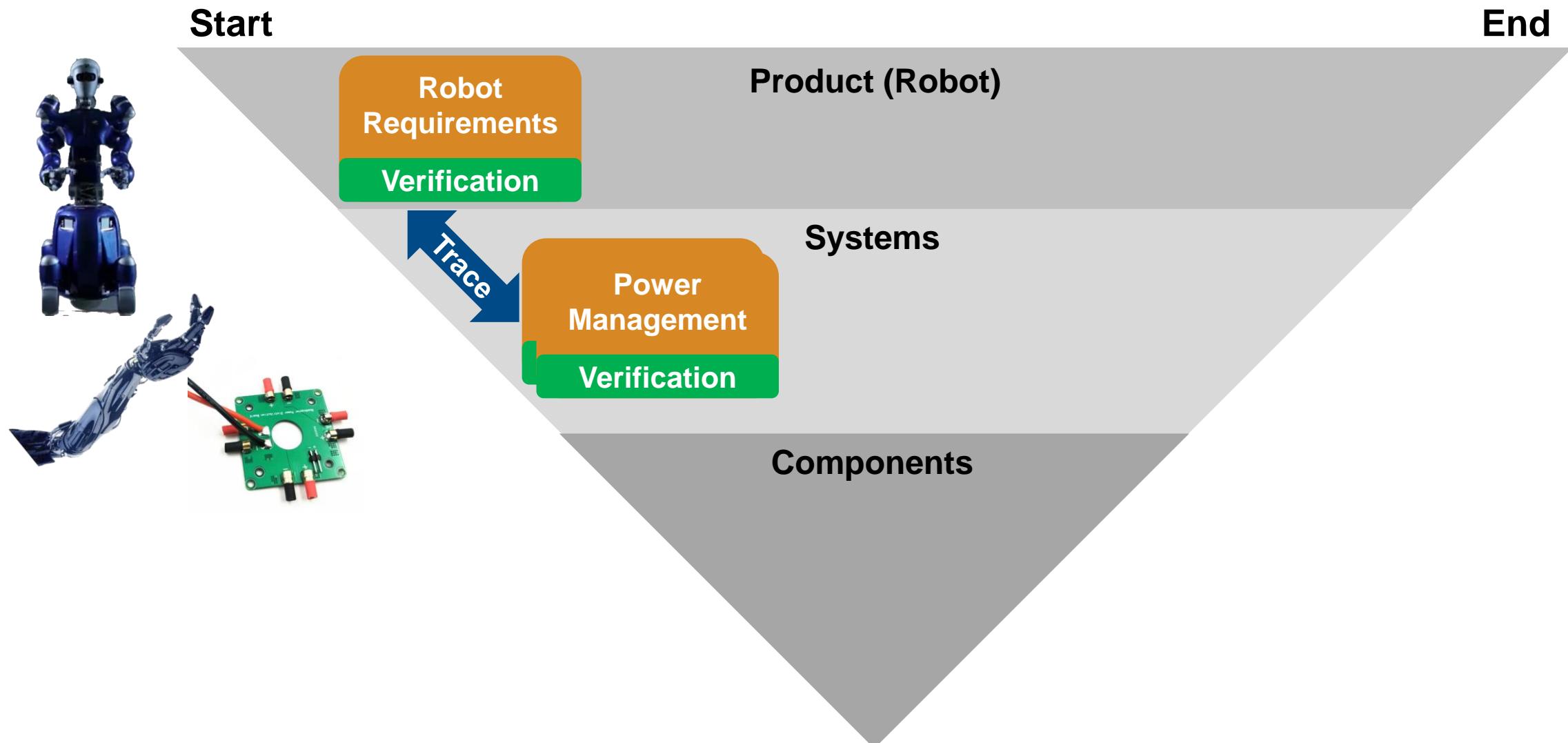
# Justin the Robot MBD Workflow



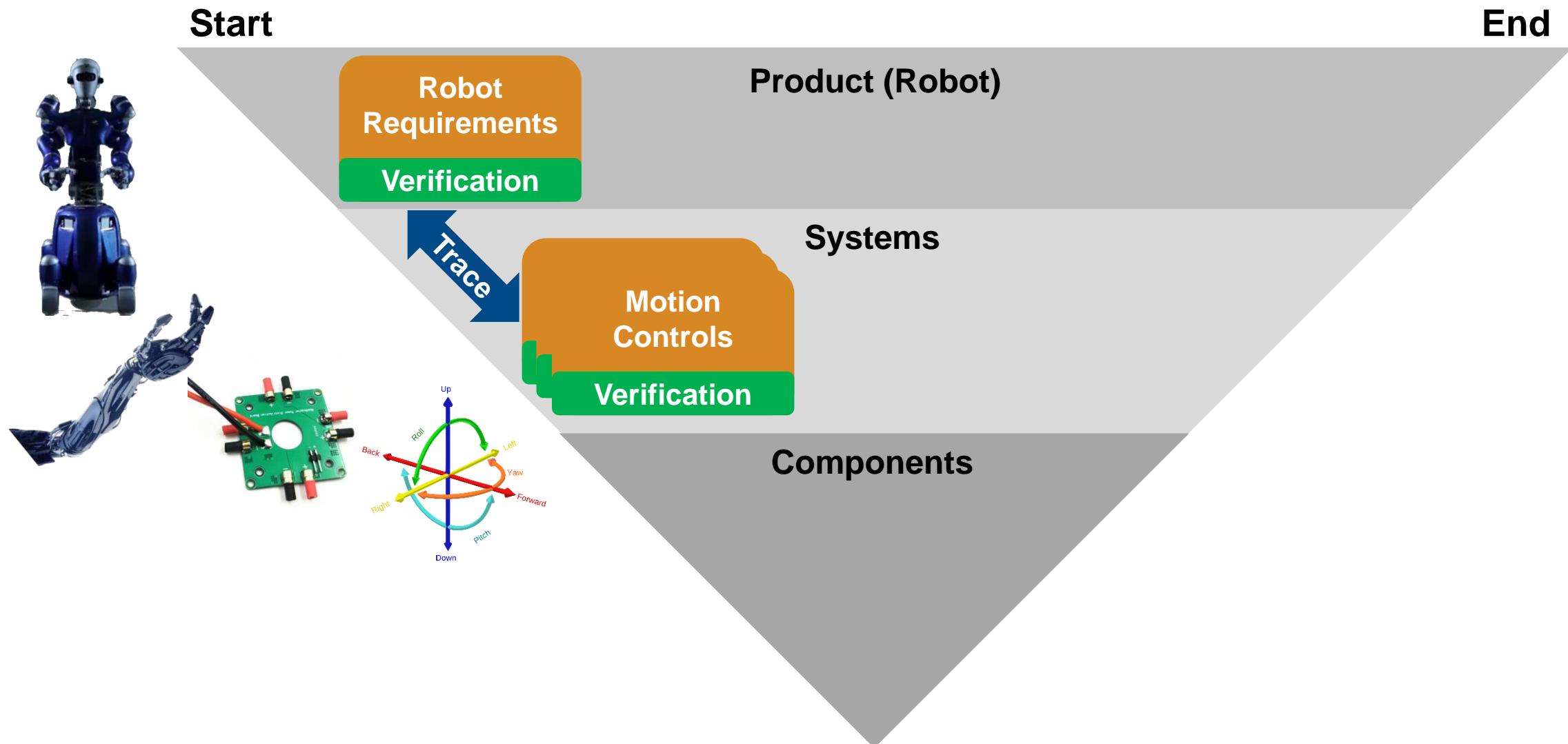
# Justin the Robot MBD Workflow



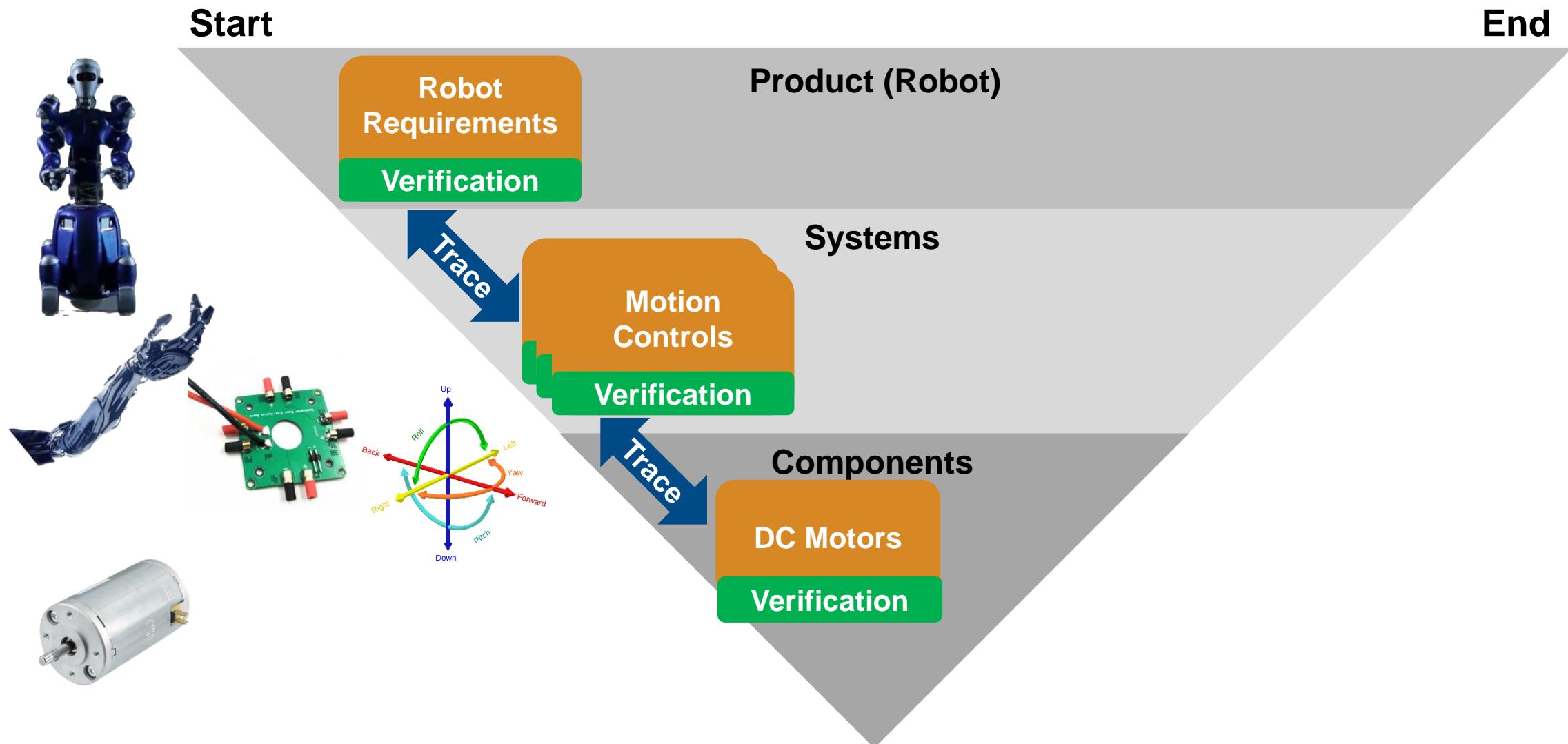
# Justin the Robot MBD Workflow



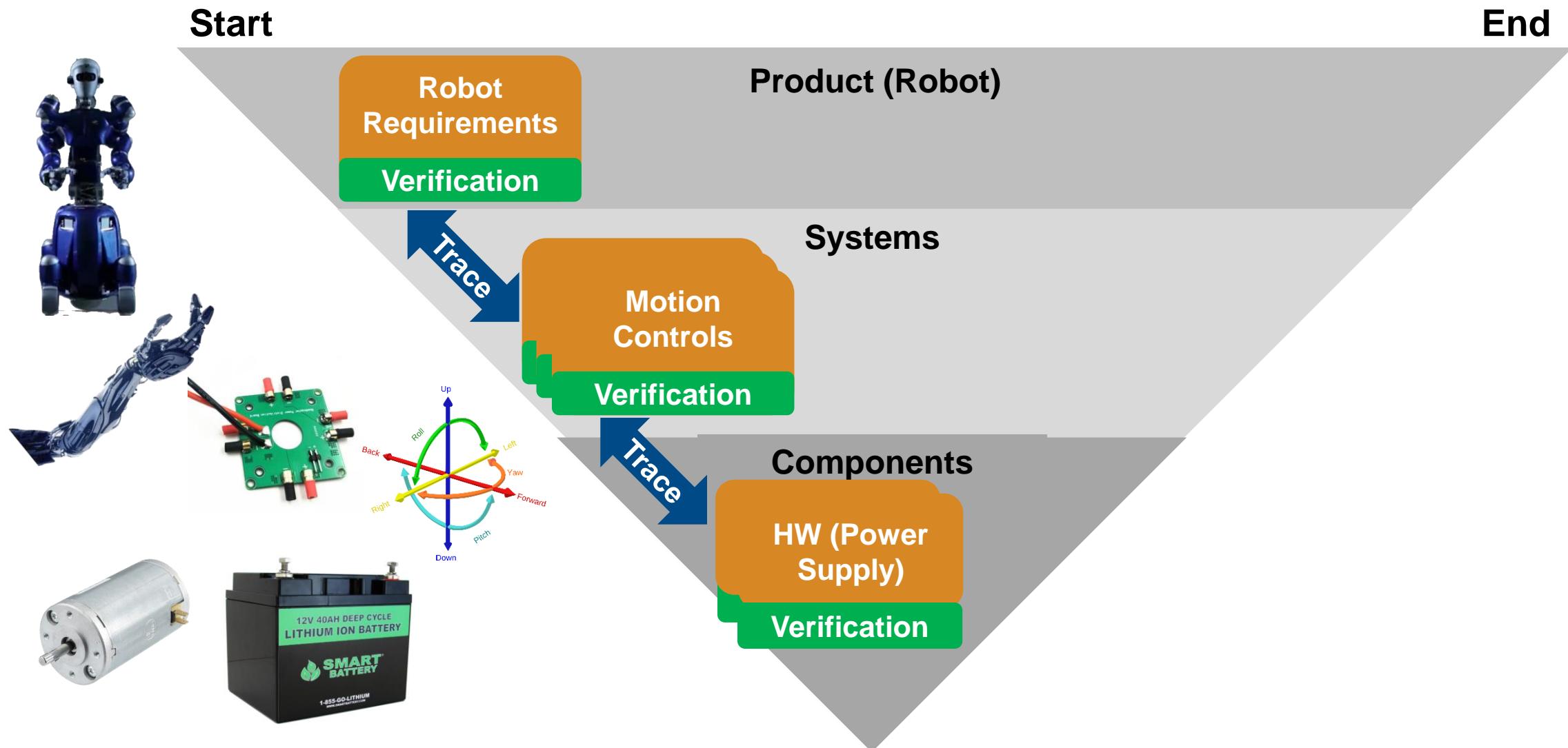
# Justin the Robot MBD Workflow



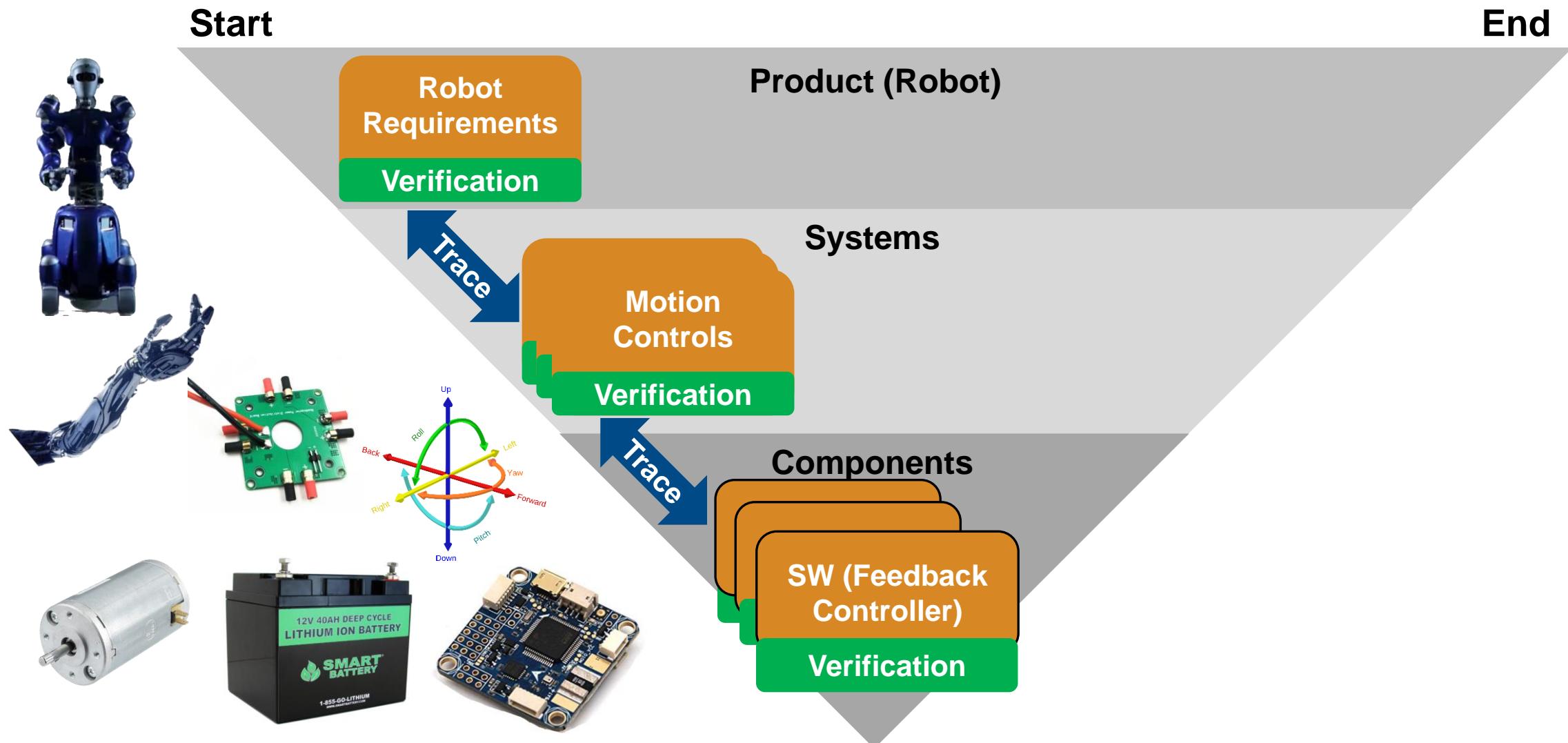
# Justin the Robot MBD Workflow



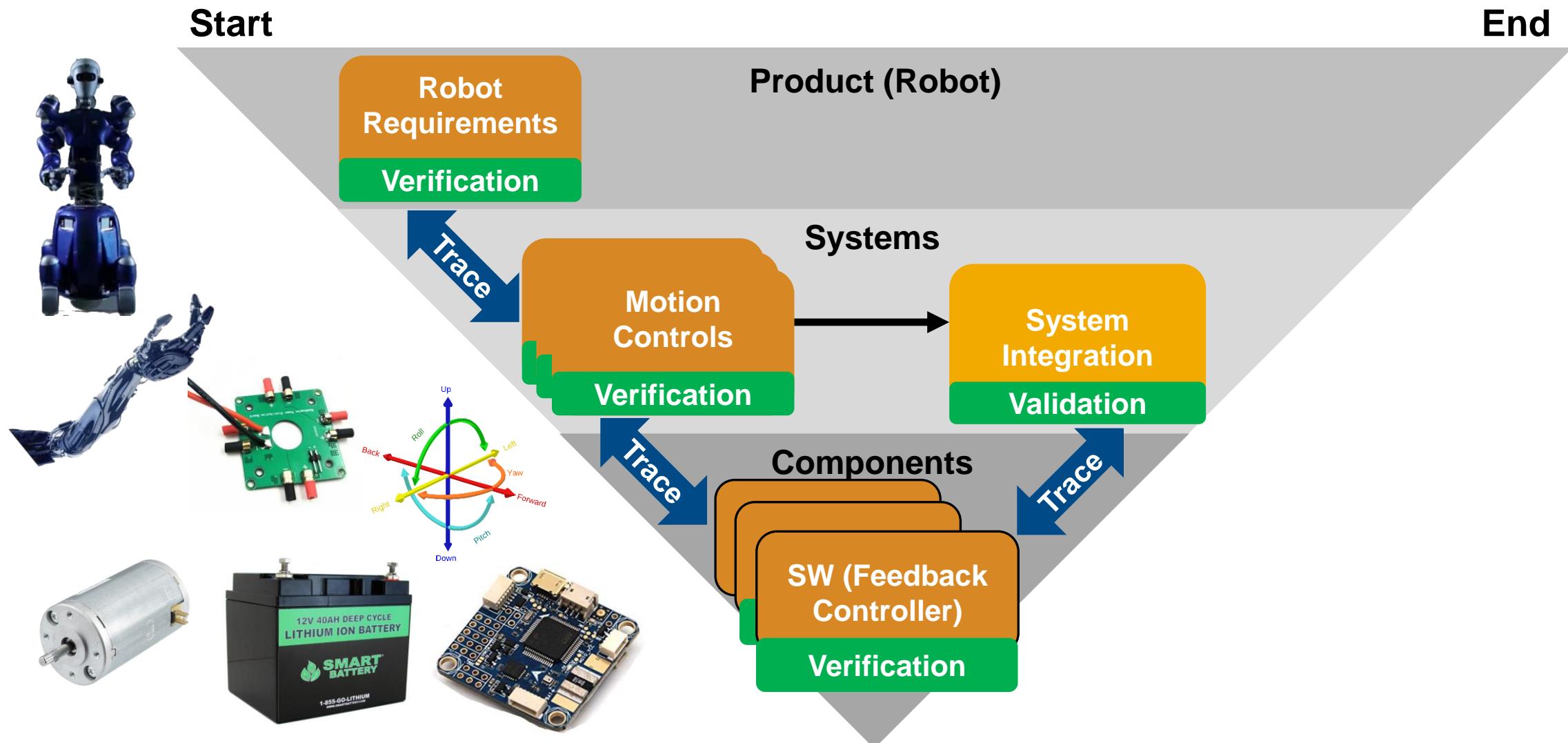
# Justin the Robot MBD Workflow



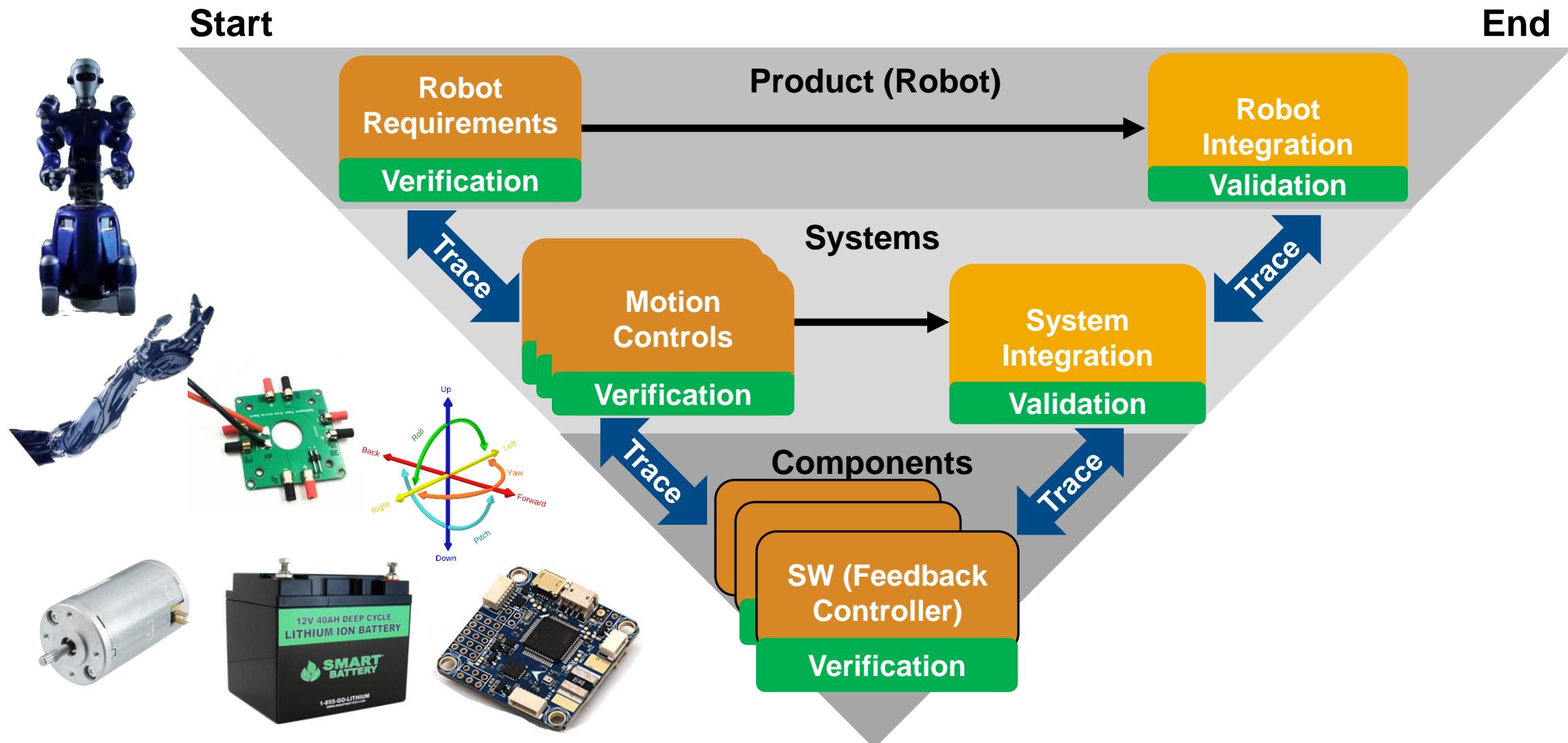
# Justin the Robot MBD Workflow



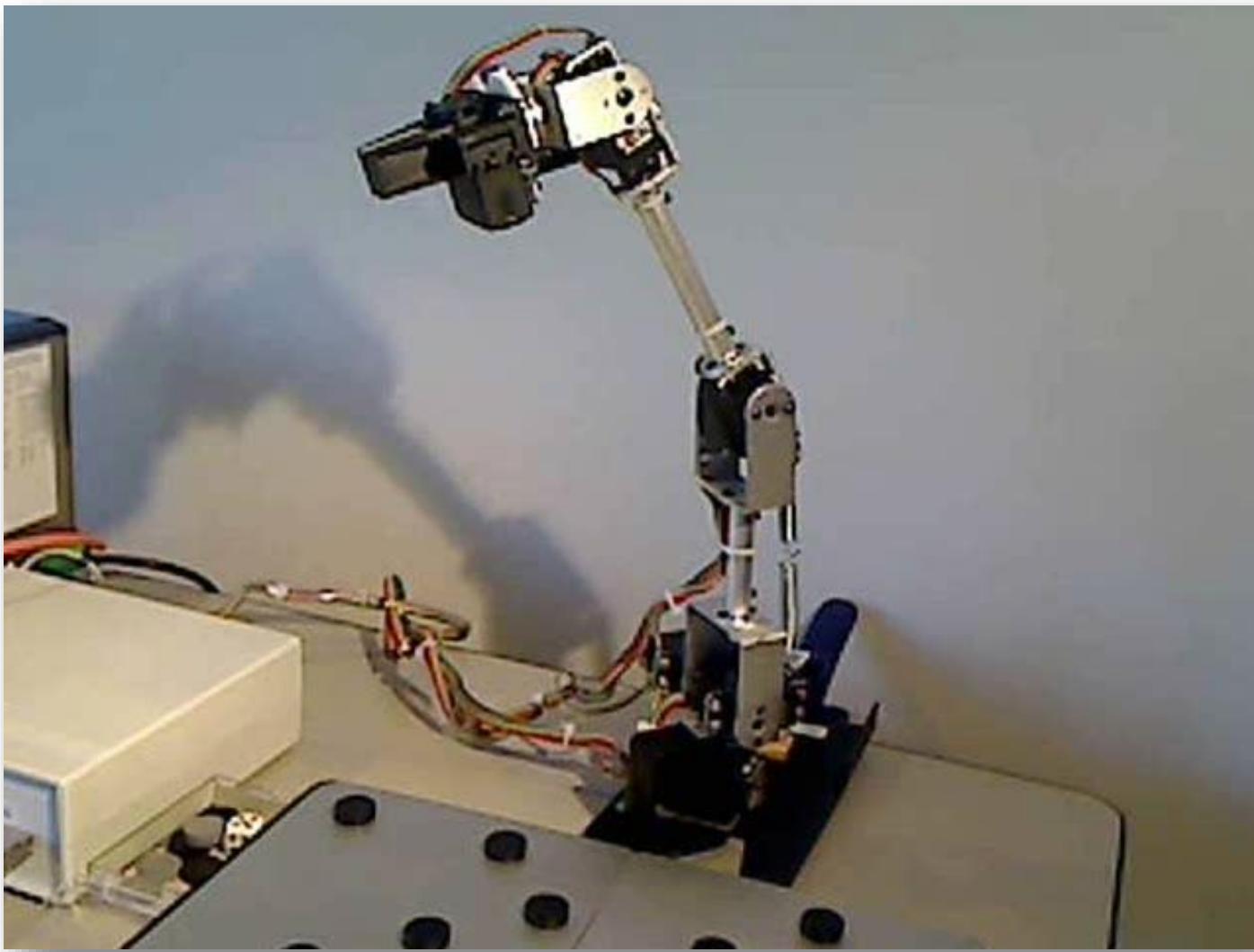
# Justin the Robot MBD Workflow



# Justin the Robot MBD Workflow

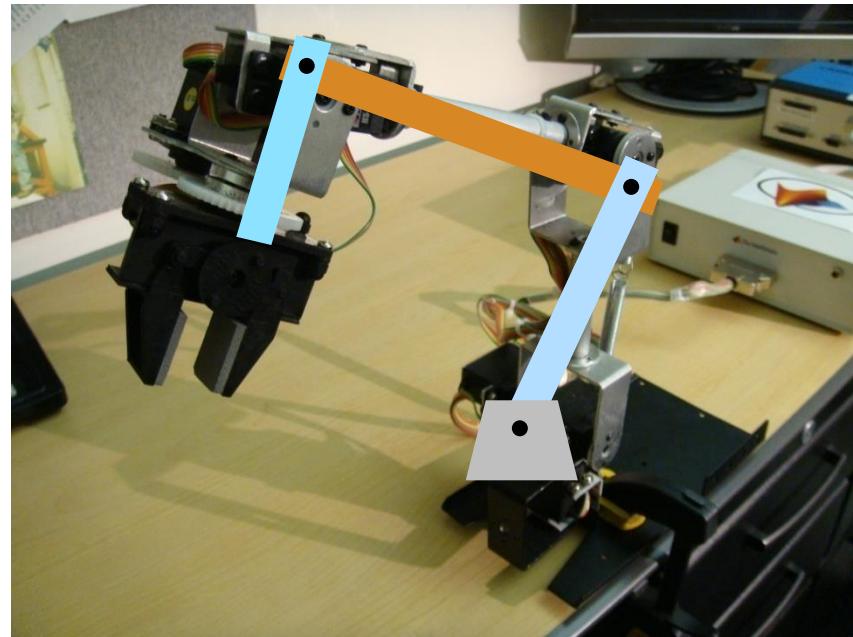


# How can you get started?

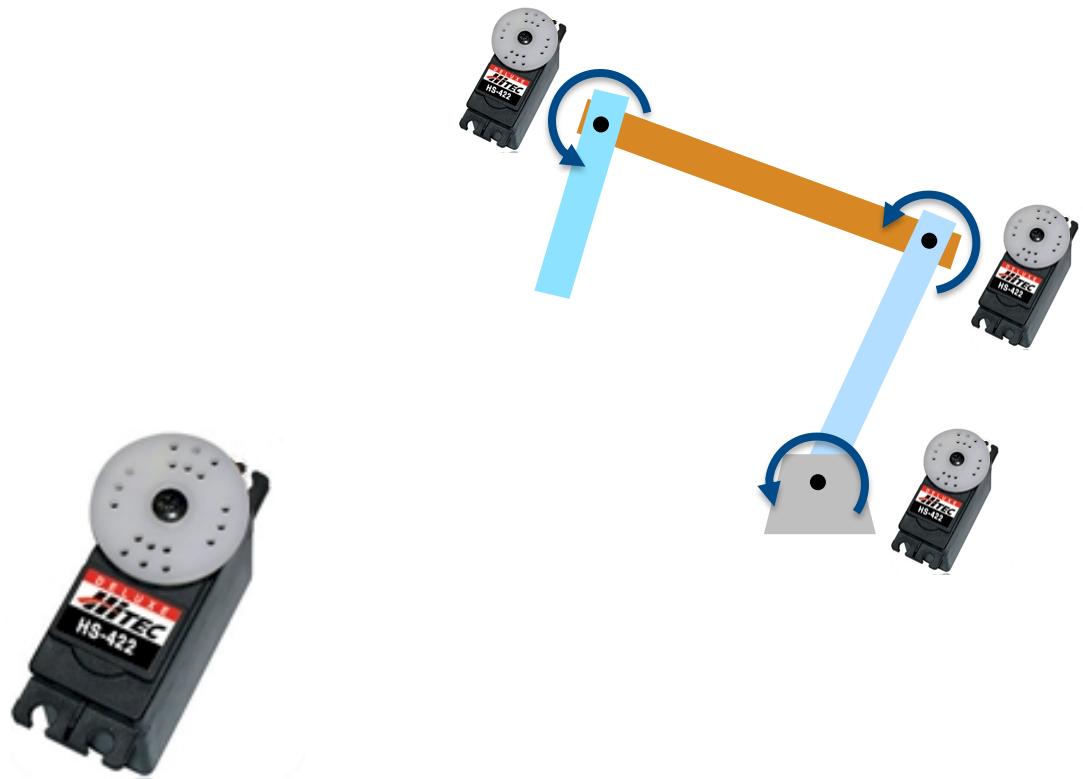


# So, where do we really start?

- If possible, break down a big problem into “smaller”(i.e., more manageable) problems
- Use a “divide and conquer” approach



And, for our robot to be of any use, it has to be able to move...



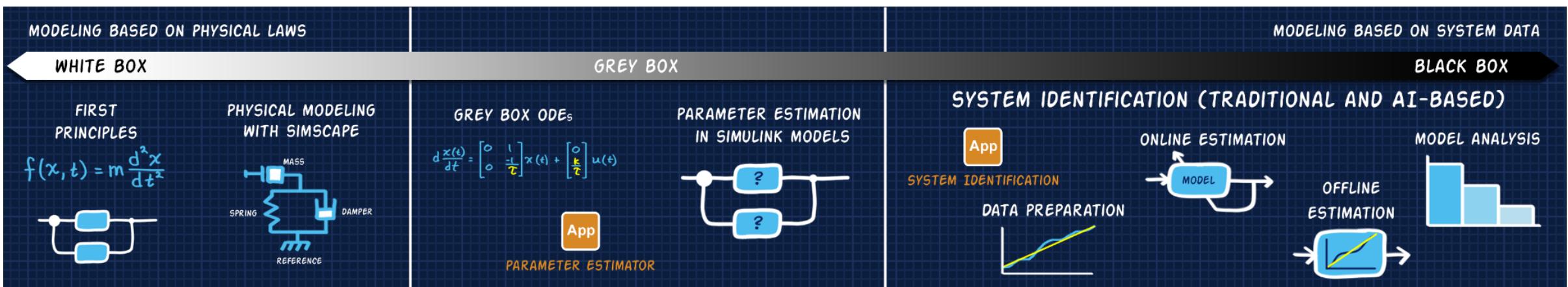
We should include the effects of actuator and sensor dynamics in the simulation model

# Different Approaches for Modeling Dynamic Systems

- *How much/what kind of information do we know?*

## Model Parameters

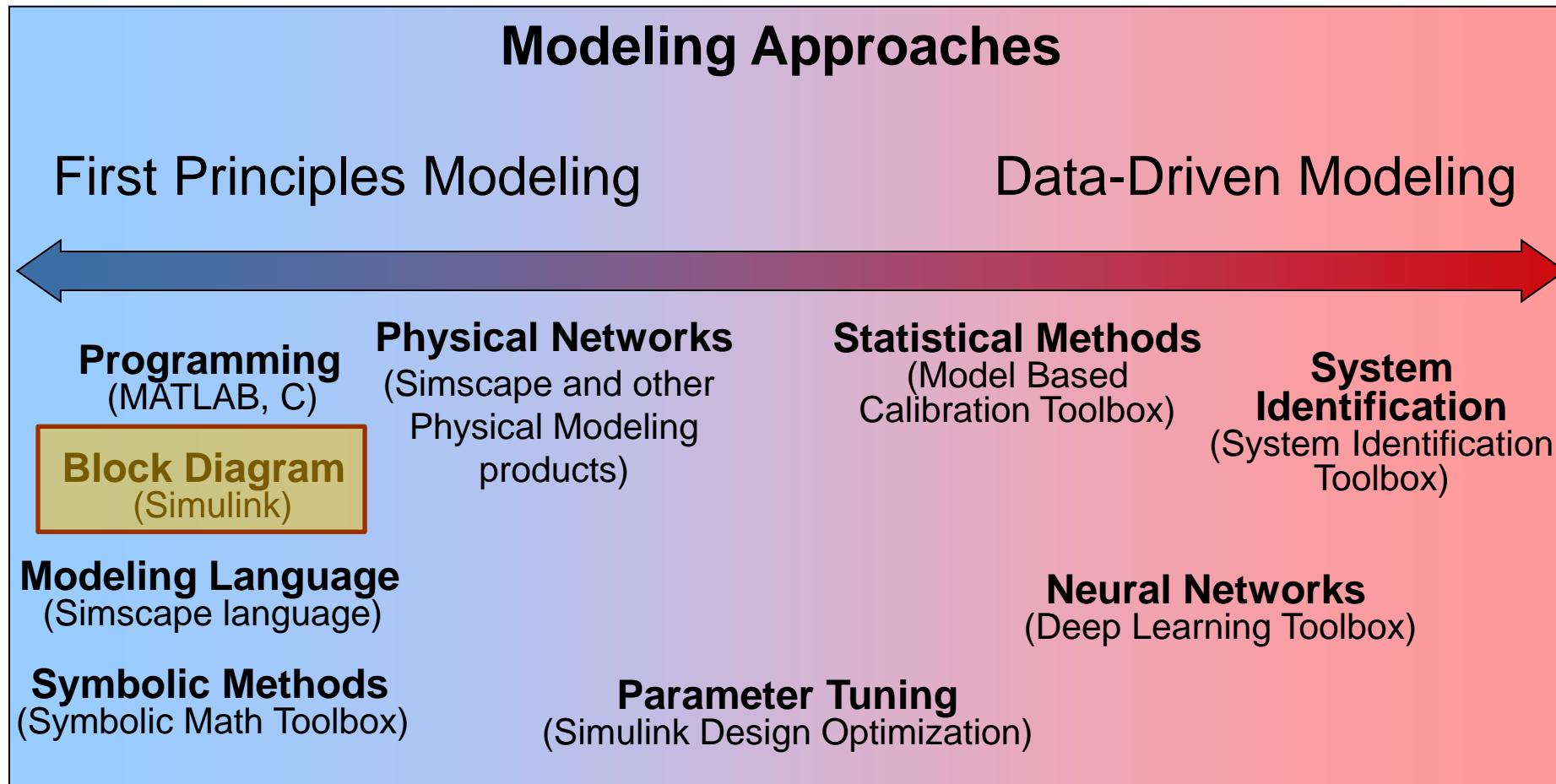
Determine model parameters through first principles, grey box, and data-driven methods.



# Agenda

- Simulink Overview
- Introduction to Model-Based Design
- Various ways of modeling a DC Motor
  - Mathematical Modeling with Simulink
  - Physical Modeling with Simscape
- Control System design
- Automatic Code Generation
- Resources and Support

# Different Approaches for Modeling Dynamic Systems



# Modeling a DC Motor in Simulink



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

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

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

*Electrical*

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

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

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

*Mechanical*

# Modeling a DC Motor in Simulink

*Based on its equations:*

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

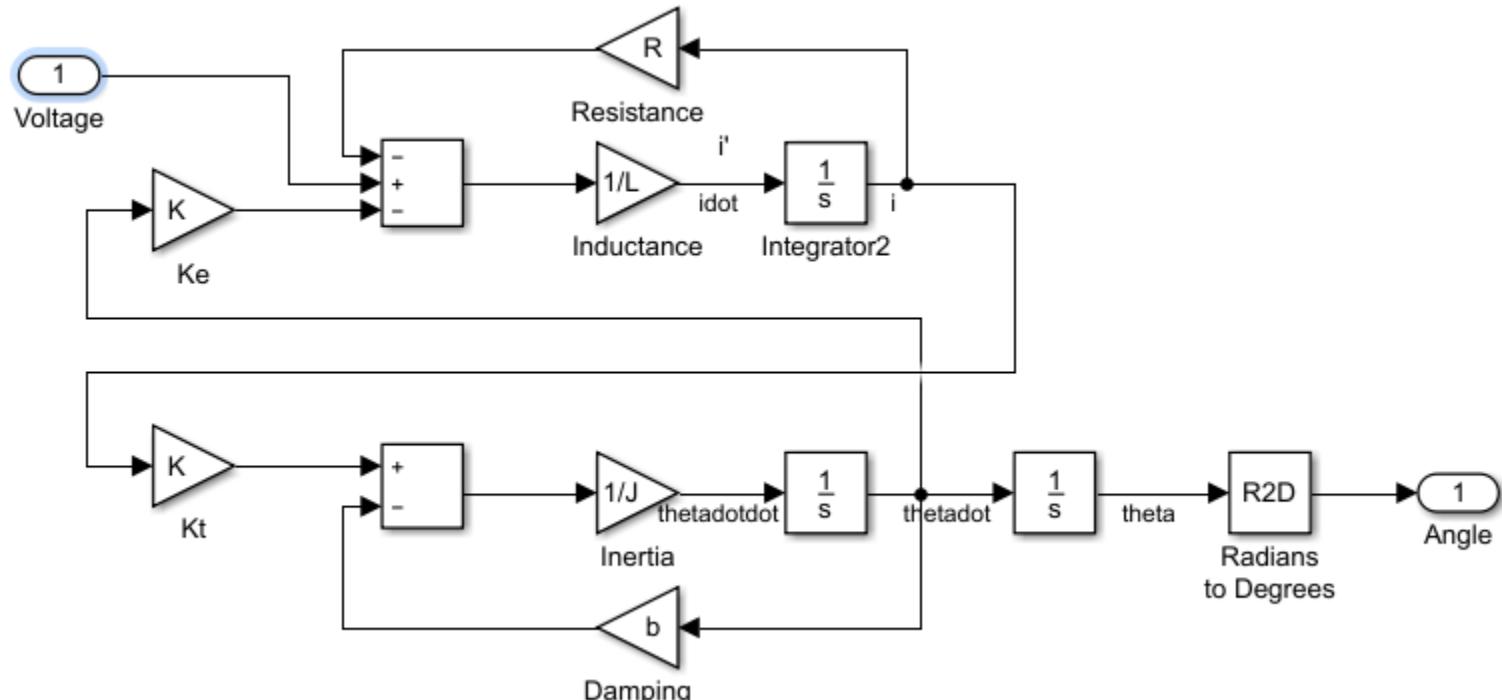
$$\frac{di}{dt} = \frac{1}{L} (V - K\omega - iR)$$

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

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

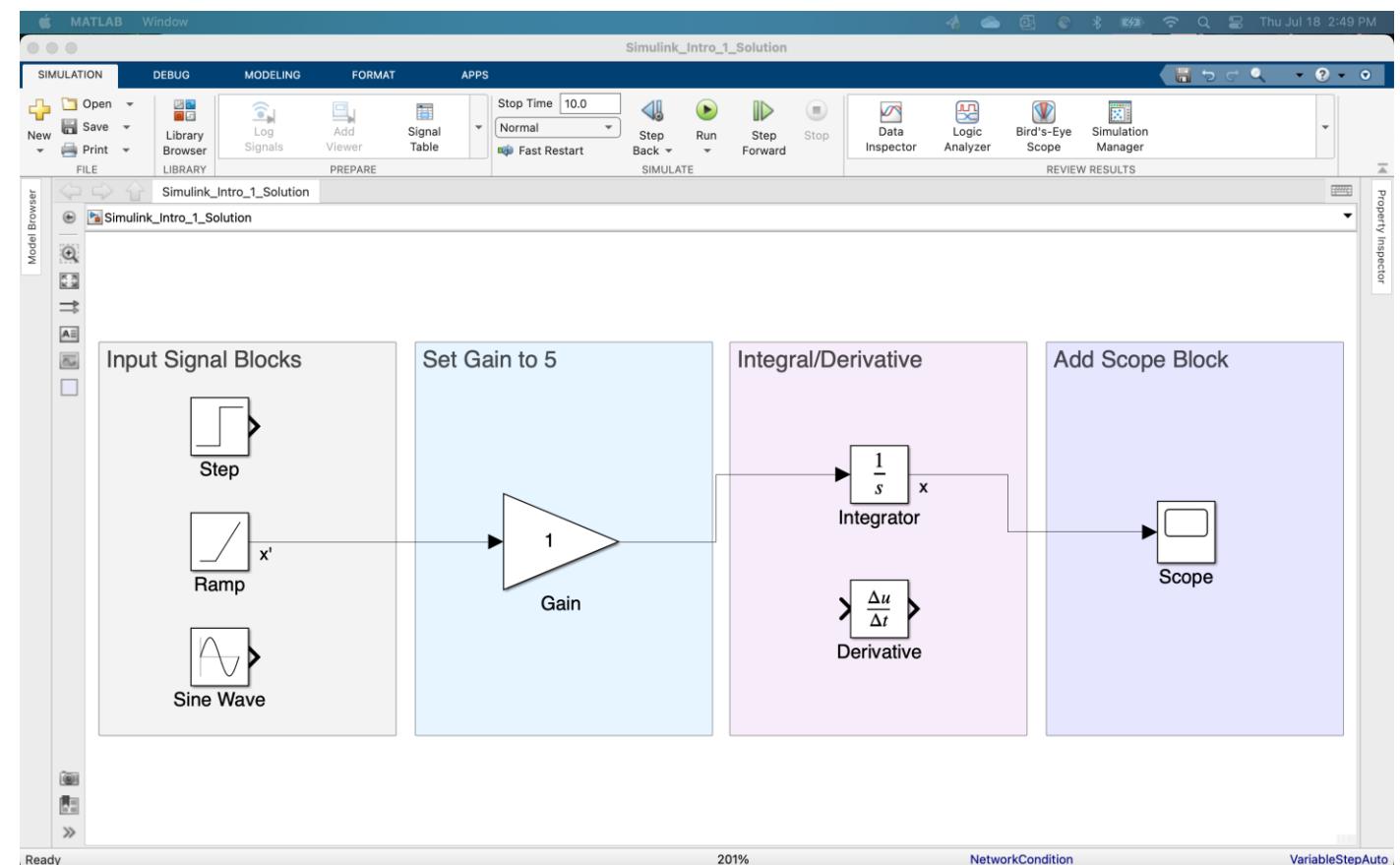
$$\frac{d\omega}{dt} = \frac{1}{J} (Ki - b\omega - T_{Load})$$

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



# Exercise 1: Getting our feet wet with Simulink (10 mins)

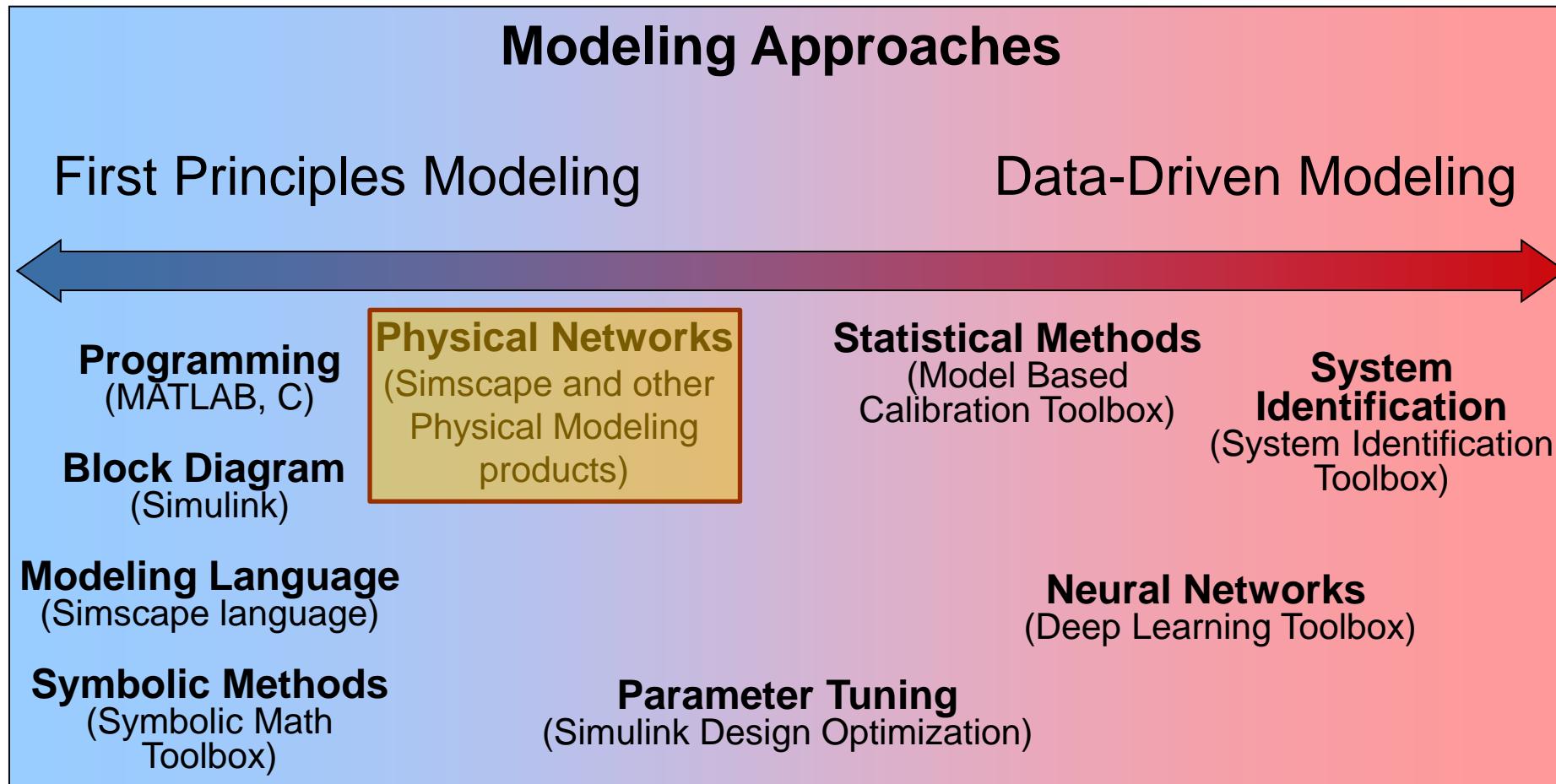
- Open: ***Simulink\_Intro.slx***
- Connect any one of the ‘Input Signal Blocks’ to the Gain.
- Decide if you want to integrate or differentiate and connect to the corresponding block
- Add a scope block and connect. You can also connect the input block to see both signals



# Agenda

- Simulink Overview
- Introduction to Model-Based Design
- Various ways of modeling a DC Motor
  - Mathematical Modeling with Simulink
  - **Physical Modeling with Simscape**
- Control System design
- Automatic Code Generation
- Resources and Support

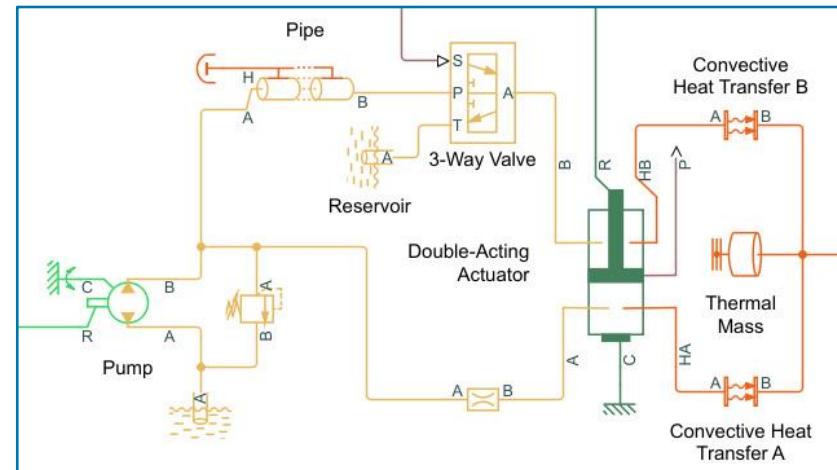
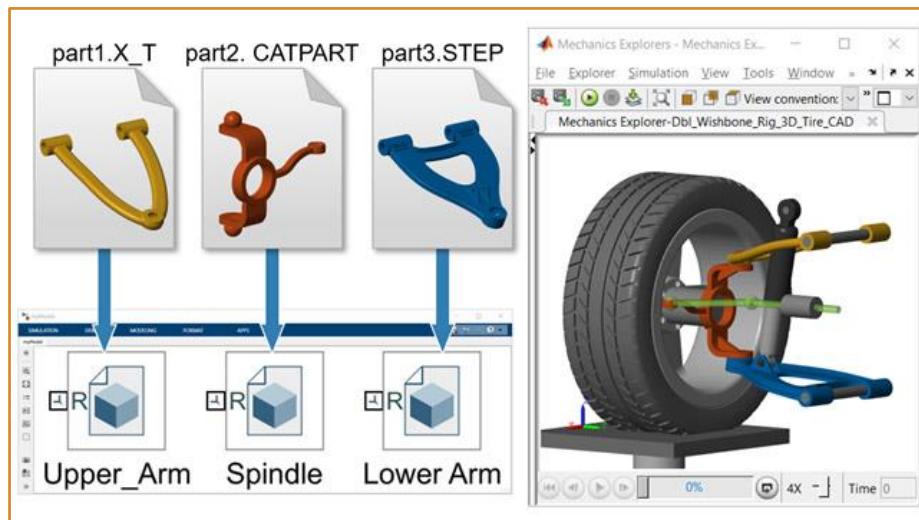
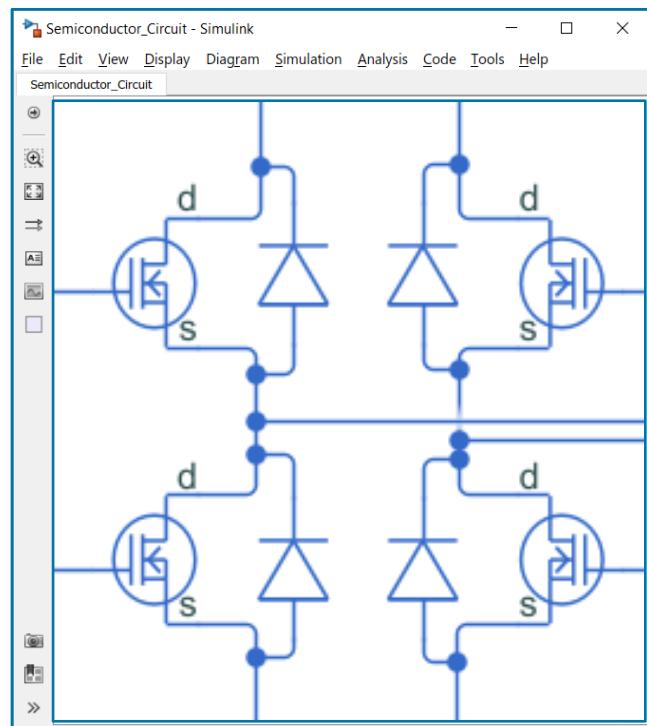
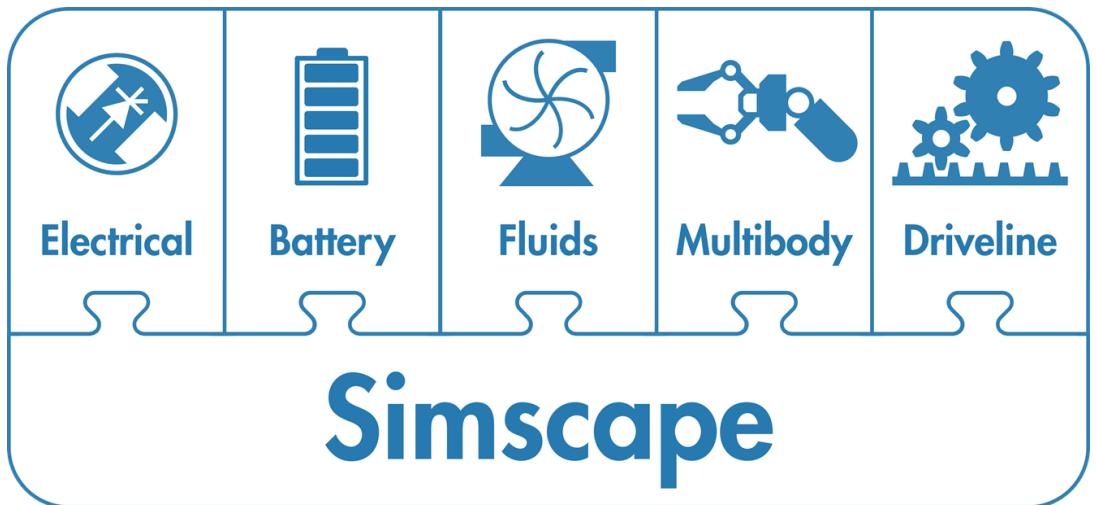
# Different Approaches for Modeling Dynamic Systems



# Simscape

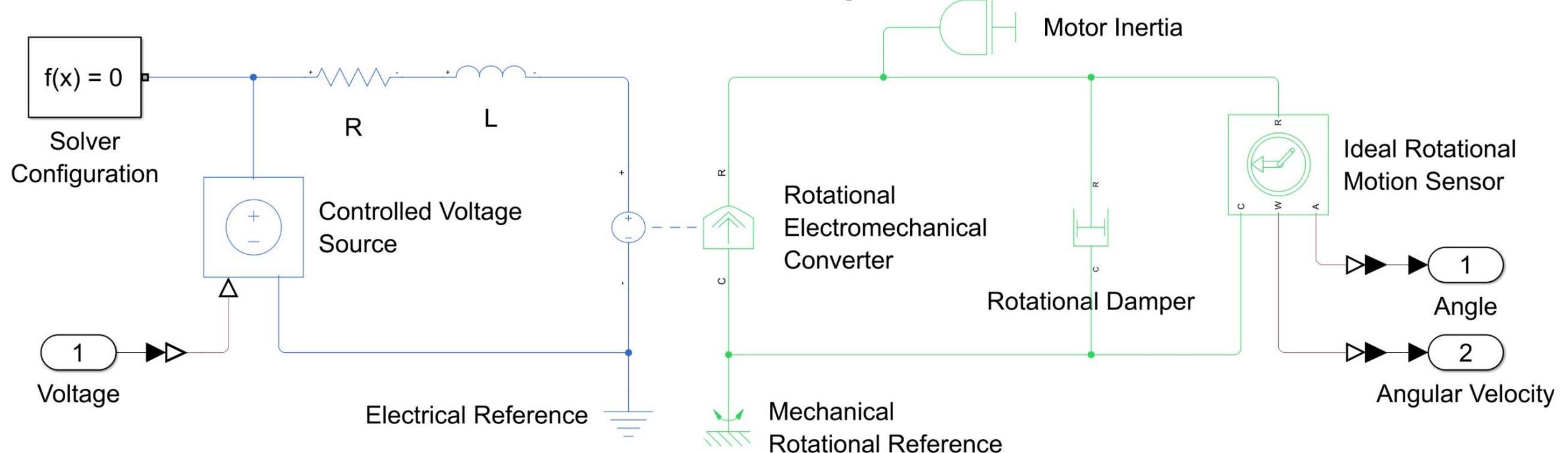
## Overview

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



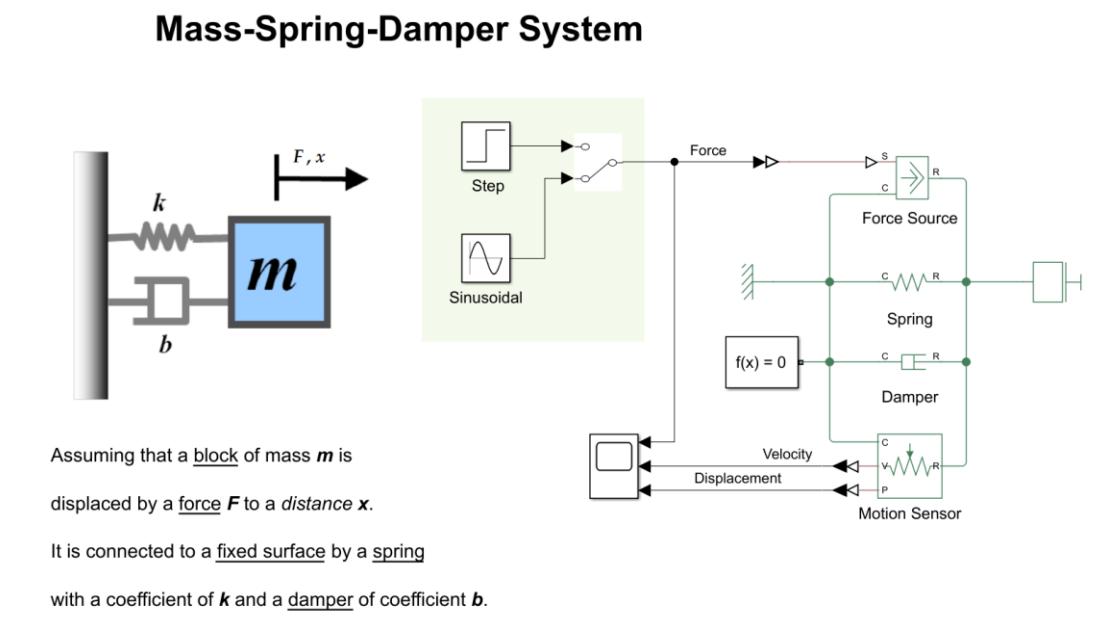
# Model Physical Network of DC Motor using Simscape

## DC Motor in Simscape



## Exercise 2: Mass-Spring Damper

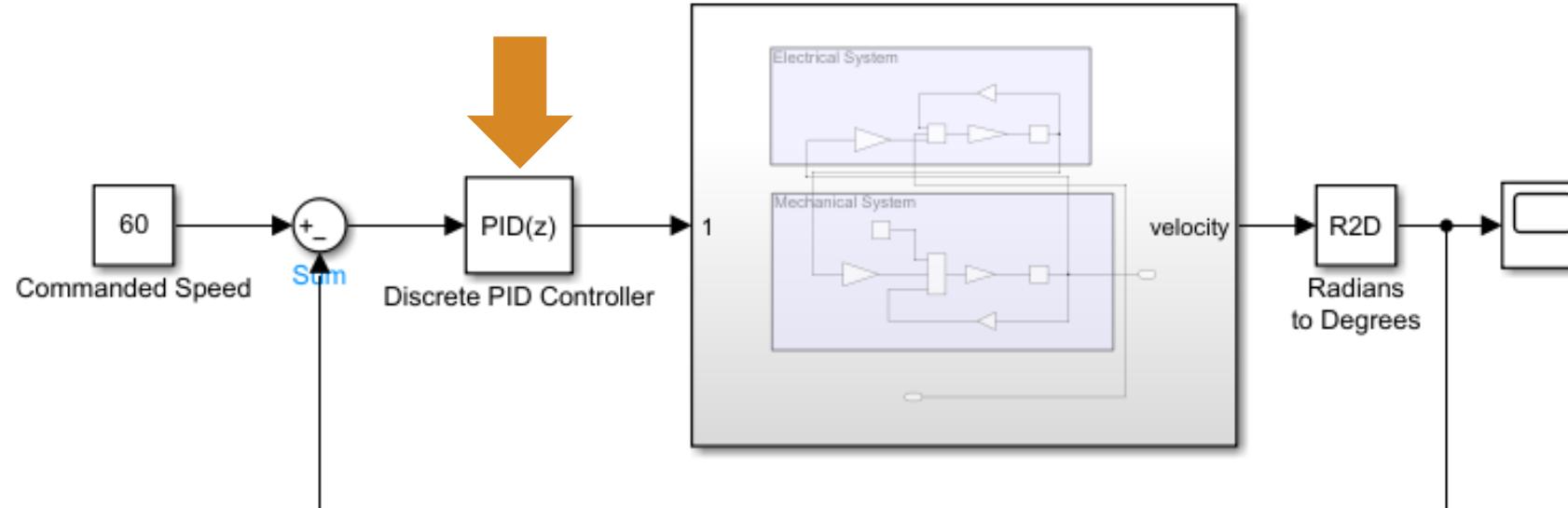
- Open: **Mass\_Spring\_Damper.slx**
- Add Translational Spring and Mass blocks
- Edit parameters and view results



# Agenda

- Simulink Overview
- Introduction to Model-Based Design
- Various ways of modeling a DC Motor
  - Mathematical Modeling with Simulink
  - Physical Modeling with Simscape
- **Control System design**
- Automatic Code Generation
- Resources and Support

# Controlling Your System

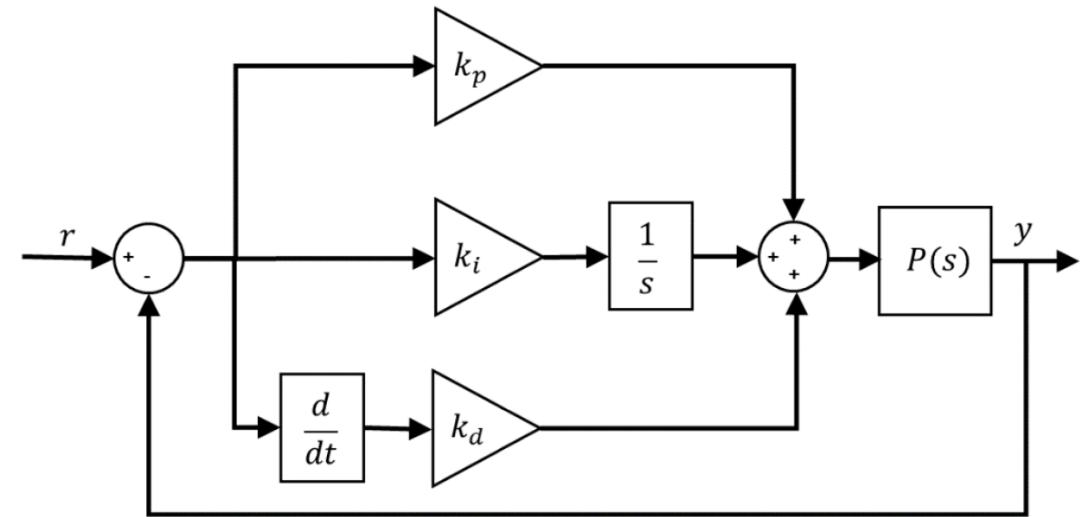


- Modeling the physical system is part of the job.
- What about the controls?

# PID Controller – Proportional, Integral, Differential Controller

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau)d\tau + k_d \frac{d e(t)}{dt}$$

$$U(s) = \left( k_p + \frac{k_i}{s} + k_d s \right) E(s)$$

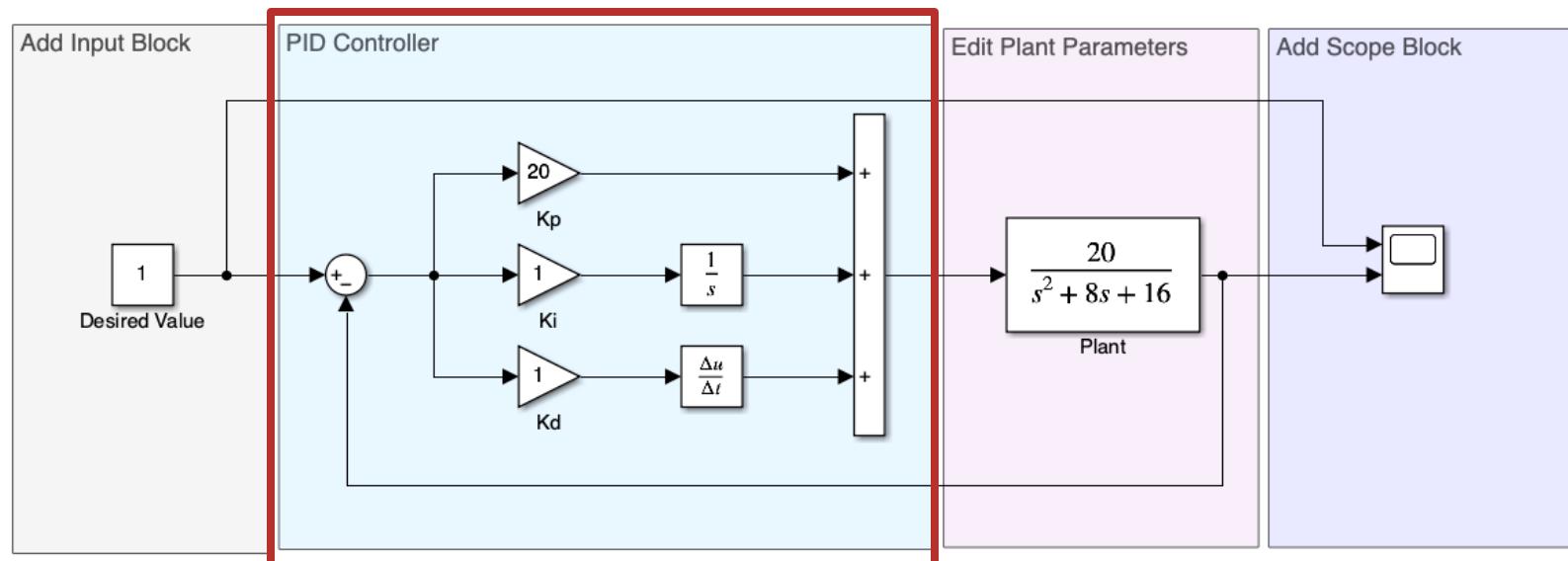
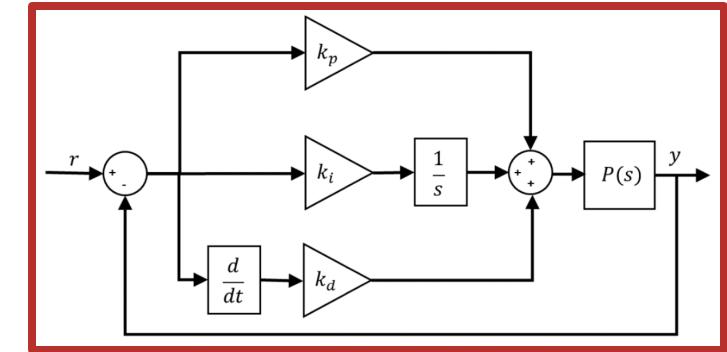


A PID controller continuously adjusts a system's output to minimize the error between a desired setpoint and a measured variable using proportional, integral, and derivative actions.

[What is a PID Controller?](#)

# Exercise 3: Build a PID controller in Simulink

- Open: ***PID\_Control.slx***
- Add a desired value block as input
- Decide the P, I, D values
- Edit the plant parameters by adding a “Transfer Function” block
- Add a scope block



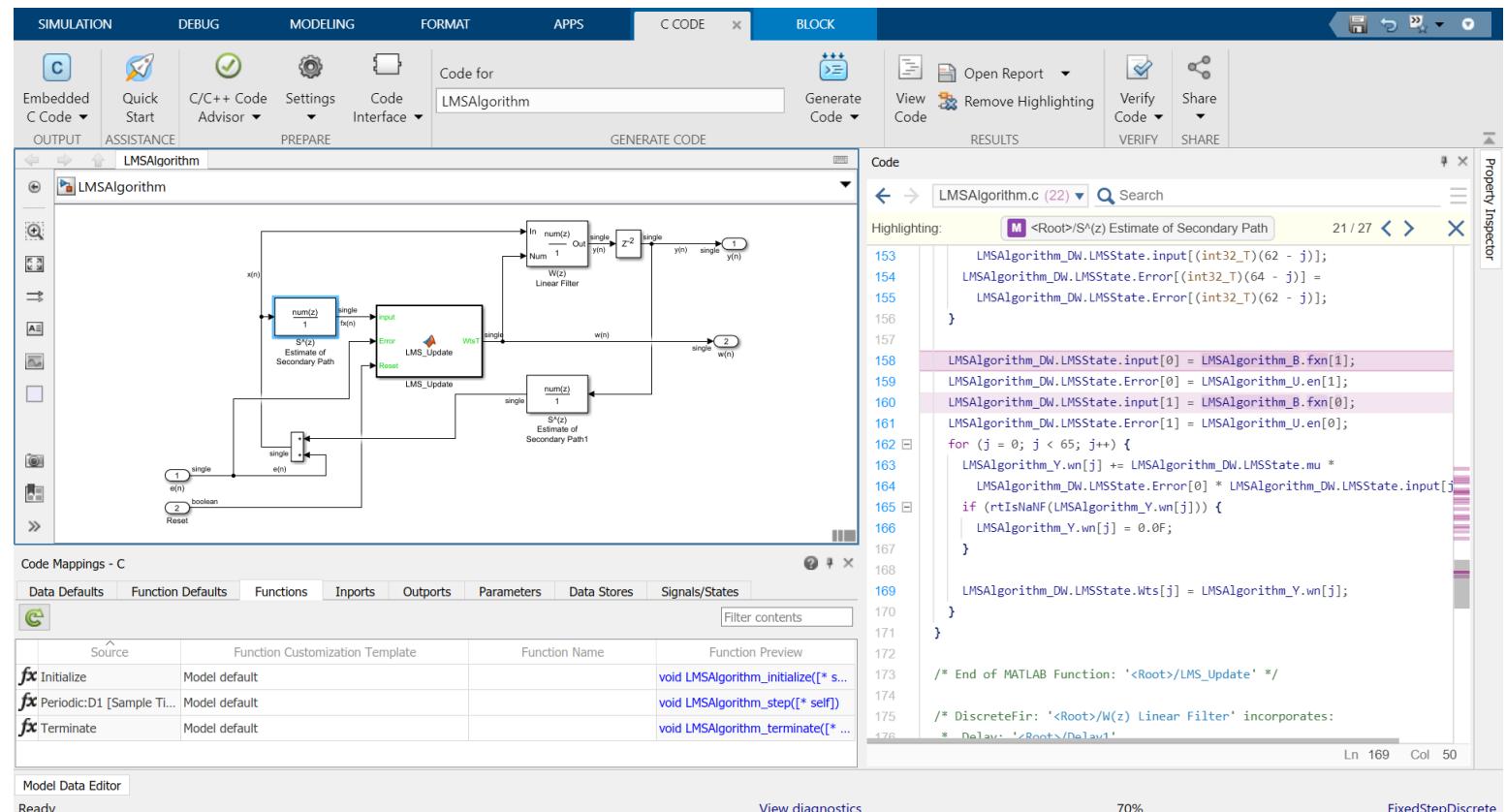
# Agenda

- Simulink Overview
- Introduction to Model-Based Design
- Various ways of modeling a DC Motor
  - Mathematical Modeling with Simulink
  - Physical Modeling with Simscape
- Control System design
- **Automatic Code Generation**
- Resources and Support

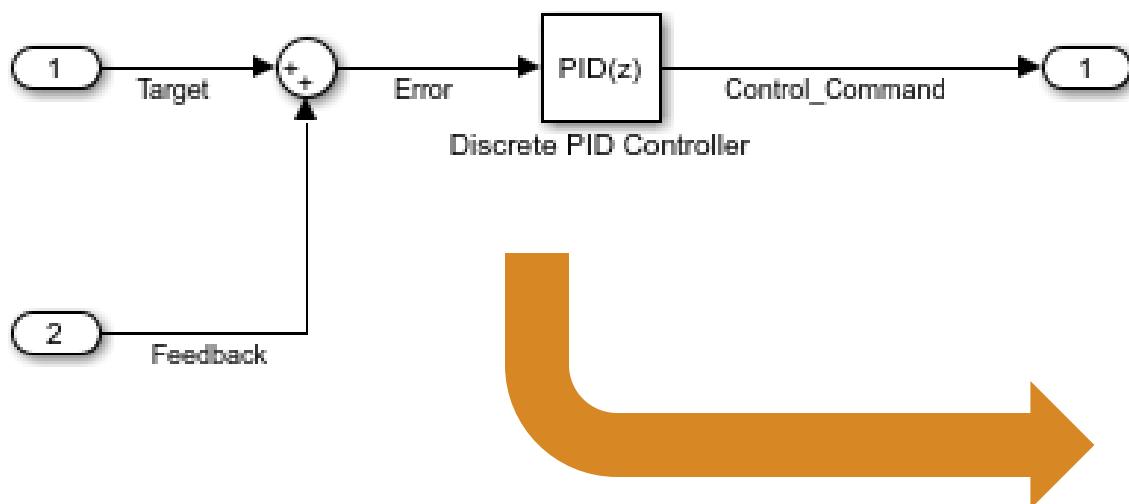
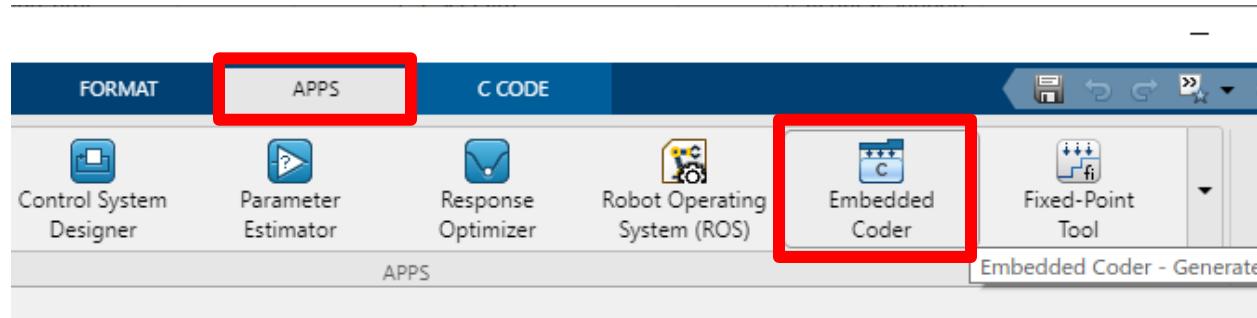
# How do we go from model to code?

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

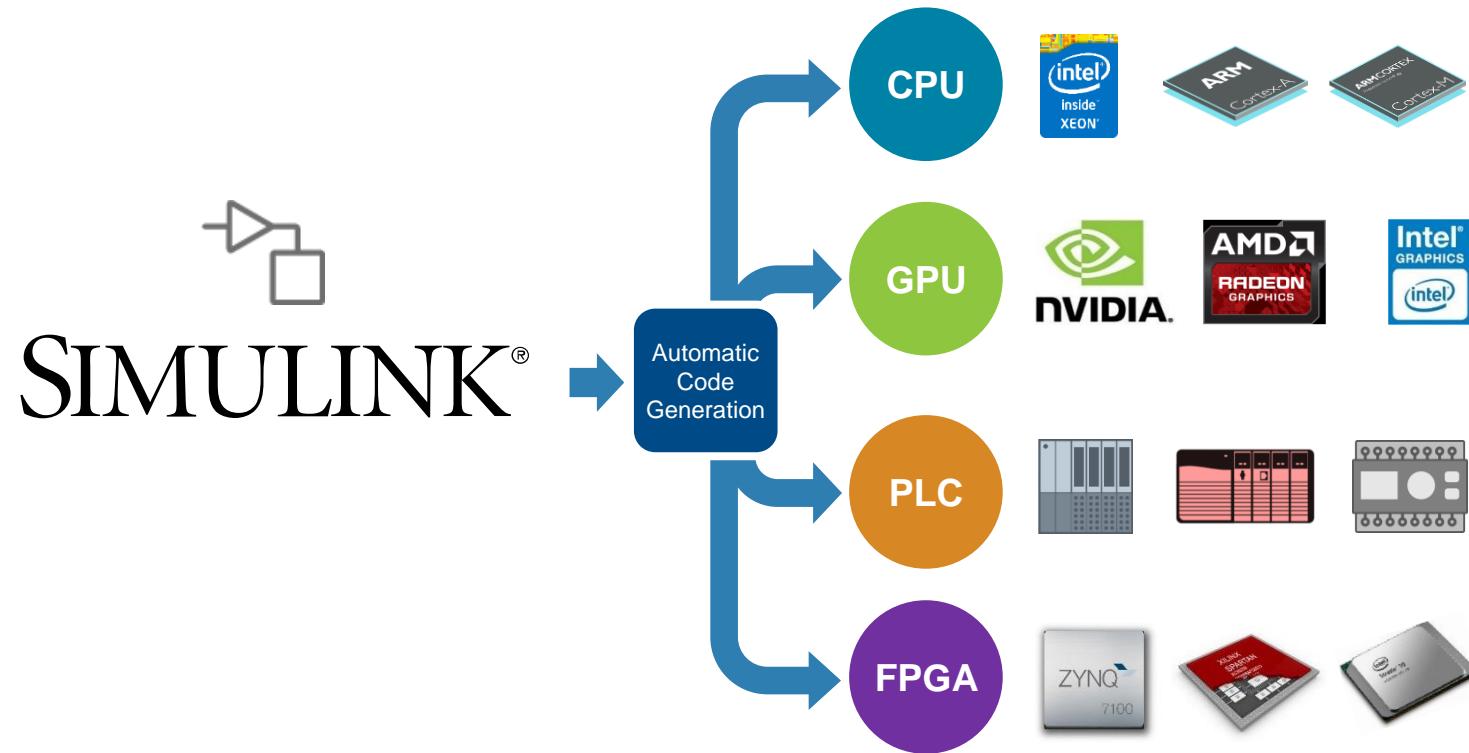
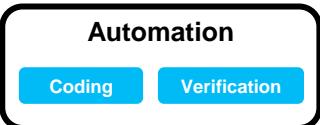


Access code generation tools  
from the Embedded Coder App

```
32  /* Model step function */
33  void codegen_example_step(void)
34 {
35     real_T rtb_Error;
36     real_T rtb_Tsamp;
37
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:
```

# How do we go from model to code?

Automate **code generation** for prototyping and production



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

# Hardware Support Packages

The screenshot shows the MATLAB R2021a interface. The top menu bar includes HOME, PLOTS, APPS, and various toolbars for New, Open, Import, Save, and workspace management. The Add-On Explorer is open, displaying 318 results for Hardware Support Packages. The left sidebar of the Add-On Explorer lists categories such as MathWorks, Community, MATLAB, Simulink, Workflows, Applications, and Disciplines. Several packages are listed, including the MATLAB Support Package for Arduino Hardware (Installed, 1571 Downloads), Legacy MATLAB and Simulink Support for Arduino (1124 Downloads), Simulink Support Package for Arduino Hardware (842 Downloads), MATLAB Support Package for Raspberry Pi Hardware (284 Downloads), MATLAB Support Package for USB Webcams (491 Downloads), Image Acquisition Toolbox Support Package for OS Generic Video Interface (332 Downloads), and MATLAB Support Package for Webcam (196 Downloads). The right side of the interface features a large image of electronic hardware and the text "Connect MATLAB and Simulink to Hardware".

<https://www.mathworks.com/hardware-support/home.html>

## Key Takeaways

- Design and simulate multi-domain systems with **Simulink** and **Simscape**
- Tune system parameters and outputs with **Control Design**
- Automatically generate code for hardware with **Coders\*** and **Support Packages**

\* *MATLAB Coder, Simulink Coder, Embedded Coder, HDL Coder, GPU Coder*

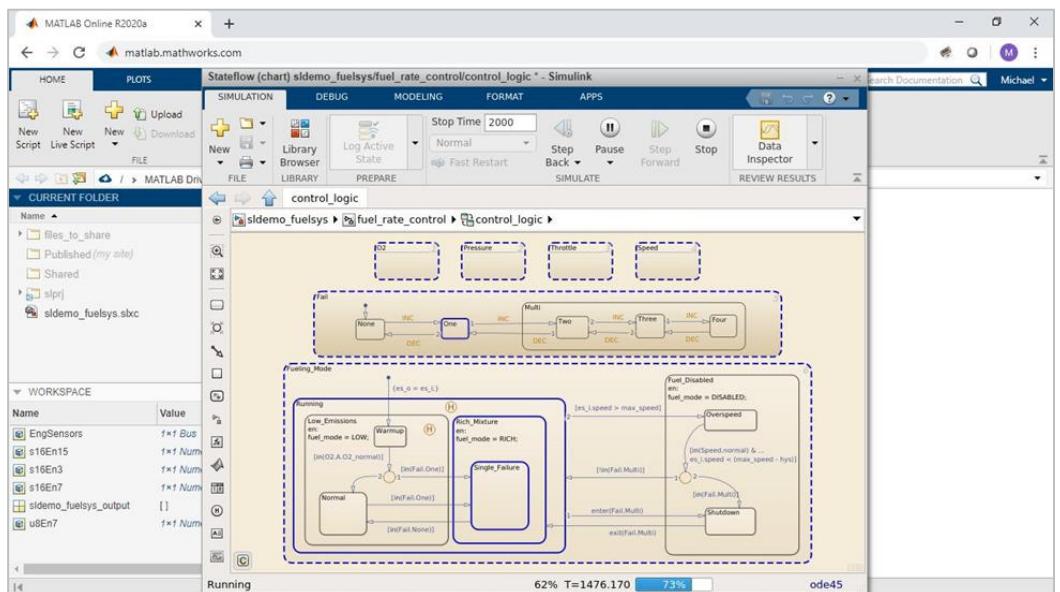
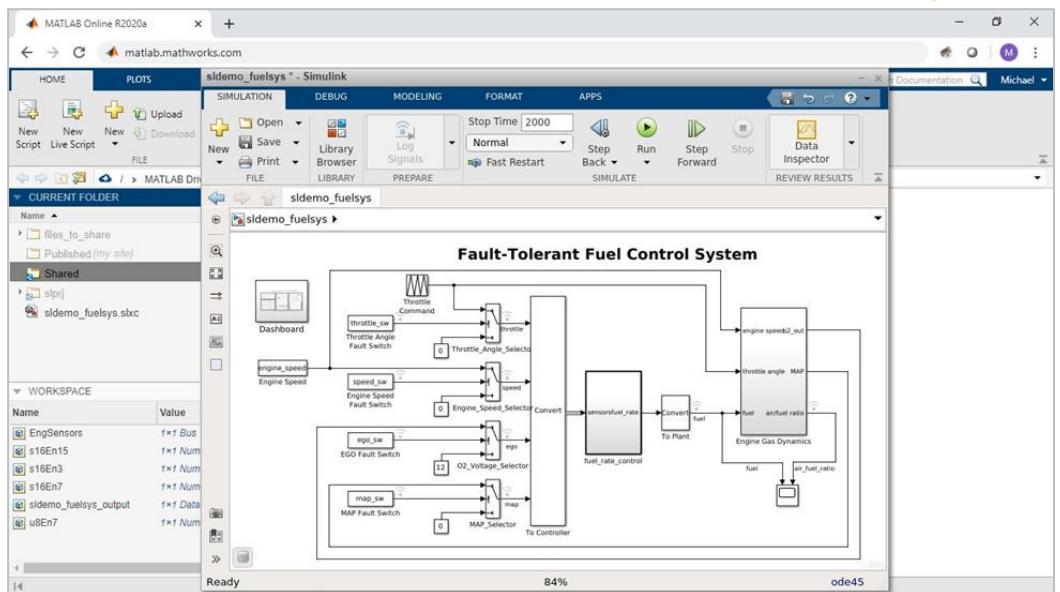
# Agenda

- Simulink Overview
- Introduction to Model-Based Design
- Various ways of modeling a DC Motor
  - Mathematical Modeling with Simulink
  - Physical Modeling with Simscape
- Control System design
- Automatic Code Generation
- **Resources and Support**

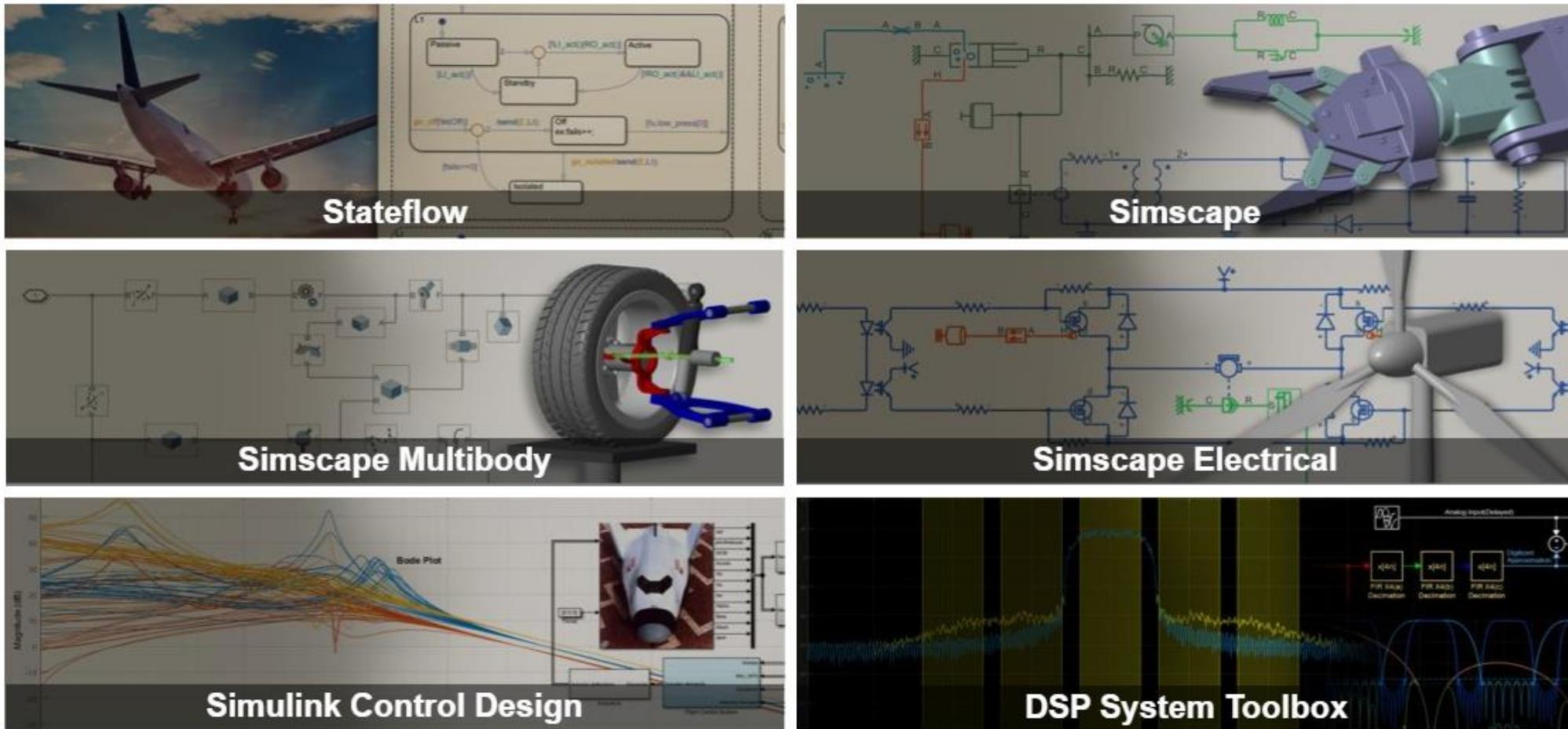
# Simulink Online

## Use Simulink through your web browser

- No downloads or installations required
- Available to anyone with a Simulink license who is eligible to use MATLAB Online
  - Most EDU licenses and Individual
- Ideal for teaching, learning, and convenient, lightweight access

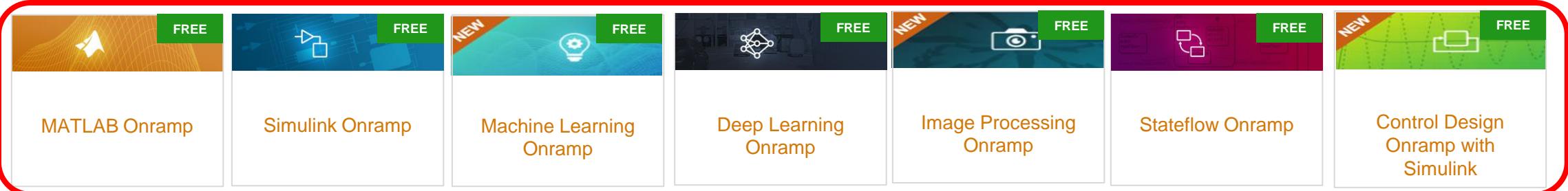


# Simulink Online: Supported Products



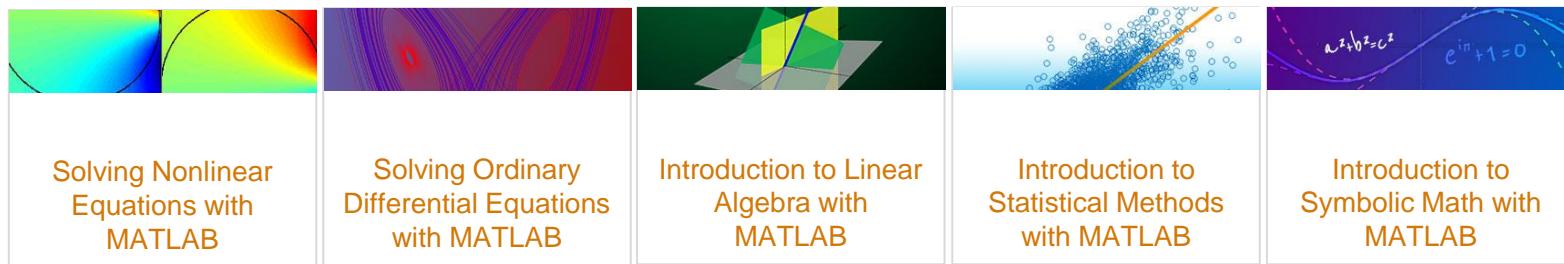
# Self-Paced Courses

FREE “getting started” content – available for everyone



## Computational Mathematics

\*Available only to users at universities that offer campus-wide training access.



## Core MATLAB



## Data Science

## Other Resources:

- [Introduction to Model-Based Design with Simulink](#)
- [Physical Modeling with Simscape](#)
- [Simulink Control Design](#)
- [Robotics and Autonomous Systems](#)
- (NEW) Multibody Simulation
- (NEW) Battery Systems

# Self-Paced Courses

## FREE “getting started” content – available for everyone

**Multibody Simulation Onramp**

[Start course](#) Share Course | [Share Certificate & Progress](#) | [Settings](#) COMPLETED

Get started modeling and simulating 3D mechanical systems in Simscape™ Multibody™. Learn to define and assemble rigid bodies into an articulated multibody system and use your skills to construct and actuate a robot drawing arm.

**Course modules**

- > Introduction 100% | 10 min
- > Multibody Simulation Concepts 100% | 15 min
- > Model Bodies 100% | 25 min
- > Position and Orient Bodies 100% | 0.5 hours
- > Assemble Bodies 100% | 0.5 hours
- > Model Contact Forces 100% | 10 min
- > Project - Robot Drawing Arm 100% | 20 min
- > Conclusion 100% | 5 min

**Related Learning**

 [Simulink Onramp](#)  
Get started quickly with the basics of Simulink.

[View all self-paced courses](#)

**About this course**

 Format: Self-paced Length: About 2.5 hours Language: English

**Recommended prerequisites**

 Simscape Onramp COMPLETED

**Features**

- Hands-on exercises with automated feedback
- Access to Simulink through your browser or desktop
- Shareable progress report and course certificate

**Authored By:**

 Duy Nguyen  
MathWorks

**Battery Pack Modeling**

[Start course](#) Share Course | [Share Certificate & Progress](#) | [Settings](#) 0% Access expires Mar 31, 2025

Learn cell-to-pack workflows for battery blocks, and thermal modeling, using the Battery Builder app and how to use Simscape™ to add a cooling plate to battery packs.

**Course modules**

- > Introduction 5 min
- > Battery Pack Modeling using Battery Builder App 15 min
- > Battery Pack Thermal Modeling using Battery Builder App 15 min
- > Adding a Cooling Plate to Battery Pack 10 min
- > Conclusion 5 min

**Related Learning**

 [Power Electronics Simulation Onramp](#)  
Learn the basics of simulating power electronics converters in Simscape.

 [Simscape Onramp](#)  
Learn the basics of simulating physical systems in Simscape.

 [Simscape Battery Onramp](#)  
Learn the basics of simulating battery systems in Simscape.

[View all self-paced courses](#)

**About this course**

 Format: Self-paced Length: About 1 hour Language: English

**Recommended prerequisites**

 Simscape Onramp COMPLETED

 Simscape Battery Onramp

**Features**

- Hands-on exercises with automated feedback
- Access to Simulink through your browser or desktop
- Shareable progress report and course certificate

**Authored By:**

 Priyanka Shukla  
MathWorks

# Try an Example

**Modeling — Examples**

**R2019a**

<b>ADC-PWM Synchronization via ADC Interrupt</b> Use the ADC block to sample an analog voltage and use the PWM block to generate a pulse waveform. This example also shows how to use	<b>LIN-Based Control of PWM Duty Cycle</b> Use the C28035 LIN Receive, LIN Transmit, and PWM blocks to generate a pulse waveform.	<b>Asynchronous Scheduling</b> Use the Texas Instruments™ c28x peripherals and Hardware Interrupt blocks to control the real-time execution of Simulink® function-call	<b>Using CAN Calibration Protocol in External Mode</b> Simulate field-oriented control (FOC) using a Permanent Magnet Synchronous Machine (PMSM) model. The model is created using	<b>Permanent Magnet Synchronous Motor Field-Oriented Control</b> Control the speed of a three-phase Permanent Magnet Synchronous Motor in a closed-loop fashion via Field-Oriented Control (FOC) using	<b>Permanent Magnet Synchronous Motor Field-Oriented Control Using CLA</b> Use the Control Law Accelerator (CLA) available on some of the TI Piccolo processors.	<b>Schedule a Multi-Rate Controller for a Permanent Magnet Synchronous...</b> Create a real-time executable for a Texas Instruments F28335 embedded target. You will build upon the algorithm specified in the
<b>Parameter Tuning and Signal Logging with Serial External Mode</b> Perform parameter tuning and data logging with a Simulink® model running in External mode on the Texas Instruments™ C2000 targets.	<b>Using the I2C Bus to Access Sensors</b> Use the I2C blocks to communicate with I2C based devices.	<b>Digital DC/DC Buck Converter Voltage Mode Control (VNC)</b> Model a controller for the DC/DC buck converter using the Embedded Coder Support Package for Texas Instruments® C2000 Processors.	<b>SPI Loopback Master Transfer</b> Configure and use SPI blocks to read and write data.	<b>Using DMA to update ePWM Duty Cycle</b> Use the IPC blocks to communicate between the two CPUs of Texas Instruments™ Delfino F2837xD using Simulink® models.	<b>Inter-Processor Communication Using IPC Blocks</b> Configure the direct memory access (DMA) parameters to modify the ePWM duty cycle. Using DMA, the sine wave data is copied from a	<b>Photovoltaic Inverter with MPPT Using Solar Explorer Kit</b> Implement a photovoltaic (PV) inverter system using the Embedded Coder® Support Package for Texas Instruments™ C2000™ Processors.

## Real-Time Code Execution Profiling

This example shows you how to use Embedded Coder™ Support Package for Texas Instruments™ C2000 Processors and Embedded Coder Support Package for Texas Instruments™ Concerto Processors for real-time execution profiling of generated code.

### Available versions of this model:

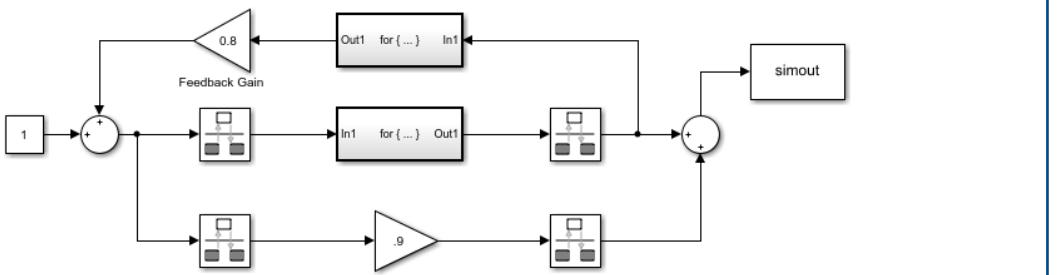
- C2000 LaunchPadXL TMS320F28377S (LAUNCHXL-F28377S)

### Model

The following figure shows the example model:

### Real-Time Code Execution Profiling

C2000 LaunchPadXL TMS320F28377S



# Examples of MathWorks Supported Hardware



Arduino



Lego EV3



Raspberry Pi



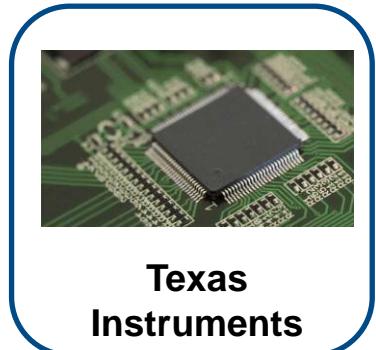
Android/iOS  
Devices



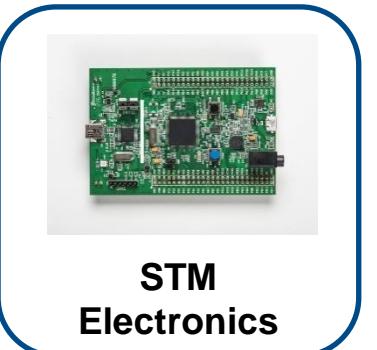
Kinect for  
Windows



BeagleBone  
Black



Texas  
Instruments



STM  
Electronics

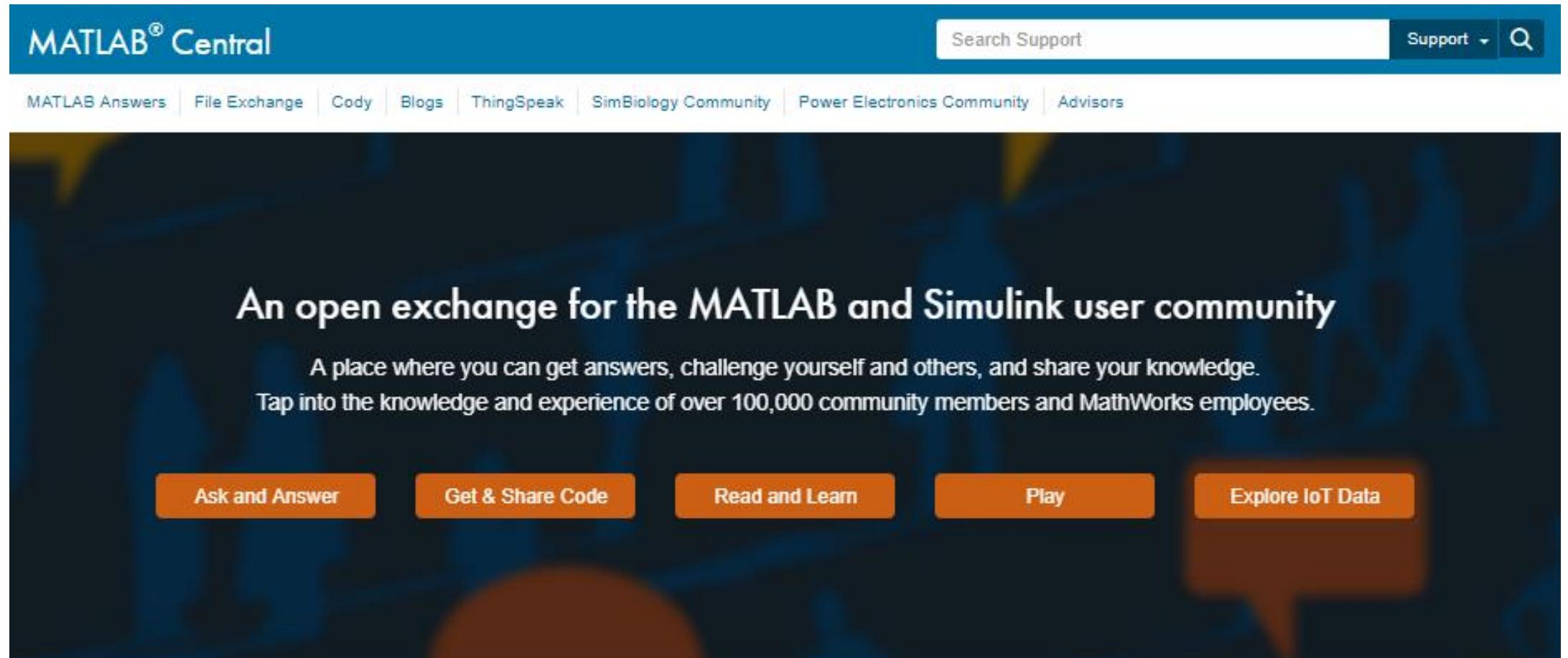


Freescale



Zynq SDR

# MATLAB Central



The screenshot shows the MATLAB Central homepage. At the top, there's a navigation bar with links for MATLAB Answers, File Exchange, Cody, Blogs, ThingSpeak, SimBiology Community, Power Electronics Community, and Advisors. There's also a search bar labeled "Search Support" and a "Support" dropdown menu with a magnifying glass icon.

**An open exchange for the MATLAB and Simulink user community**

A place where you can get answers, challenge yourself and others, and share your knowledge.  
Tap into the knowledge and experience of over 100,000 community members and MathWorks employees.

Five orange buttons are displayed horizontally: "Ask and Answer", "Get & Share Code", "Read and Learn", "Play", and "Explore IoT Data".

CONTRIBUTORS

525,000

ANSWERS PER DAY

120

DOWNLOADS PER DAY

25,000

SOLVERS PER DAY

730

# Technical Support



Products Solutions Academia Support Community Events

[Get MATLAB](#)



CA

Support

Search Support

Support ▾



Contact support

Download products

## Get Started



Download  
Products



Installation  
Help



Tutorials

## Get Help



Documentation



Examples



Answers

## Community

### File Exchange

Find and Share Code

### Blogs

Learn from Experts

### Cody

Play Coding Game

### ThingSpeak

Collect and Analyze IoT data

# MathWorks Education Application Engineers

*nsardesa@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

