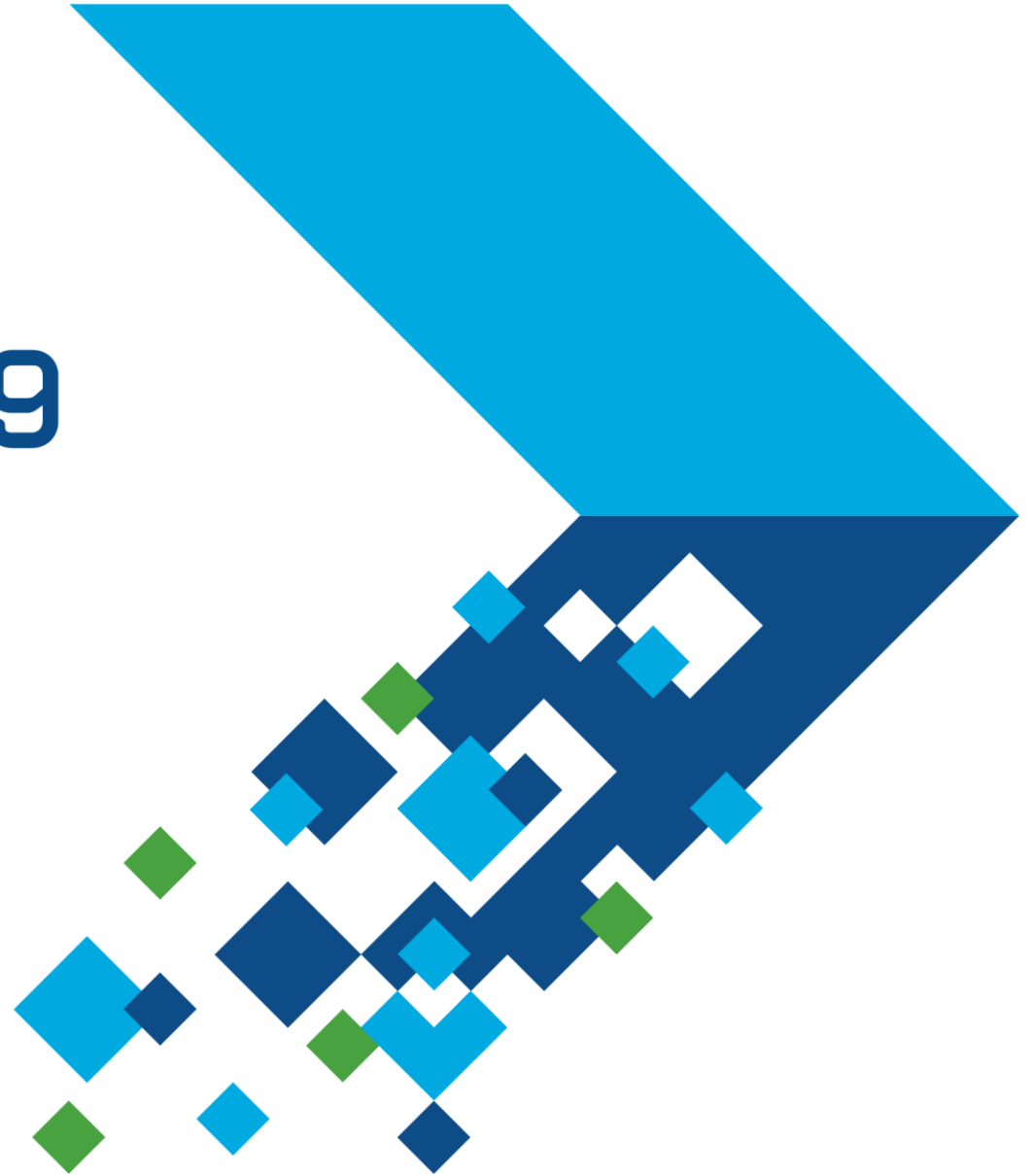


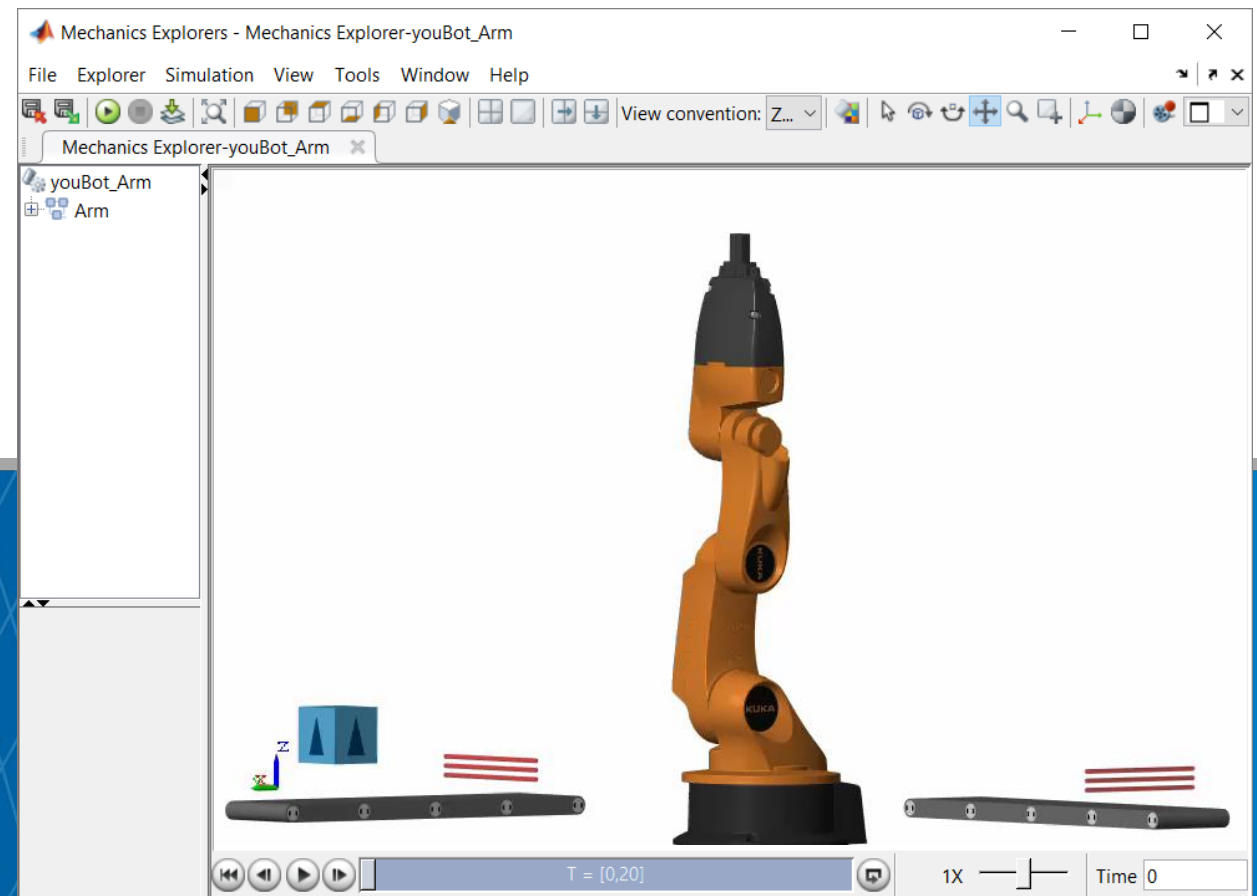
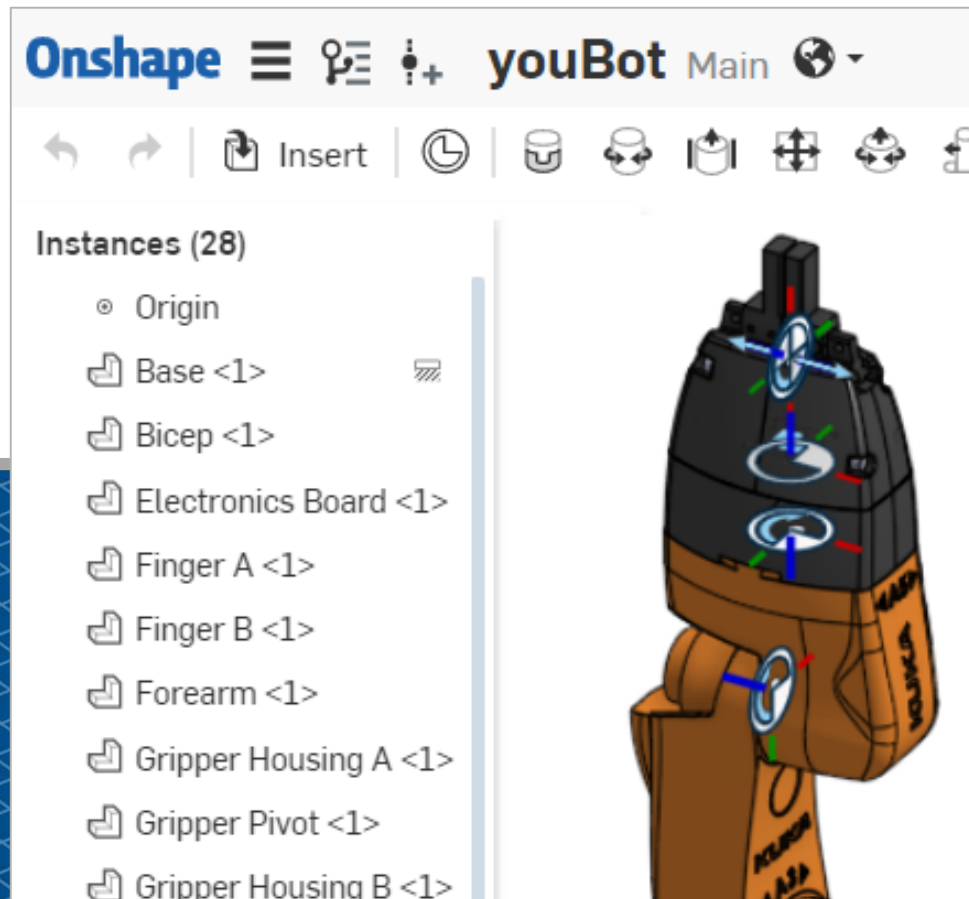
MATLAB EXPO 2019

Optimizing Robotic Systems with Simscape

Veer Alakshendra



Optimizing Robotic Systems with Simscape



In this session

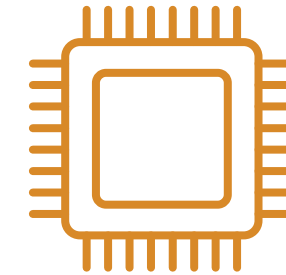
- Simscape and MATLAB enable engineers to combine CAD models with multidomain, dynamic simulation

MATLAB

In this session

- Simscape and MATLAB enable engineers to combine CAD models with multidomain, dynamic simulation

MATLAB



Simulink

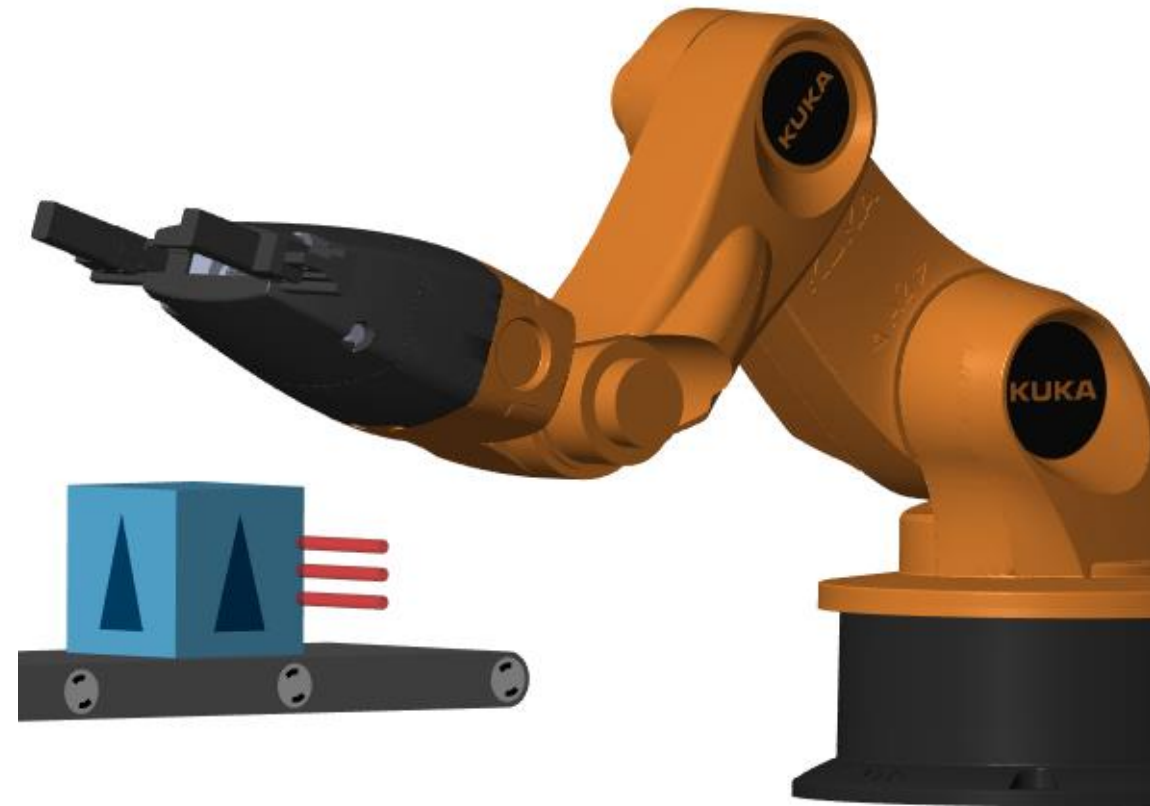


Simscape

Stateflow

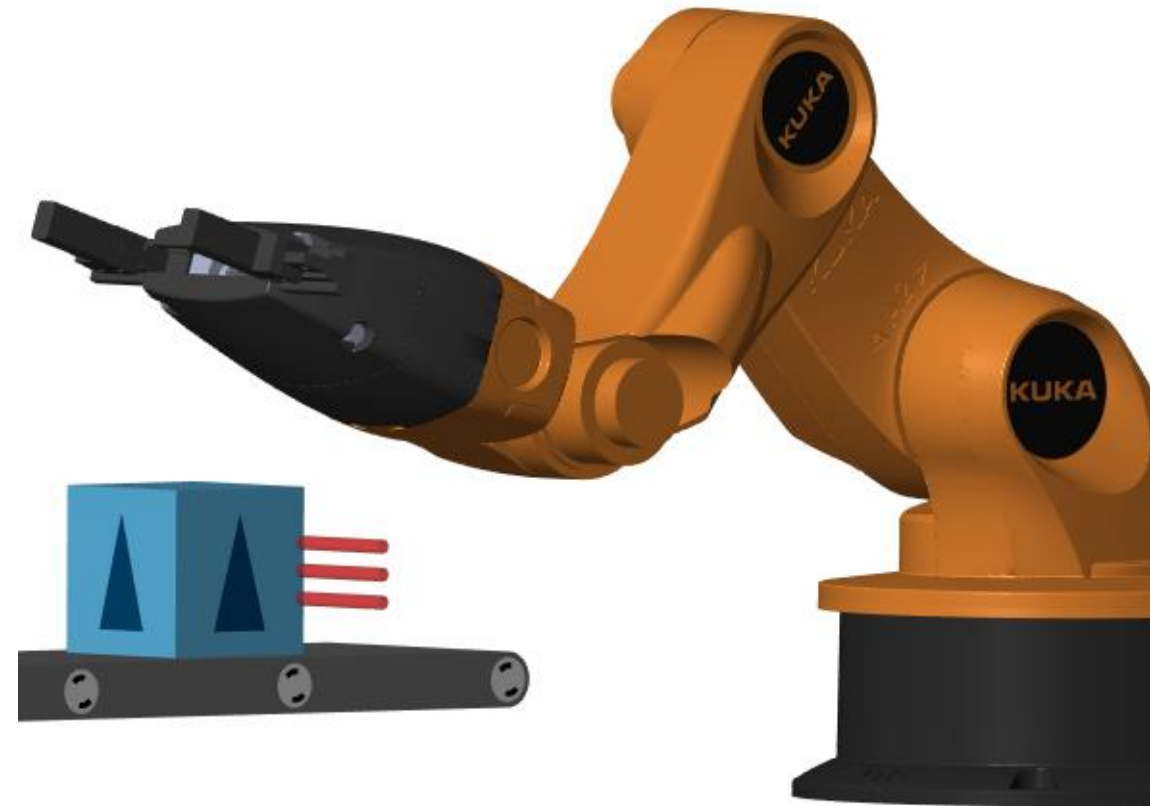
In this session:

- Simscape and MATLAB enable engineers to combine CAD models with multidomain, dynamic simulation
- Results you can achieve:
 1. Optimized mechatronic systems
 2. Improved quality of overall system
 3. Shortened development cycle



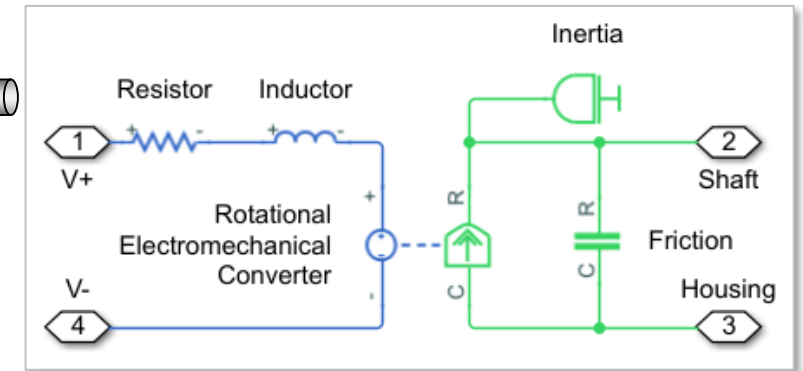
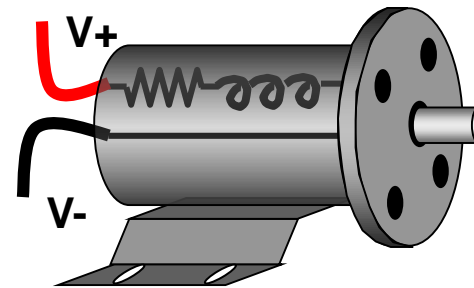
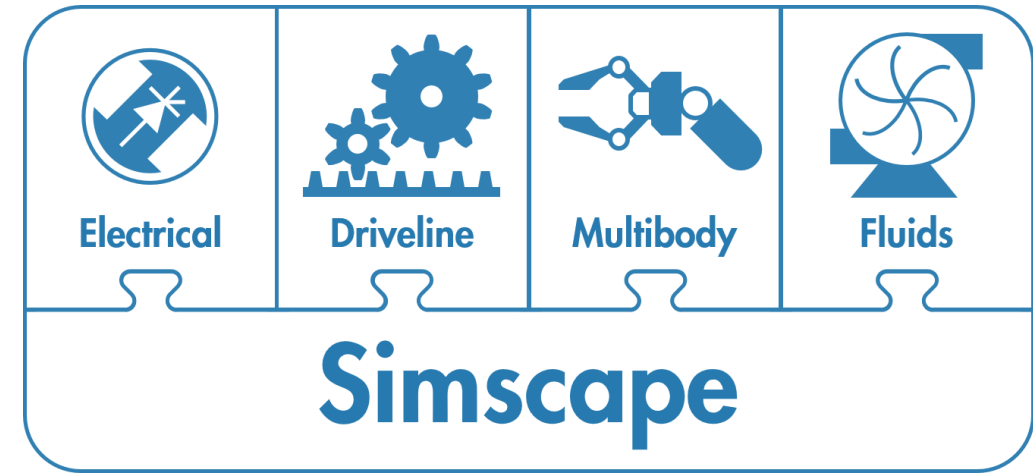
Why Combine CAD and Multidomain Dynamic Simulation?

- **Fewer iterations** on mechanical design because requirements are refined
- **Fewer mechanical prototypes** because mistakes are caught earlier
- **Reduced system cost** because components are not oversized
- **Less system downtime** because system is debugged using virtual commissioning



Simscape Overview

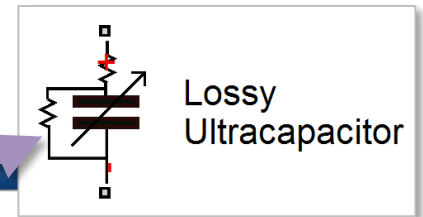
- Enables physical modeling (acausal) of multidomain physical systems
 - Assemble a schematic
 - Equations derived automatically
 - Leverage MATLAB and Simulink



$$i = (C_0 + C_v v) \frac{dv}{dt} + \frac{v}{r_d}$$

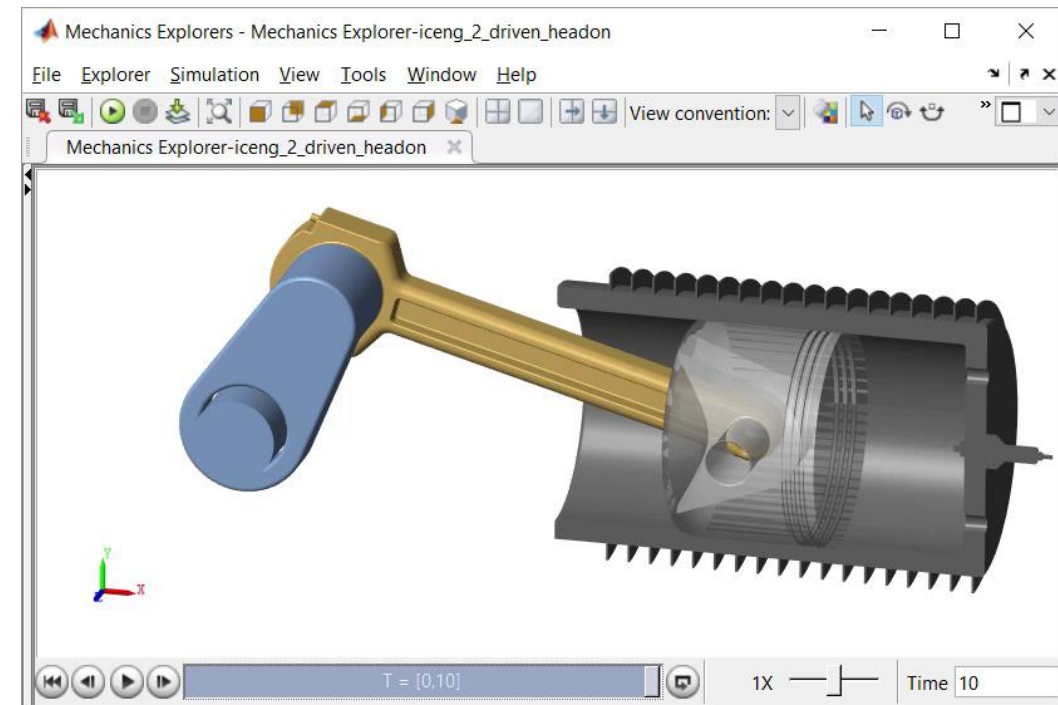
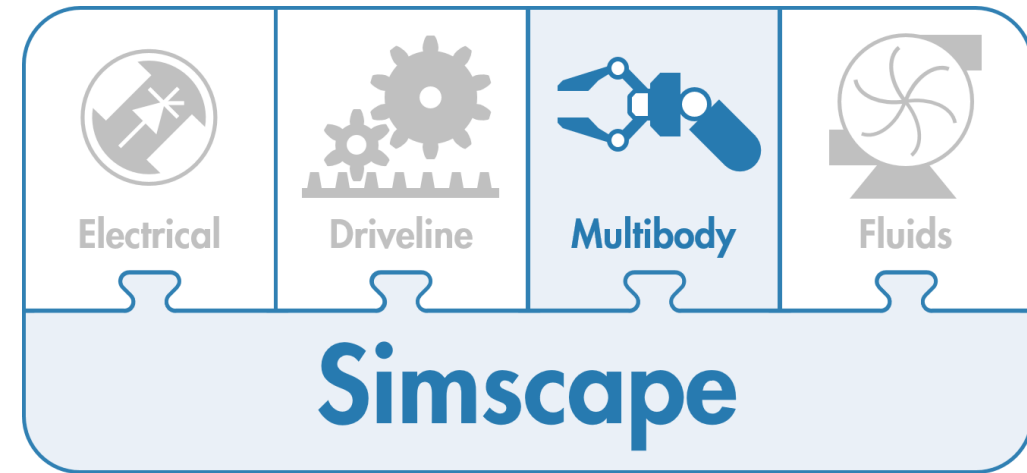
```

Editor - C:\+MyComponent\LossyUltraCapacitor.ssc
40 equations
41 i == (C0 + Cv*vc)*vc.der + vc/Rd;
42 v == vc + i*R;
43 end
    
```



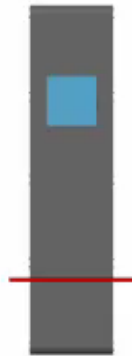
Simscape Multibody Overview

- Enables multibody simulation of 3D mechanical systems
 - Assemble bodies and joints including import from CAD
 - No need to derive and program equations



Design Challenge

System:

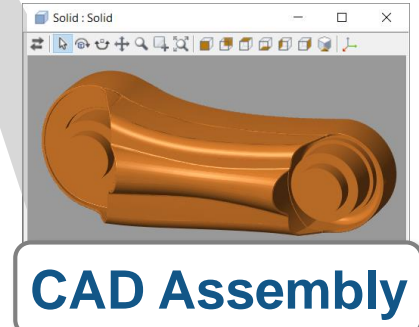
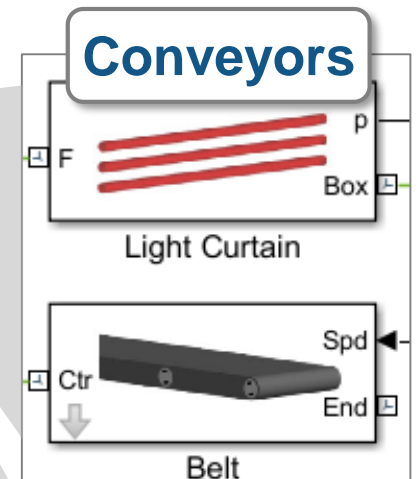
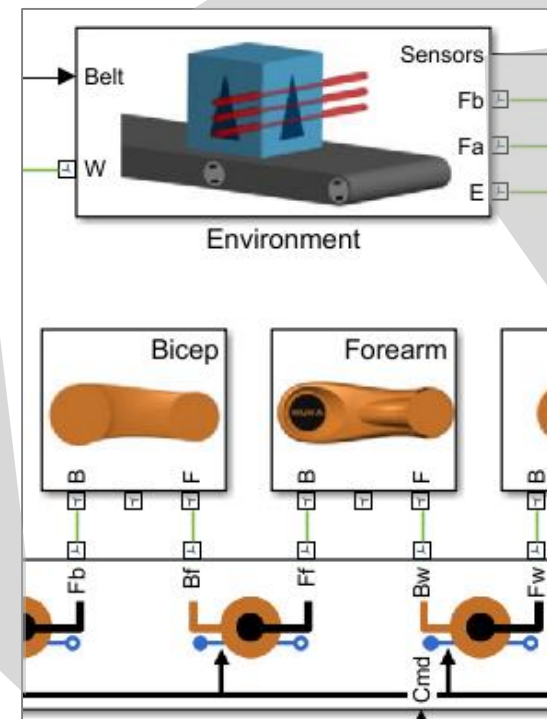
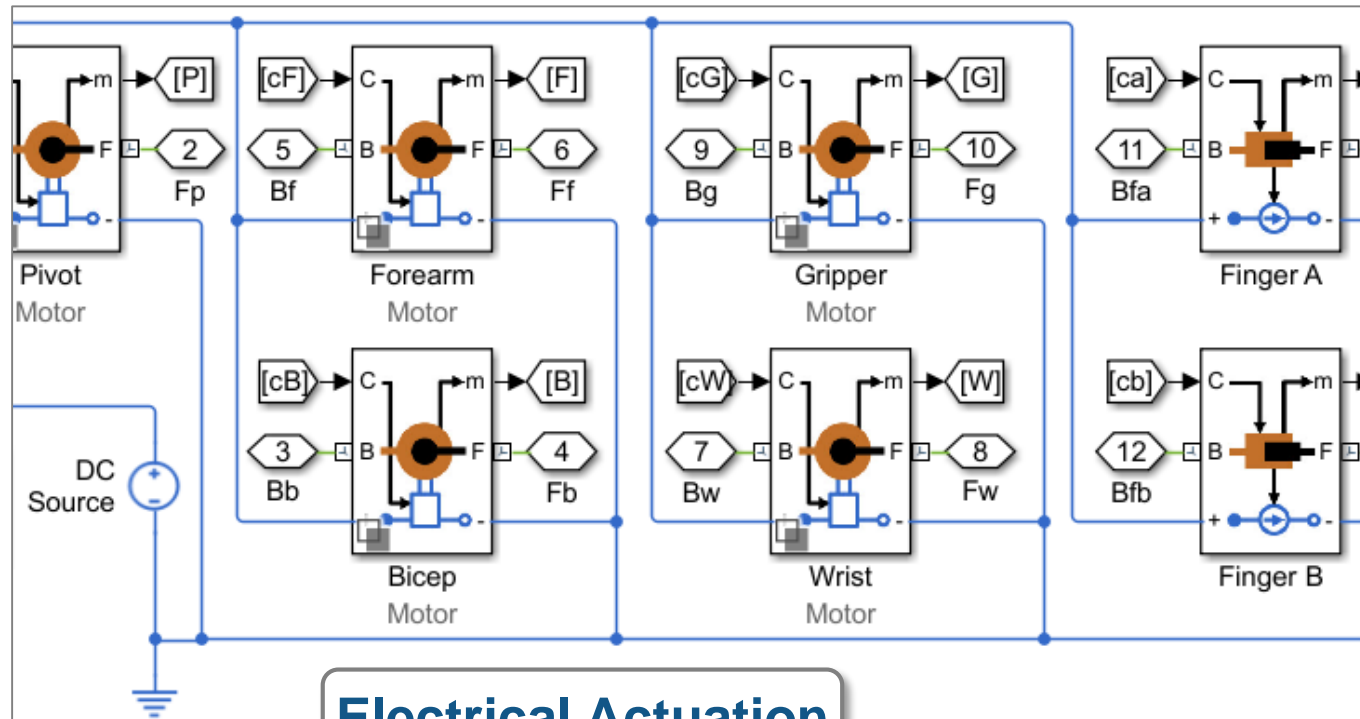
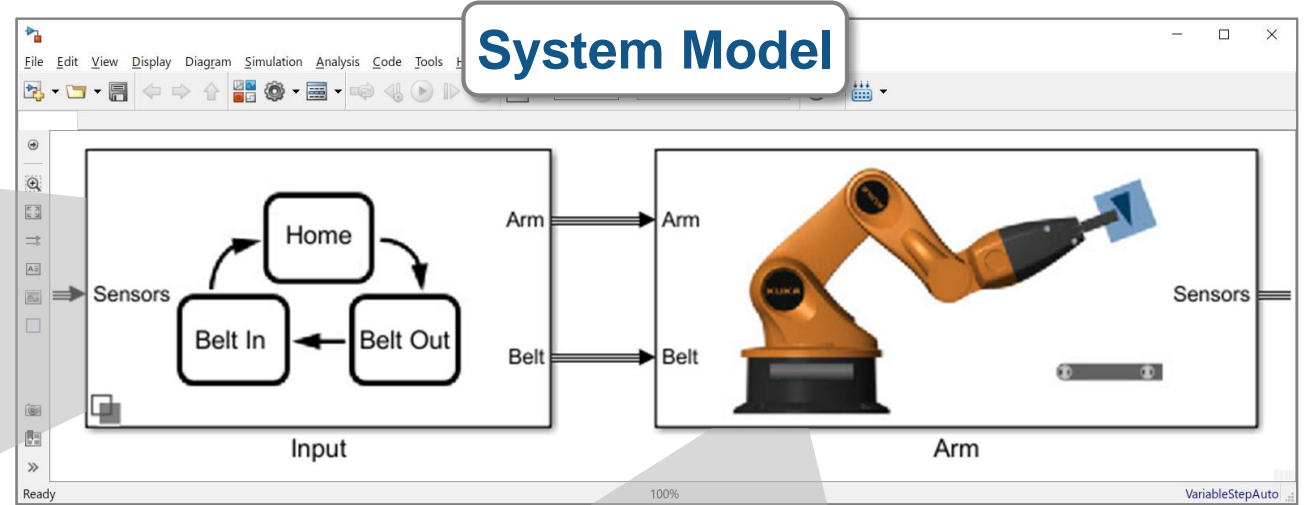
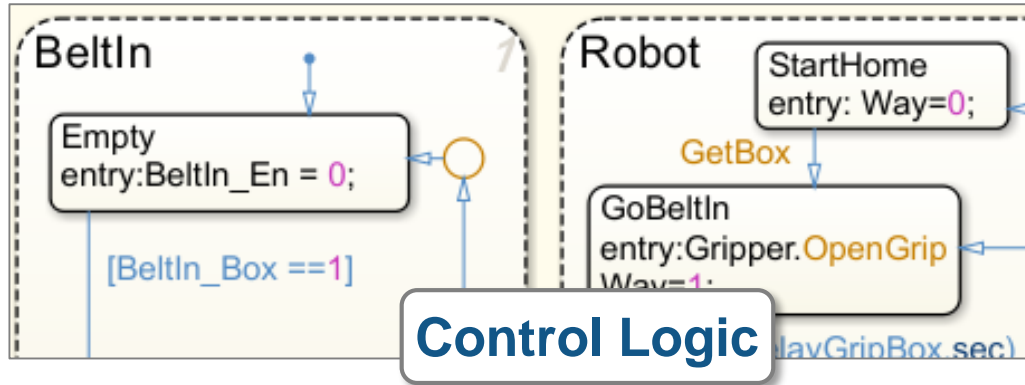


Challenge: Select motors and define controls for robot and conveyor belts.

Solution: Import CAD model into Simscape; use simulation to define actuator requirements and control logic

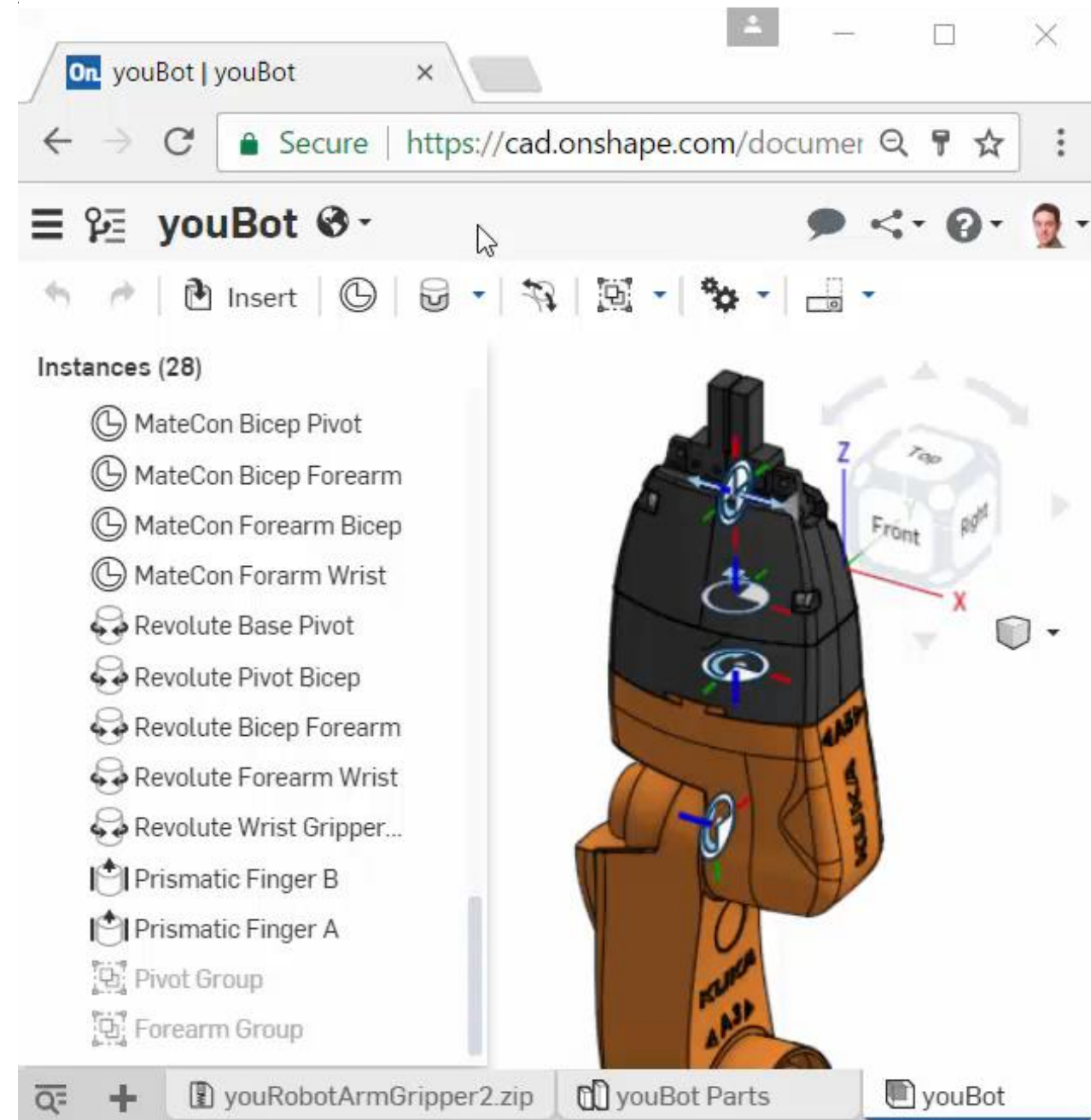
1. Import CAD Model
2. Determine Motor Requirements
3. Integrate Electrical Actuators
4. Minimize Power Consumption
5. Develop Control Logic

System Model



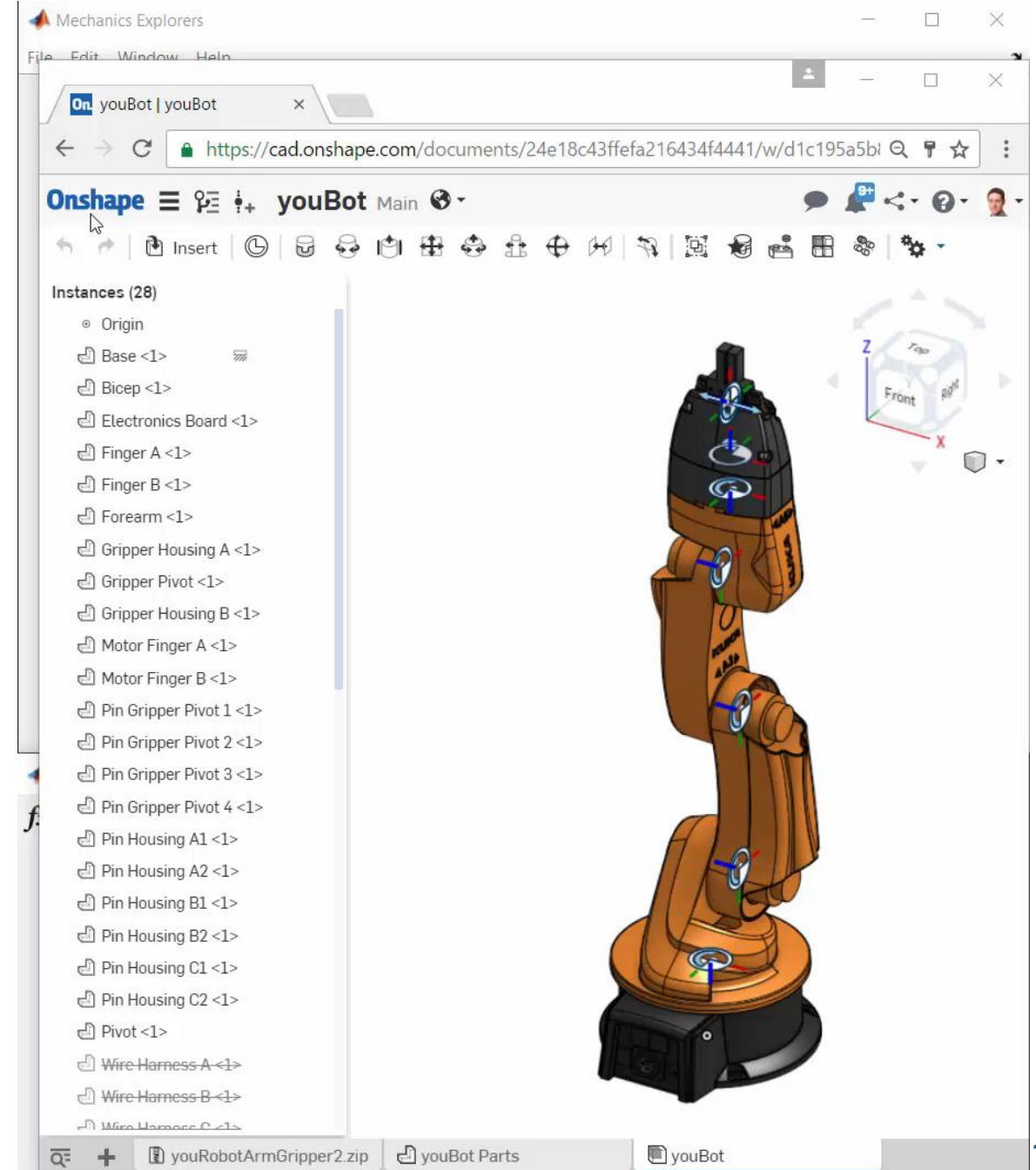
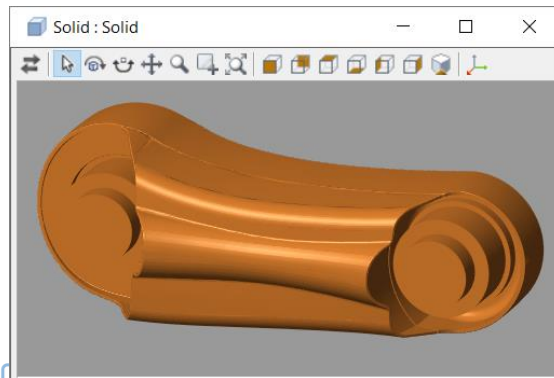
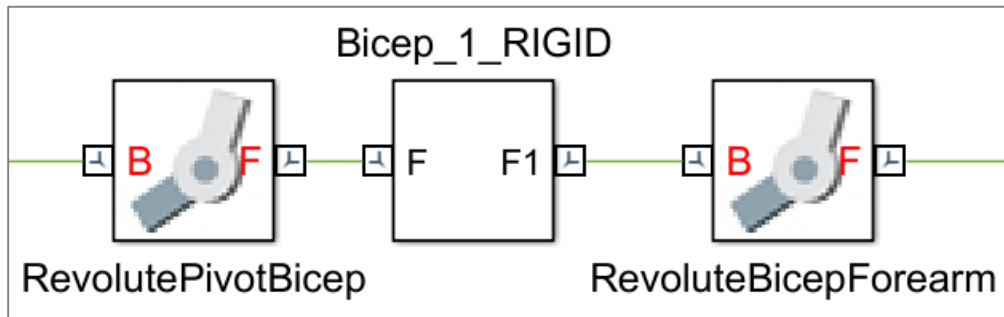
Kuka Robot

- 5 degrees of freedom, and a gripper
- Key advantage of Onshape: Ability to directly define joints
 - Exact mapping to constraints used in multibody simulation
- System engineer reuses mechanical design in dynamic simulation



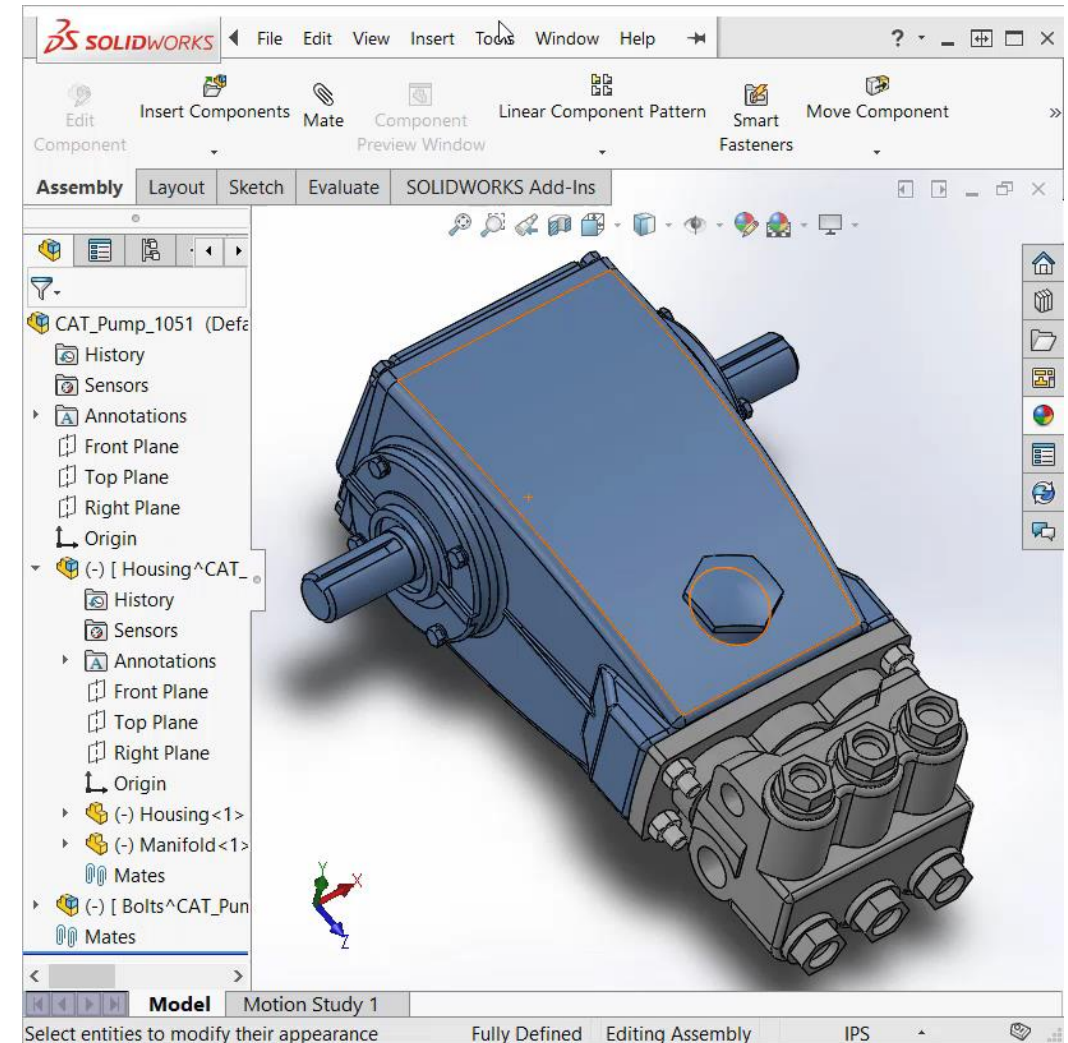
1. Import Model from CAD

- Convert CAD assembly to dynamic simulation model for use within Simulink
 - Mass, inertia, geometry, and joints



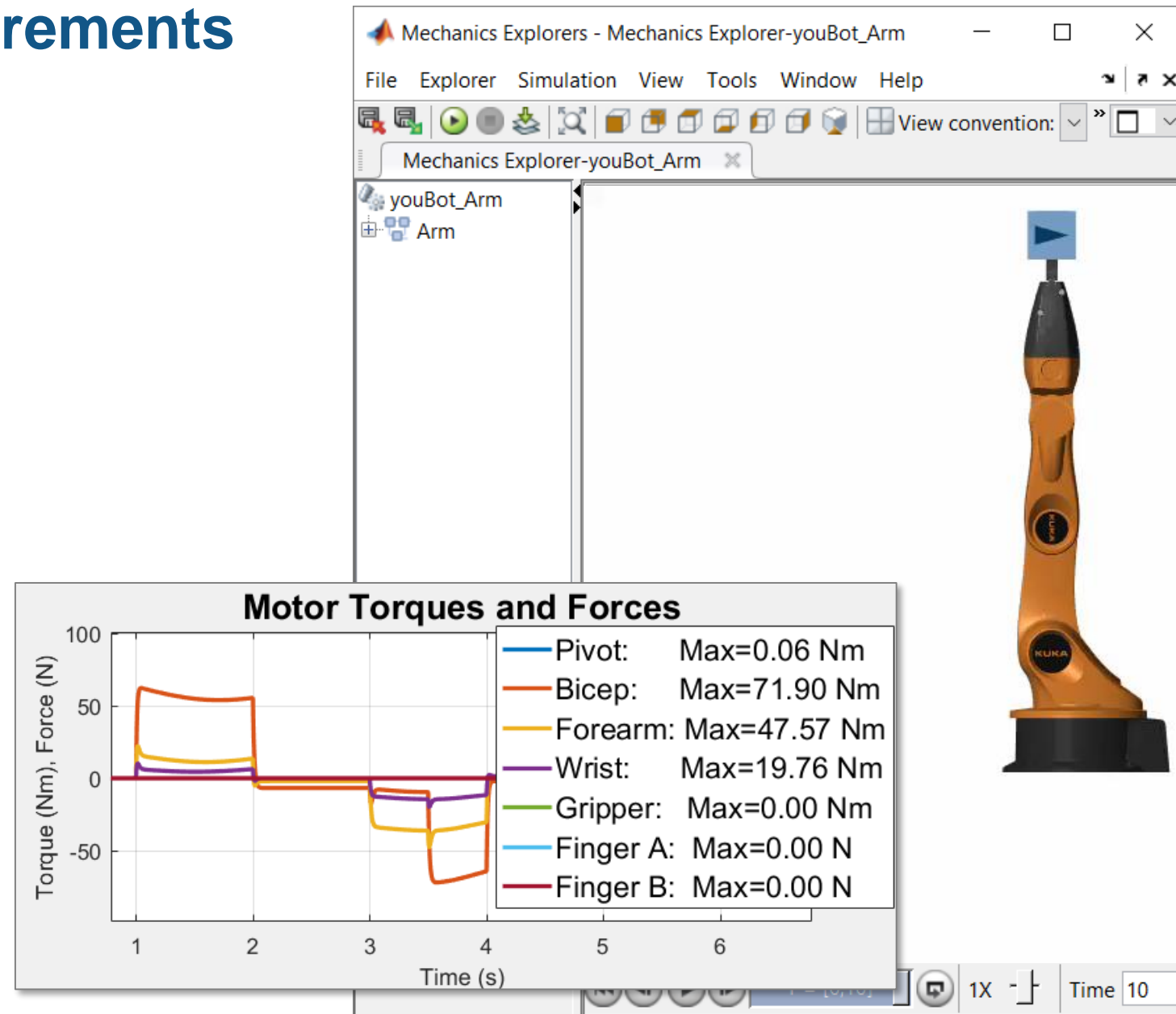
Simscape Multibody CAD Import

- Import CAD assemblies
 - Part definitions
 - Converts mate definitions to joints
 - SOLIDWORKS, Inventor, Onshape, and PTC Creo® (Pro/ENGINEER®)
- Import CAD Parts
 - CATIA, NX, SolidEdge, and others
 - STEP files



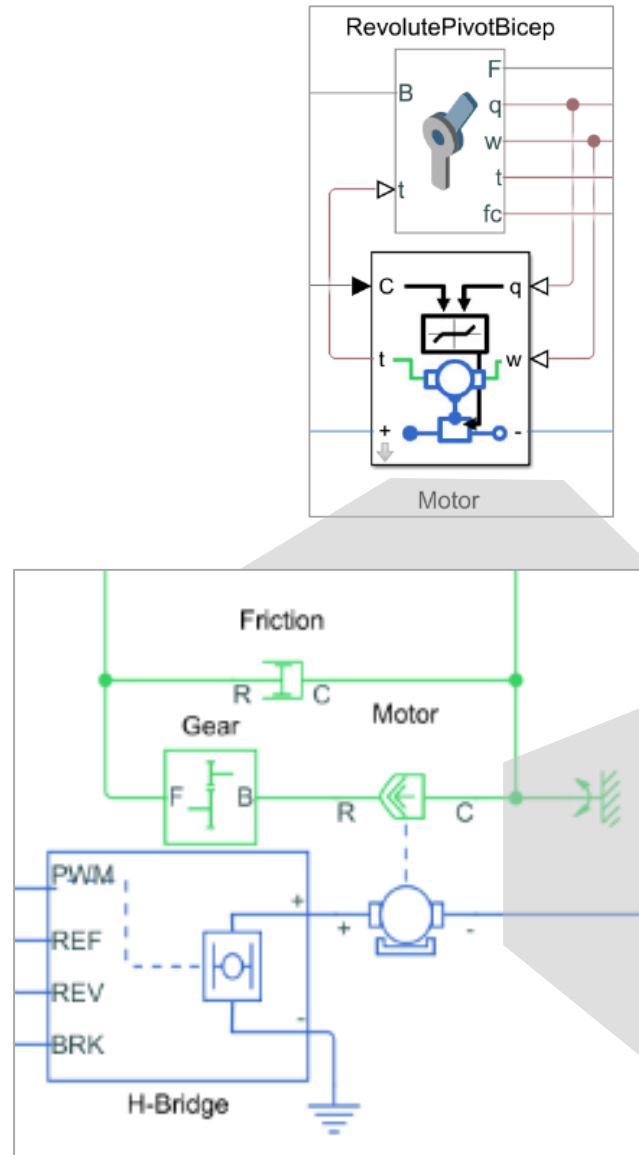
2. Determine Motor Requirements

- Define and run a set of tests
 - Maximum payload, speed
 - Worst case friction levels
 - Full range of movement
- Use dynamic simulations to calculate required torque and bearing forces
- If design changes, automatically rerun tests and re-evaluate results



3. Integrate Electrical Actuators

- Add motors, drive circuitry, gears, and friction
- Choose motors based on torque requirements
- Assign parameters directly from data sheets



Motor Data

251601

Characteristics

Terminal resistance	Ω	0.978
Terminal inductance	mH	0.573
Torque constant	mNm / A	33.5
Speed constant	rpm / V	285
Speed / torque gradient	rpm / mNm	8.32
Mechanical time constant	ms	11.8
Rotor inertia	gcm ²	135

Electrical Torque

Mechanical

Model parameterization: Circuit parameters

Armature resistance: 0.978 Ohm

Armature inductance: 0.573 mH

Torque constant: 33.5 mN*m/A

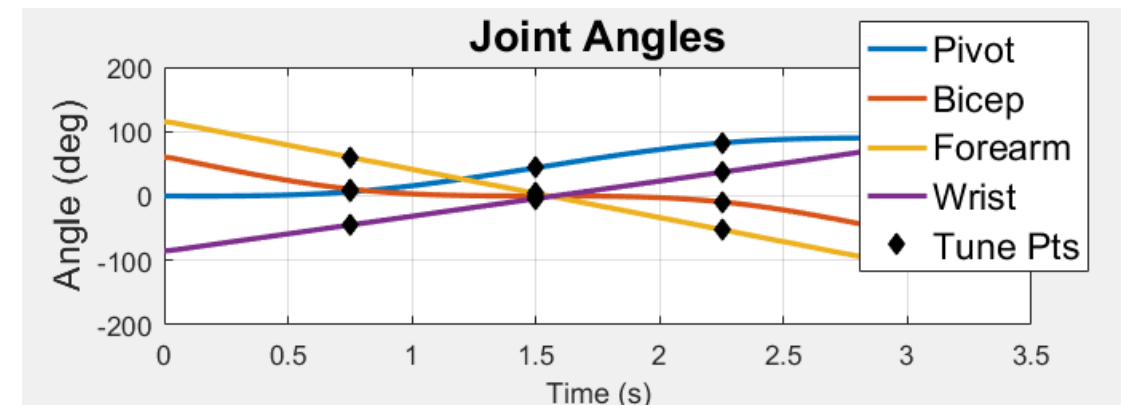
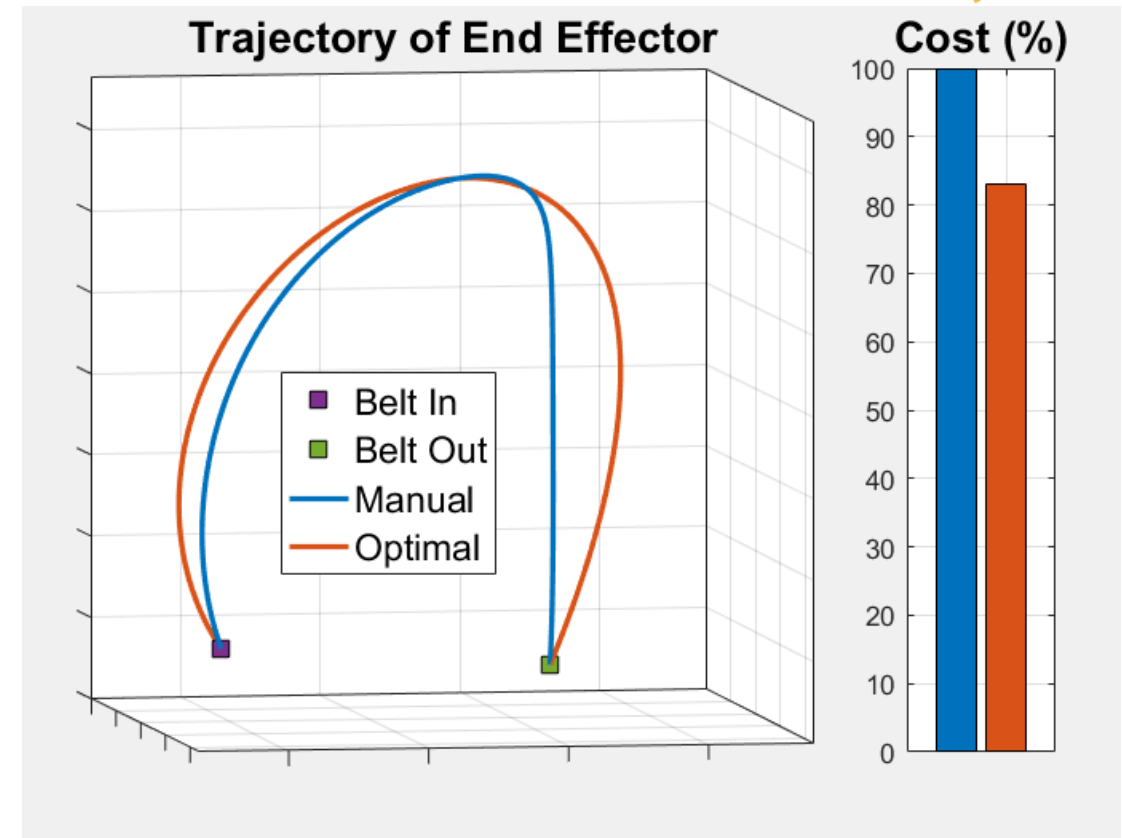
4. Minimize Power Consumption

Model:

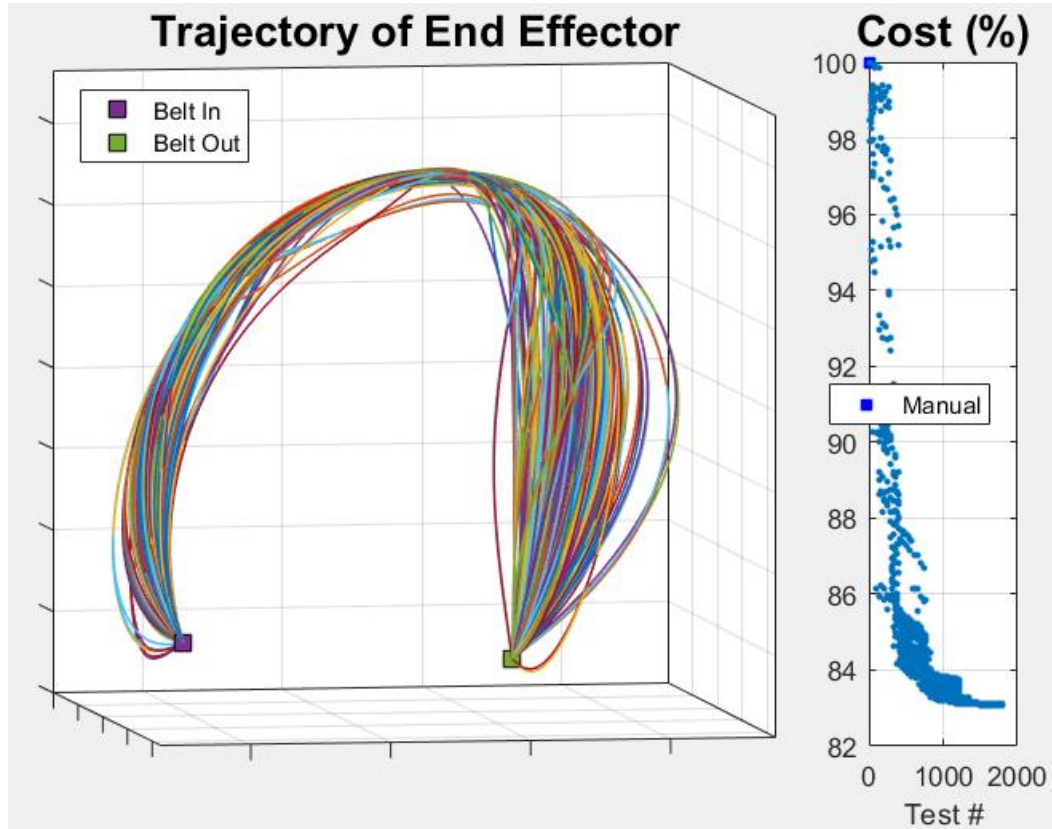


Challenge: Identify arm trajectory that minimizes power consumption.

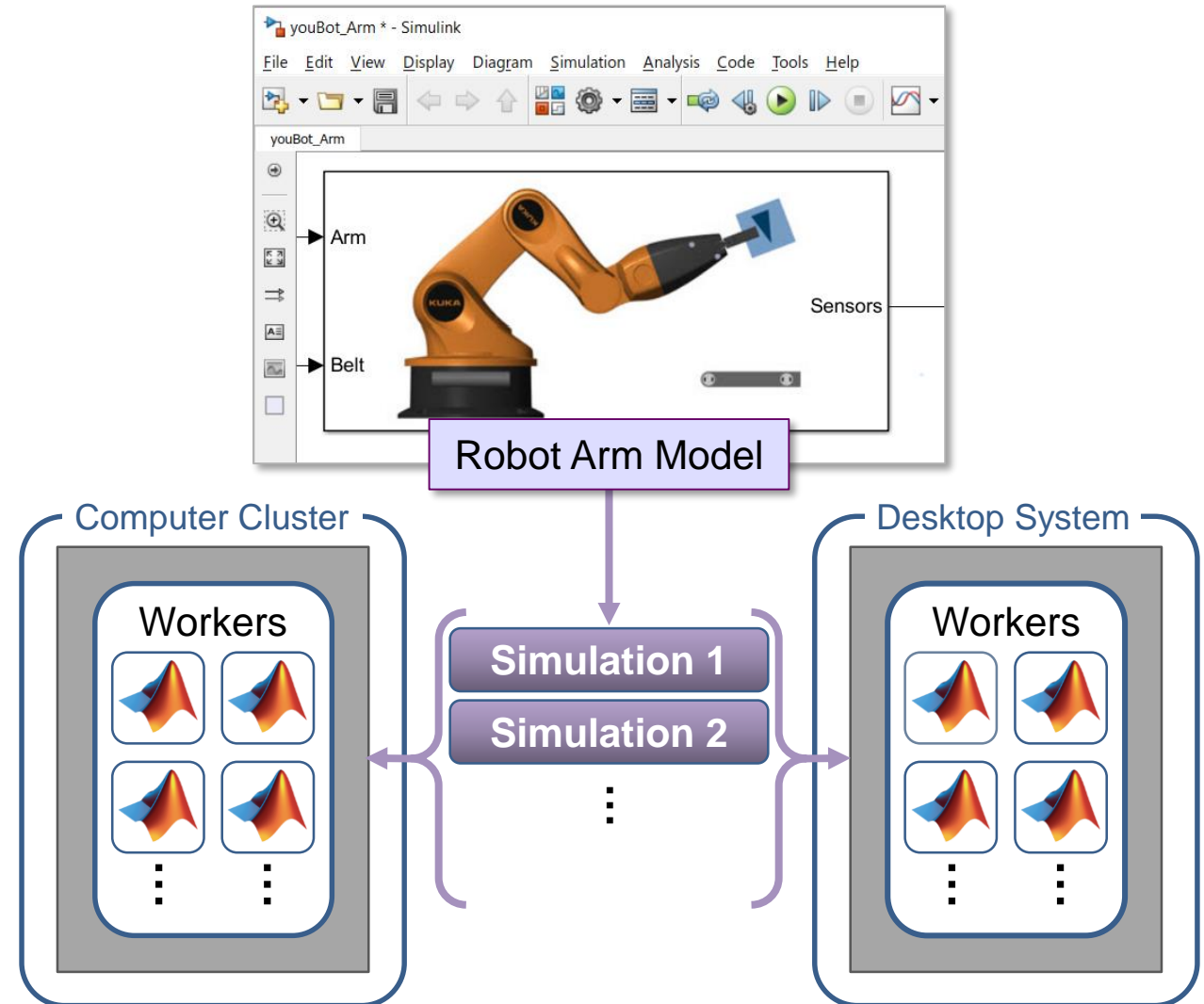
Solution: Use dynamic simulation to calculate power consumption, and use optimization algorithms to tune trajectory.



Accelerate Design Iterations Using Parallel Computing



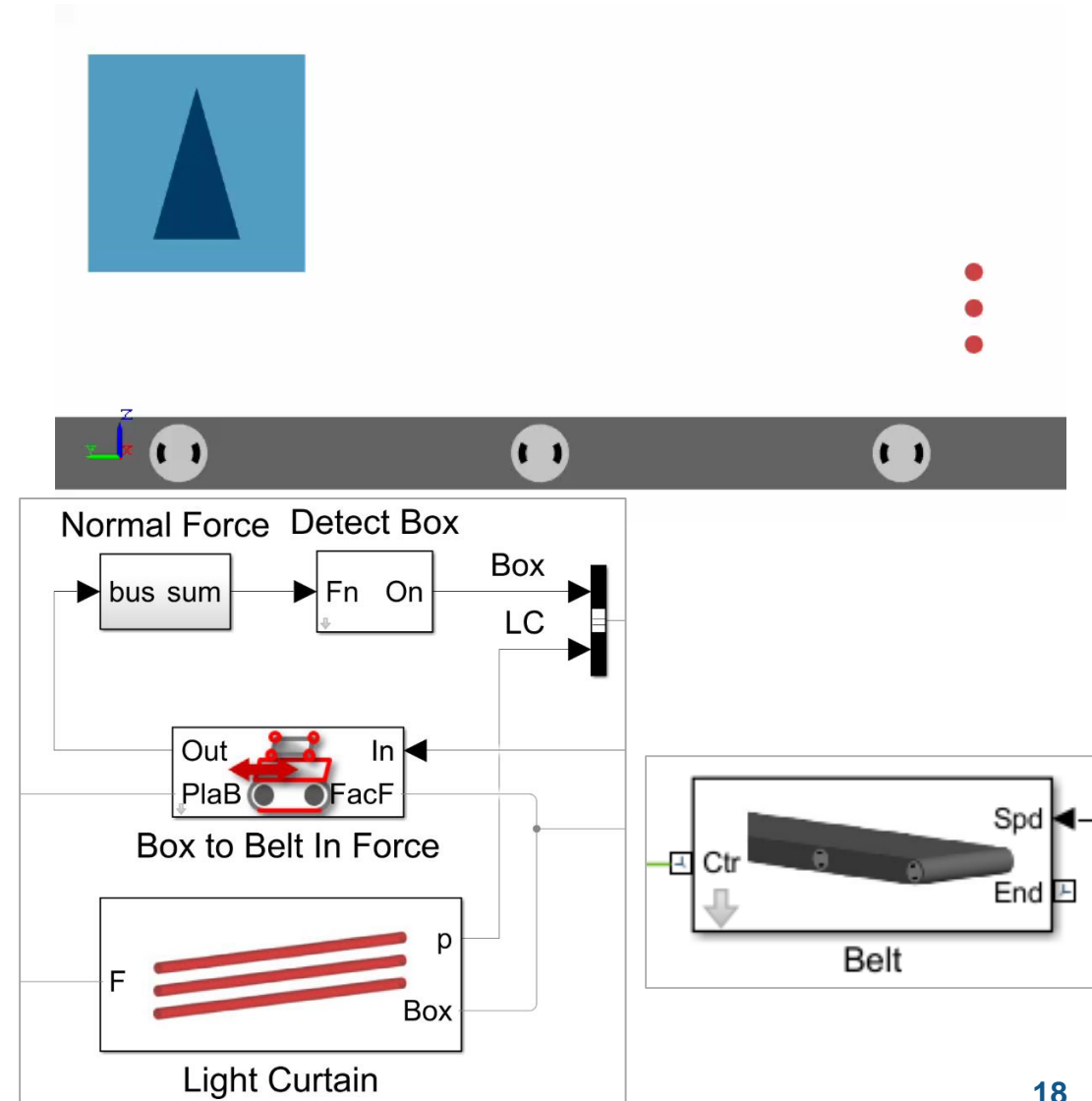
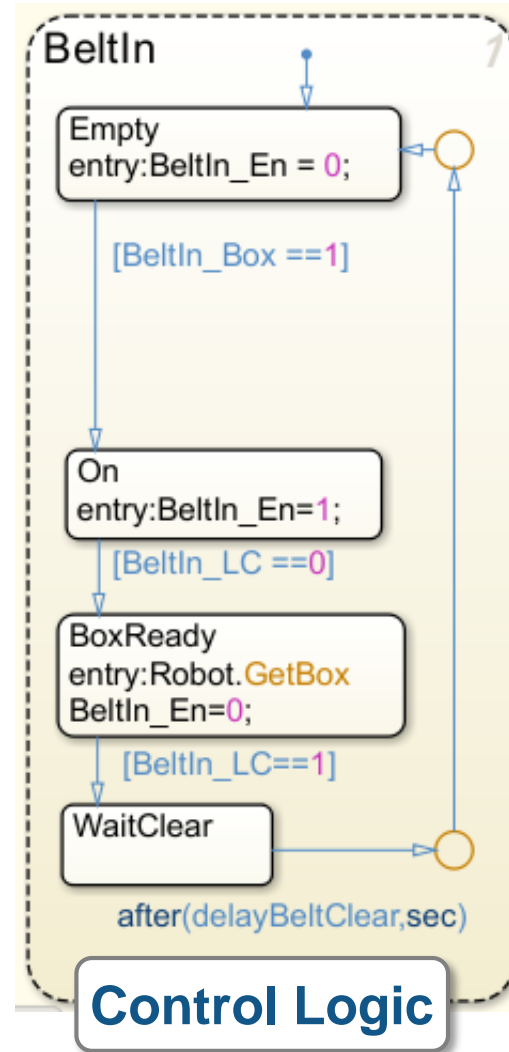
This optimization task required nearly 2000 simulations.



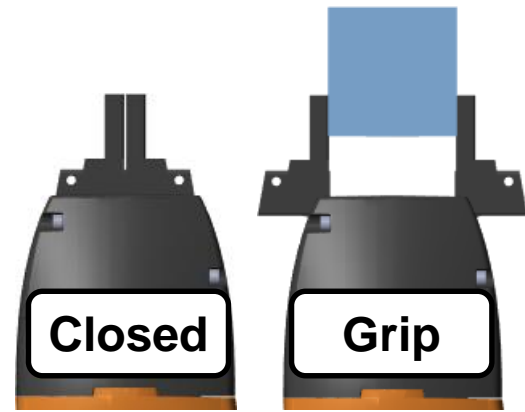
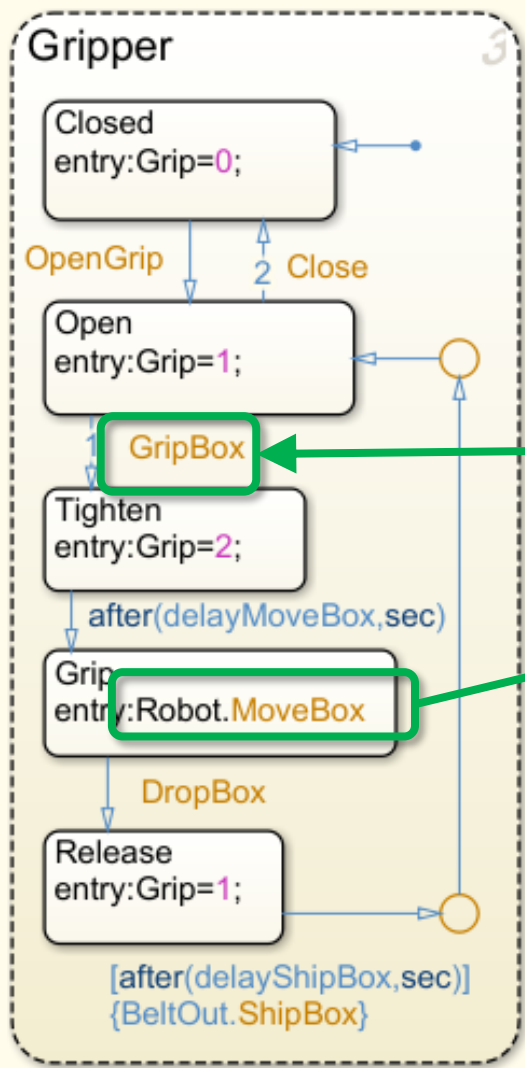
Running simulations in parallel speeds up your testing process.

5. Design Control Logic for Arm and Conveyor Belts

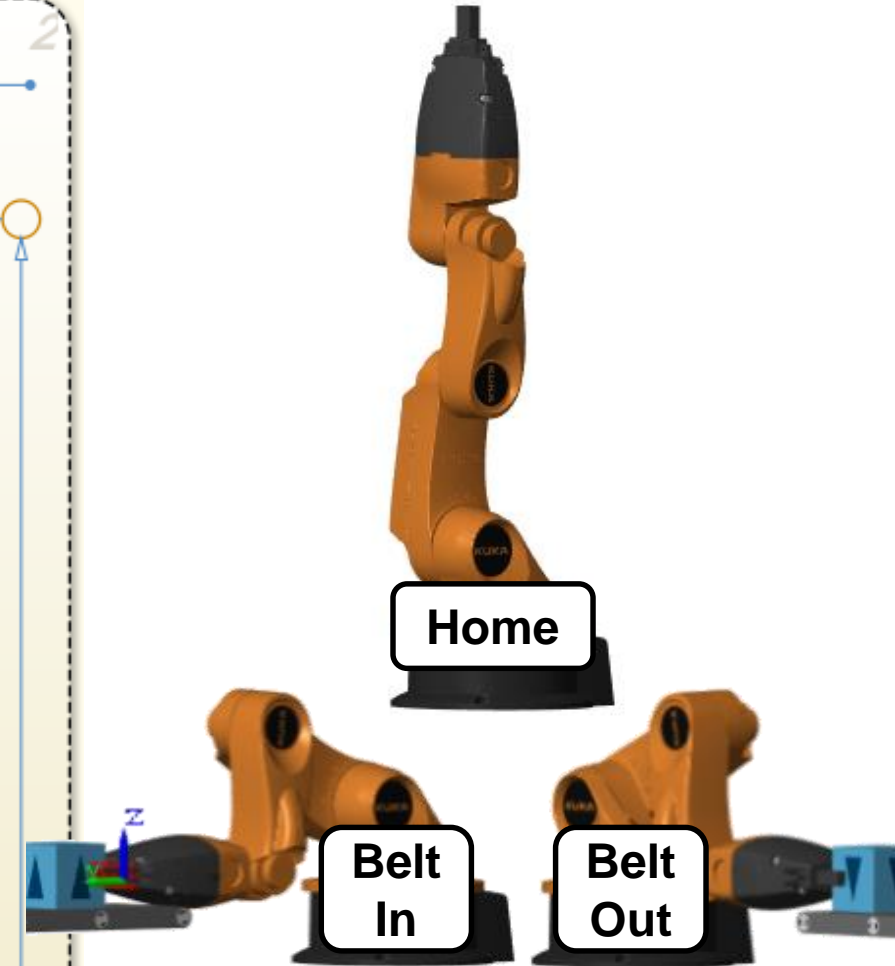
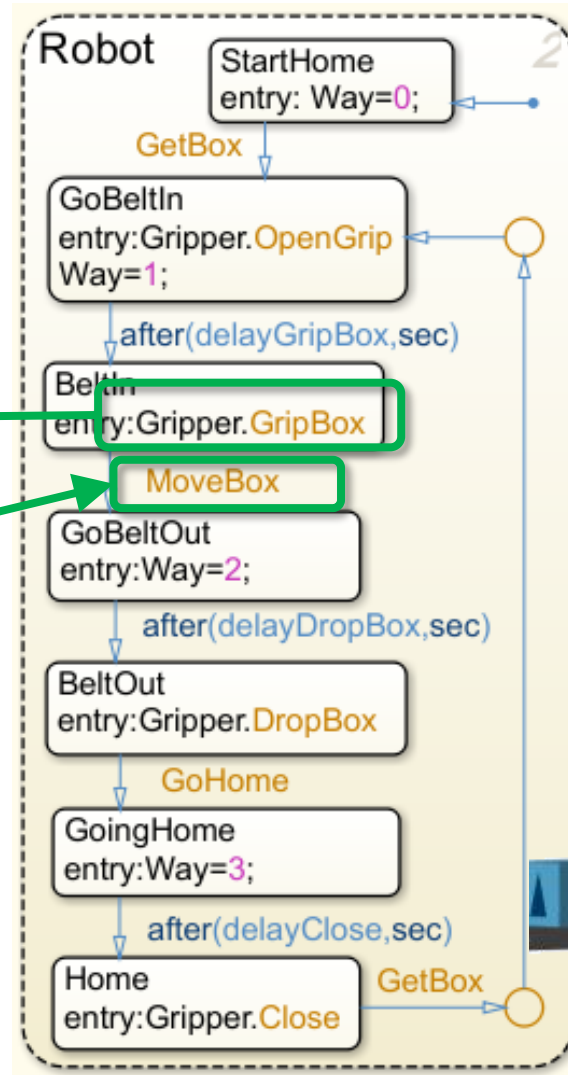
- Sense quantities within model that govern system events
- Design logic using a state chart
- Use outputs of logic to control models of system components



5. Design Control Logic for Arm and Conveyor Belts



State charts depend on each other



5. Design Control Logic for Arm and Conveyor Belts

Stateflow (chart) youBot_Arm/Input/Control/Logic* - Simulink

File Edit View Display Chart Simulation Analysis Code Tools Help

Logic

BeltIn

- Empty entry: BeltIn_En = 0;
- [BeltIn_Box == 1]
- On entry: BeltIn_En = 1;
- [BeltIn_LC == 0]
- BoxReady entry: Robot.GetBox; BeltIn_En = 0;
- [BeltIn_LC == 1]
- WaitClear
- after(delayBeltClear, sec)

Robot

- StartHome entry: Way = 0;
- GetBox
- GoBeltIn entry: Gripper.OpenGrip; Way = 1;
- after(delayGripBox, sec)
- BeltIn entry: Gripper.GripBox
- MoveBox
- GoBeltOut entry: Way = 2;
- after(delayDropBox, sec)
- BeltOut entry: Gripper.DropBox
- GoHome
- GoingHome entry: Way = 3;
- after(delayClose, sec)
- Home entry: Gripper.Close; GetBox

Gripper

- Closed entry: Grip = 0;
- OpenGrip
- Open entry: Grip = 1;
- GripBox
- Tighten entry: Grip = 2;
- after(delayMoveBox, sec)
- Grip entry: Robot.MoveBox
- DropBox
- Release entry: Grip = 1;
- [after(delayShipBox, sec)]
- [BeltOut.ShipBox]
- Close

BeltOut

- Empty entry: BeltOut_En = 0;
- [BeltOut_Box == 1]
- WaitRelease
- ShipBox
- On entry: BeltOut_En = 1;
- [BeltOut_LC == 0]
- BoxReady entry: Robot.GoHome; BeltOut_En = 0;
- [BeltOut_LC == 1]
- WaitClear
- after(delayBeltClear, sec)

Running 100% ode15s

Mechanics Explorers - Mechanics Explorer-youBot_Arm

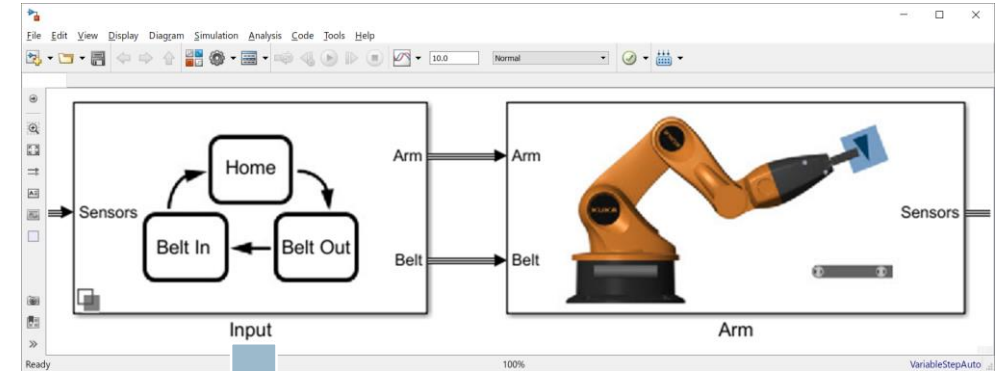
File Explorer Simulation View Tools Window Help

Mechanics Explorer-youBot_Arm

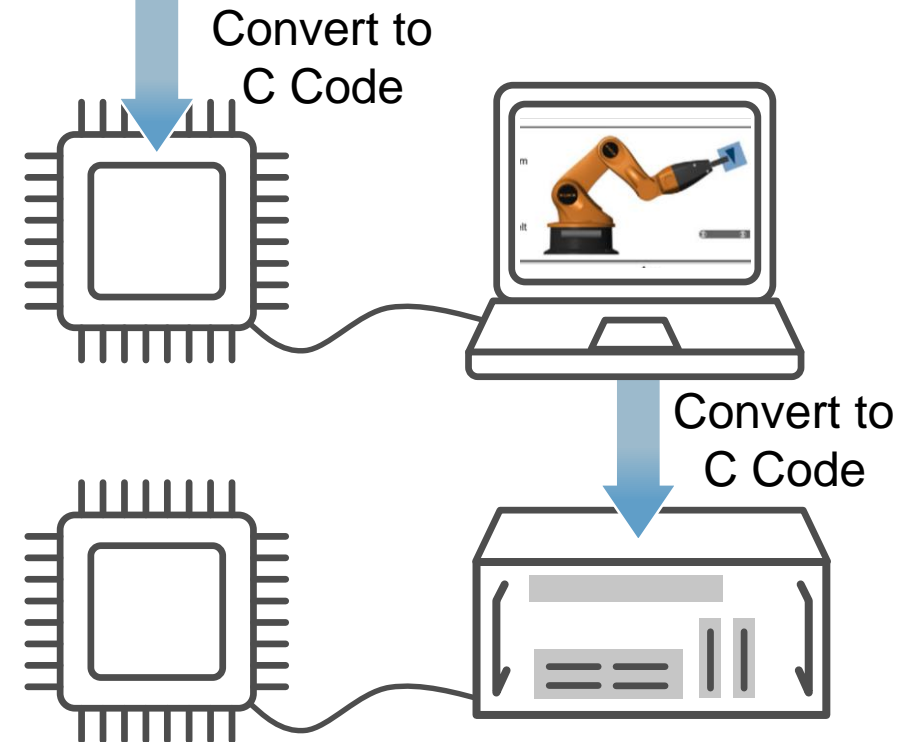
0% 1X Time 0

Test Production Control Software

- Automatically convert algorithms to production code
 - C Code, IEC 61131-3 Code
- Incrementally test the effect of each conversion step
 - Fixed-point math
 - Latency on production controller
- Use the same plant model
 - Test without expensive hardware prototypes



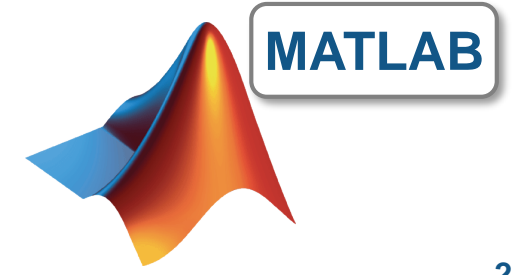
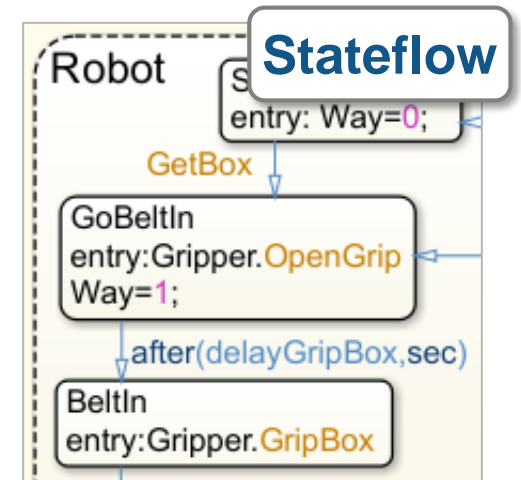
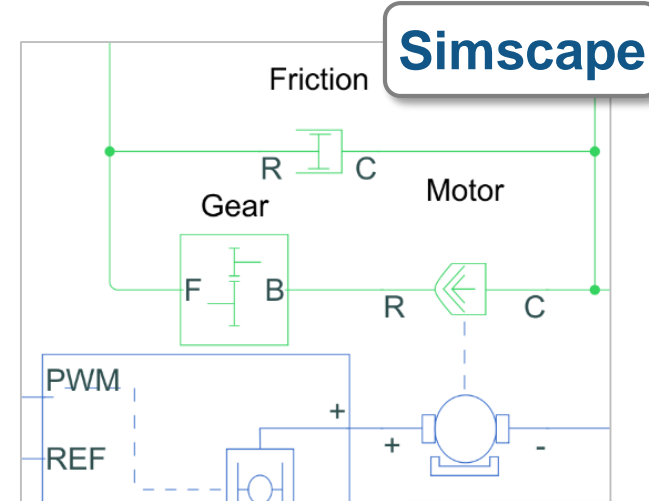
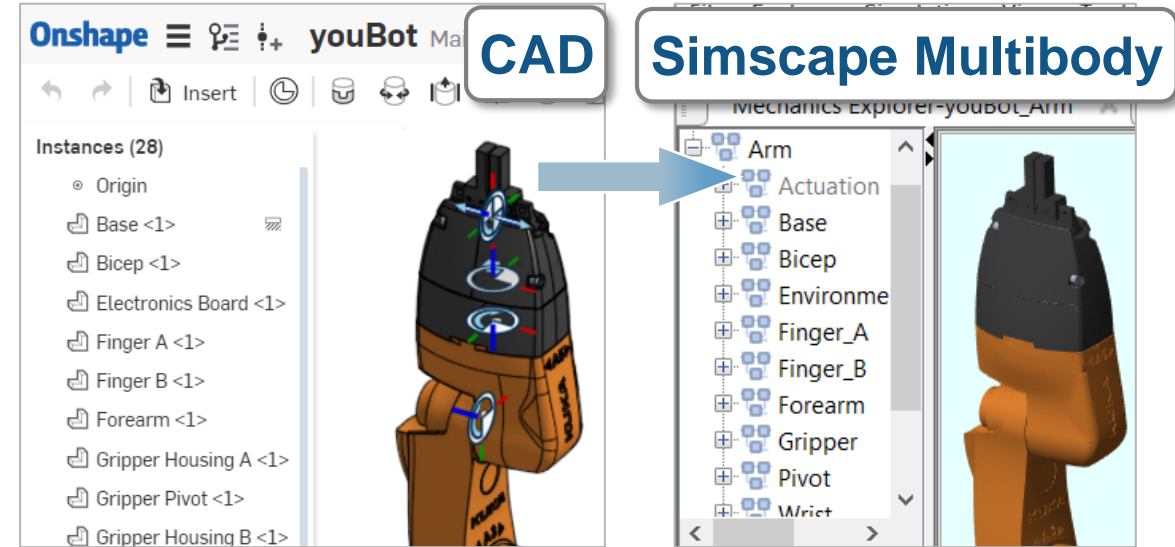
Processor-in-the-Loop (PIL)



Hardware-in-the-Loop (HIL)

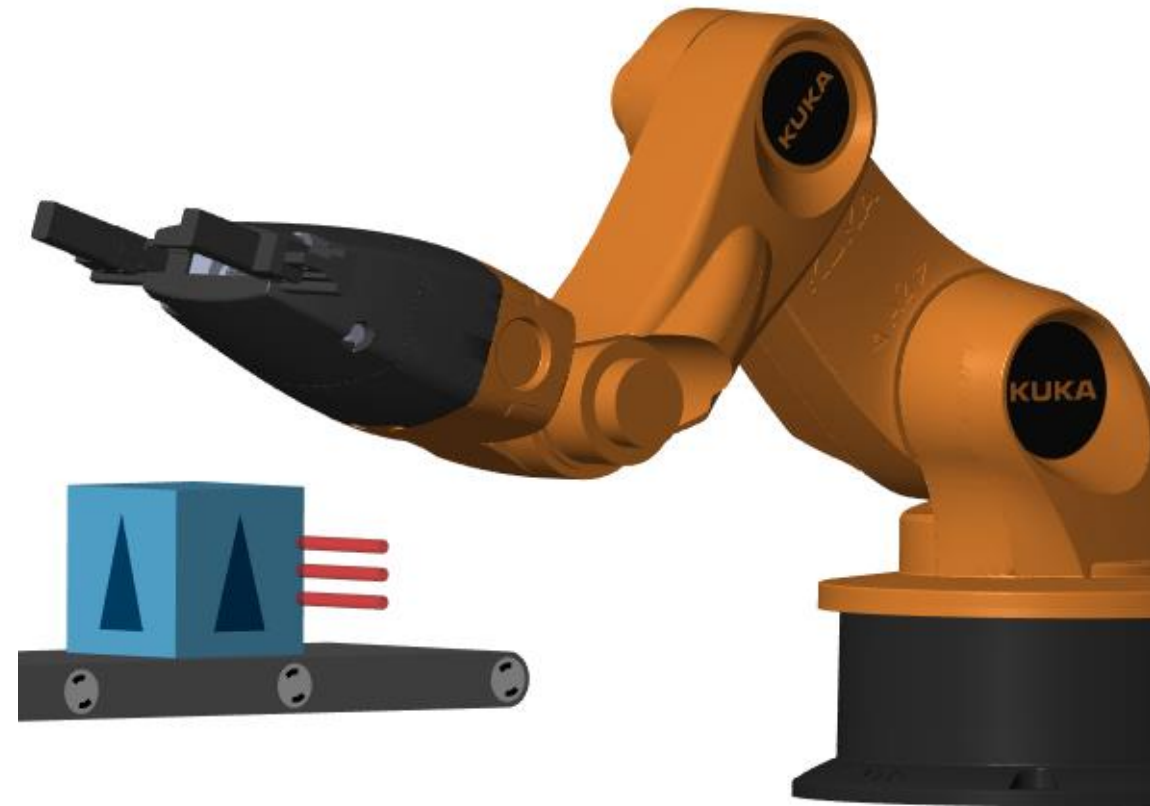
How we did it

- Convert CAD assemblies into dynamic simulation models with **Simscape Multibody**
- Add electric actuators with **Simscape** and control logic using **Stateflow**
- Optimize system using **MATLAB**
- Perform dynamic simulation in **Simulink**



Summary

- Simscape and MATLAB enable engineers to combine CAD models with multidomain, dynamic simulation
- Results:
 1. Optimized mechatronic systems
 2. Improved quality of overall system
 3. Shortened development cycle





Modeling Physical Systems with Simscape

- Create models in various physical domains
- Combine Simulink and Simscape models
- Model energy transfer between different physical domains
- Create user-defined Simscape components

Thank You

Email: valakshe@mathworks.com

LinkedIn: <https://www.linkedin.com/in/veer-alakshendra-b5324222/>