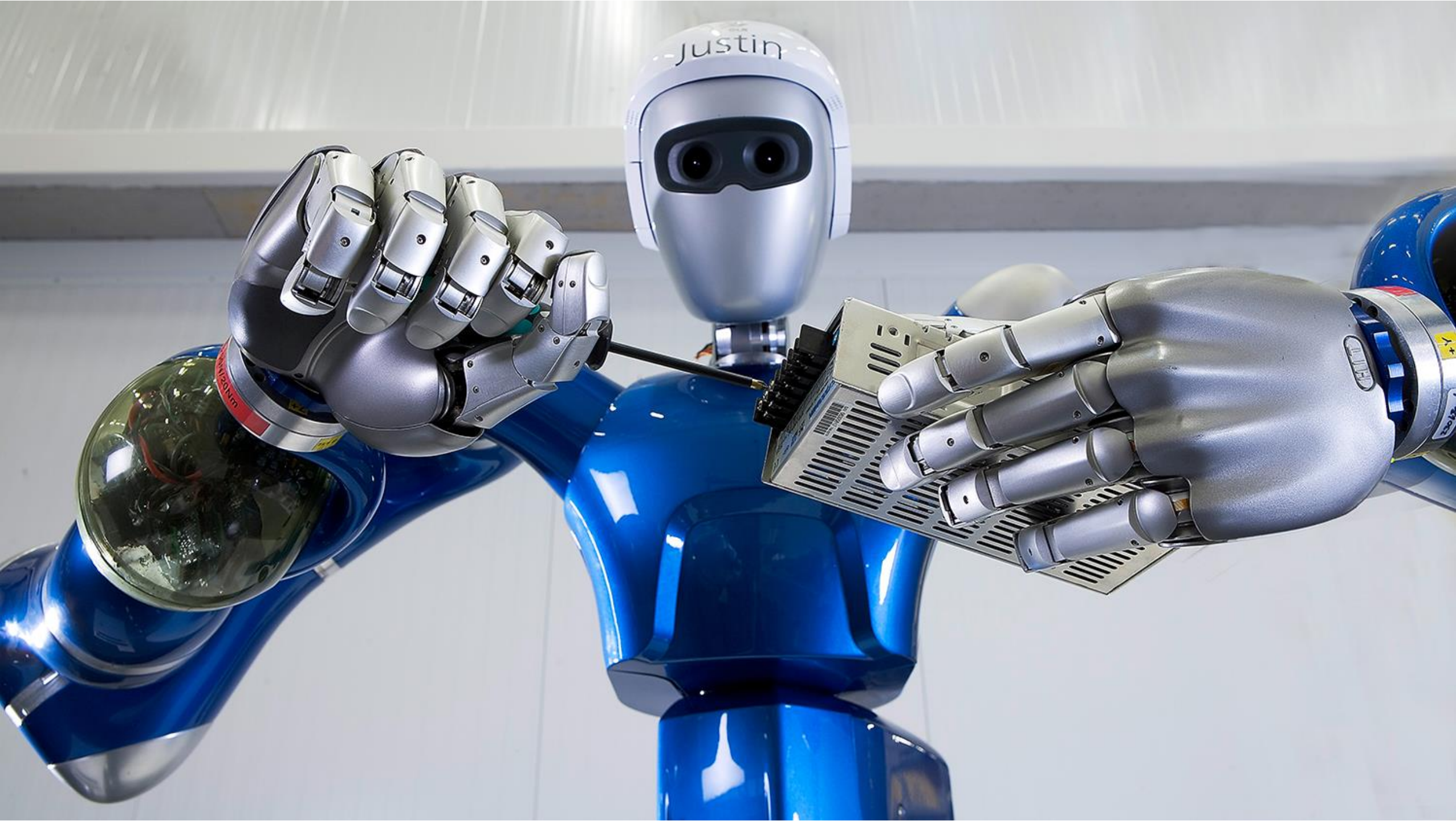


# MATLAB EXPO 2019

Developing Autonomous Robots with  
MATLAB and Simulink

Veer Alakshendra









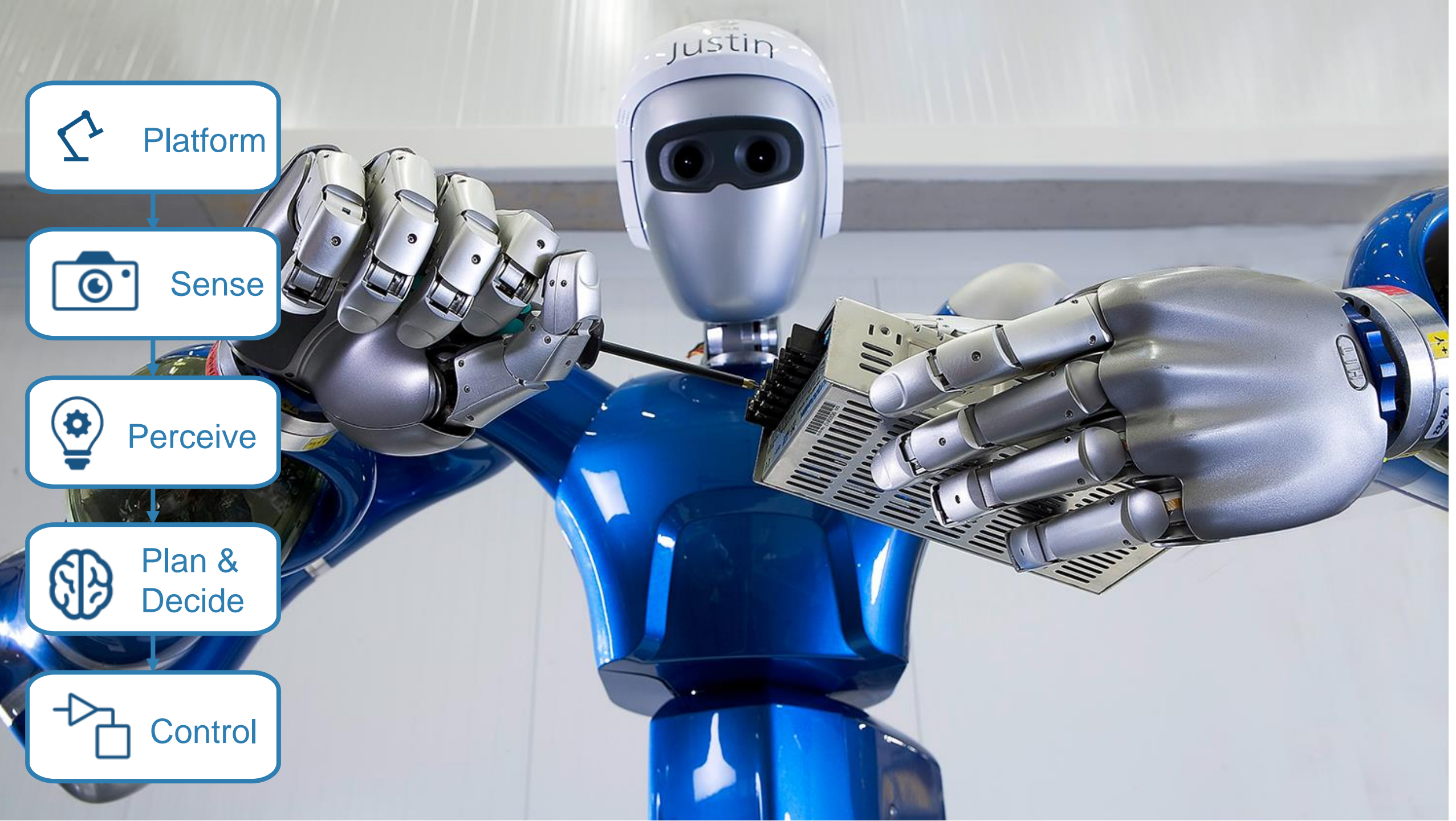
# Challenges with Autonomous Robotics Systems

Applying Multidomain Expertise

Complexity of Algorithms

End-to-End workflows





Platform



Sense



Perceive

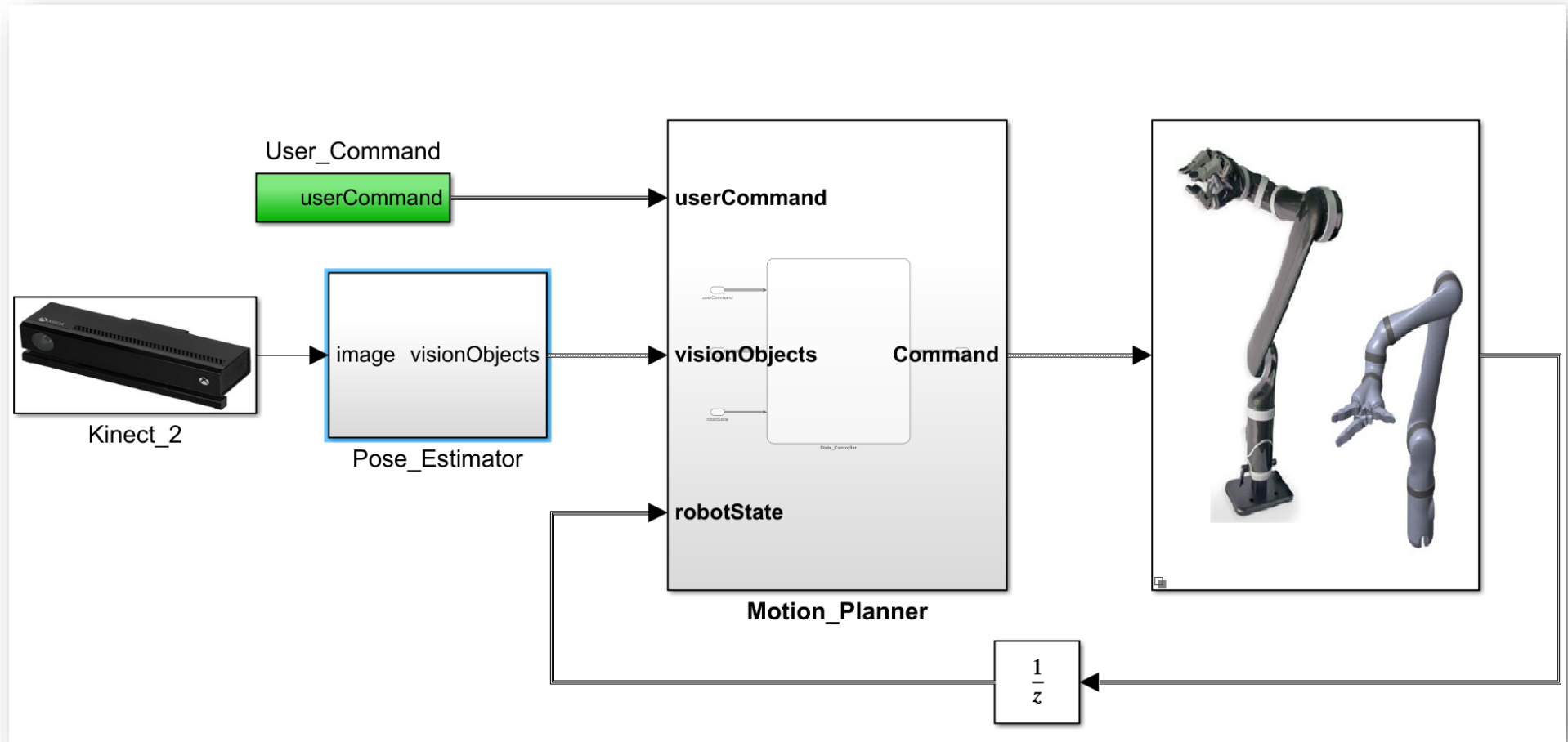
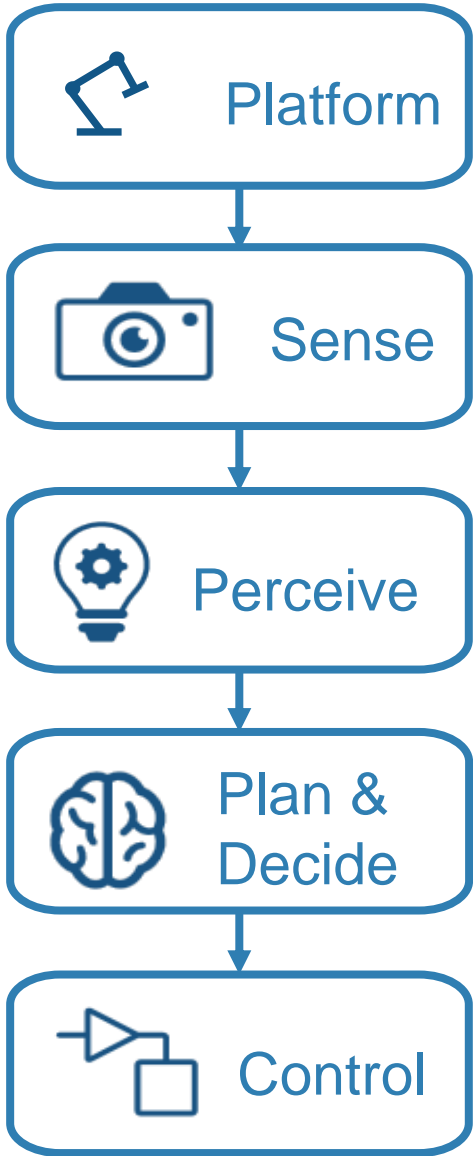


Plan &  
Decide

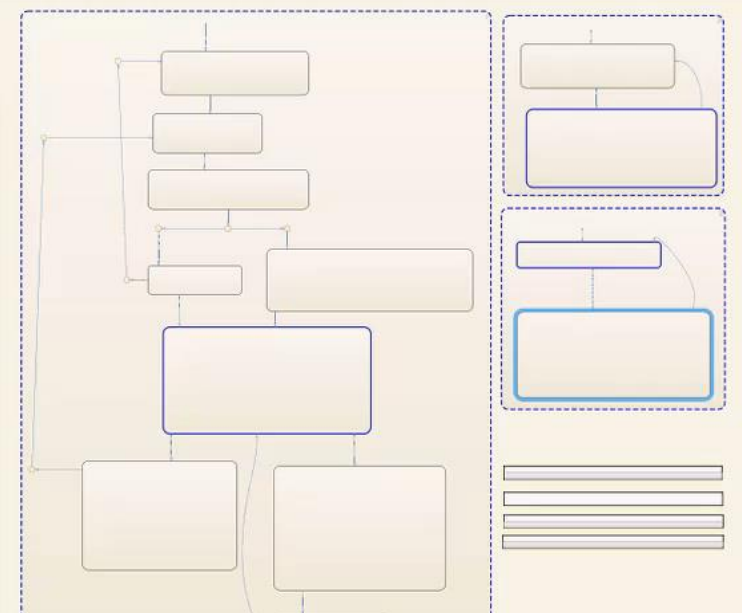
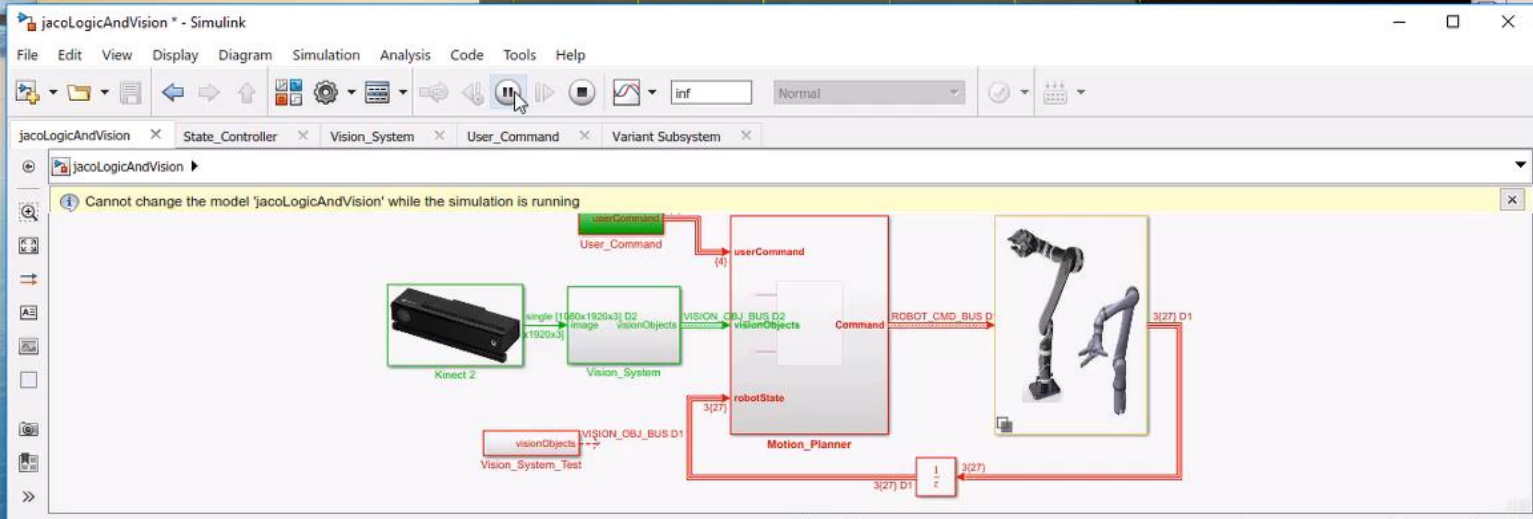
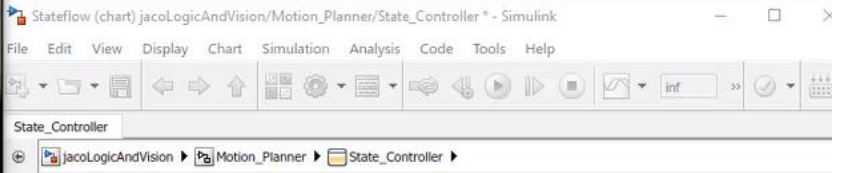
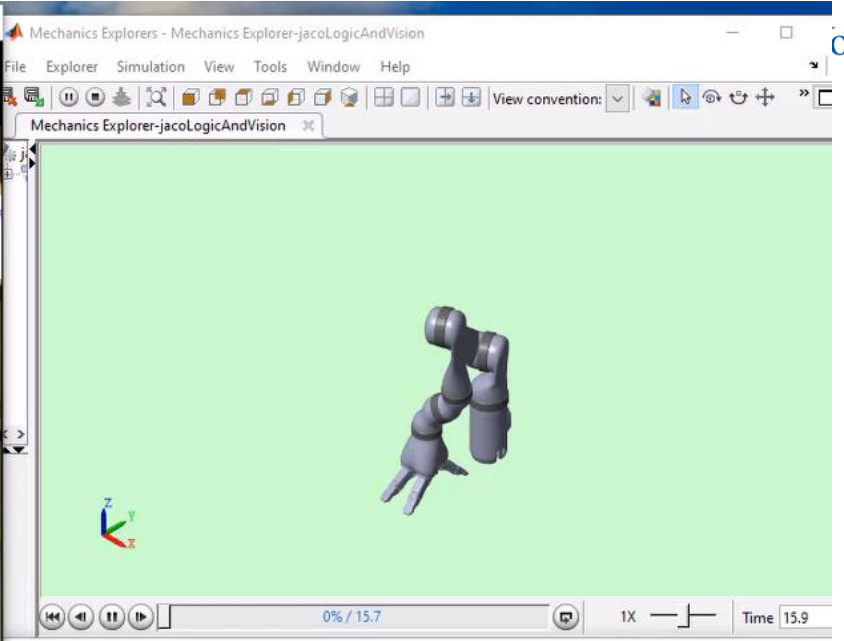
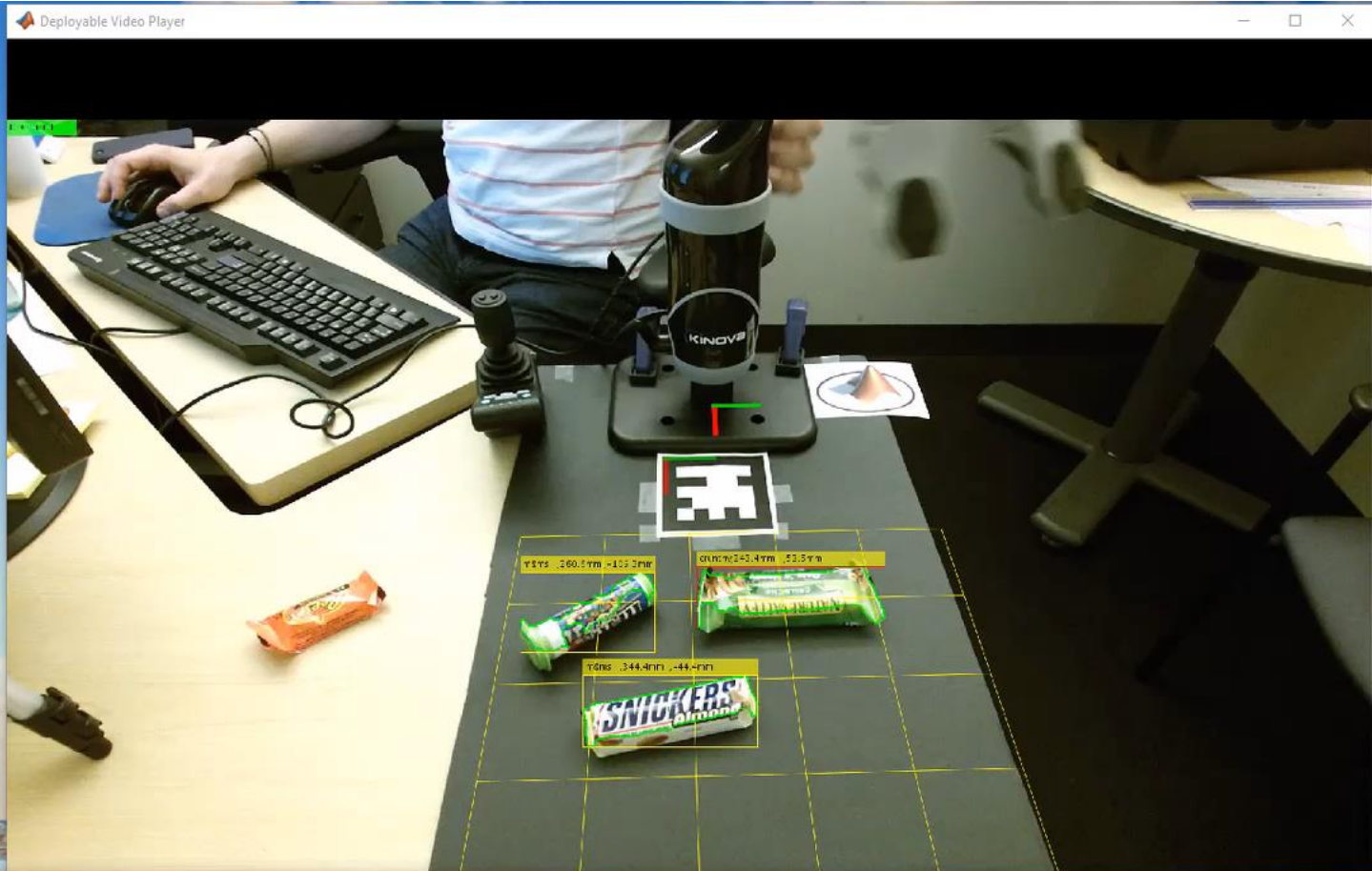


Control

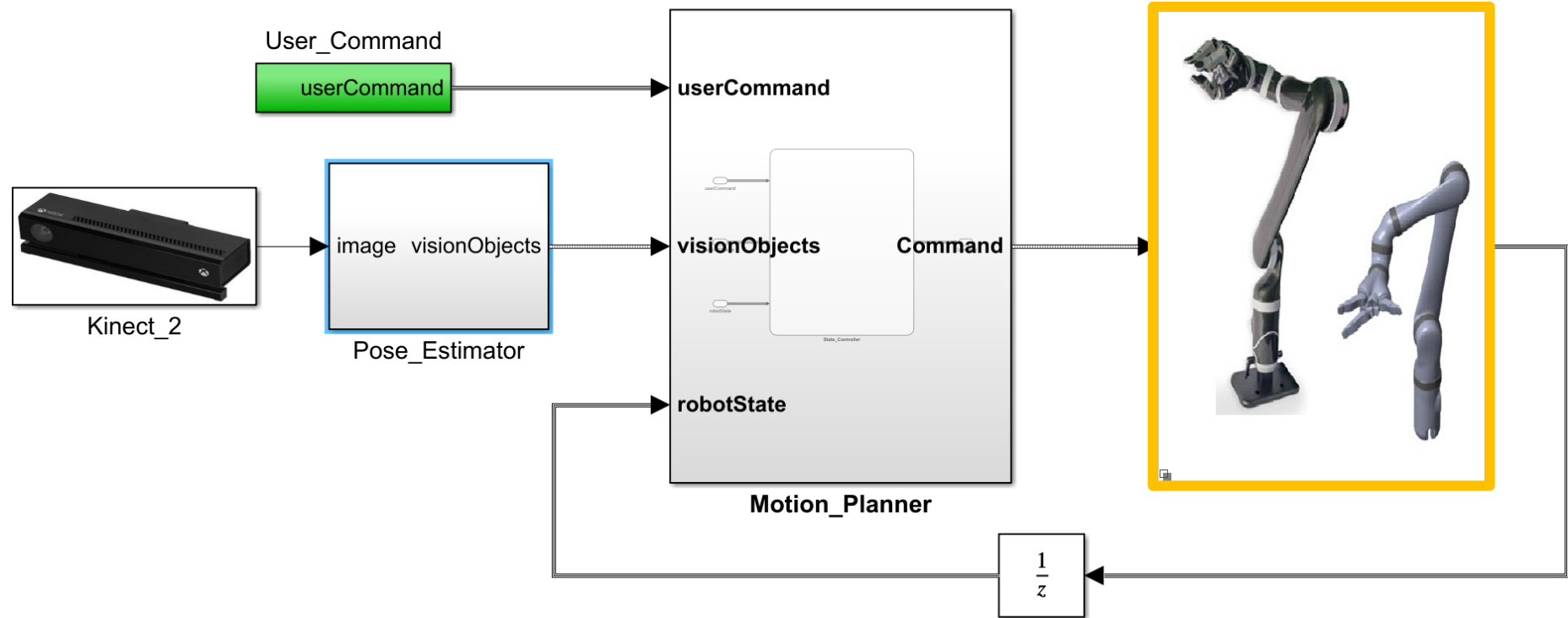
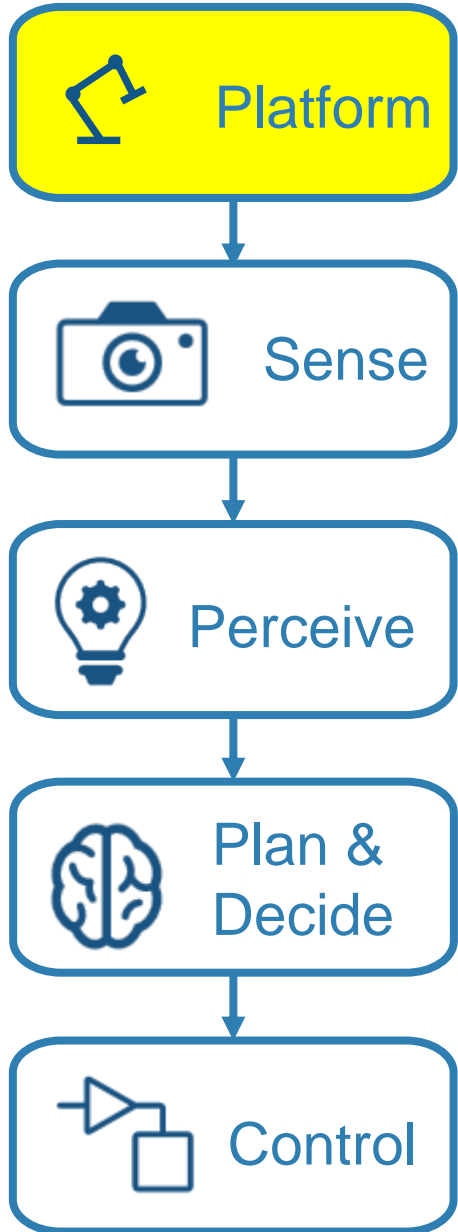
# Today: Design Pick and Place Application







# Today: Design Pick and Place Application





# Platform Design

*How to create a model of my system that suits my needs?*

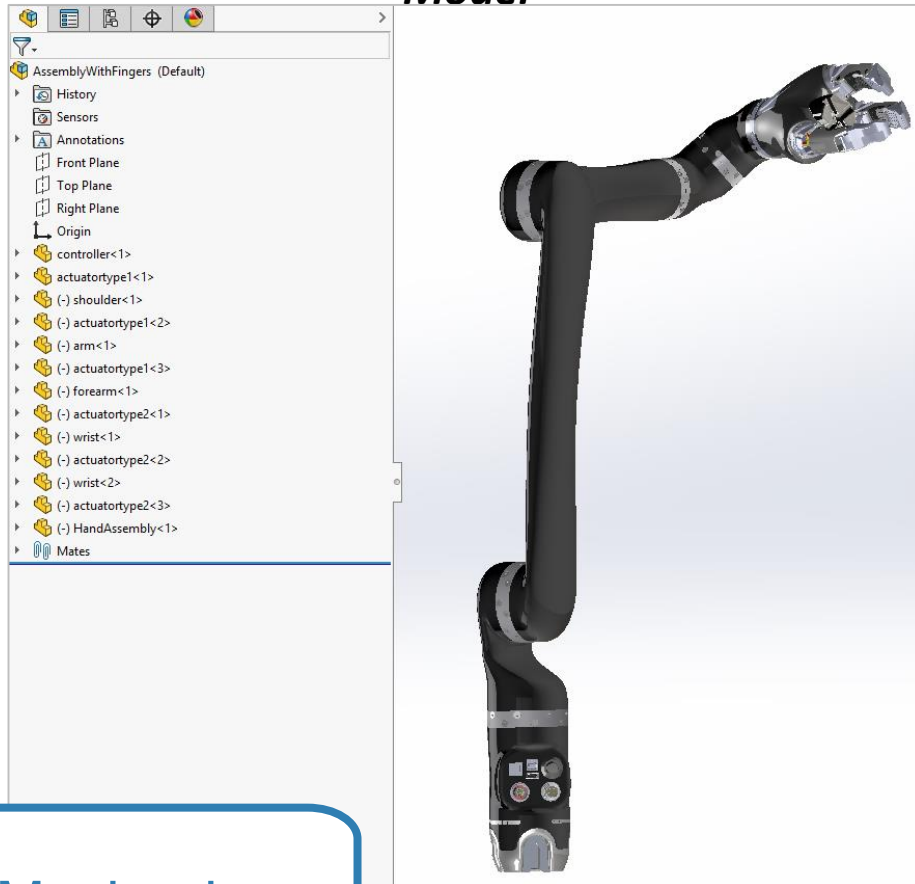
Mechanics

Actuators

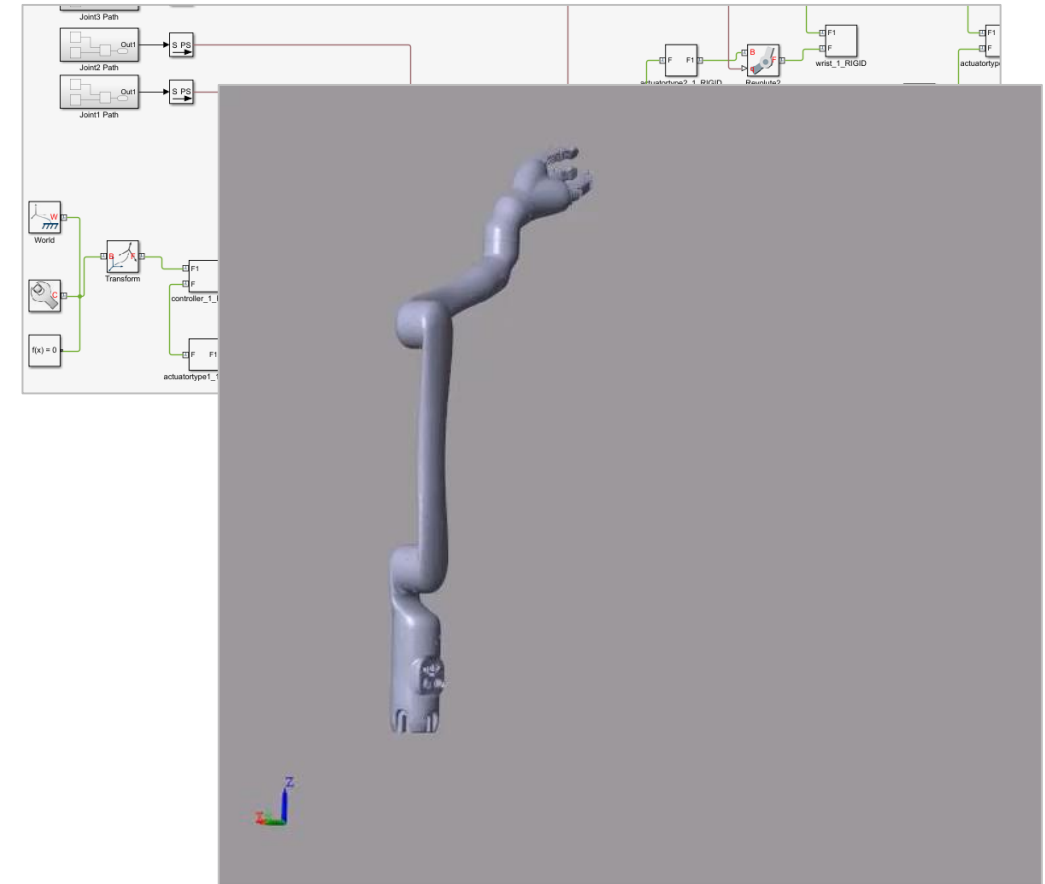
Environment

# Import models from common CAD Tools

*SolidWorks Model*



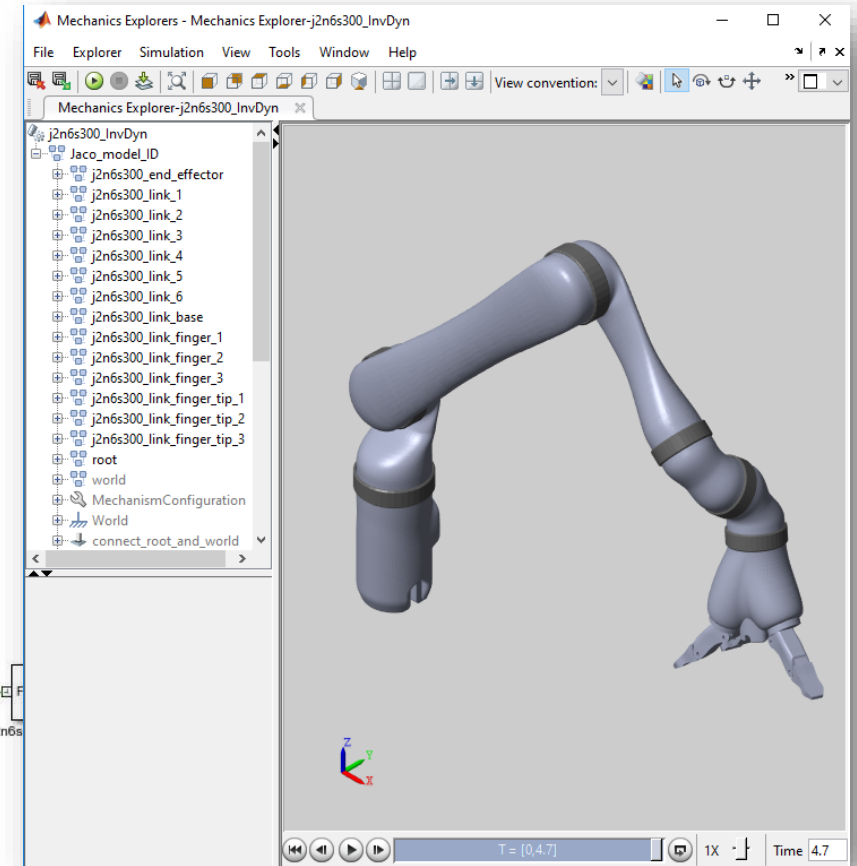
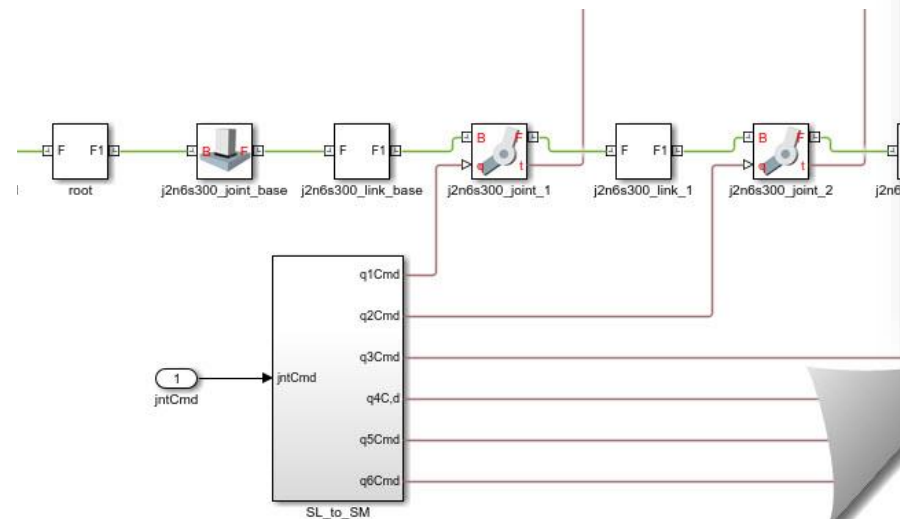
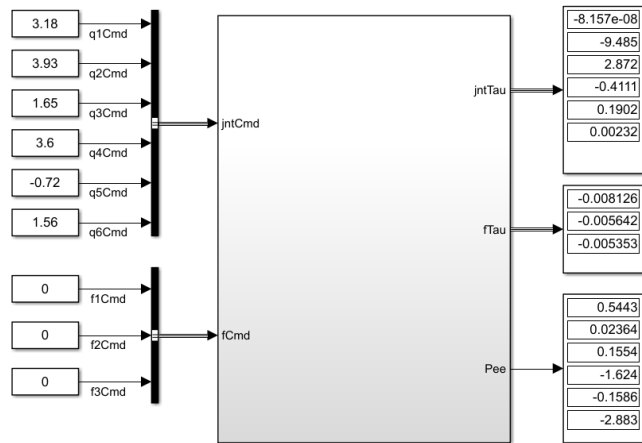
*Simscape Multibody Model*



Mechanics

# Mechanics: One line import from URDF

```
%% Import robot from URDF
smimport('j2n6s300_standalone_stl.urdf');
```





MATLAB R2019a

HOME | PLOTS | APPS

New Script | New Live Script | New | Open | Compare | Import Data | Save Workspace | New Variable | Open Variable | Favorites | Run and Time | Simulink | Layout | Preferences | Set Path | Add-Ons | Help | Community | Request Support | Learn MATLAB

FILE | VARIABLE | CODE | SIMULINK | ENVIRONMENT | RESOURCES

C:\D drive\ Matlab and Simulink files \ robotarm2 \ Source \ Model \ kinova\_description \ urdf

Current Folder:
 

- two\_arm\_robot\_example\_standalone.xacro
- m1n6s300.xacro
- m1n6s300\_standalone.xacro
- m1n6s200.xacro
- m1n6s200\_standalone.xacro
- m1n4s200.xacro
- m1n4s200\_standalone.xacro
- kinova\_inertial.xacro
- kinova\_finger\_set.xacro
- kinova\_common.xacro
- kinova\_gazebo
- j2s7s300.xacro
- j2s7s300\_standalone.xacro
- j2s6s300.xacro
- j2s6s300\_standalone\_stl.urdf
- j2s6s300\_standalone.xacro
- j2s6s300\_standalone.urdf
- j2n7s300.xacro
- j2n7s300\_standalone.xacro
- j2n6s300.xacro
- j2n6s300\_standalone\_stl.urdf
- j2n6s300\_standalone.xacro
- j2n6s300\_standalone.urdf
- j2n6s200.xacro
- j2n6s200\_standalone.xacro
- j2n6s200\_standalone.urdf**
- j2n4s300.xacro
- j2n4s300\_standalone.xacro
- slprj

Command Window:
 

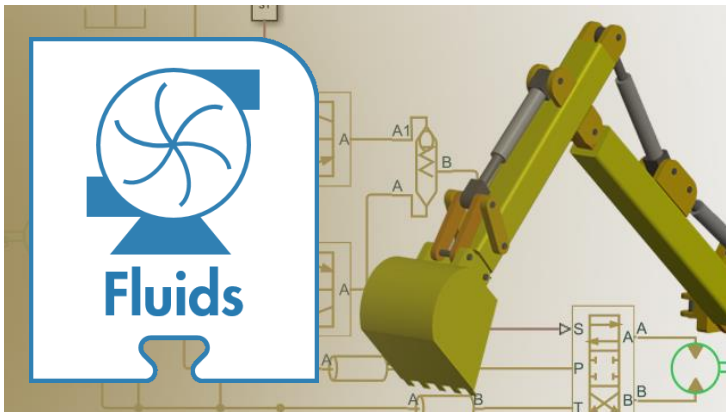
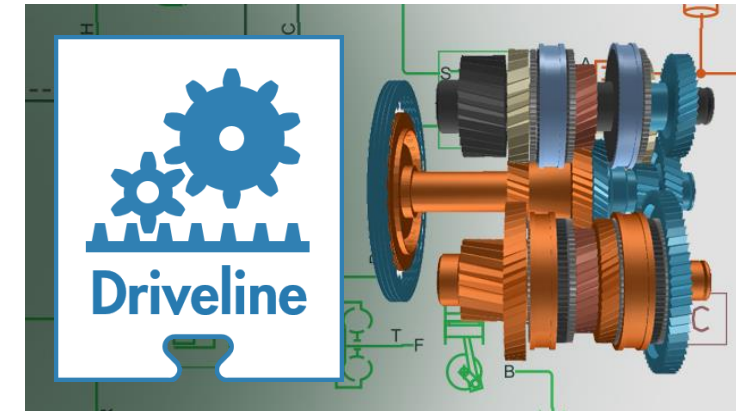
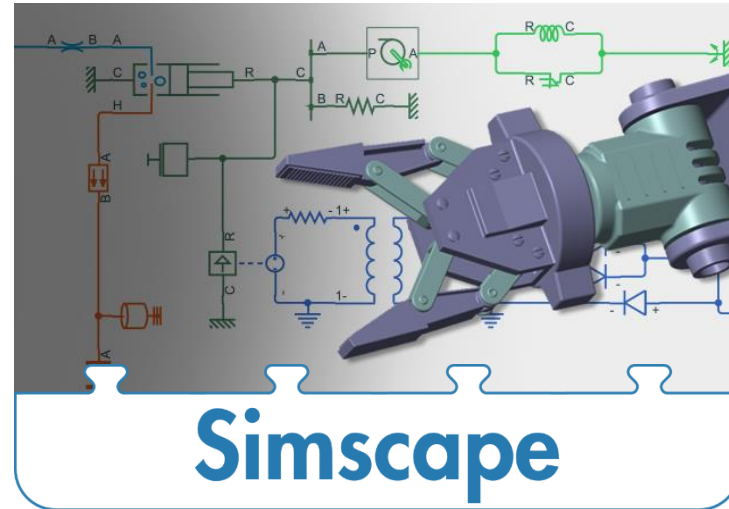
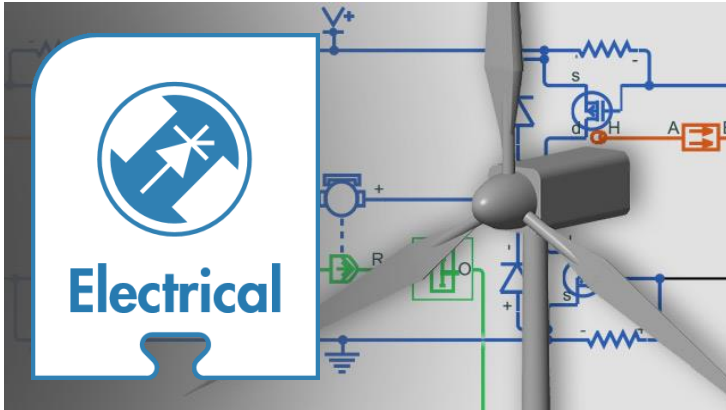
```
fx >> smimport('j2n6s300_standalone_stl.urdf')
```

Workspace:
 

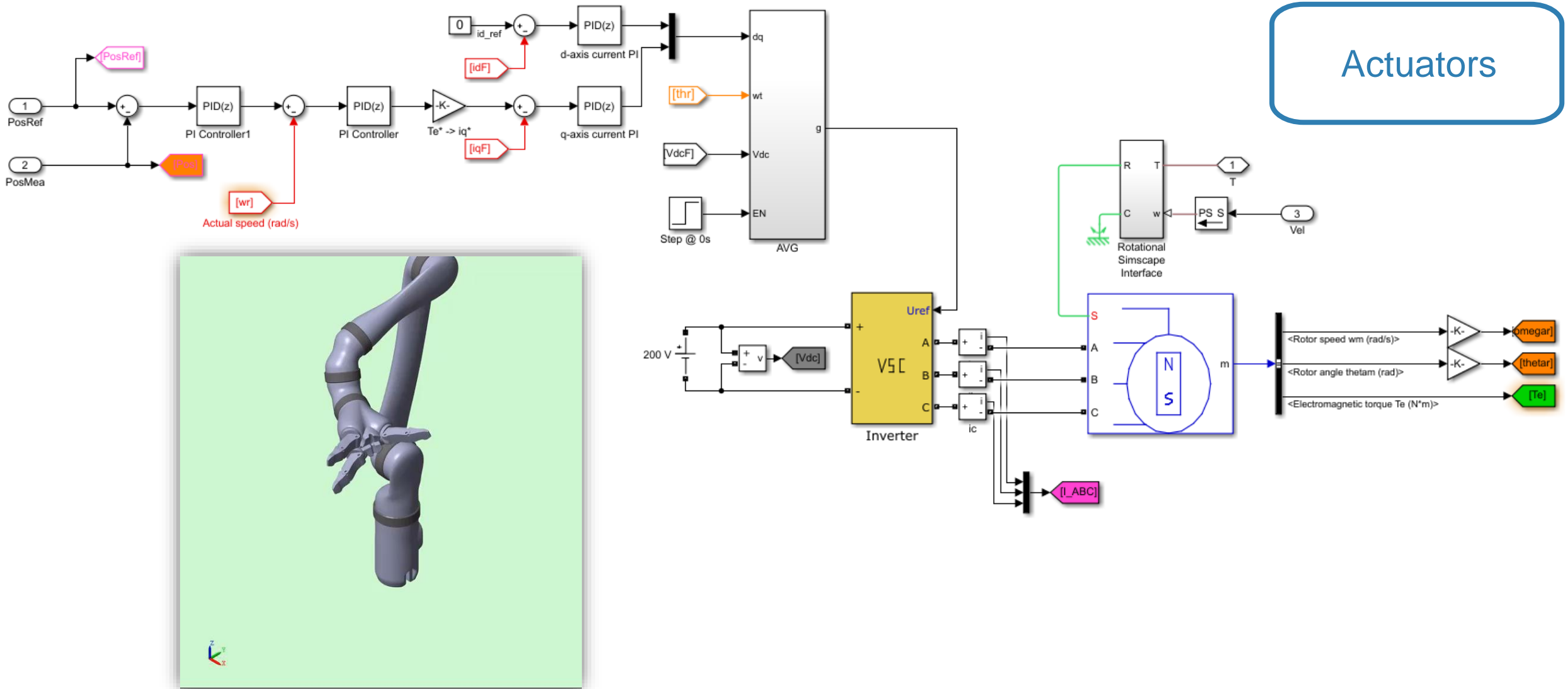
Name	Value

j2n6s200\_standalone.urdf (URDF File)
   
 No details available

# Actuators: Model other domains

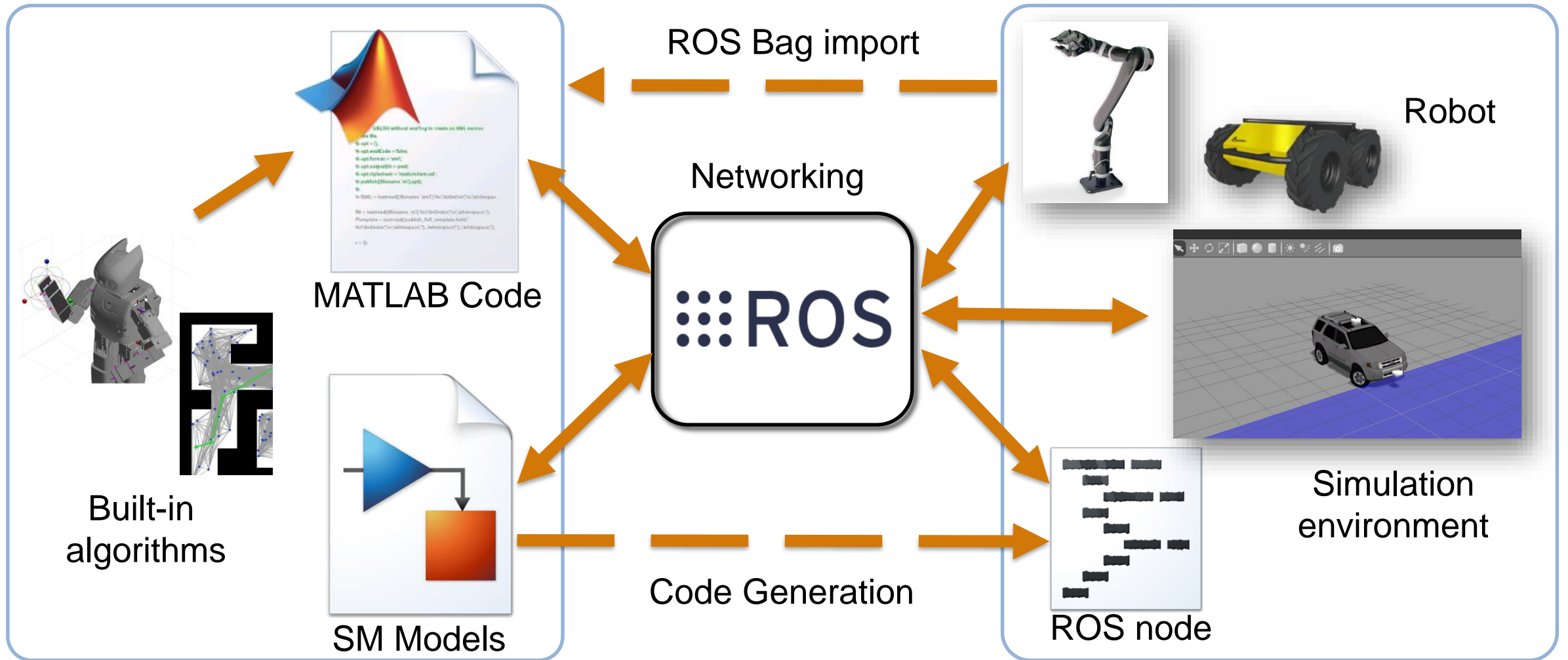


# Modeling Actuators

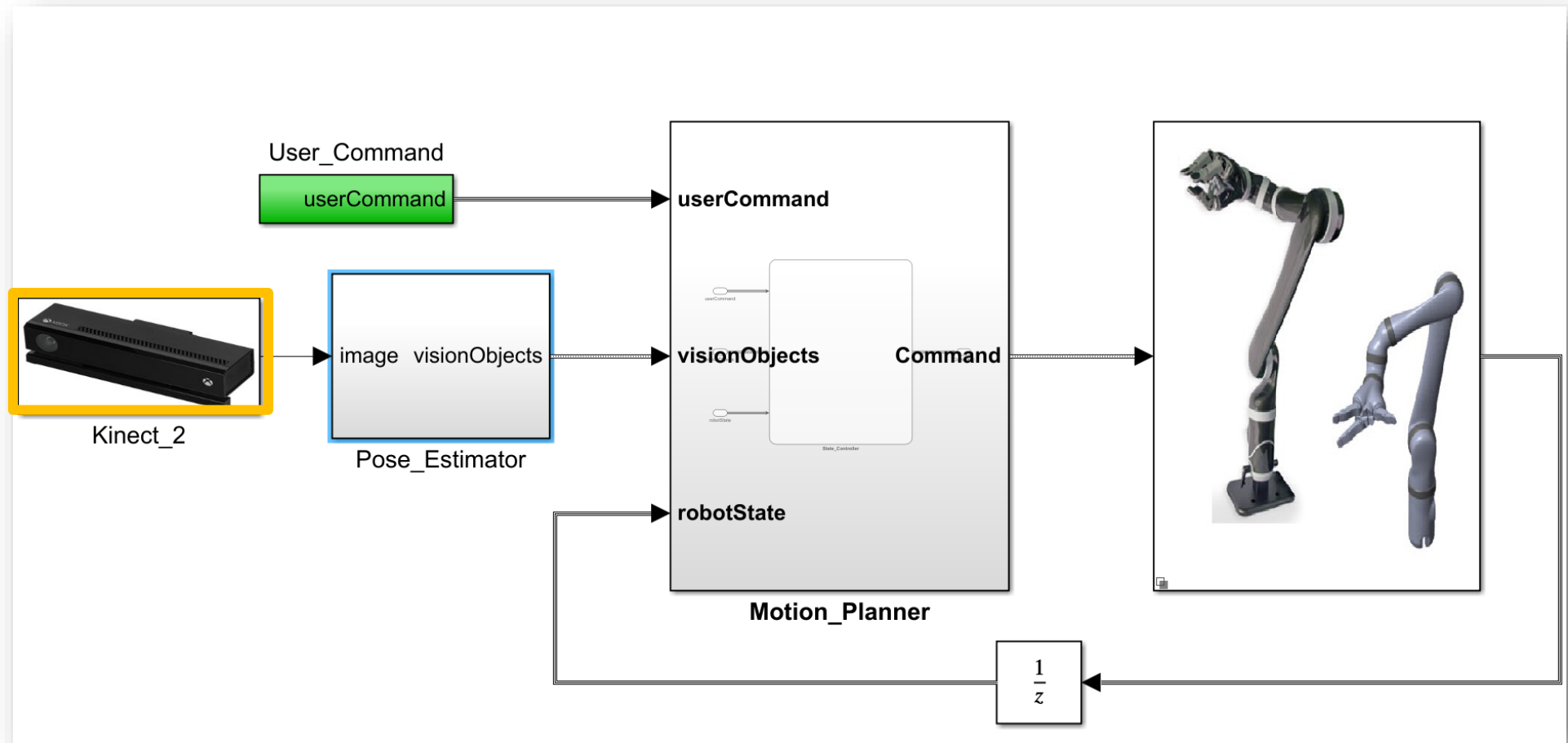
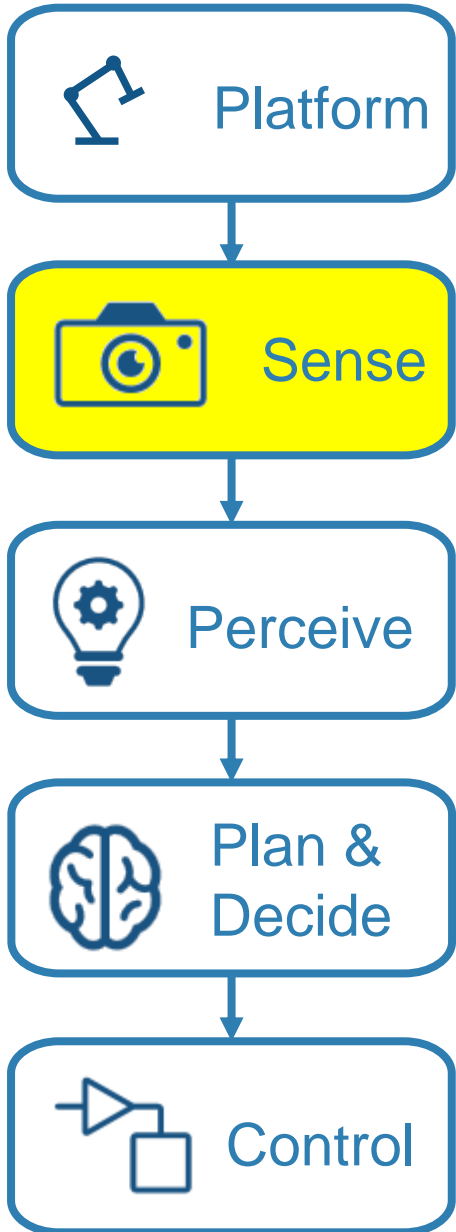




# Environment: Connect MATLAB and Simulink with ROS

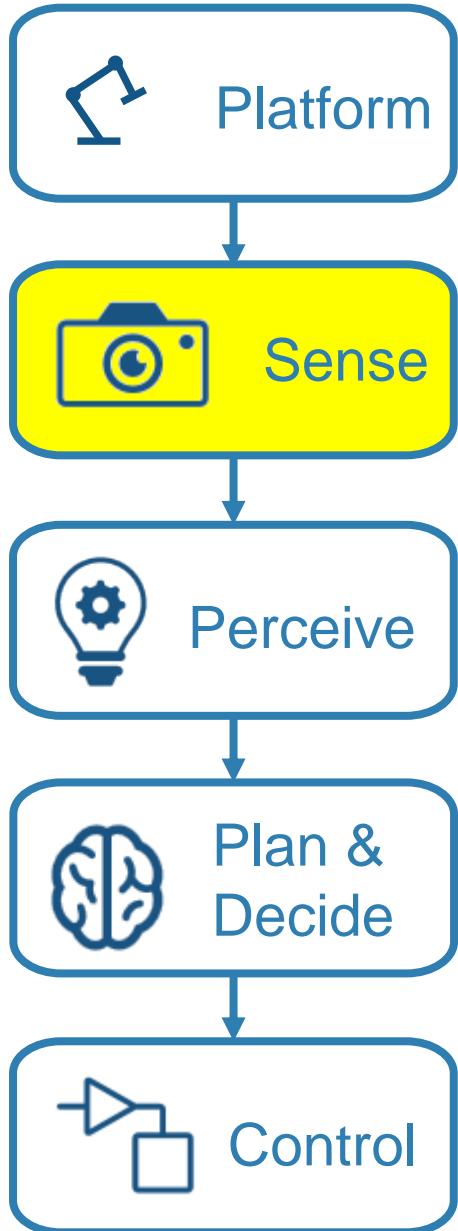


# Design Pick and Place Application

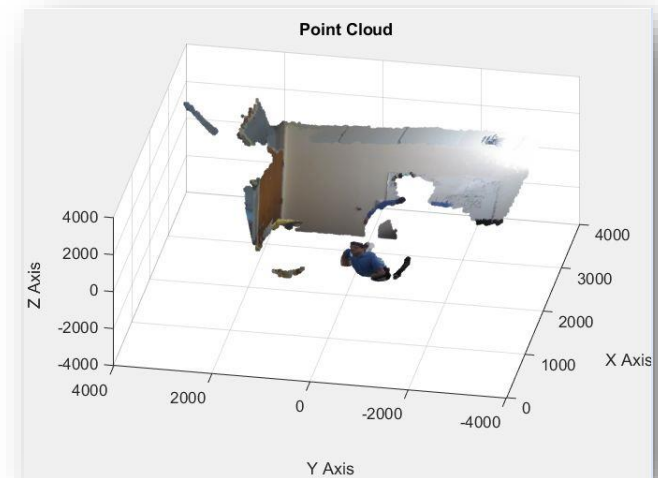


Demo

# Design Pick and Place Application

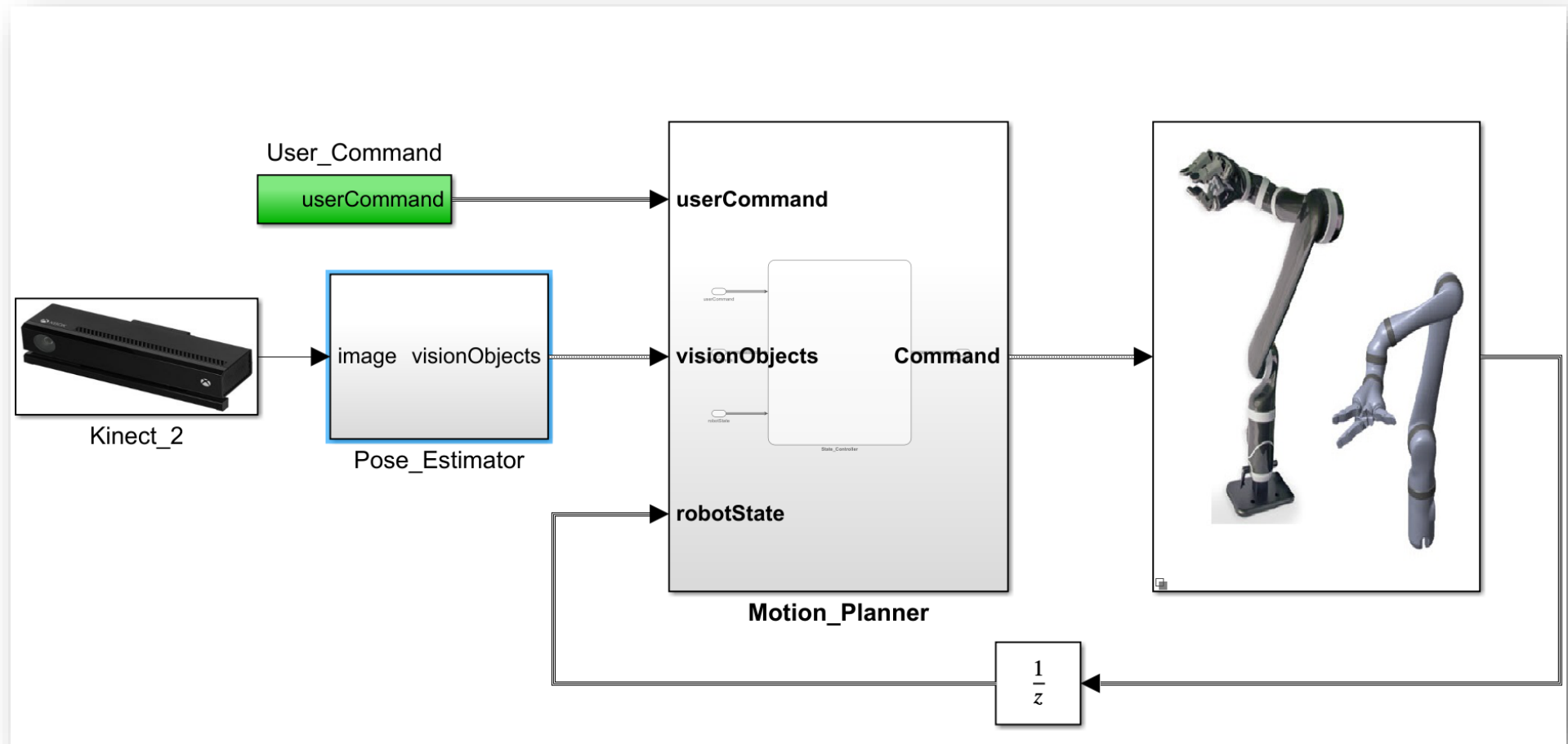
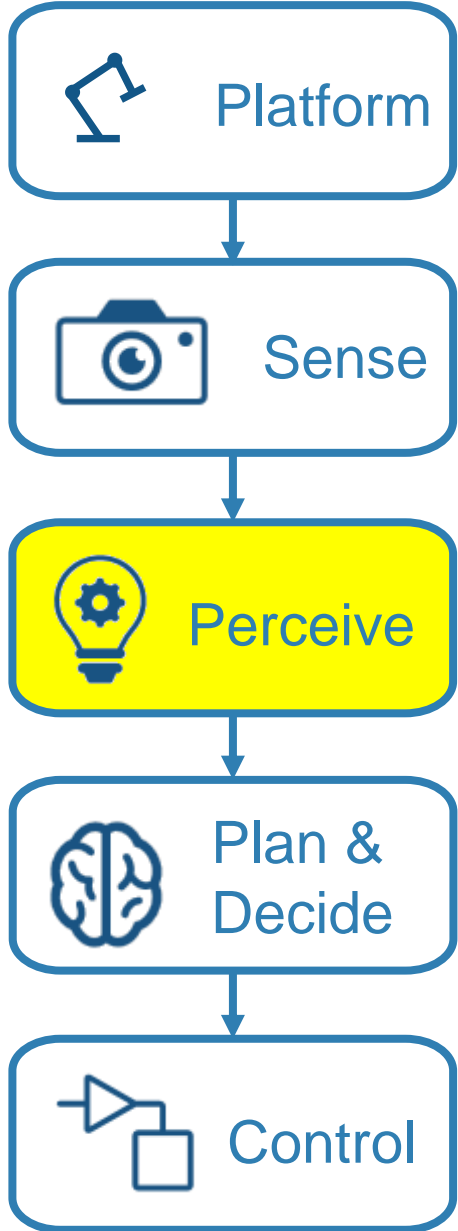


- Support for Common Sensors
- Image analysis
- Apps
- Image enhancement
- Visualizing Point Clouds



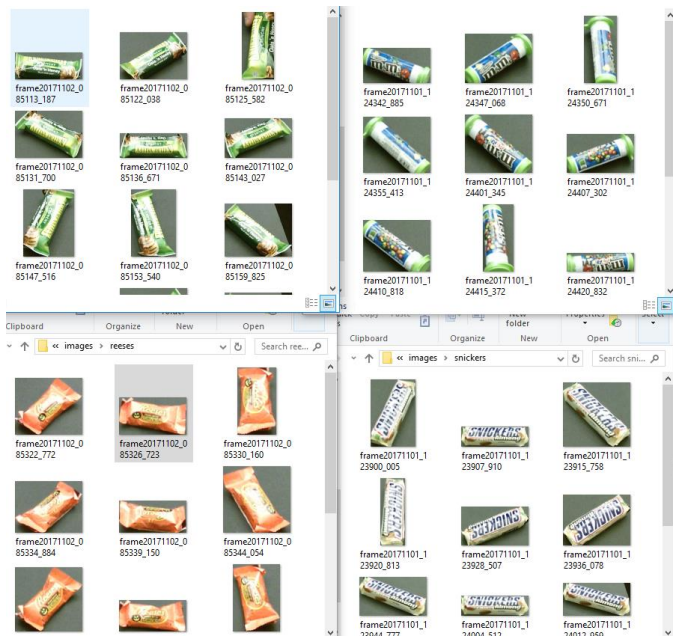


# Today: Design Pick and Place Application



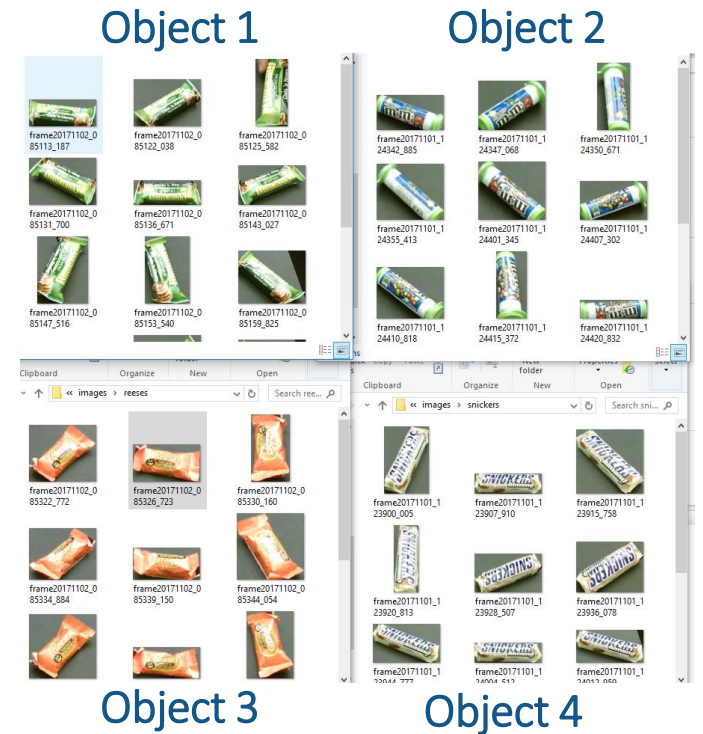
# Object Classifier and Pose Estimator

Images



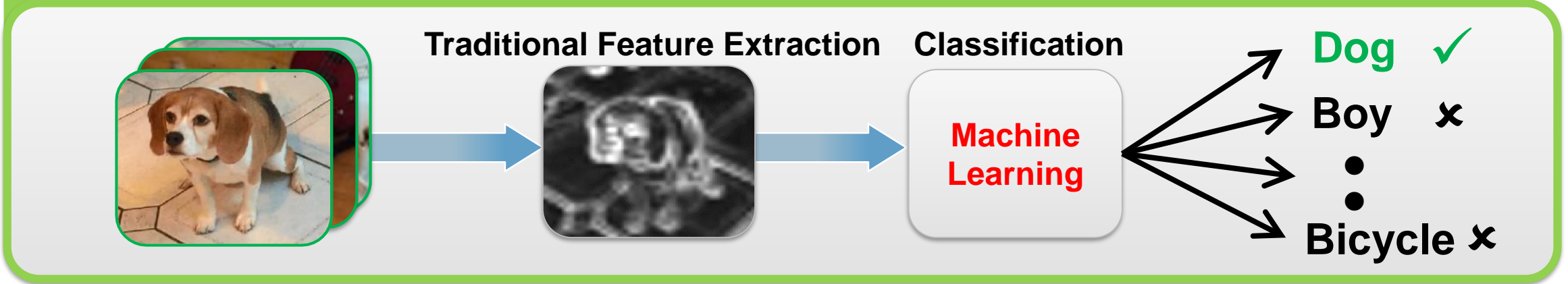
Pose Estimator

Labels and Poses

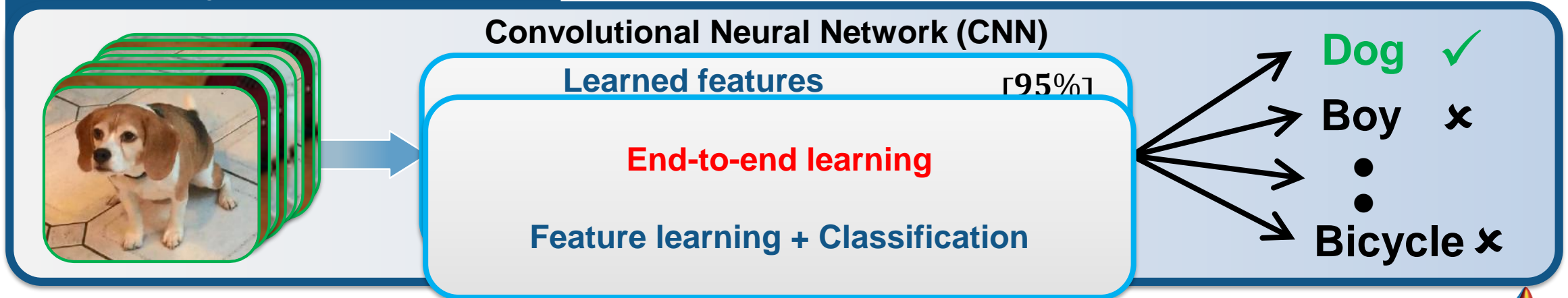


# MATLAB makes machine learning easy and accessible

## Traditional Machine Learning approach

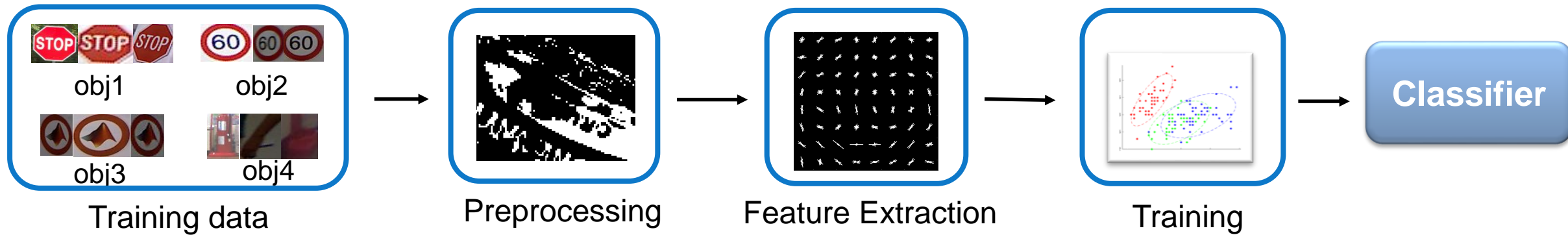


## Deep Learning approach





# Complex workflows made easy with MATLAB



```

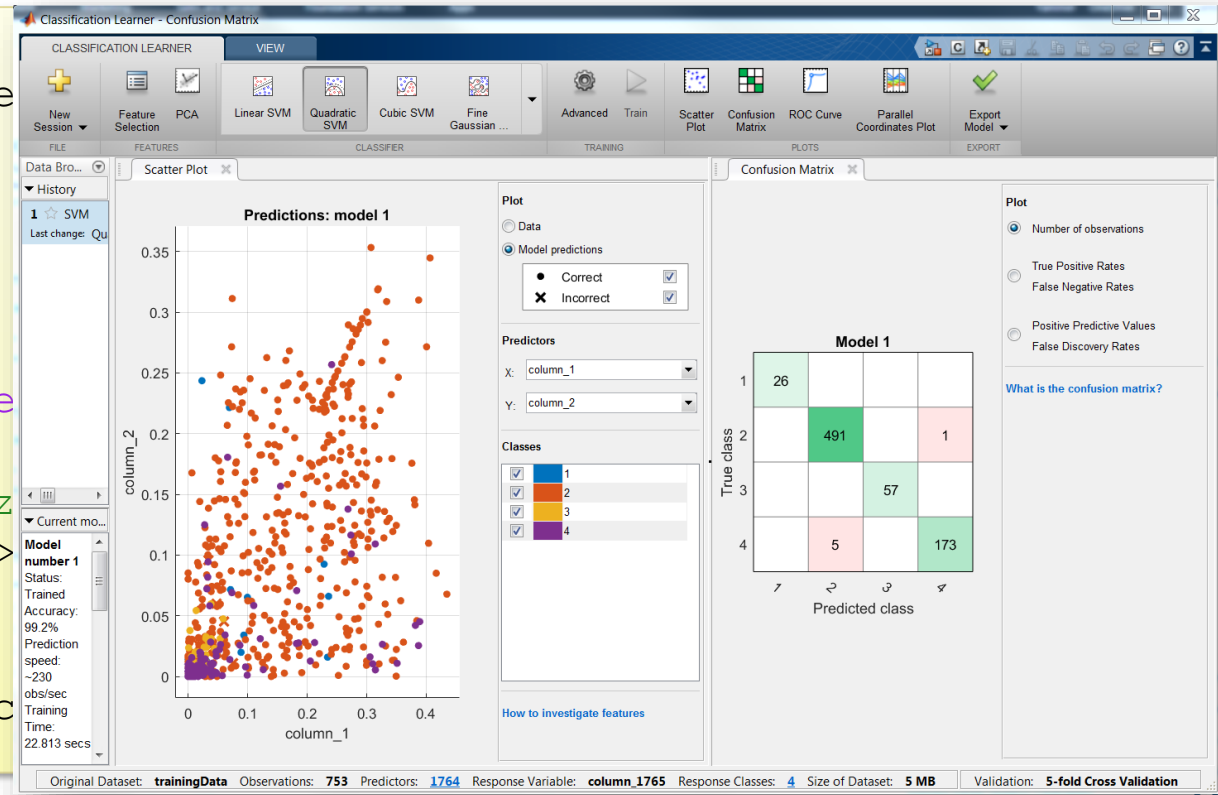
% Detect regions
BW = createMask(videoFrame

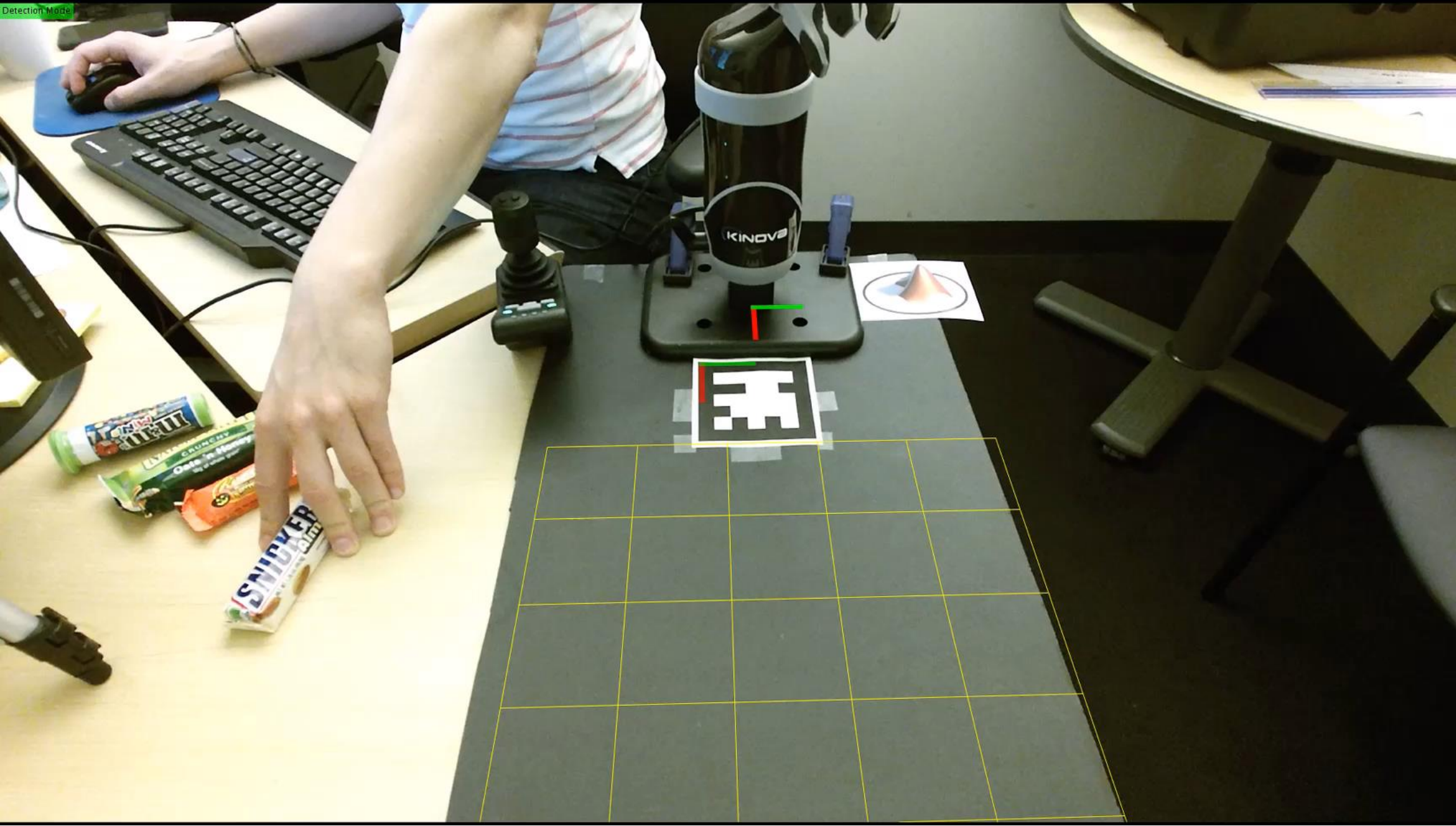
% Fill image regions
BW = imfill(BW, 'holes');

% Get bounding boxes
stats = regionprops('table

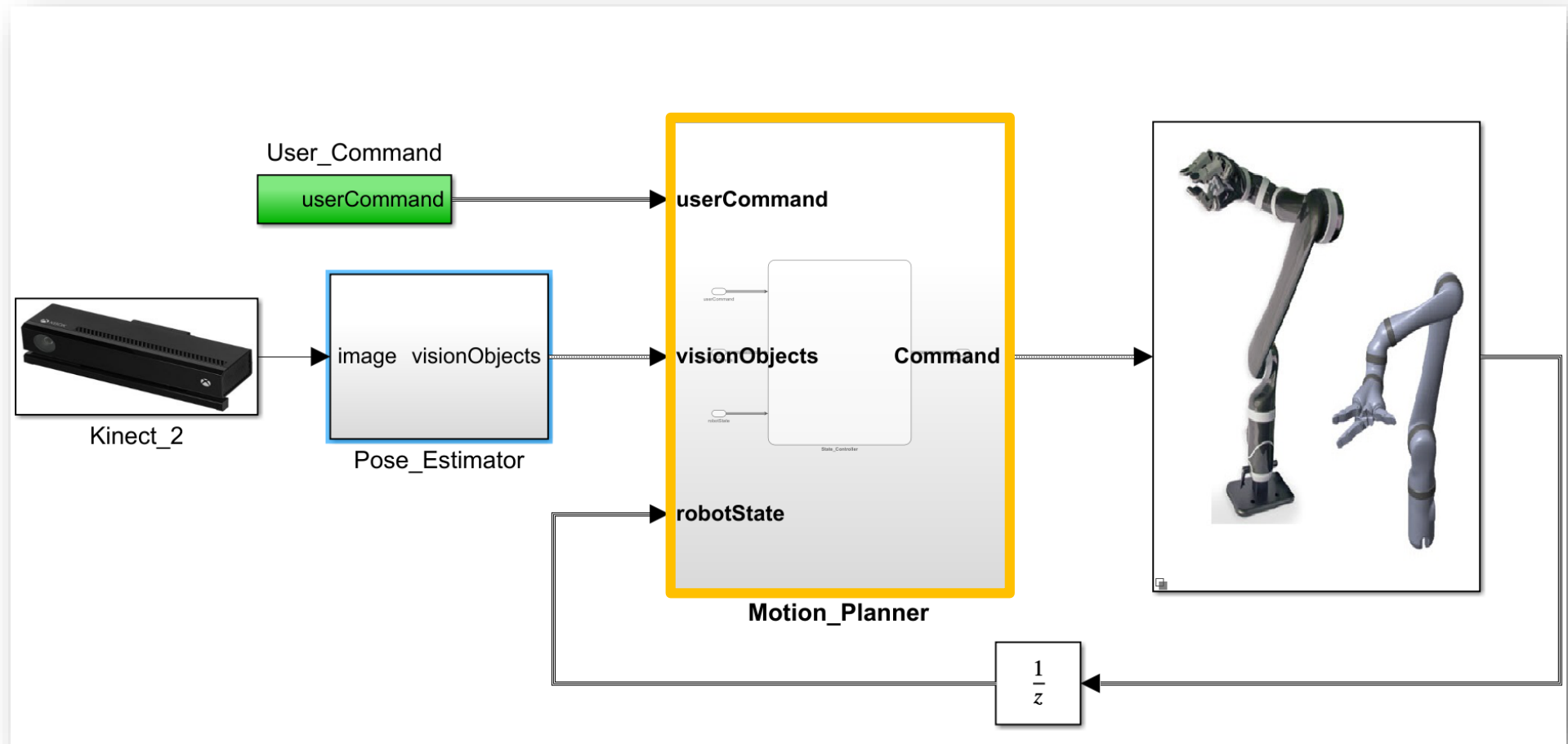
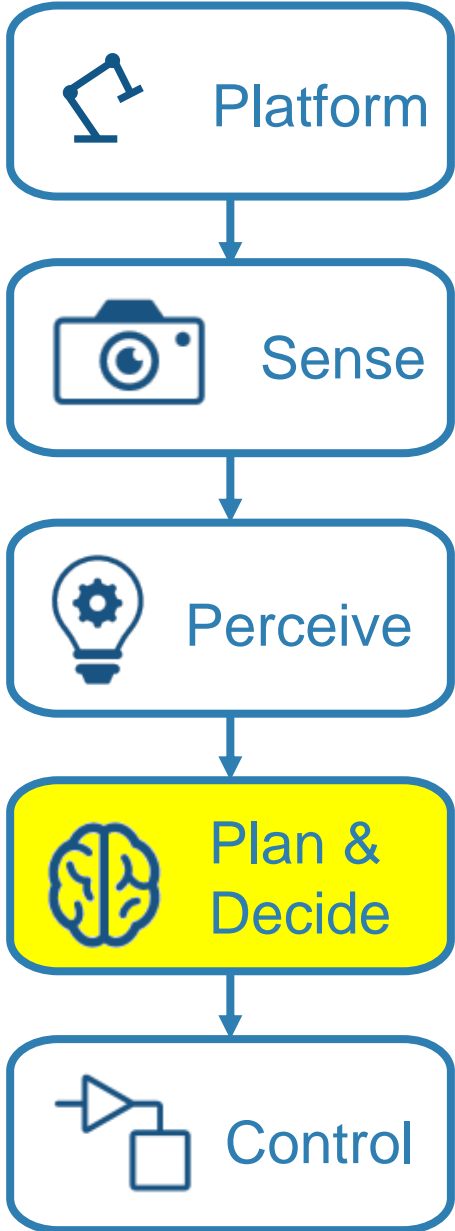
% Filter based on area siz
targetIndex = stats.Area >

% Get bounding boxes from
testFeatures(k,:) = extrac
    
```

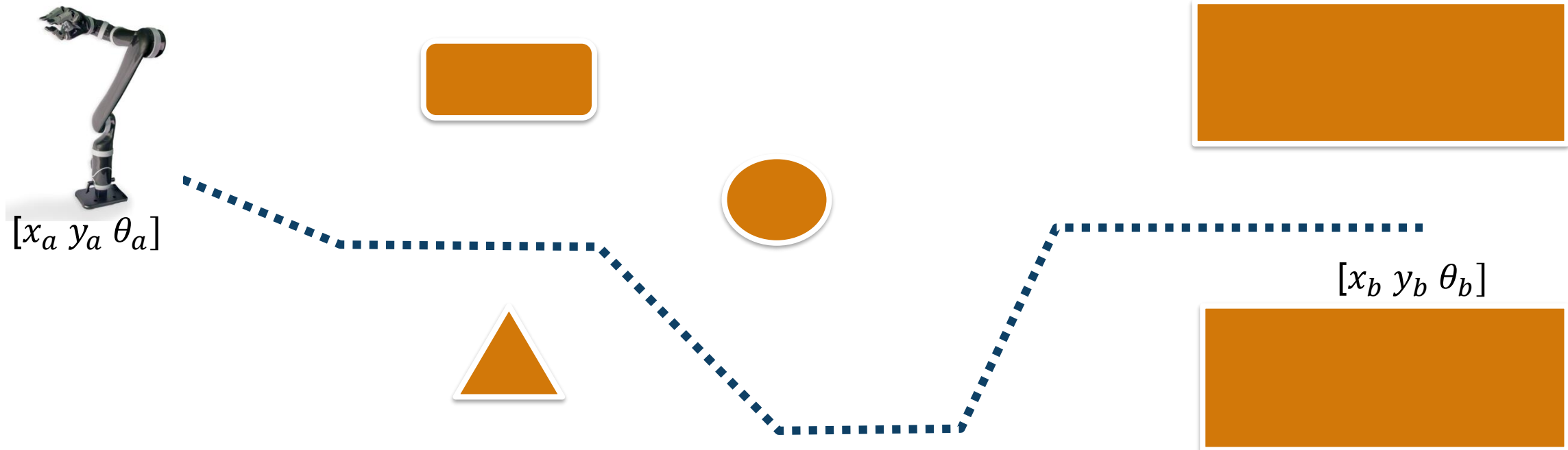
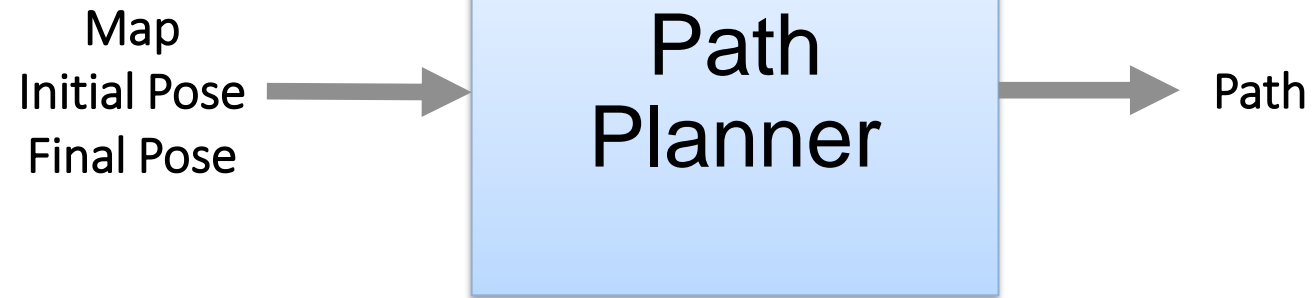




# Design Pick and Place Application

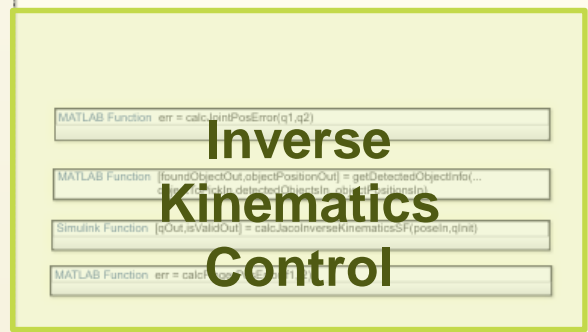
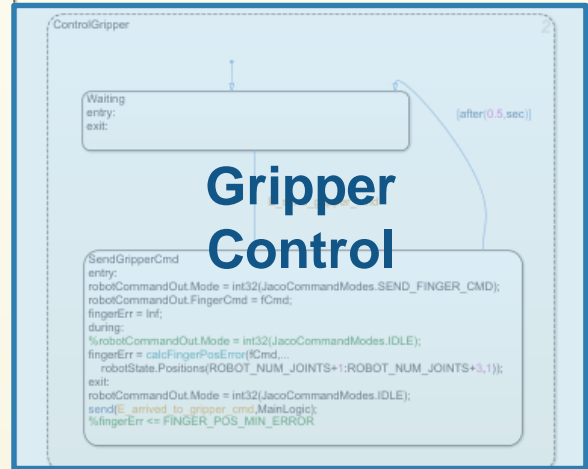
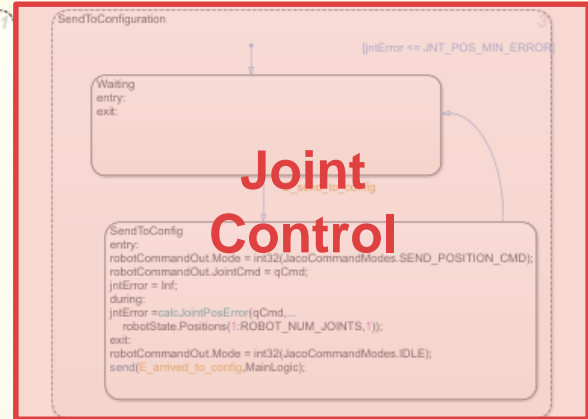
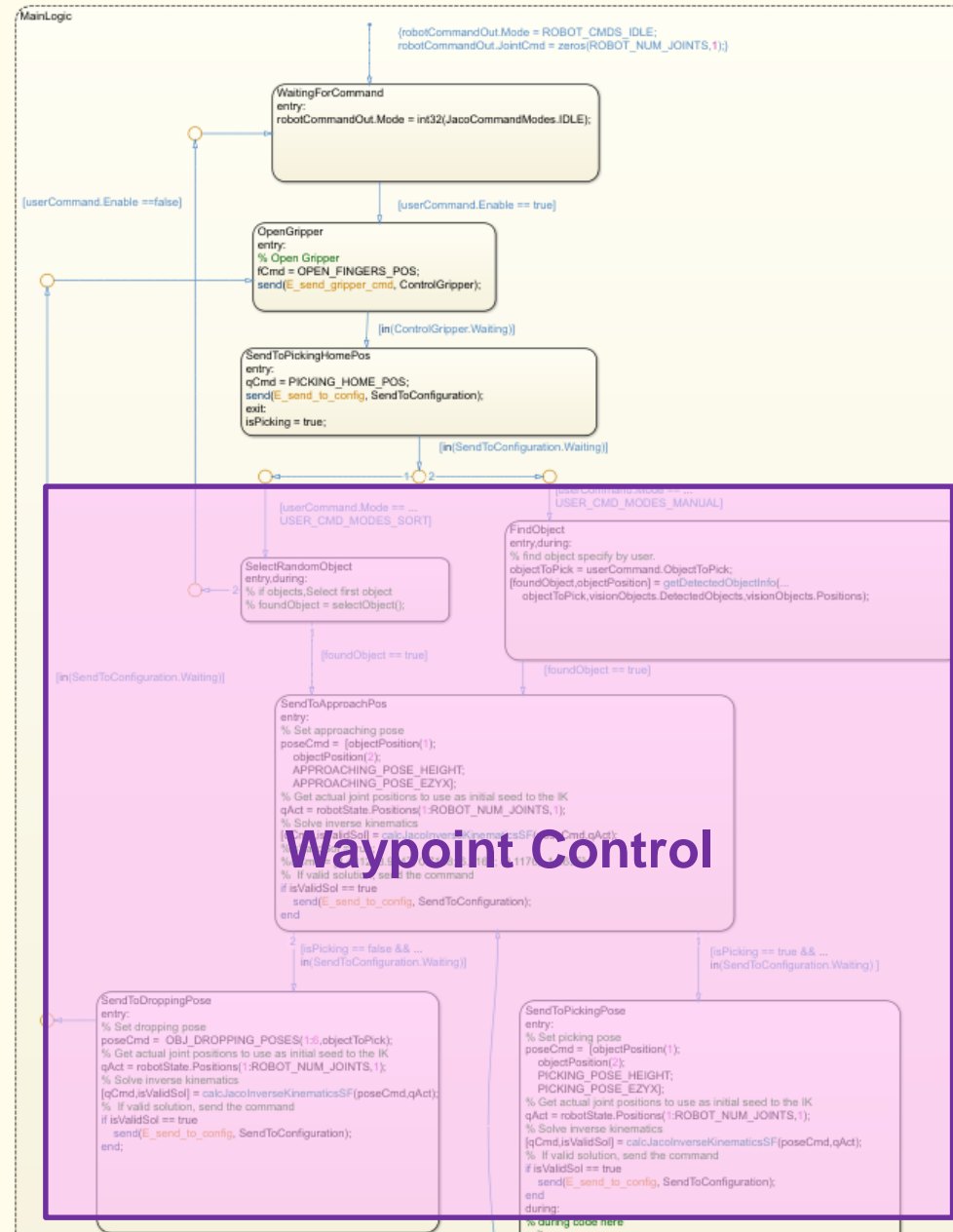


# Planning: Find a path

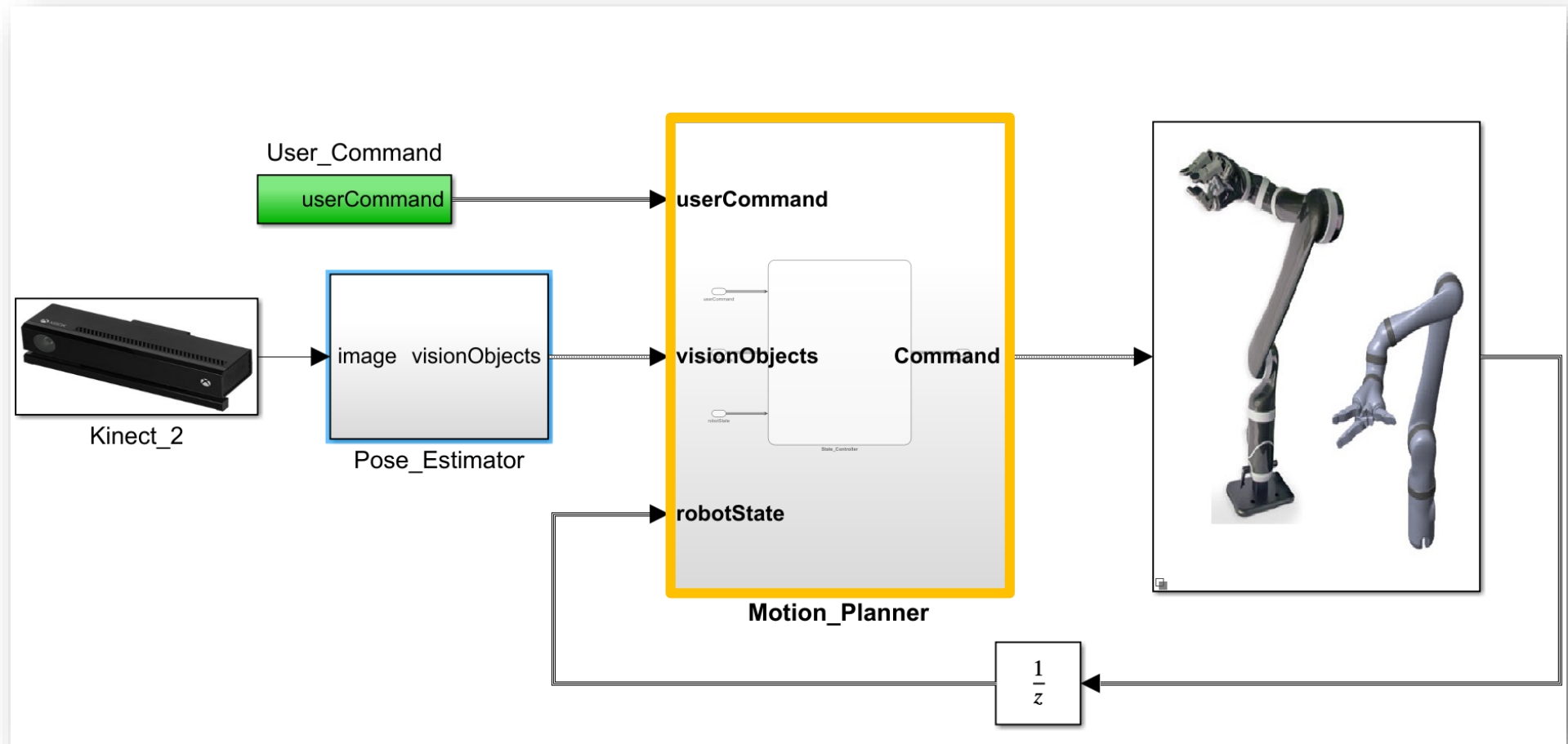
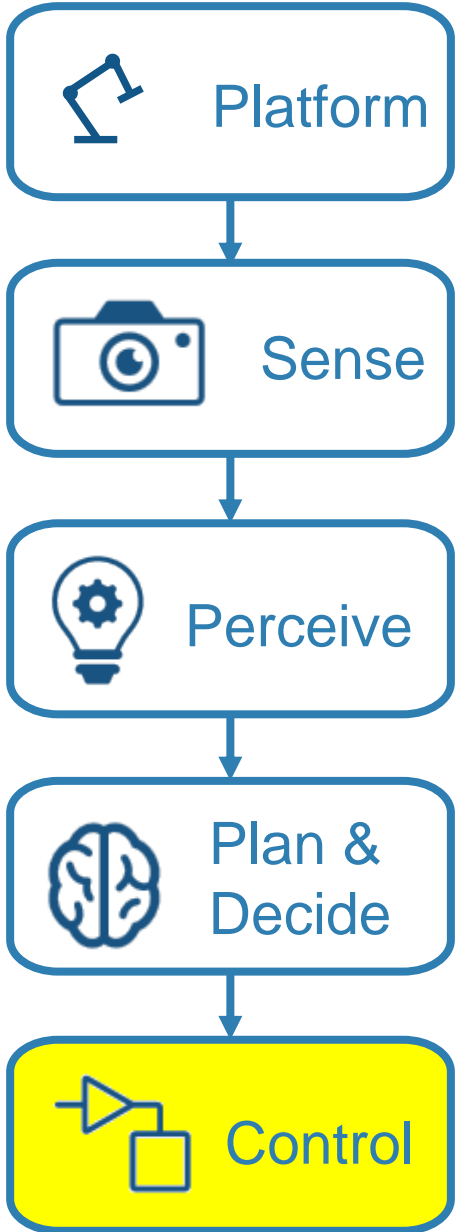




# Plan with Stateflow



# Design Pick and Place Application

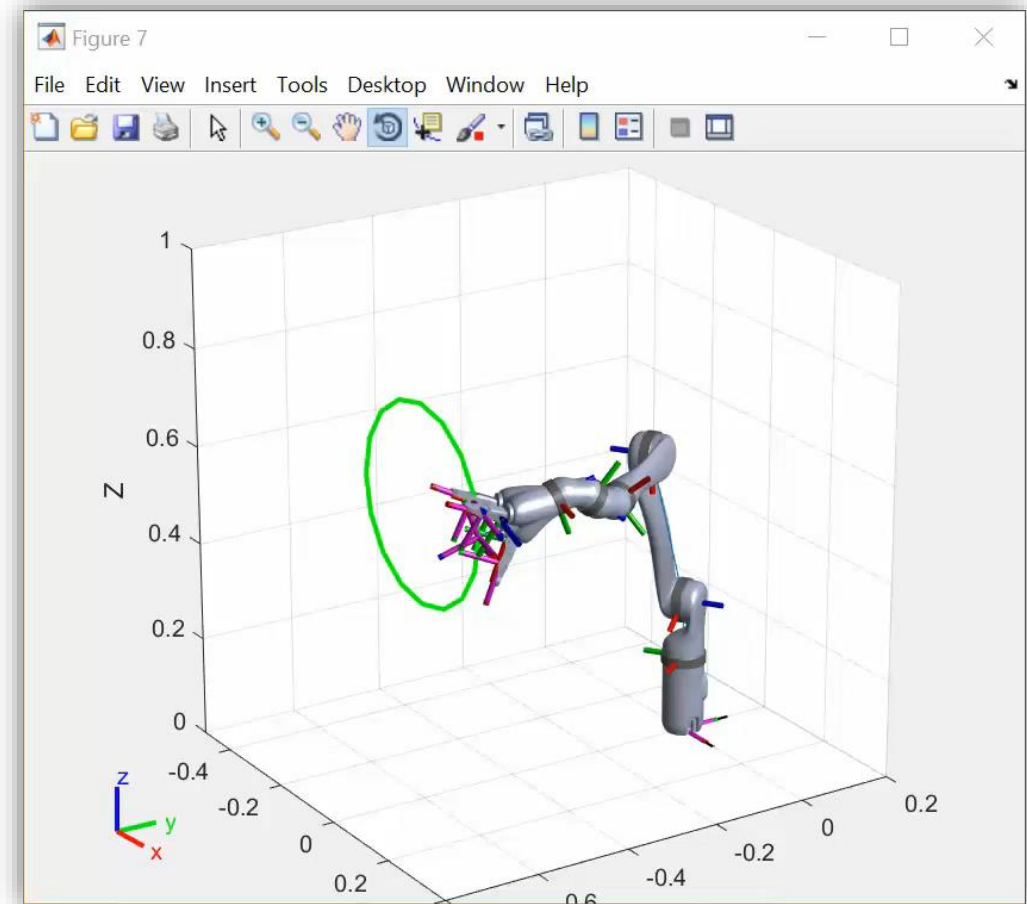


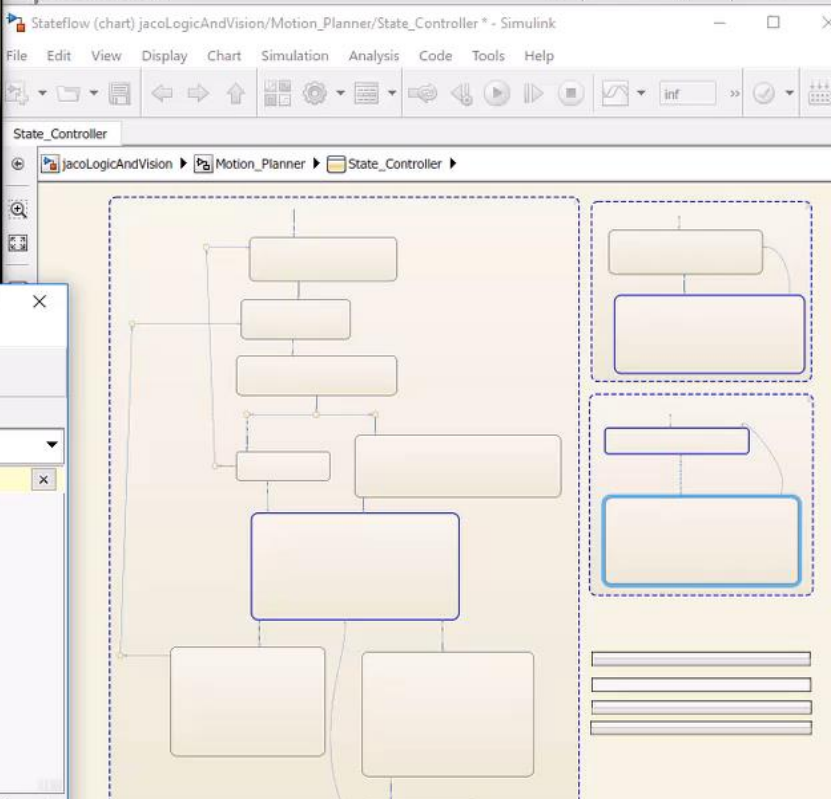
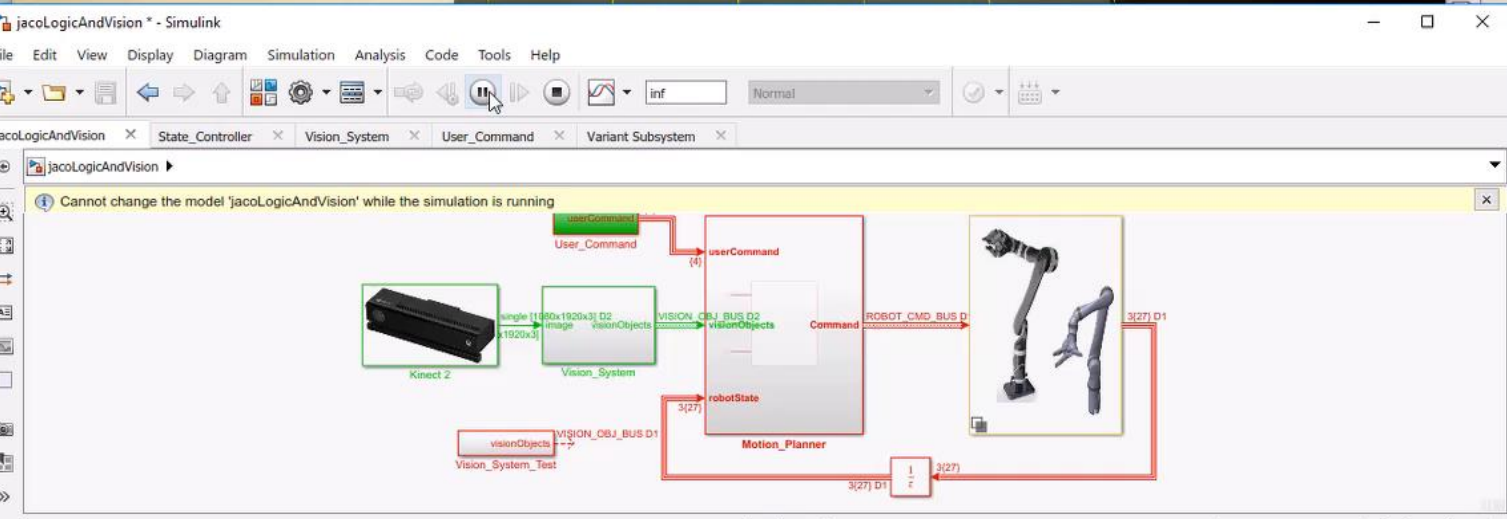
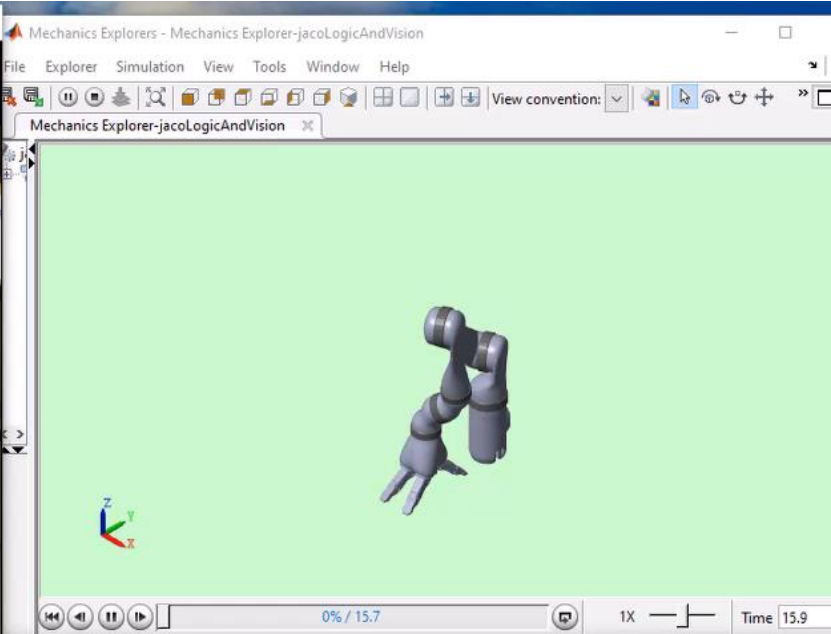
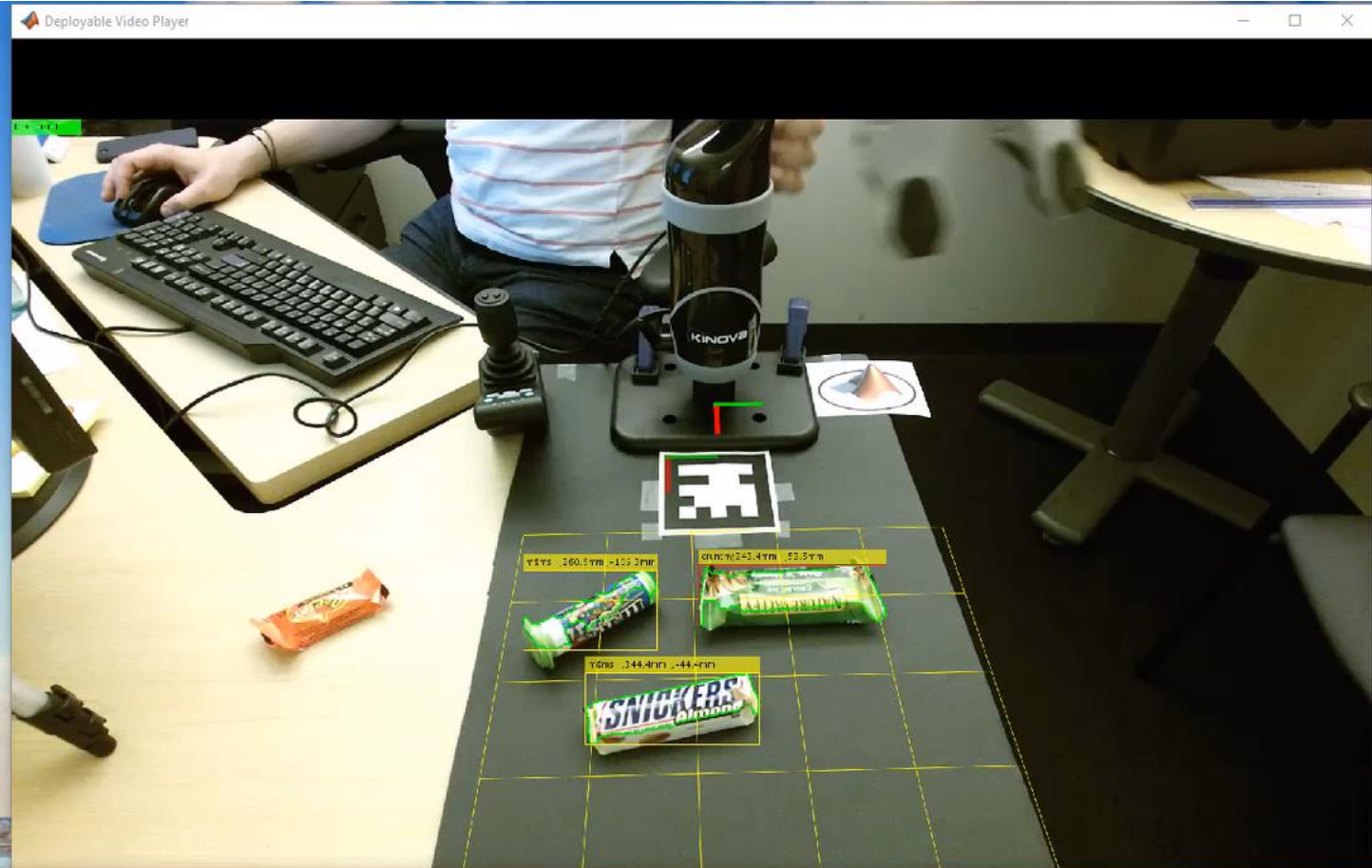
# Control: Explore Built In Functions: Inverse Kinematics

```
% Create ik solver object
ik=robotics.InverseKinematics('RigidBodyTree',
                             jaco)

% Disable random restarts
ik.SolverParameters.AllowRandomRestart = false;

% Parameters to pass to the solver
weights = [1, 1, 1, 1, 1, 1];
q_init = 0.1*ones(numel(q_home),1);
```





## Key Takeaway of this Talk

Success in developing an autonomous robotics system requires:

- Multi-domain simulation
- Trusted tools which make complex workflows easy and integrate with other tools
- Model-based design



# Clearpath Robotics Accelerates Algorithm Development for Industrial Robots

## Challenge

Shorten development times for laser-based perception, computer vision, fleet management, and control algorithms used in industrial robots

## Solution

Use MATLAB to analyze and visualize ROS data, prototype algorithms, and apply the latest advances in robotics research

## Results

- Data analysis time cut by up to 50%
- Customer communication improved
- Cutting-edge SDV algorithms quickly incorporated

[Link to user story](#)



An OTTO self-driving vehicle from Clearpath Robotics.

*“ROS is good for robotics research and development, but not for data analysis. MATLAB, on the other hand, is not only a data analysis tool, it’s a data visualization and hardware interface tool as well, so it’s an excellent complement to ROS in many ways.”*  
- Iliia Baranov, Clearpath Robotics

 **MathWorks®** | *Training Services*

## Deep Learning with MATLAB

This two-day course provides a comprehensive introduction to practical deep learning using MATLAB®.

### Topics include:

- Importing image and sequence data
- Using convolutional neural networks for image classification, regression, and object detection
- Using long short-term memory networks for sequence classification and forecasting
- Modifying common network architectures to solve custom problems
- Improving the performance of a network by modifying training options

```
% Thank you
```