MATLAB EXPO 2019

使用MATLAB和Simulink开发自主机器人

Jing Wu

主要内容

成功的开发一个自主机器人系统需要：

1. 使用新技术进行多域仿真
2. 使用可信赖的工具，可以将复杂的工作流程简化并与其它工具集成
3. 使用基于模型的设计作为开发方式

开发自主机器人系统面临的挑战

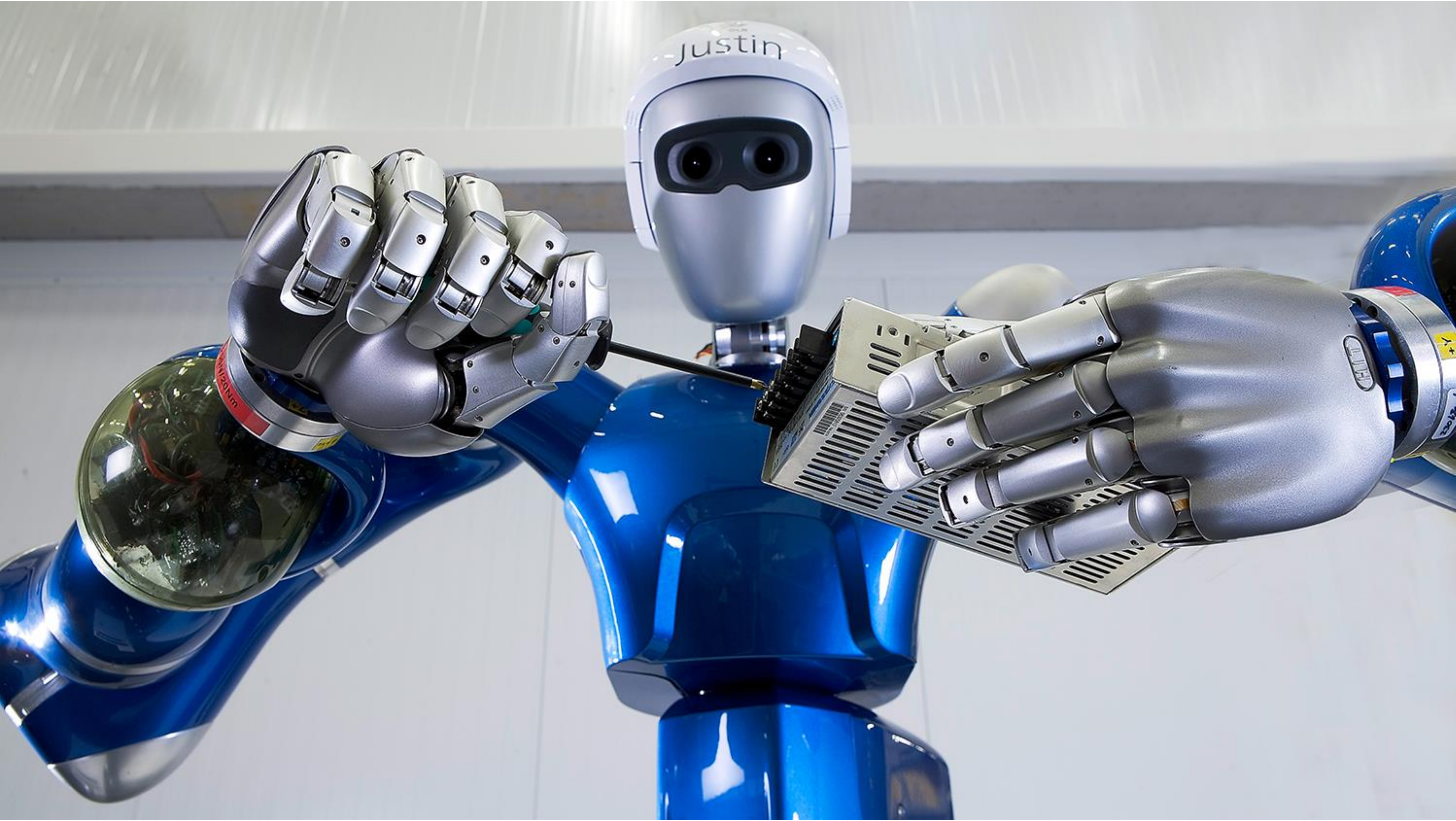
多域专业知识的应用

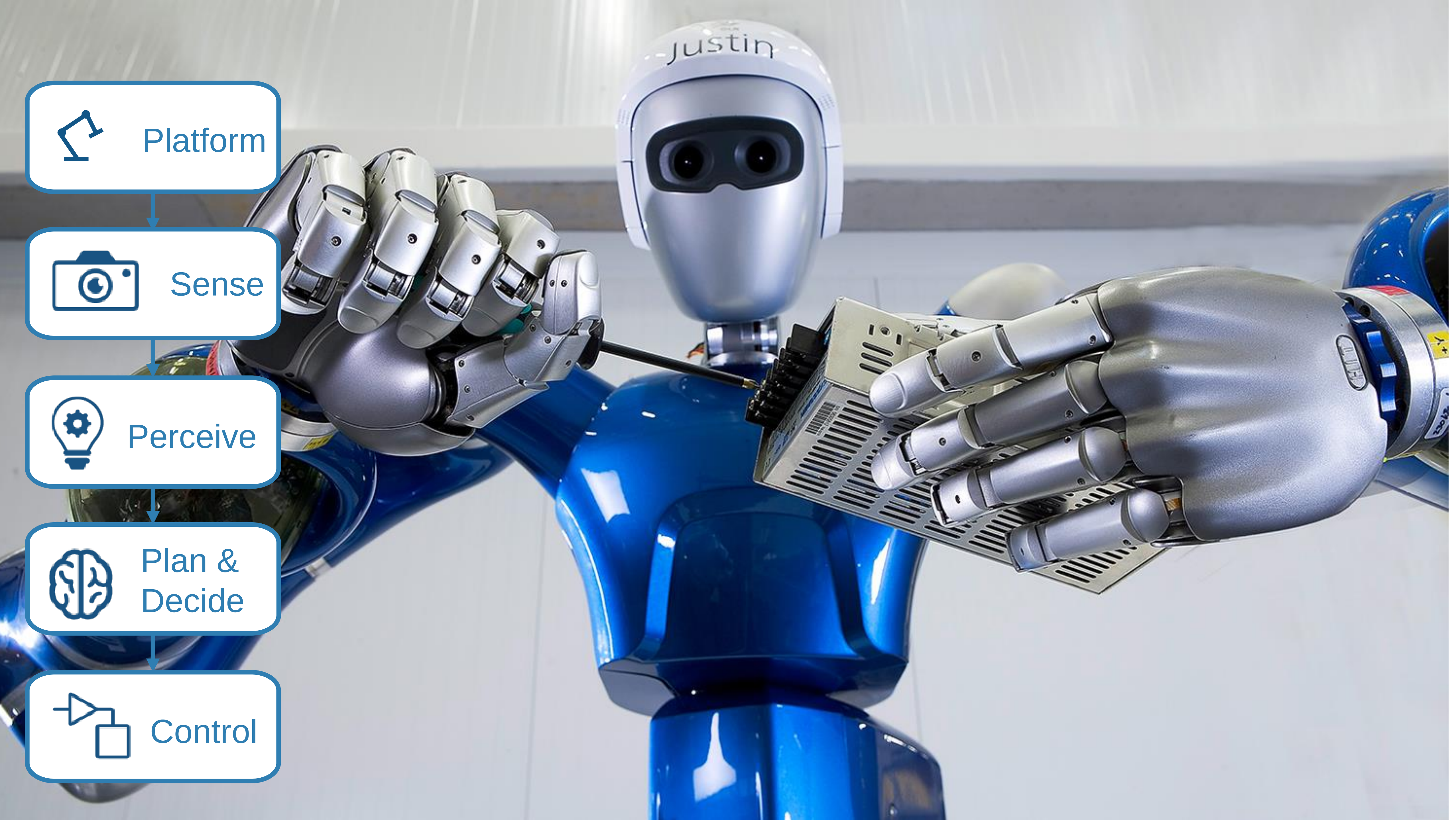
算法的复杂性

端到端工作流

知识产权保护

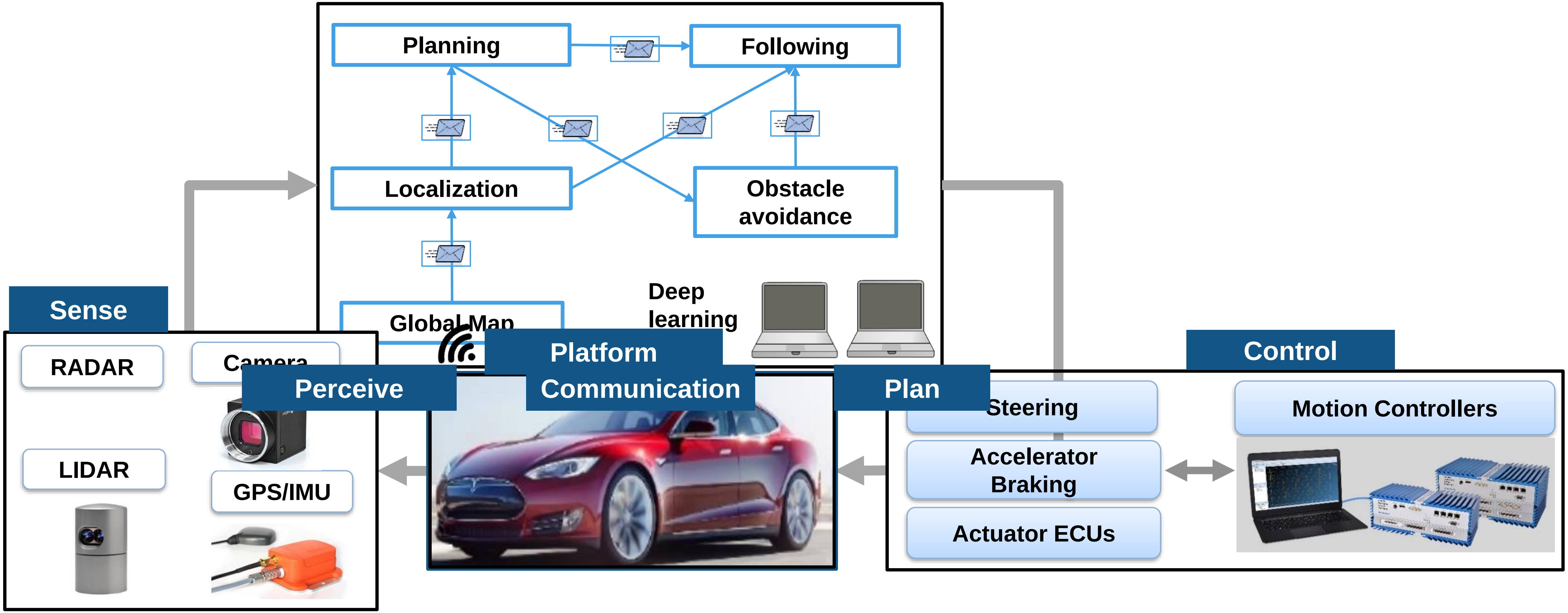
成功的机器人是什么样的?


Justin

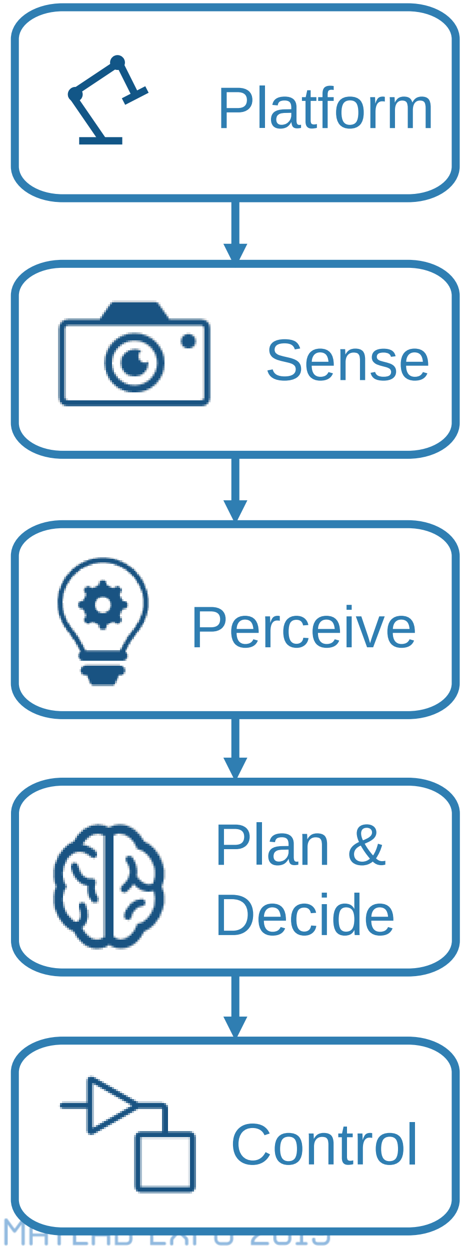


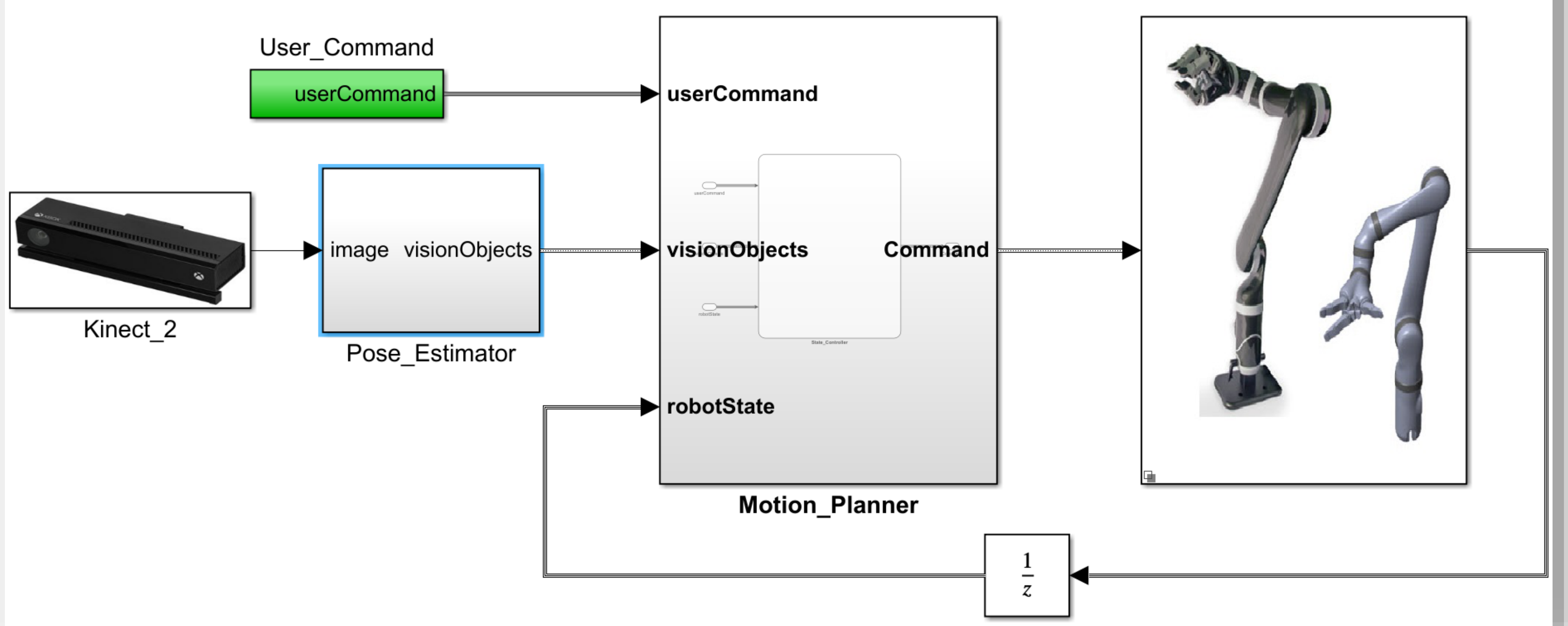
Justin
Platform
Sense
Perceive
Plan &
Decide
Control

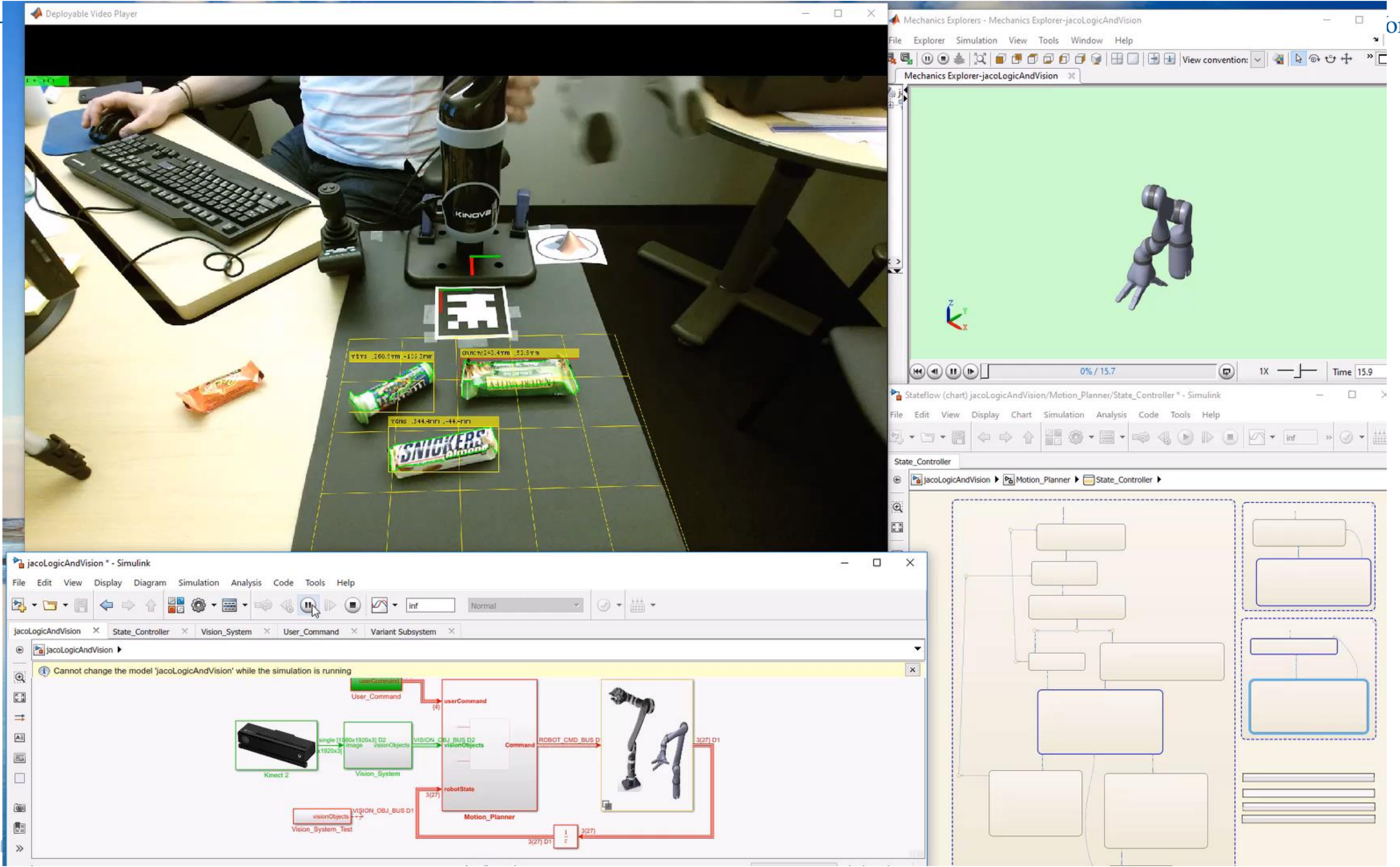
其它例子: 自主驾驶汽车


Planning
Following
Localization
Obstacle avoidance
Global Map
Deep learning
Sense
RADAR
Camera
LIDAR
GPS/IMU
Perceive
Platform
Communication
Plan
Control
Steering
Accelerator Braking
Actuator ECUs
Motion Controllers

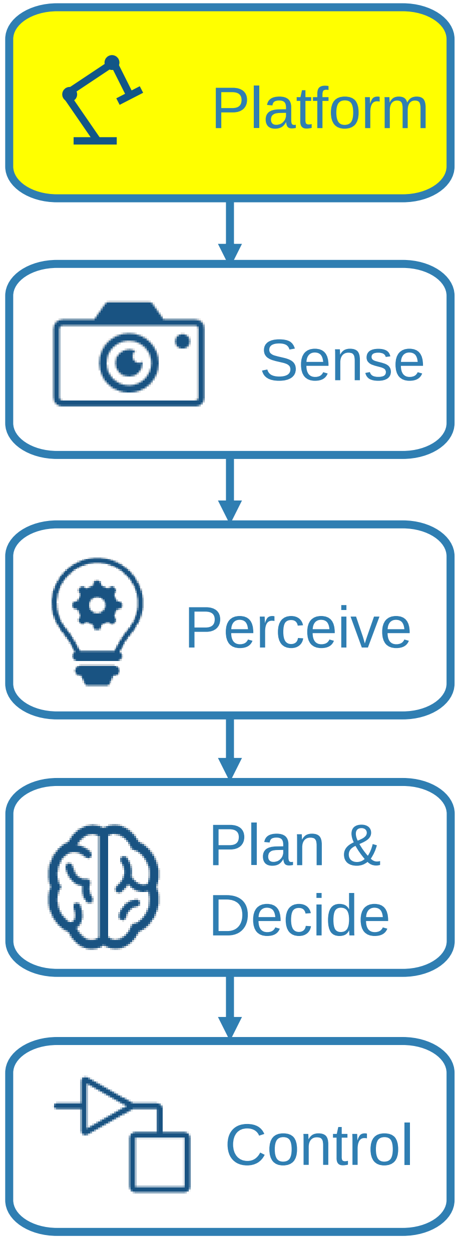
今天：设计抓取放置应用

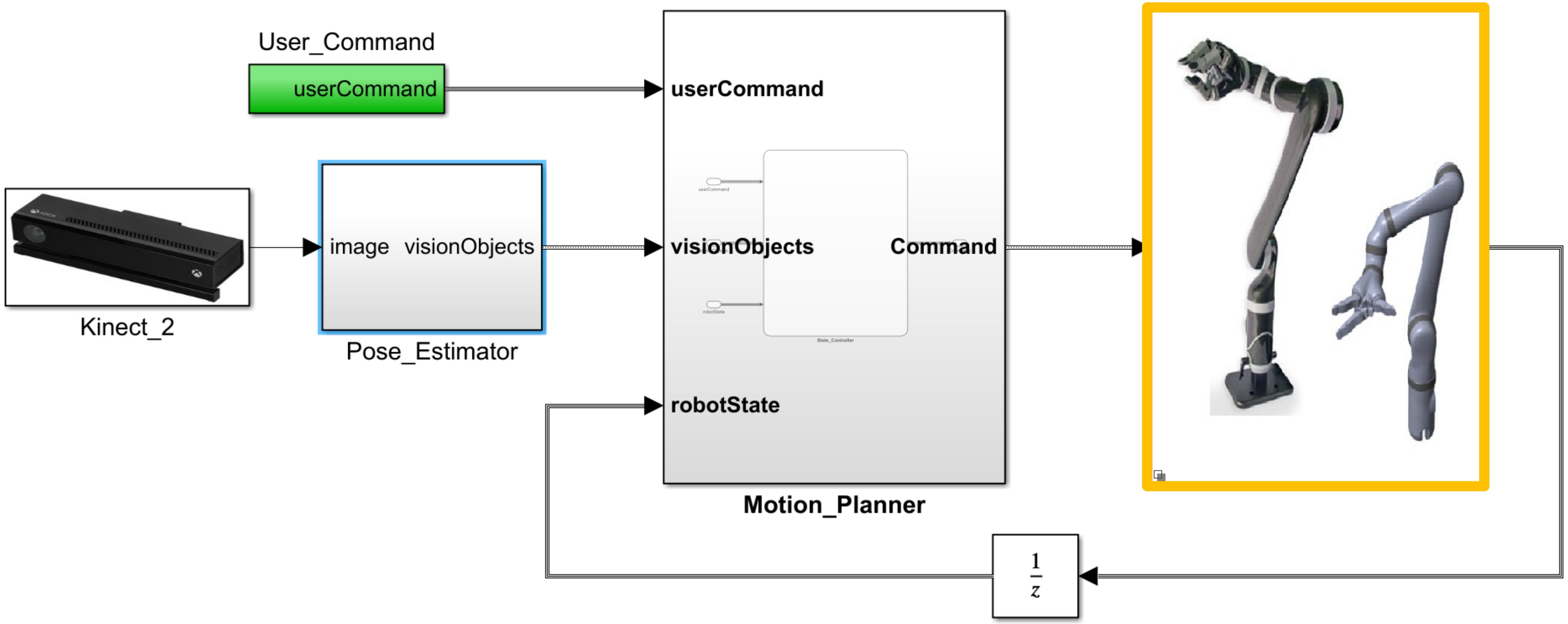

Platform
Sense
Perceive
Plan &
Decide
Control


User_Command
userCommand
userCommand
Kinect_2
image
visionObjects
Pose_Estimator
visionObjects
Command
robotState
Motion_Planner
1
z


Deployable Video Player
Mechanics Explorers - Mechanics Explorer-jacoLogicAndVision
File Explorer Simulation View Tools Window Help
View convention:
Mechanics Explorer-jacoLogicAndVision
0% / 15.7
1X
Time 15.9
Stateflow (chart) jacoLogicAndVision/Motion_Planner/State_Controller * - Simulink
File Edit View Display Chart Simulation Analysis Code Tools Help
State_Controller
jacoLogicAndVision
Motion_Planner
State_Controller
jacoLogicAndVision * - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
inf
Normal
jacoLogicAndVision
State_Controller
Vision_System
User_Command
Variant Subsystem
Cannot change the model 'jacoLogicAndVision' while the simulation is running
User_Command
userCommand
Kinect 2
Vision_System
visionObjects
Command
robotState
Motion_Planner
ROBOT_CMD_BUS D
VISION_OBJ_BUS D1
Vision_System_Test
SNICKERS

今天：设计抓取放置应用


Platform
Sense
Perceive
Plan &
Decide
Control


User_Command
userCommand
image
visionObjects
Kinect_2
Pose_Estimator
userCommand
visionObjects
Command
robotState
Motion_Planner
1
z

平台设计

如何创建一个满足要求的系统模型？

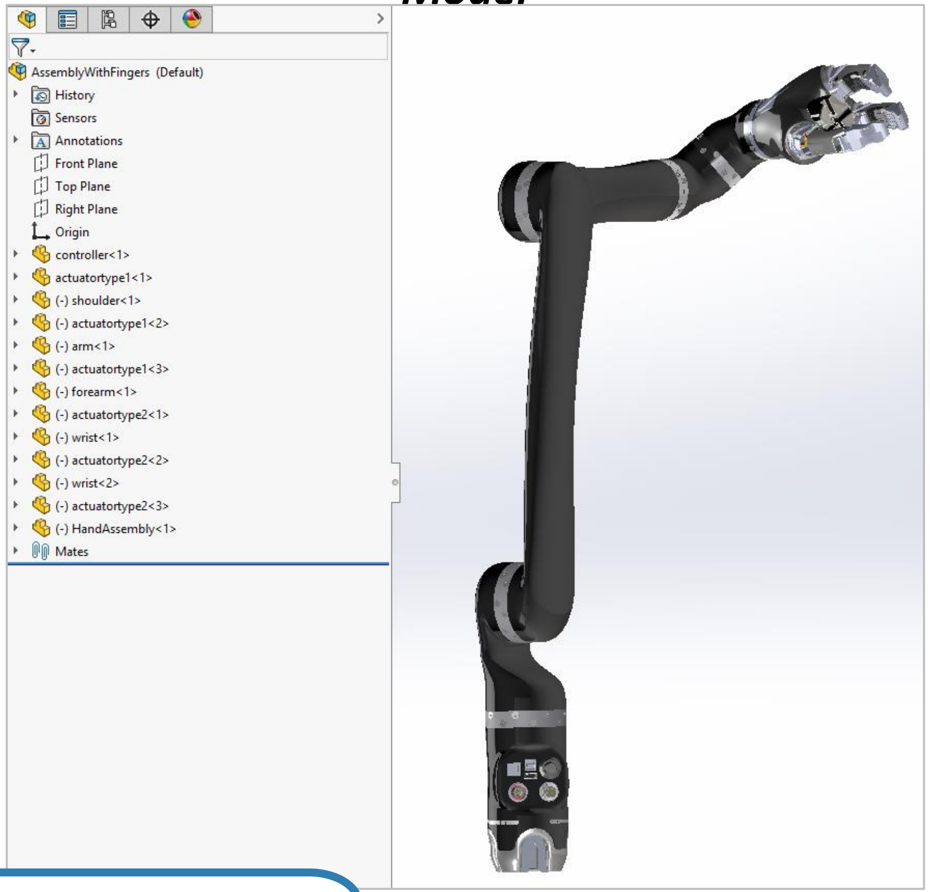
Mechanics

Actuators

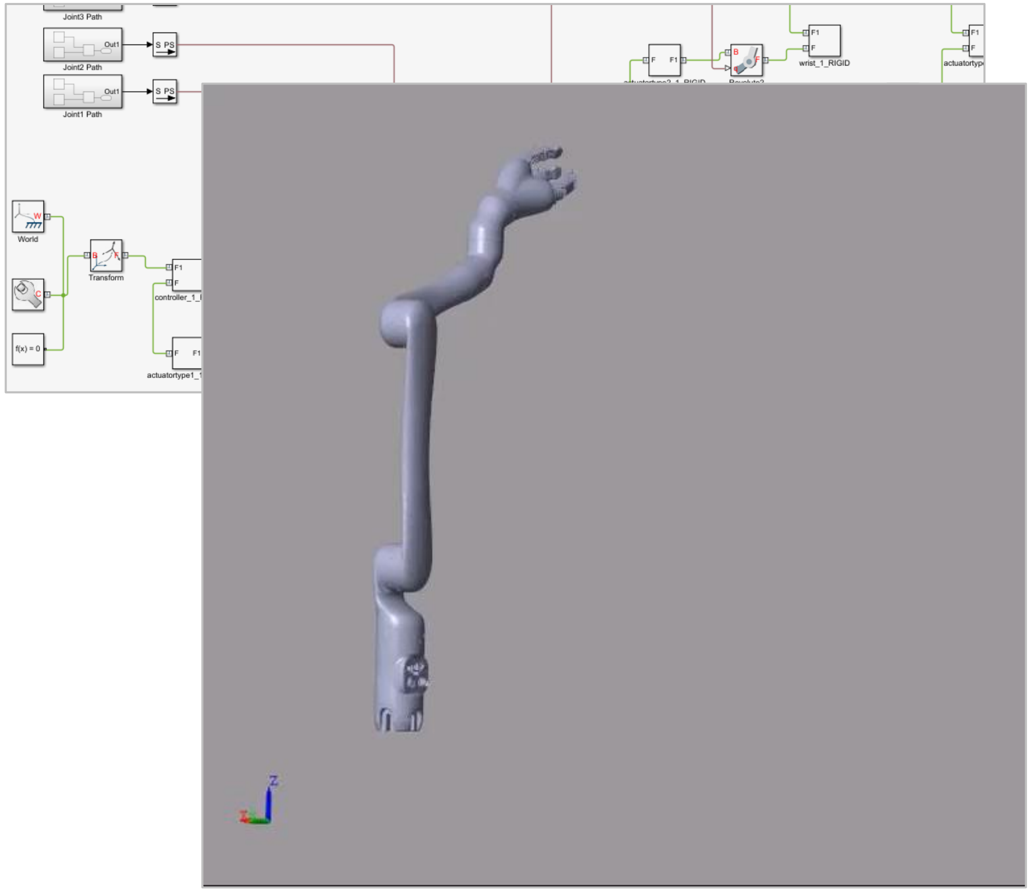
Environment

从通用CAD工具导入模型

SolidWorks Model



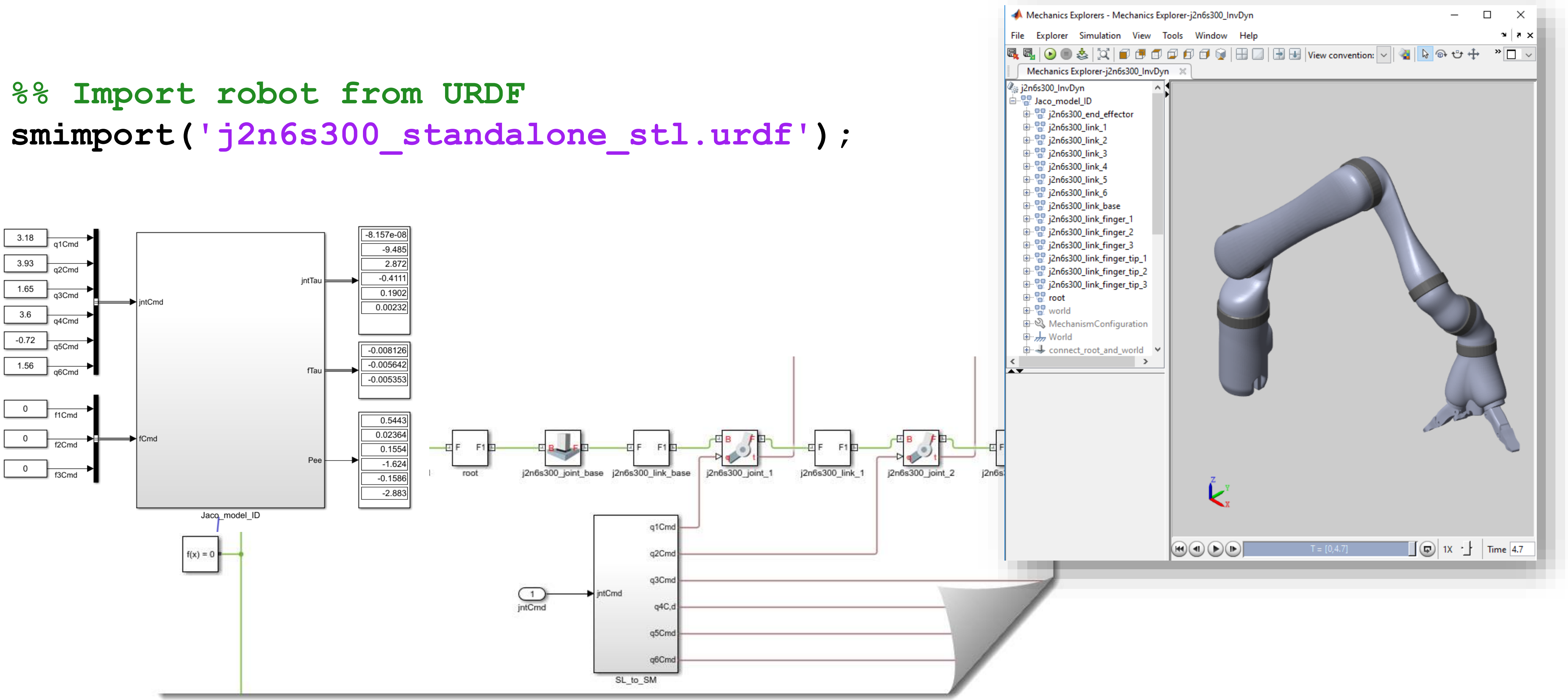
Simscape Multibody Model



Mechanics

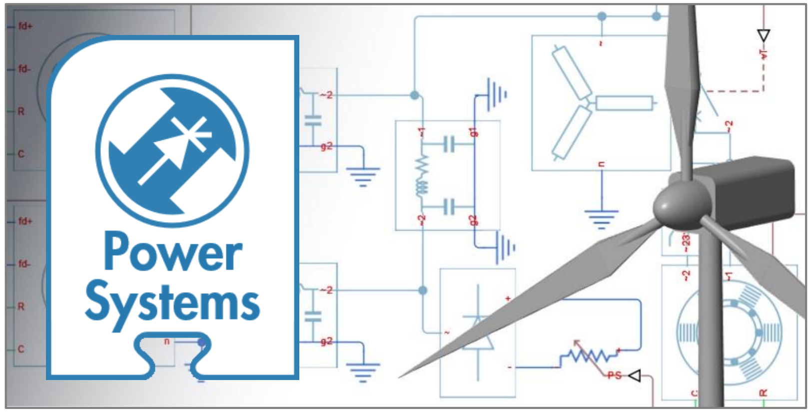
力学：一条指令即可从URDF文件导入

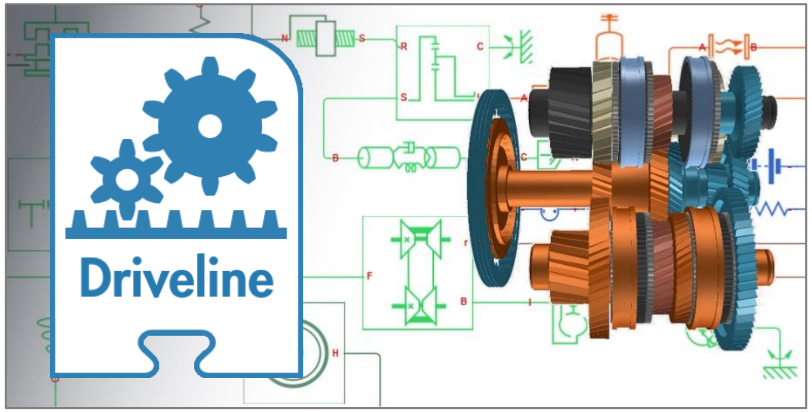
```
%% Import robot from URDF
smimport('j2n6s300_standalone_stl.urdf');
```

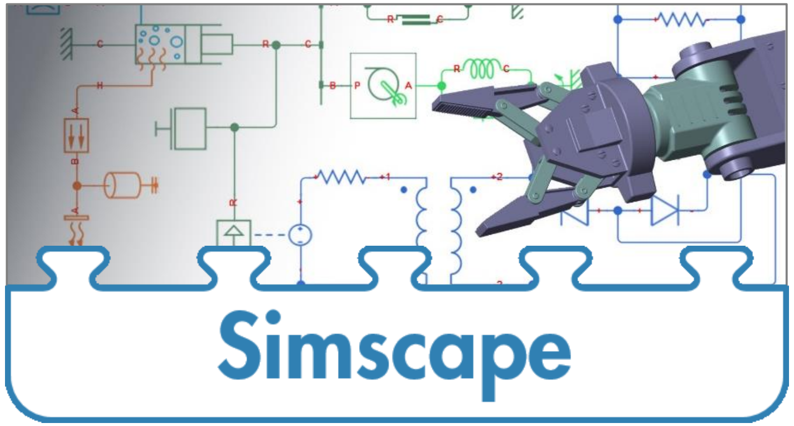


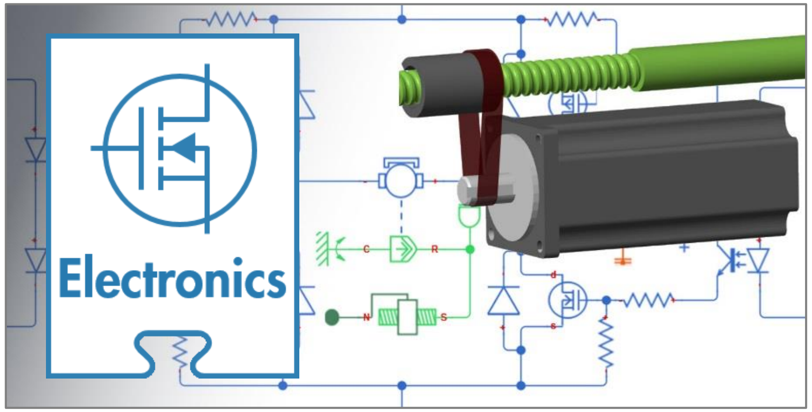
Demo

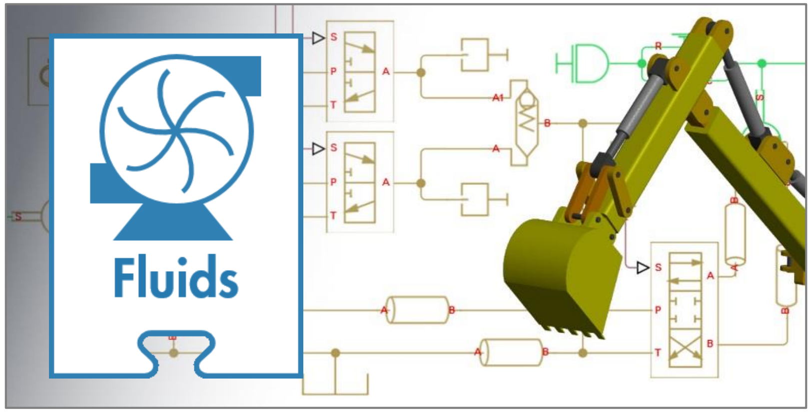
关节驱动器：对其他物理域建模

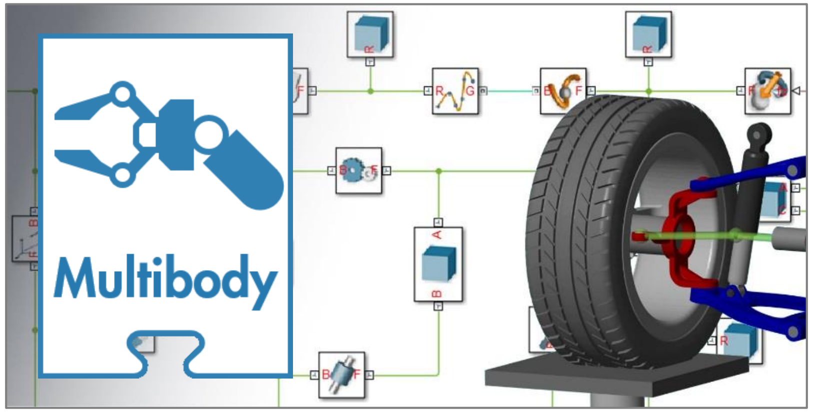






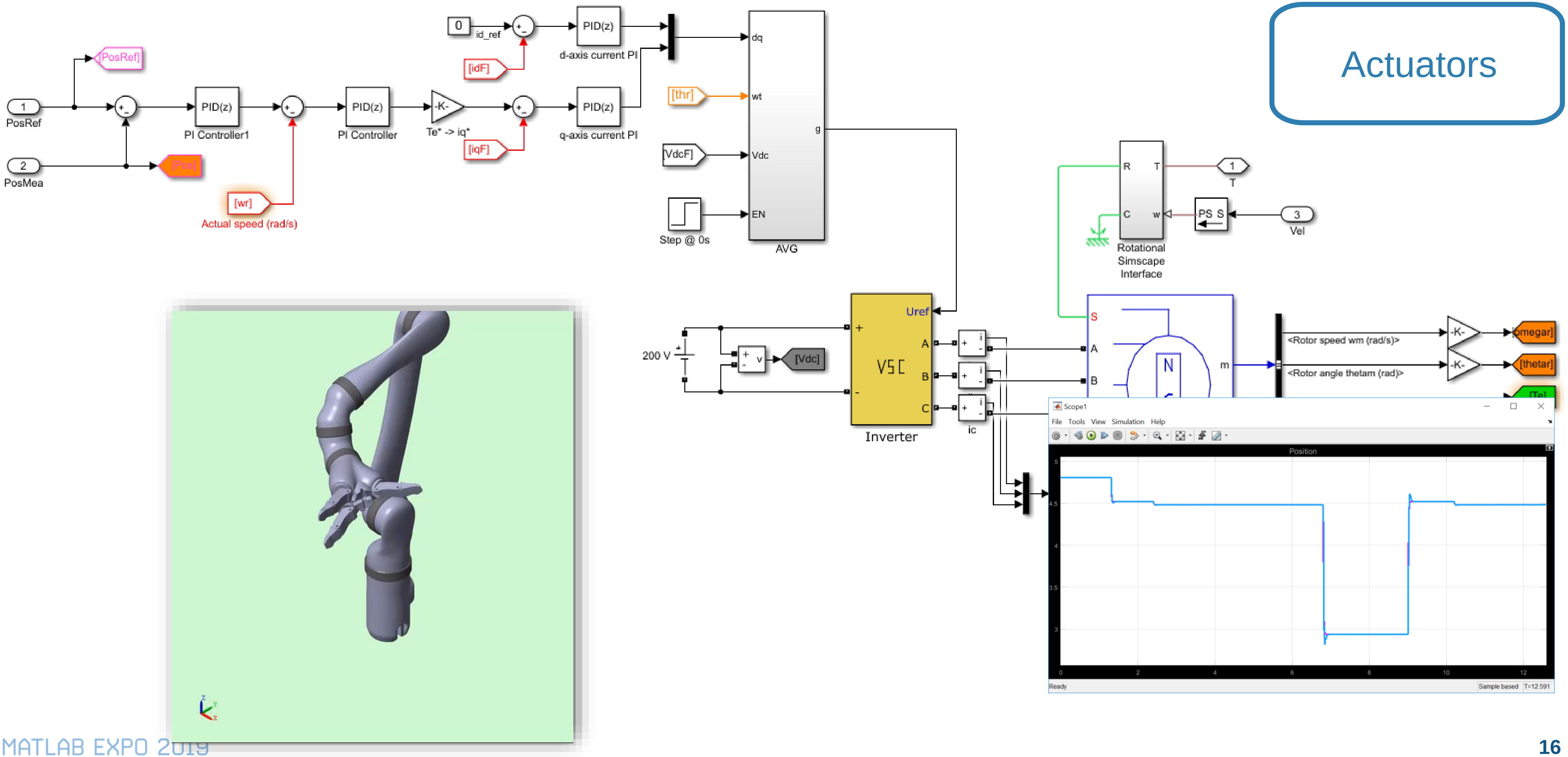




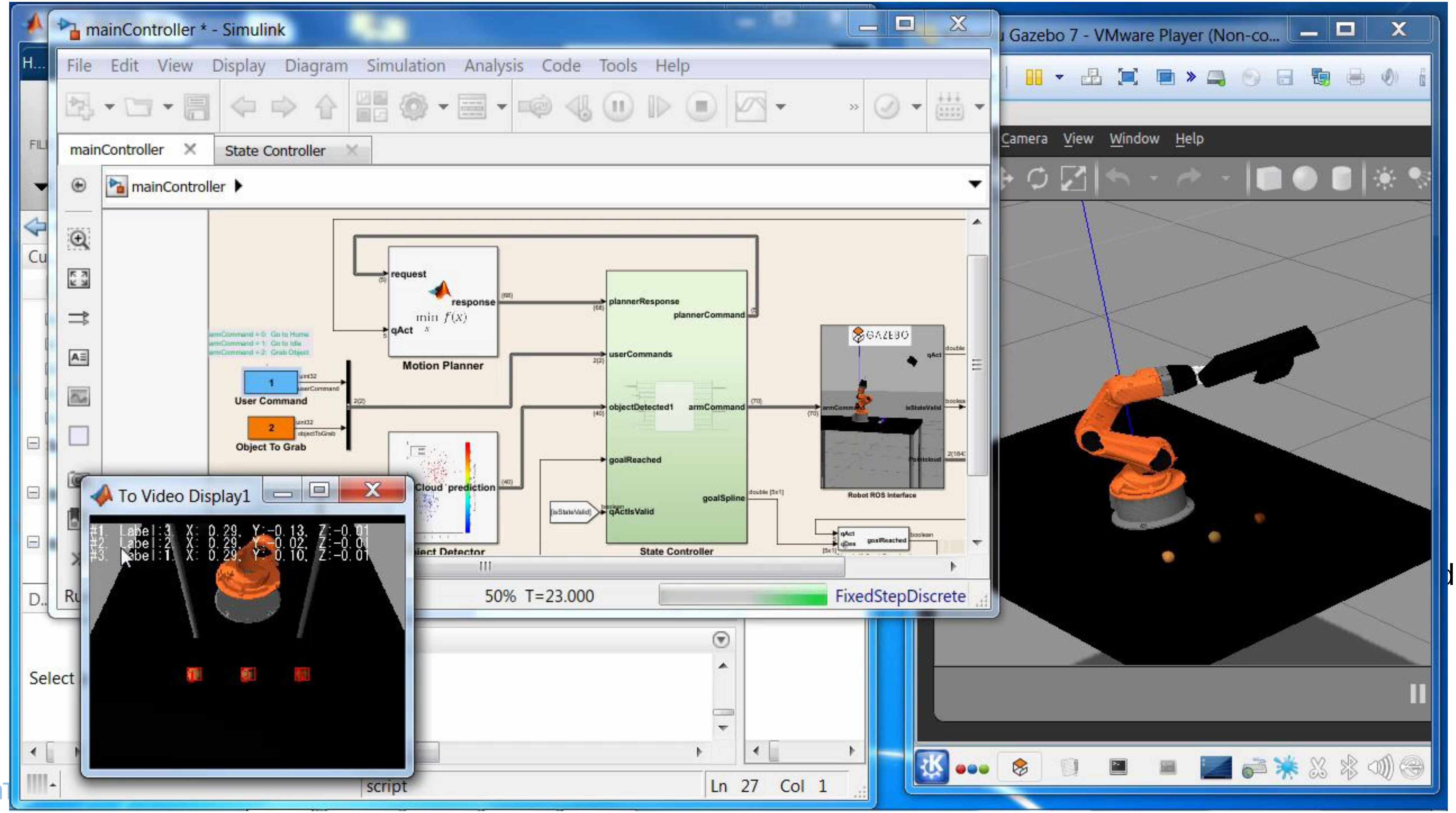


关节驱动器建模

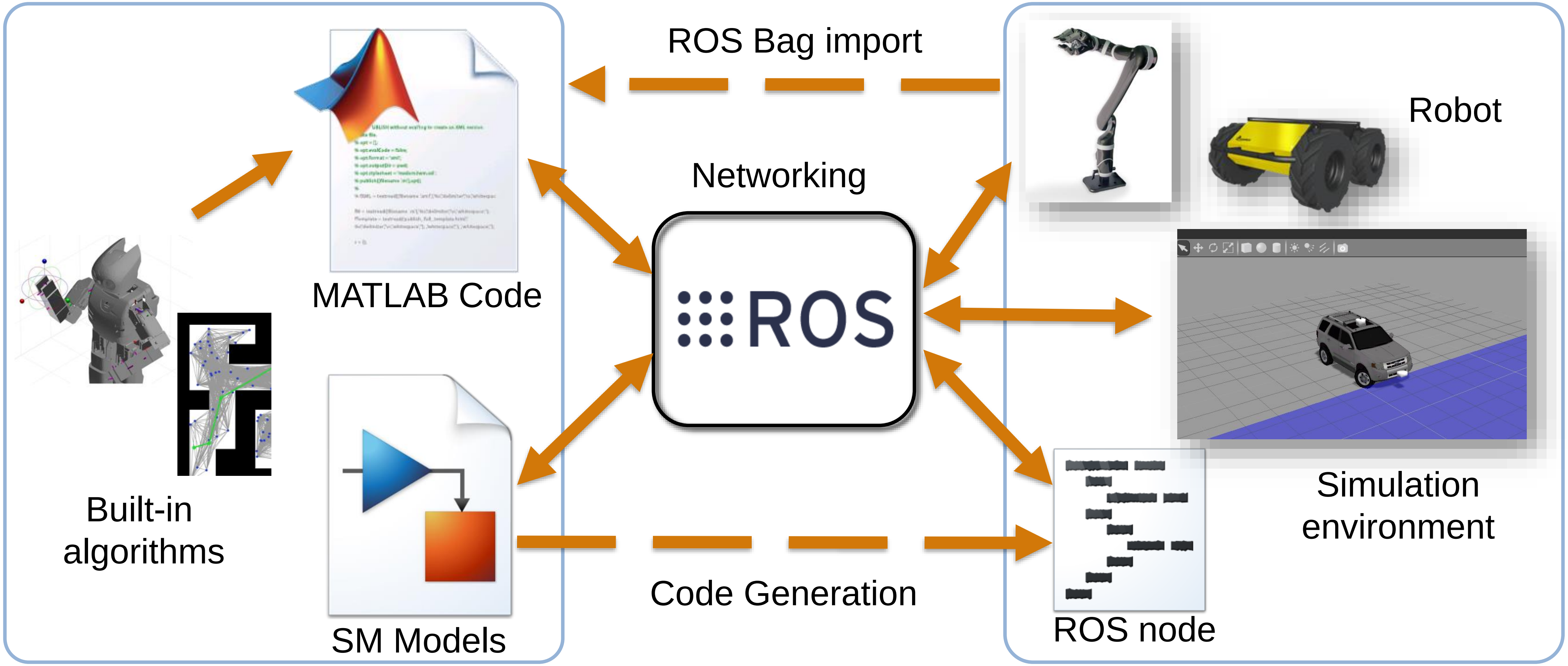
Actuators



环境：与外部机器人仿真环境连接

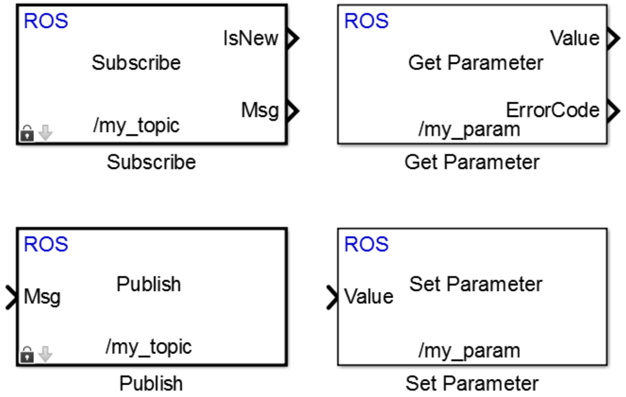


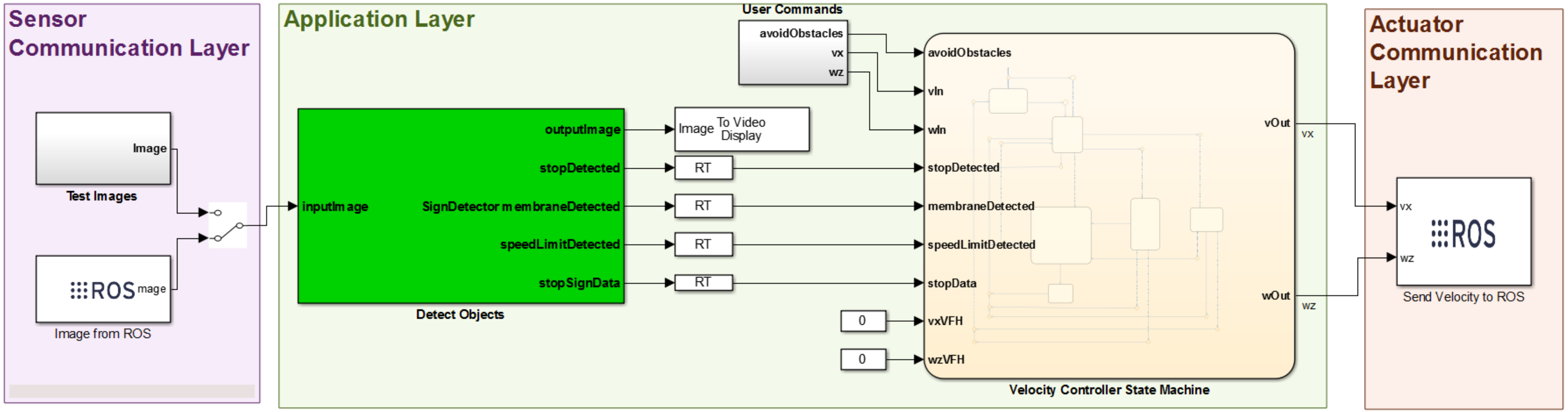
环境：将MATLAB和Simulink与ROS连接



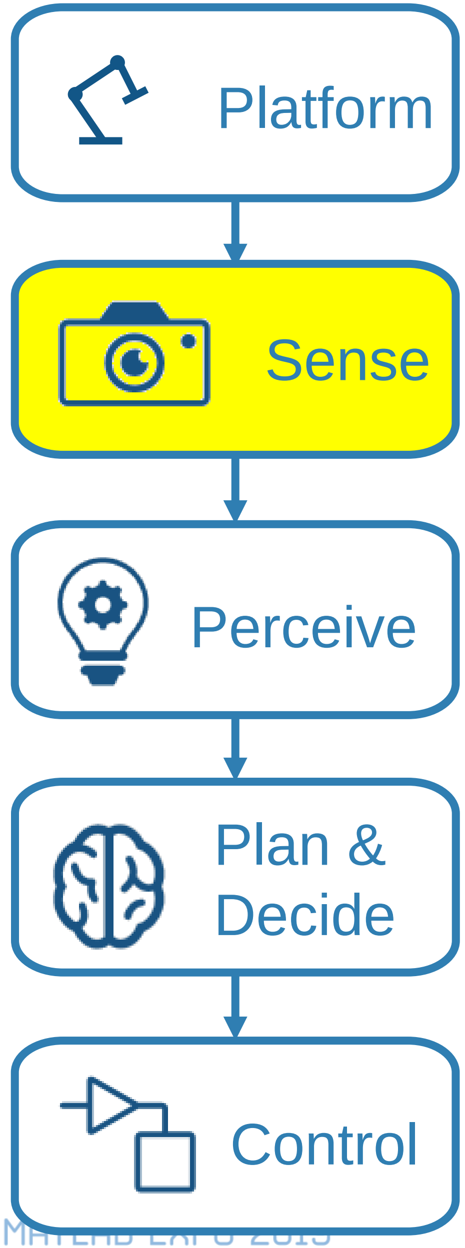
与ROS联合仿真

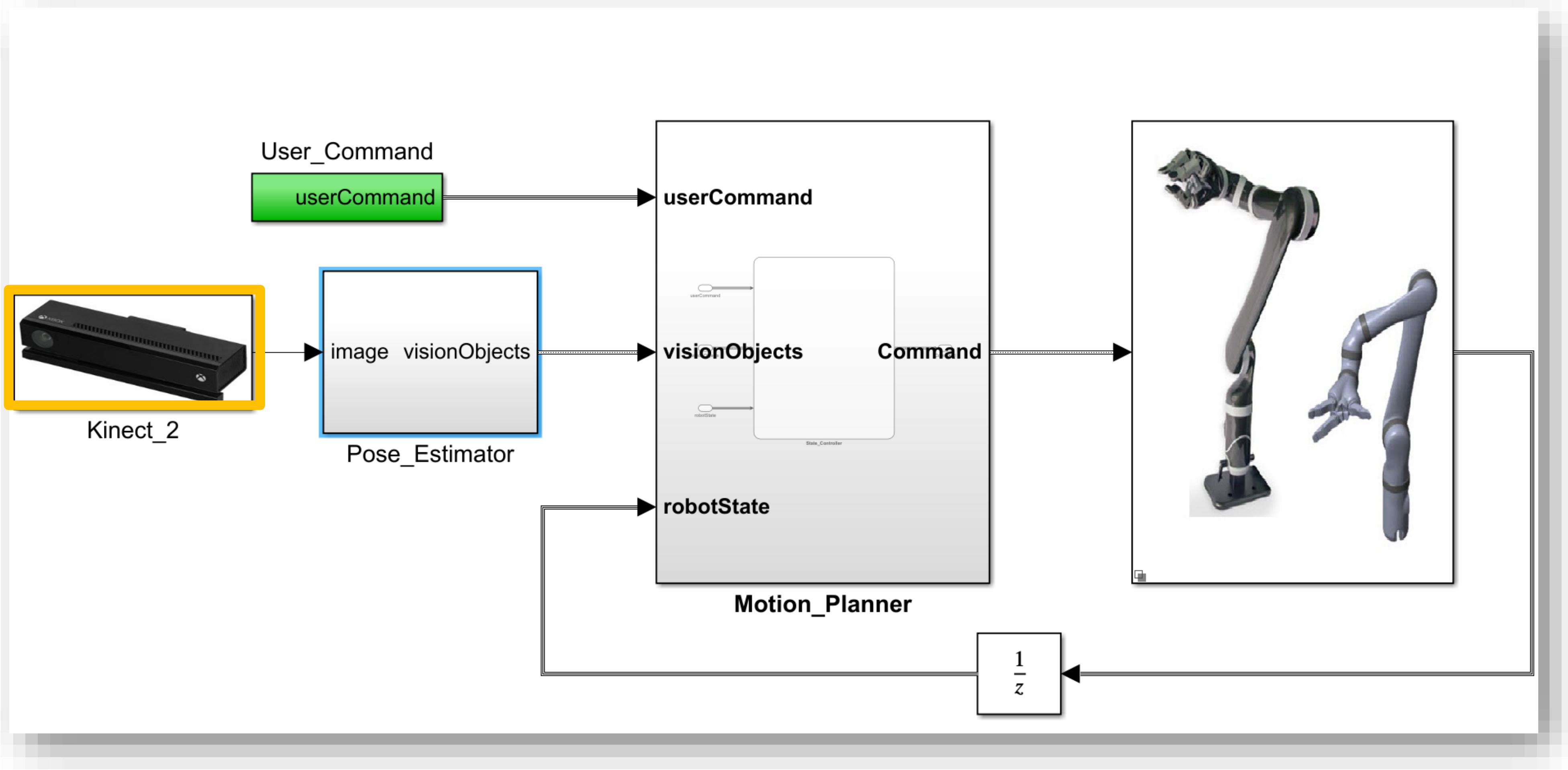
```
%% Connect to ROS
rosinit '192.168.204.144';

%% Create subscribers
imSub = rossubscriber('/camera/rgb/image_raw');
scanSub = rossubscriber('/scan');

%% Create publisher
[velPub, velMsg] = rospublisher('/husky_velocity_controller/cmd_vel');
```





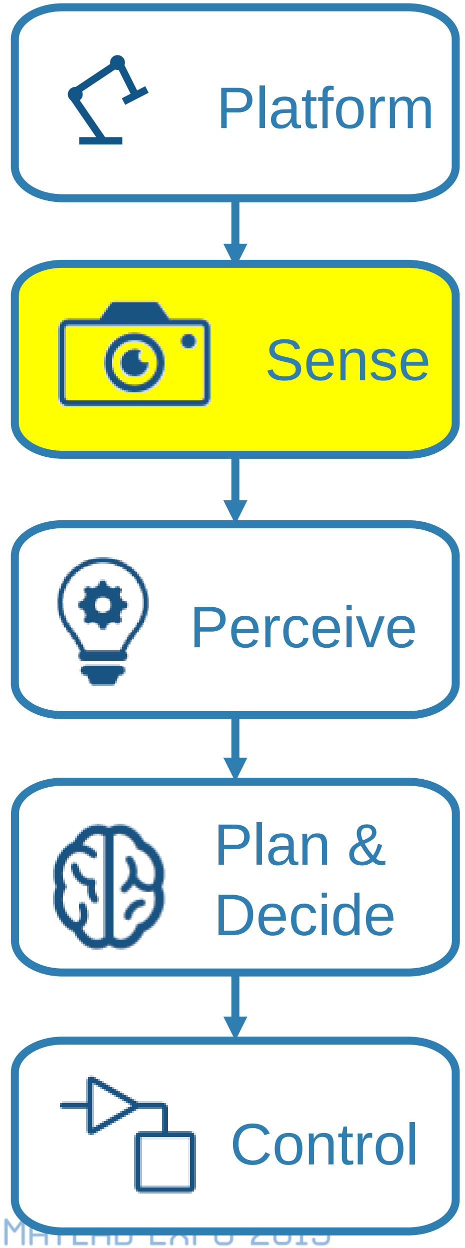
设计抓取放置应用



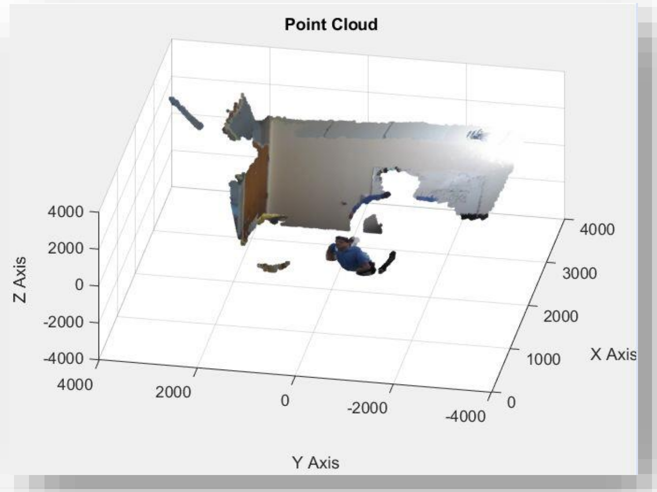


Demo

设计抓取放置应用

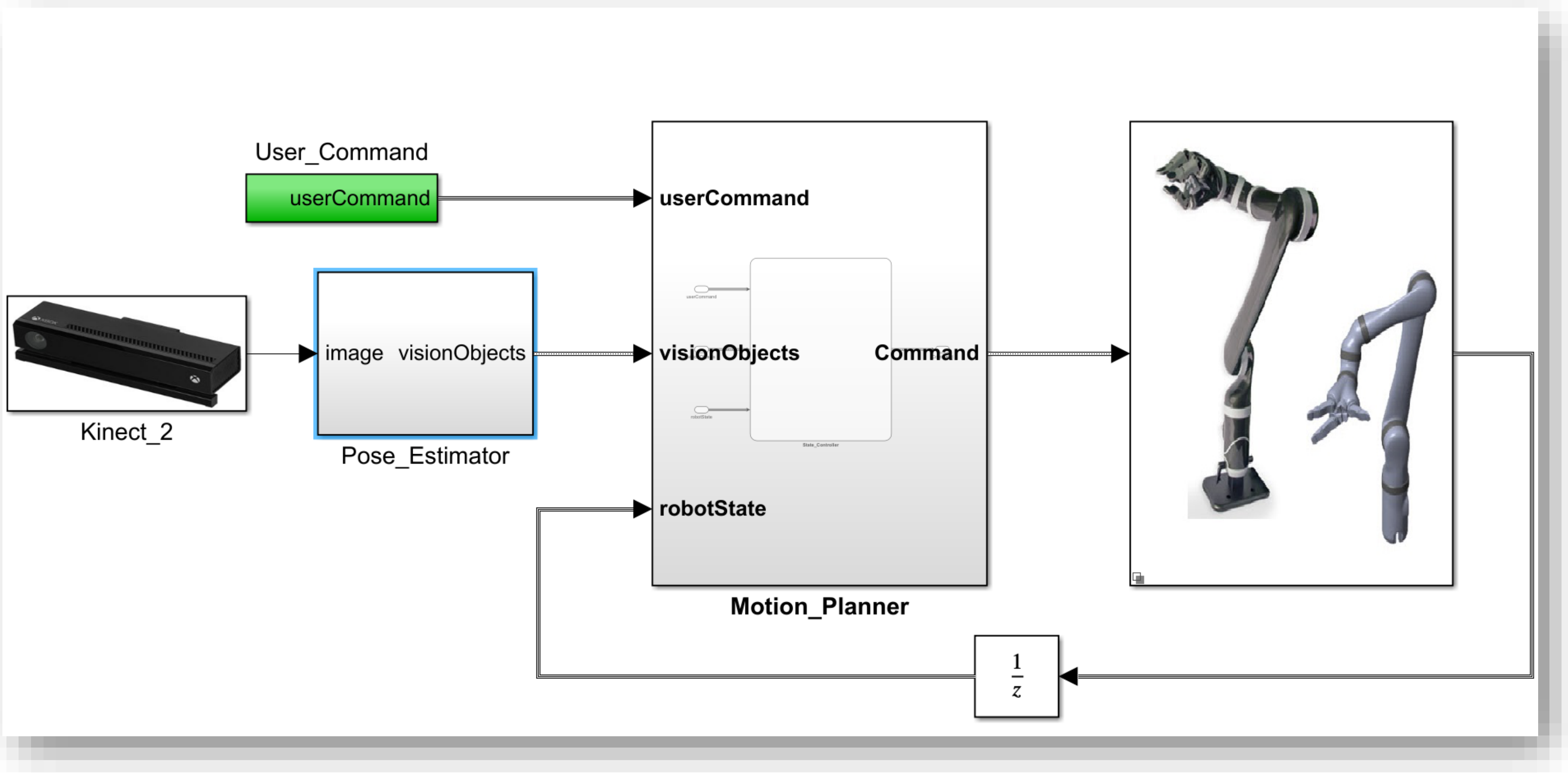


- 支持通用传感器
- 图像分析
- Apps应用
- 图像增强
- 可视化点云



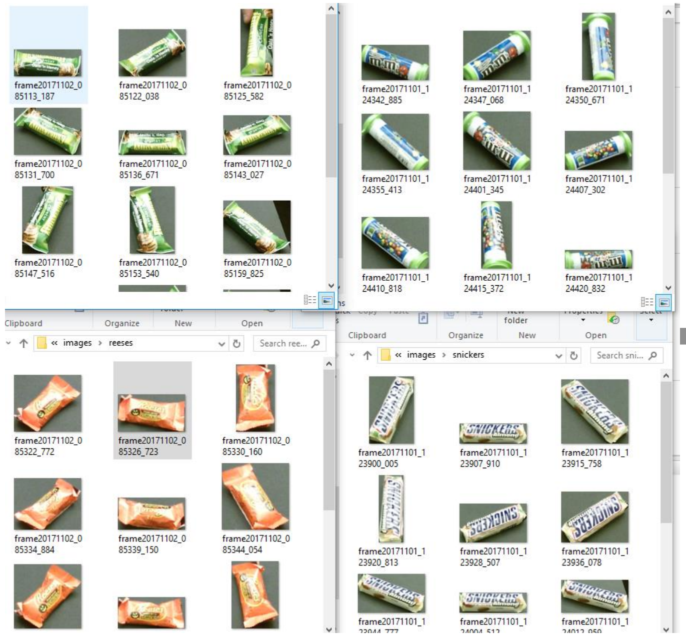
今天：设计抓取放置应用

- Platform
- Sense
- **Perceive**
- Plan & Decide
- Control



目标分类器和位姿估计

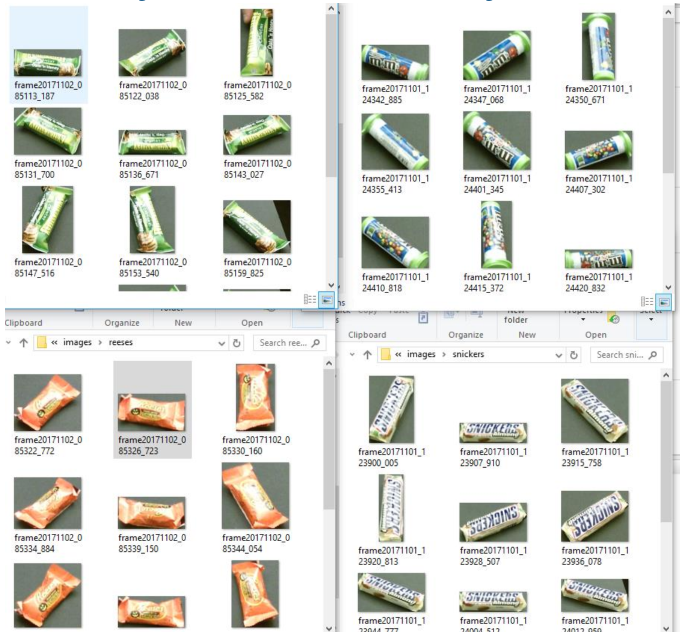
Images



Pose Estimator

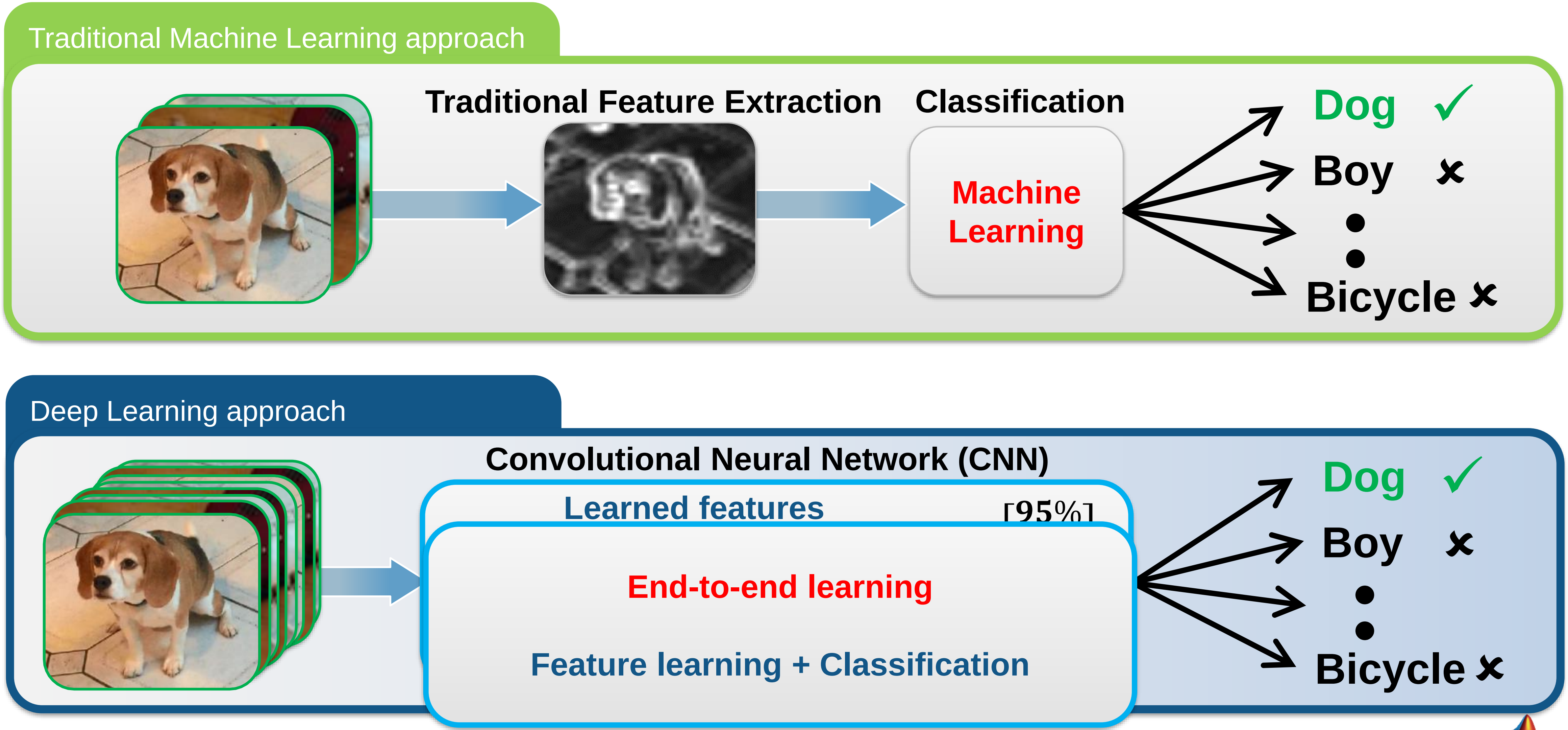
Labels and Poses

Object 1 Object 2

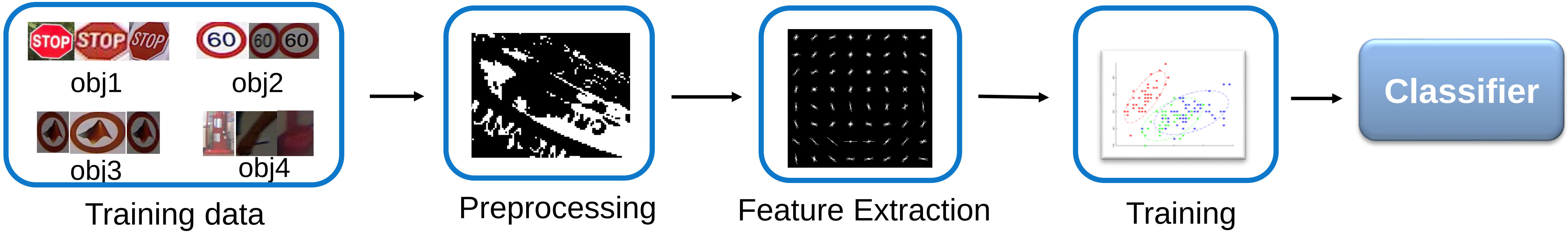


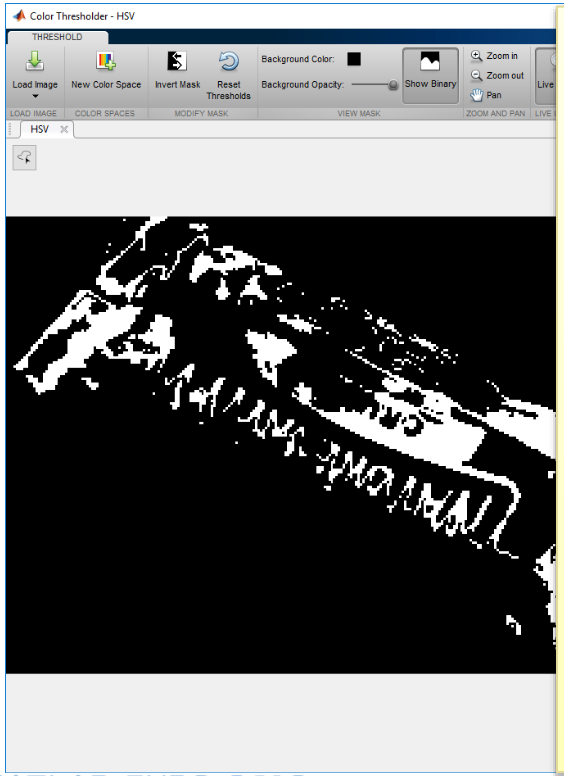
Object 3 Object 4

MATLAB让机器学习简单和易用

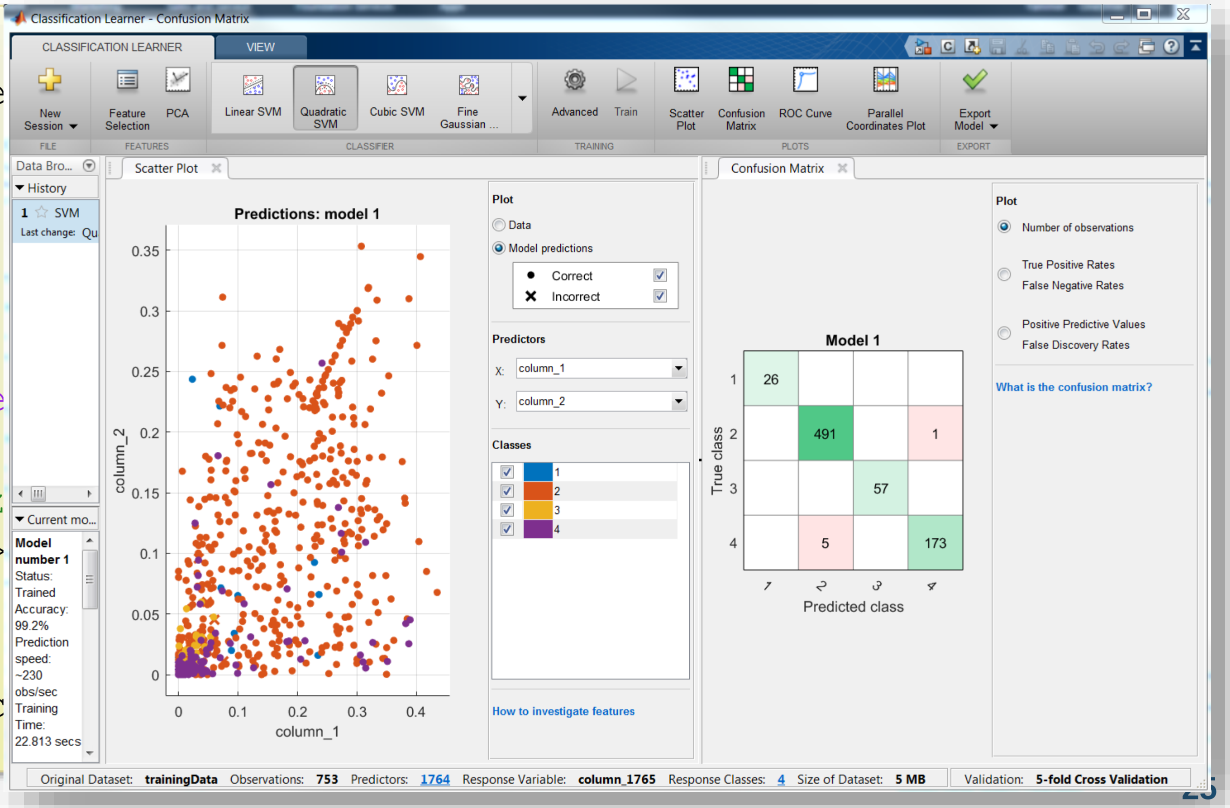


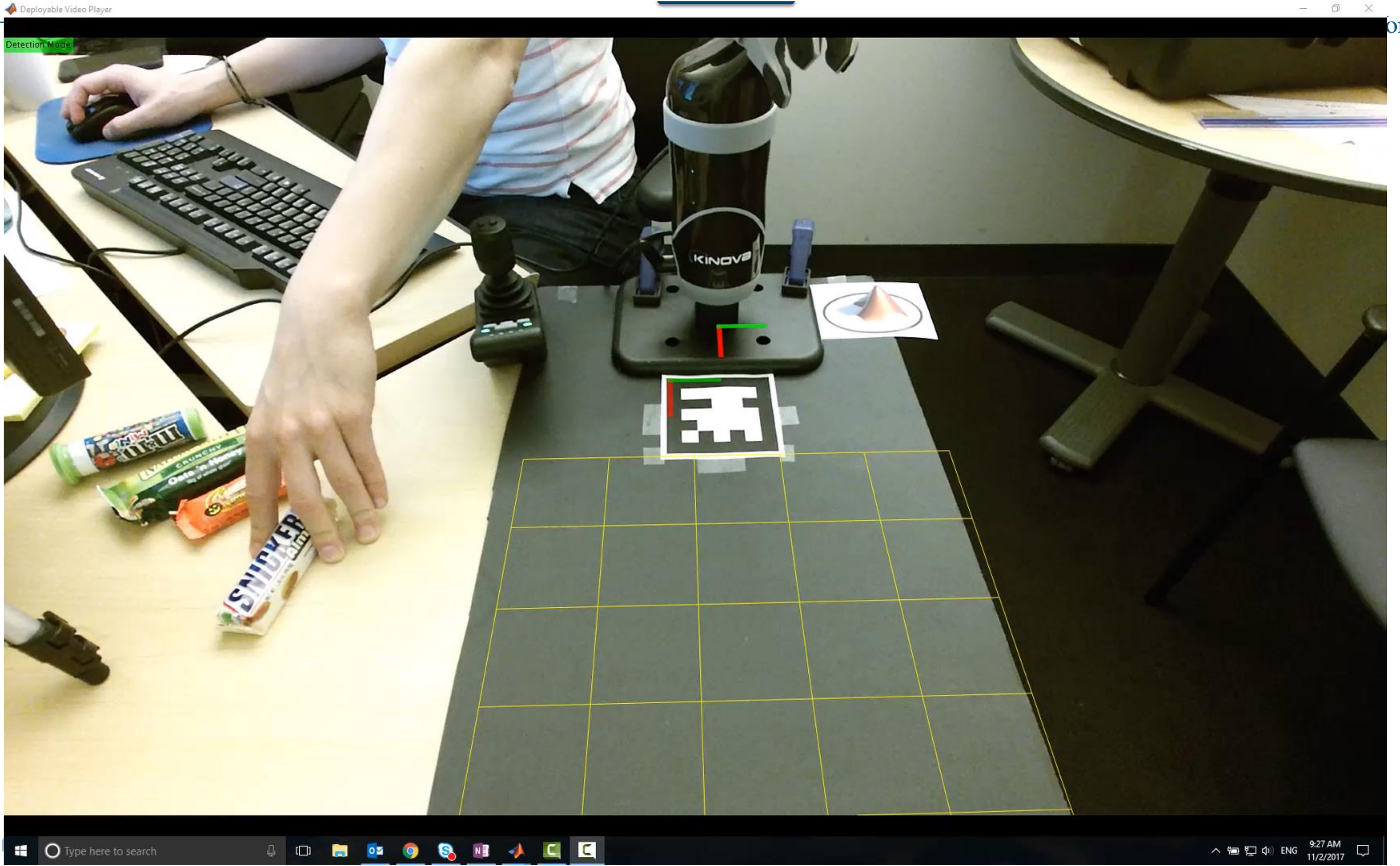
MATLAB让复杂工作流简化





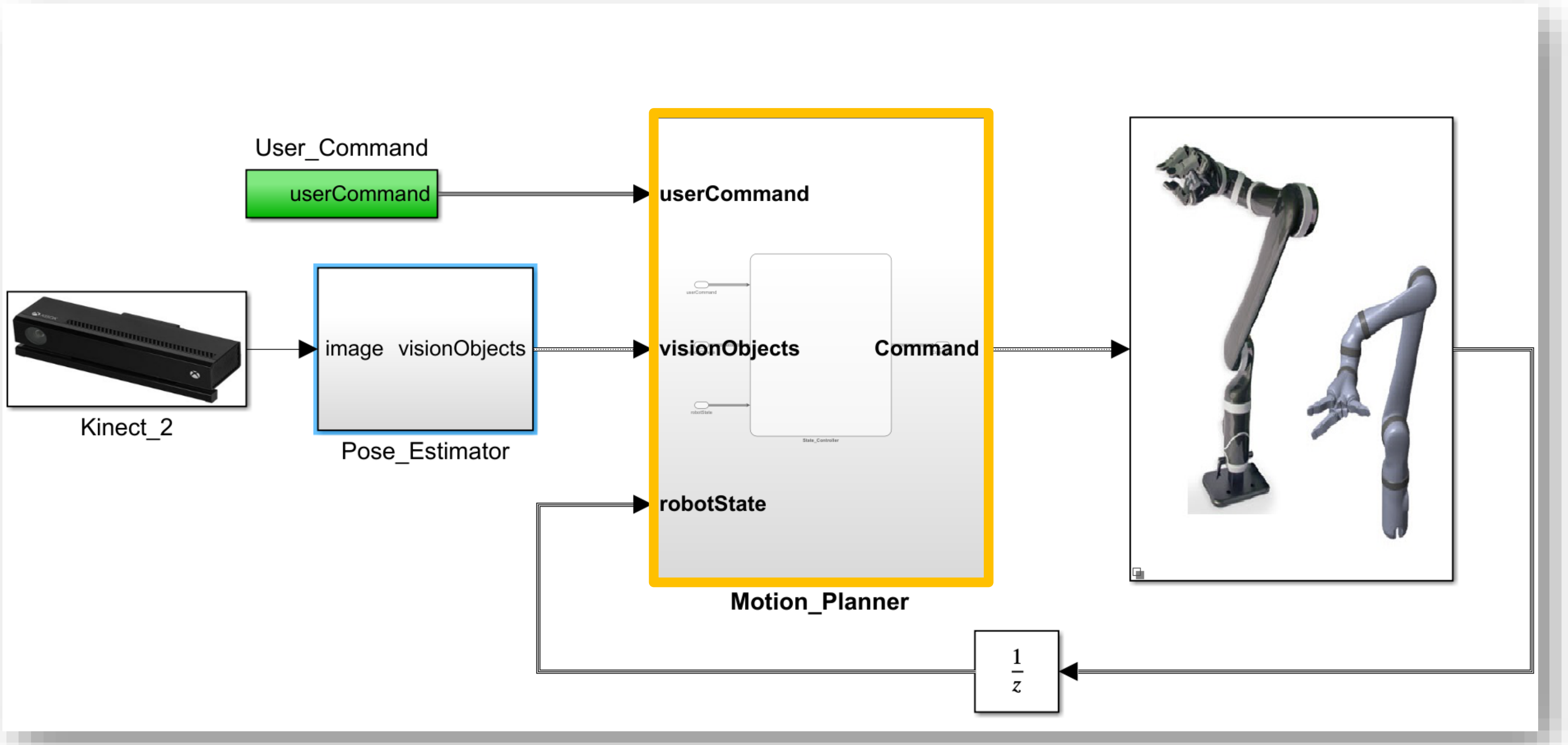

```
% Detect regions
BW = createMask(videoFrame
% Fill image regions
BW = imfill(BW,'holes');
% Get bounding boxes
stats = regionprops('table
% Filter based on area siz
targetIndex = stats.Area >
% Get bounding boxes from
testFeatures(k,:) = extrac
```



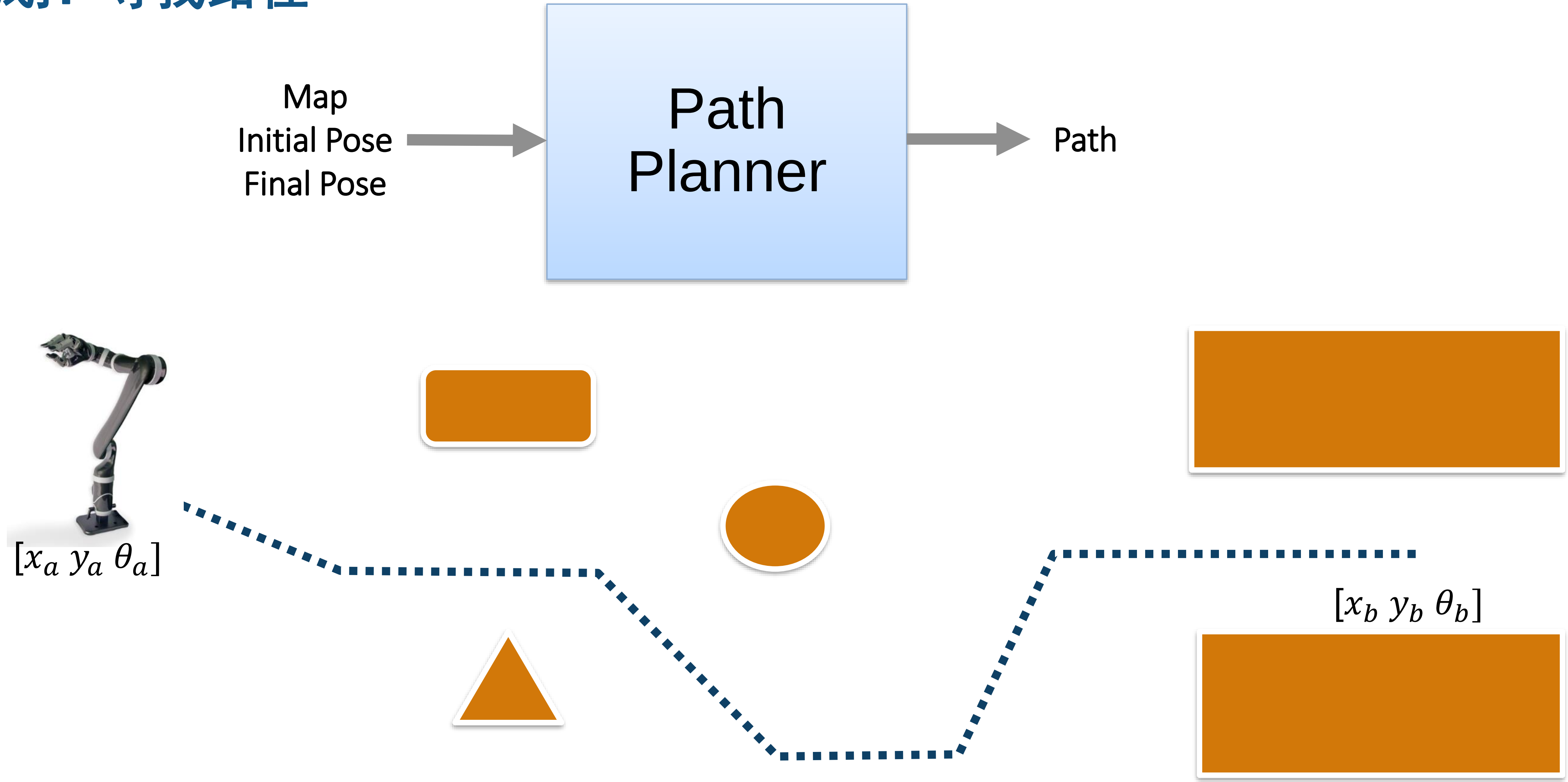

Deployable Video Player
Detection Mode
KINOVA
Type here to search
ENG
9:27 AM
11/2/2017

设计抓取放置应用

- Platform
- Sense
- Perceive
- **Plan & Decide**
- Control

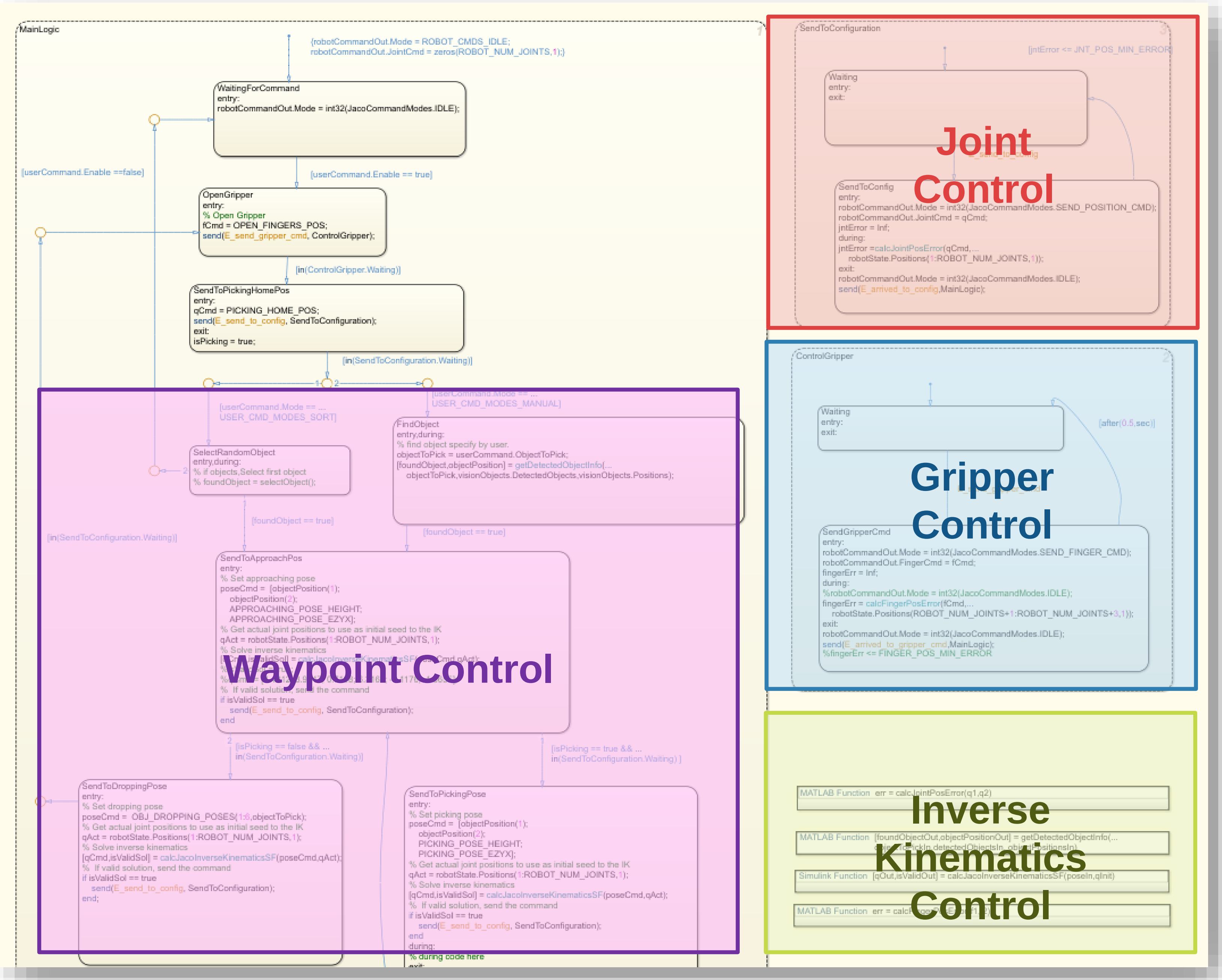


规划：寻找路径

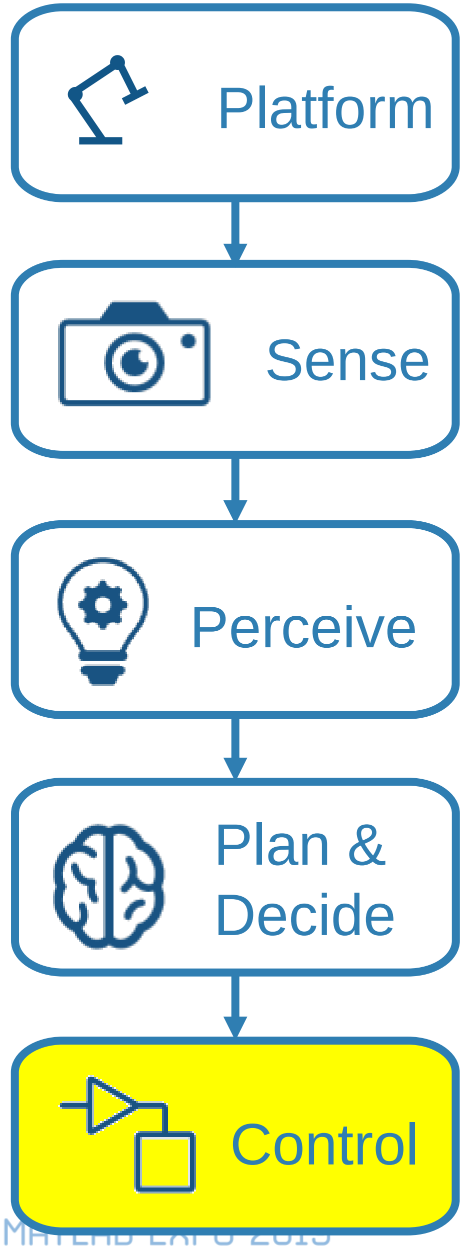


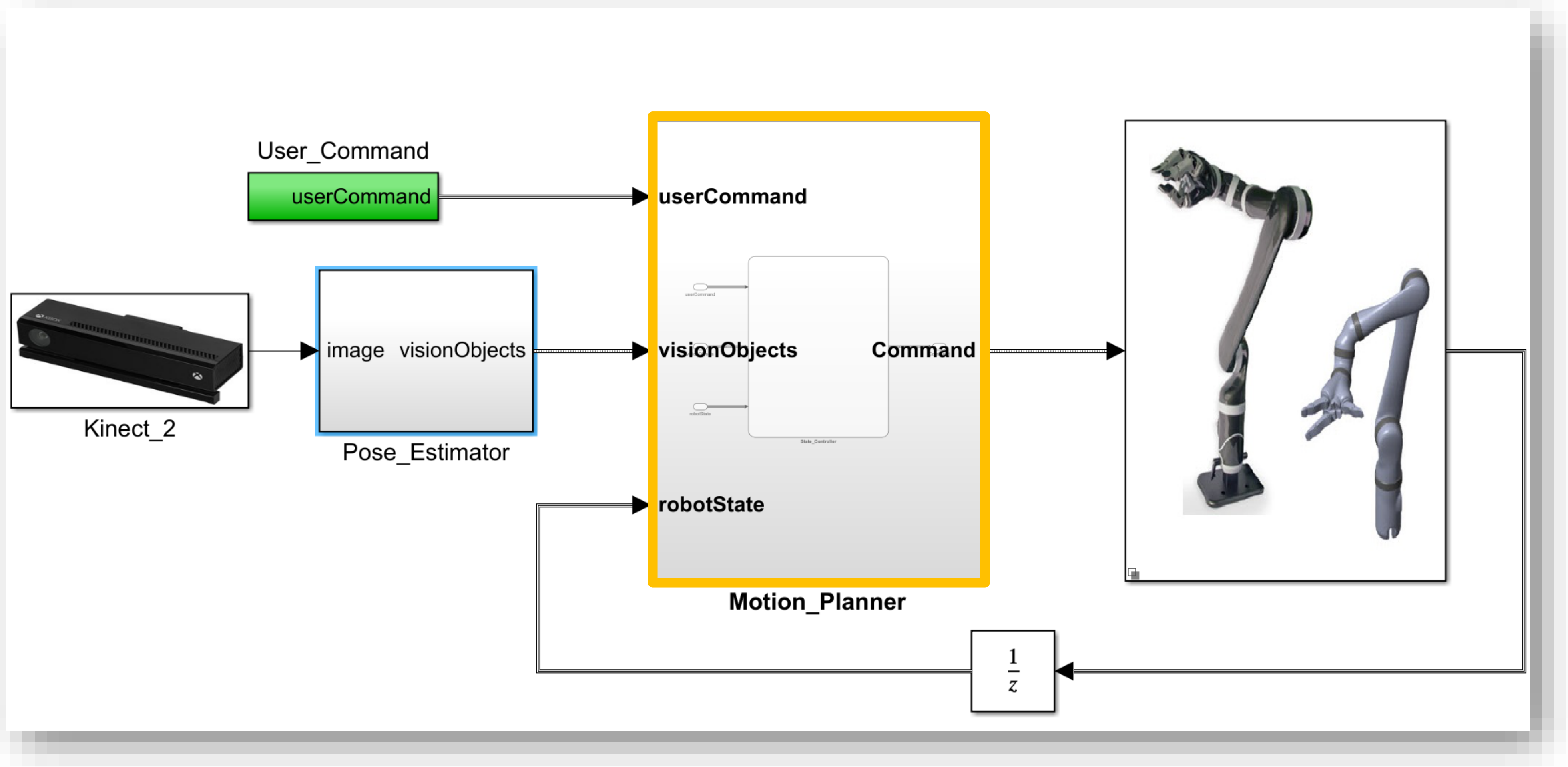
Map
Initial Pose
Final Pose
Path Planner
Path
$[x_a\ y_a\ \theta_a]$
$[x_b\ y_b\ \theta_b]$

使用Stateflow规划



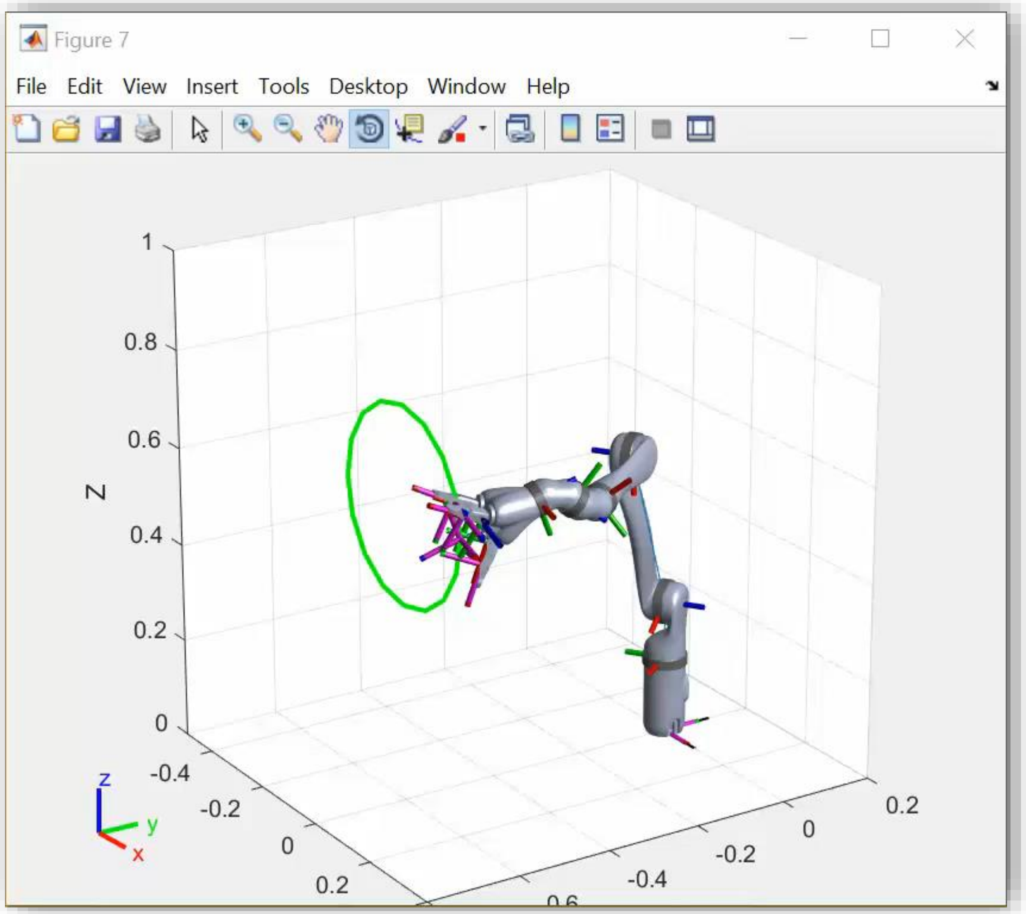
设计抓取放置应用


Platform
Sense
Perceive
Plan &
Decide
Control


User_Command
userCommand
userCommand
Kinect_2
image
visionObjects
Pose_Estimator
visionObjects
Command
robotState
Motion_Planner
1
z

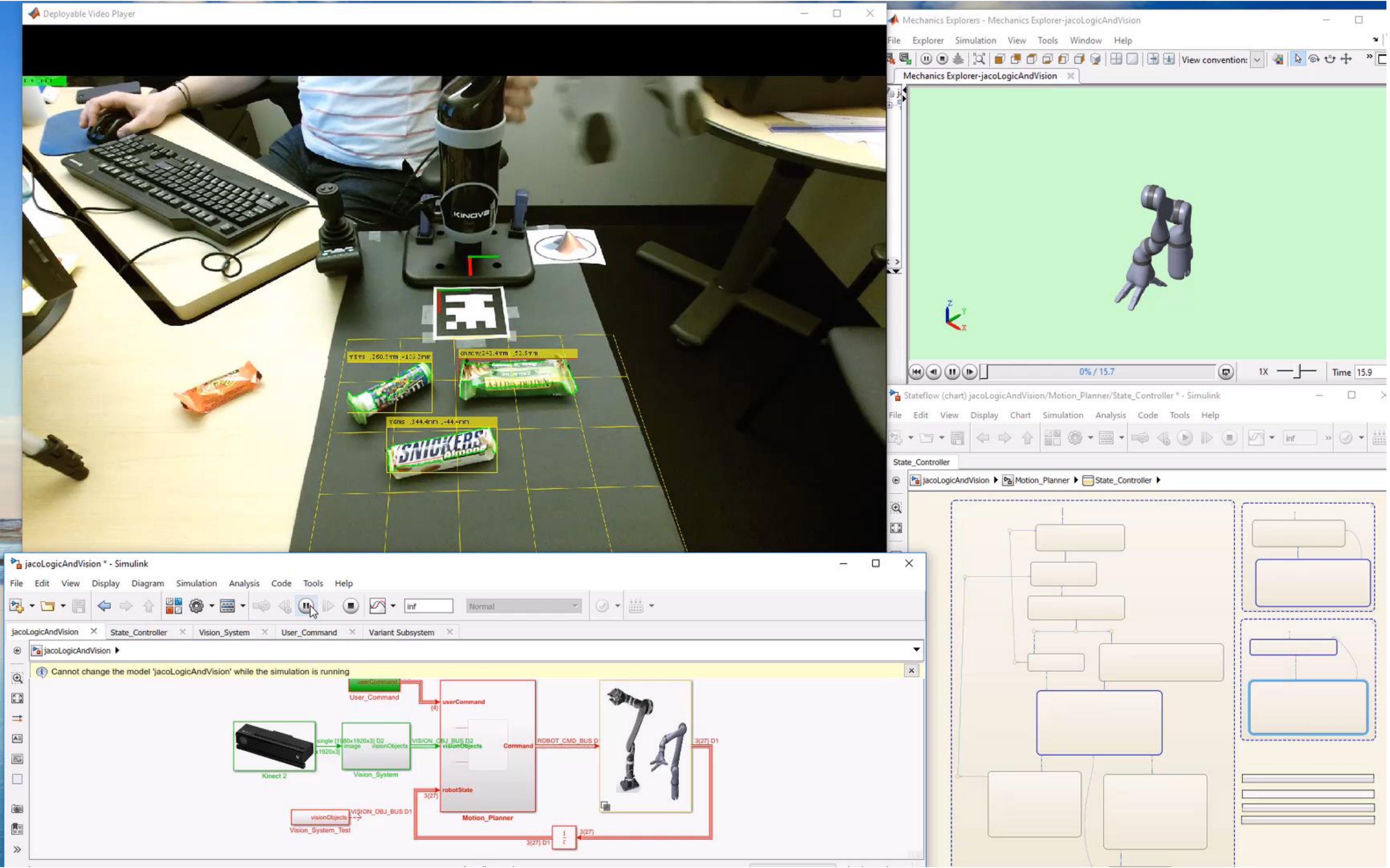
控制:探索内置函数：反向运动学

```
% 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);
```

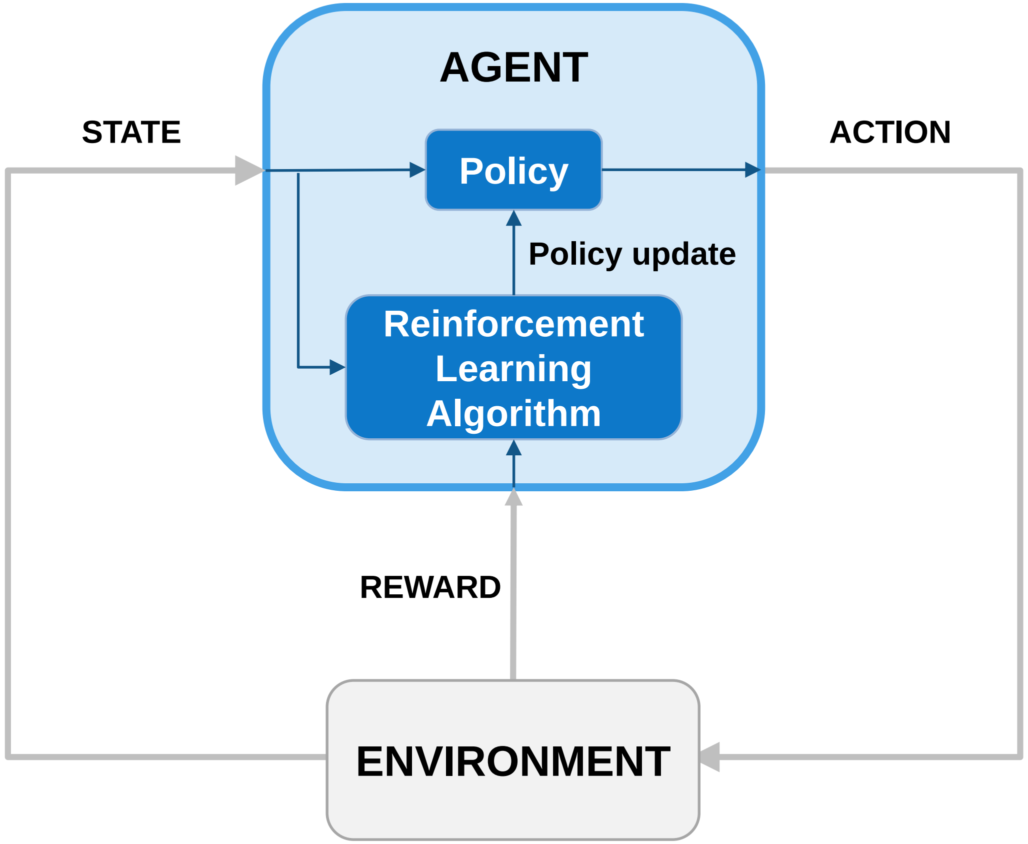


Demo

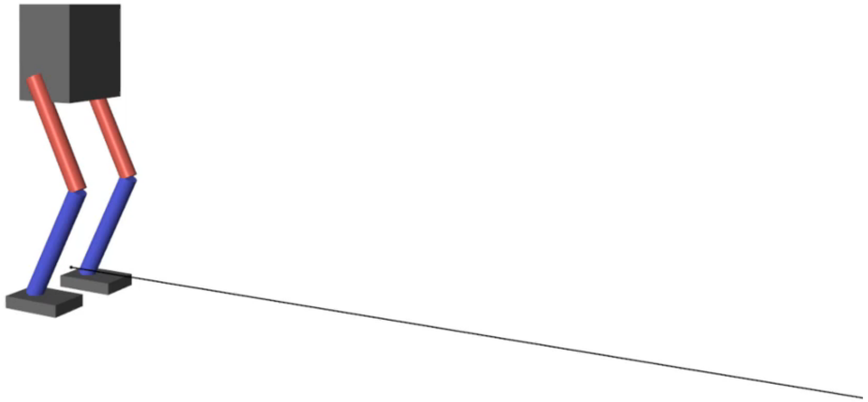
MathWorks®


Deployable Video Player
KINOVA
SNICKERS
Mechanics Explorers - Mechanics Explorer-jacoLogicAndVision
File Explorer Simulation View Tools Window Help
View convention:
Mechanics Explorer-jacoLogicAndVision
0% / 15.7
1X
Time 15.9
Stateflow (chart) jacoLogicAndVision/Motion_Planner/State_Controller * - Simulink
File Edit View Display Chart Simulation Analysis Code Tools Help
State_Controller
jacoLogicAndVision
Motion_Planner
State_Controller
jacoLogicAndVision * - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
inf
Normal
jacoLogicAndVision
State_Controller
Vision_System
User_Command
Variant Subsystem
Cannot change the model 'jacoLogicAndVision' while the simulation is running
userCommand
User_Command
Kinect 2
Vision_System
visionObjects
Command
ROBOT_CMD_BUS D1
robotState
Motion_Planner
VISION_OBJ_BUS D1
Vision_System_Test

高级控制: 强化学习



New in R2019a!



总结

成功开发一个自主机器人系统需要：

- 多域仿真
- 使用可信赖的工具，可以将复杂的工作流程简化并与其它工具集成
- 基于模型设计

德国宇航中心 (DLR) 机器人和机电中心采用基于模型设计开发自主类人机器人

DLR's humanoid robot Agile Justin autonomously performing a complex construction task.

Challenge

Develop control systems for a two-armed mobile humanoid robot with 53 degrees of freedom

Solution

Use Model-Based Design with MATLAB and Simulink to model the controllers and plant, generate code for HIL testing and real-time operation, optimize trajectories, and automate sensor calibration

Results

- Programming defects eliminated
- Complex functionality implemented in hours
- Advanced control development by students enabled

"Model-Based Design and automatic code generation enable us to cope with the complexity of Agile Justin's 53 degrees of freedom. Without Model-Based Design it would have been impossible to build the controllers for such a complex robotic system with hard real-time performance."

Berthold Bäuml

DLR

Link to user story

ClearPath Robotics为工业机器人加速算法开发

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

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."

- Ilia Baranov, Clearpath Robotics

Link to user story

Voyage为自动驾驶出租车开发纵向控制

Challenge

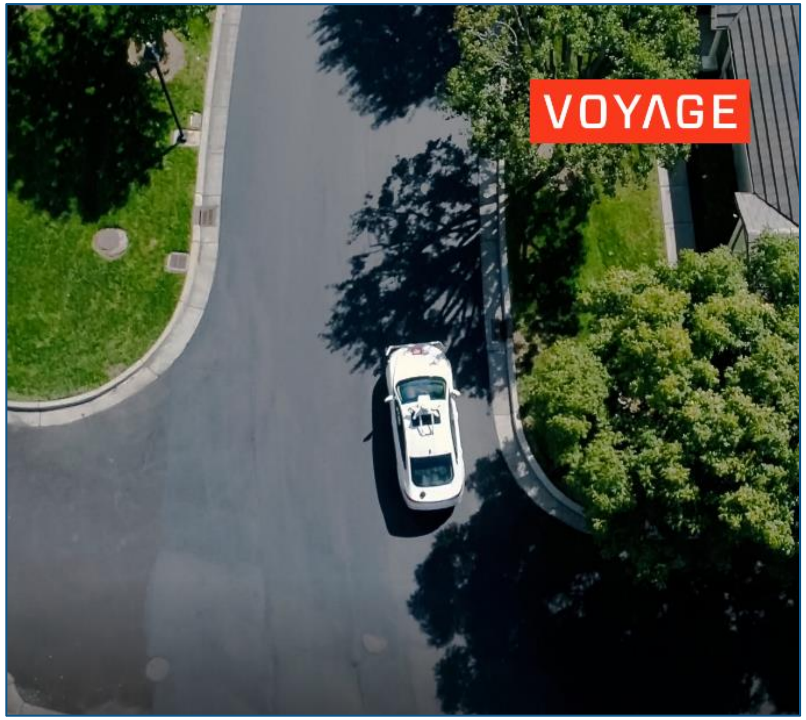
Develop a controller for a self-driving car to follow a target velocity and maintain a safe distance from obstacles

Solution

Use Simulink to design a longitudinal model predictive controller and tuned parameters based on experimental data imported into MATLAB using Robotics System Toolbox. Deploy the controller as a ROS node using Robotics System Toolbox. Generate source code using Simulink Coder into a Docker Container.

Results

- Development speed tripled
- Easy integration with open-source software
- Simulink algorithms delivered as production software



Voyage's self driving car in San Jose, California.

"We were searching for a prototyping solution that was fast for development and robust for production. We decided to go with Simulink for controller development and code generation, while using MATLAB to automate development tasks."

- Alan Mond, Voyage

Link to user story

Festo采用基于模型设计开发创新型机械臂

The Festo Bionic Handling Assistant. Image © Festo AG.

Challenge

Design and implement a control system for a pneumatic robotic arm

Solution

Use Simulink and Simulink PLC Coder to model, simulate, optimize, and implement the controller on a programmable logic controller

Results

- Complex PLC implementation automated
- Technology and innovation award won
- New business opportunities opened

"Using Simulink for Model-Based Design enables us to develop the sophisticated pneumatic controls required for the Bionic Handling Assistant and other mechatronic designs. With Simulink PLC Coder, it is now much easier to get from a design to a product."

Dr. Rüdiger Neumann
Festo

Link to user story

% Thank you