MATLAB EXPO 2018

为机器人和自主系统开发算法

郭文彦

本次演讲的亮点

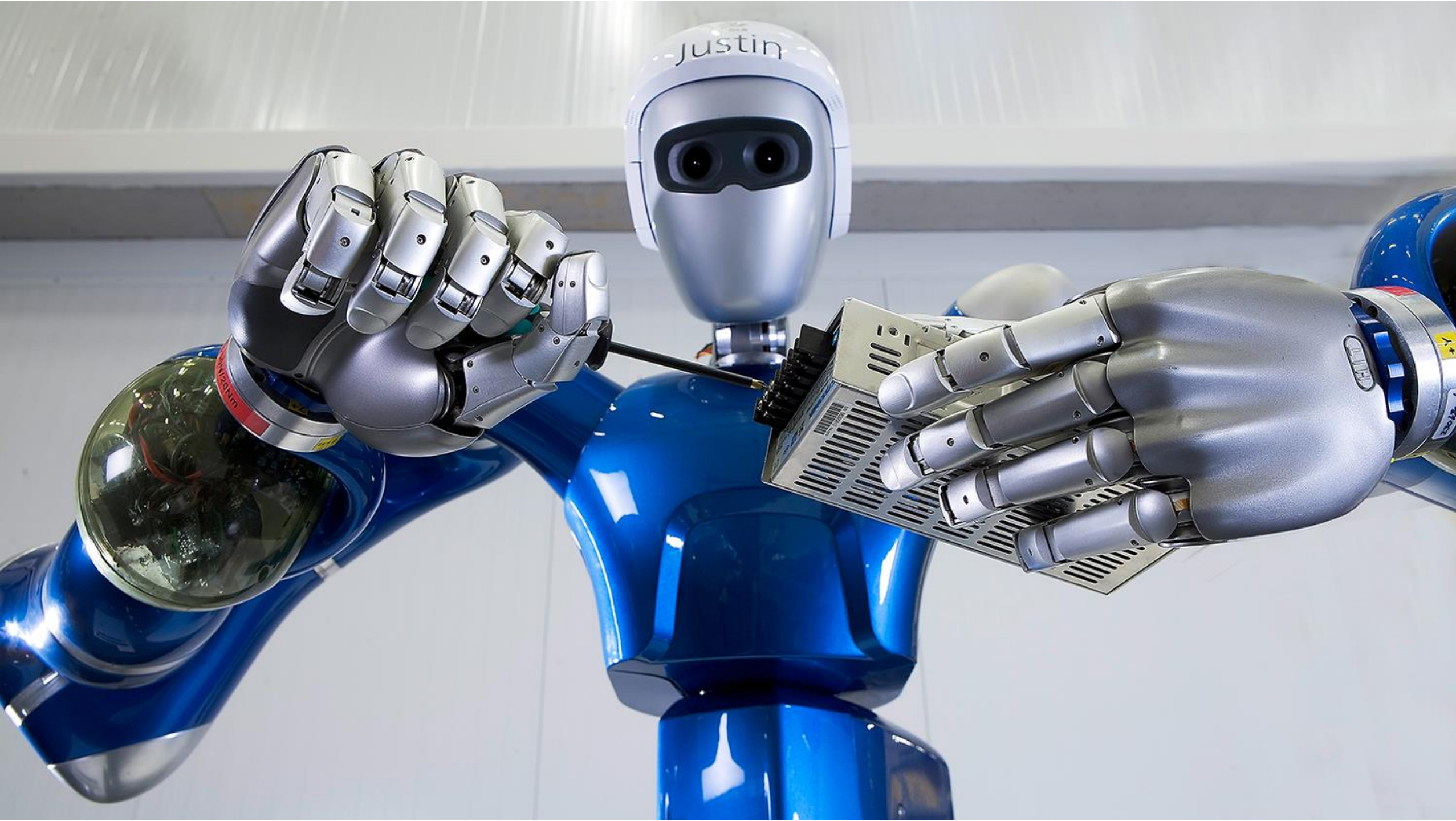
成功的开发一个自主机器人系统需要：

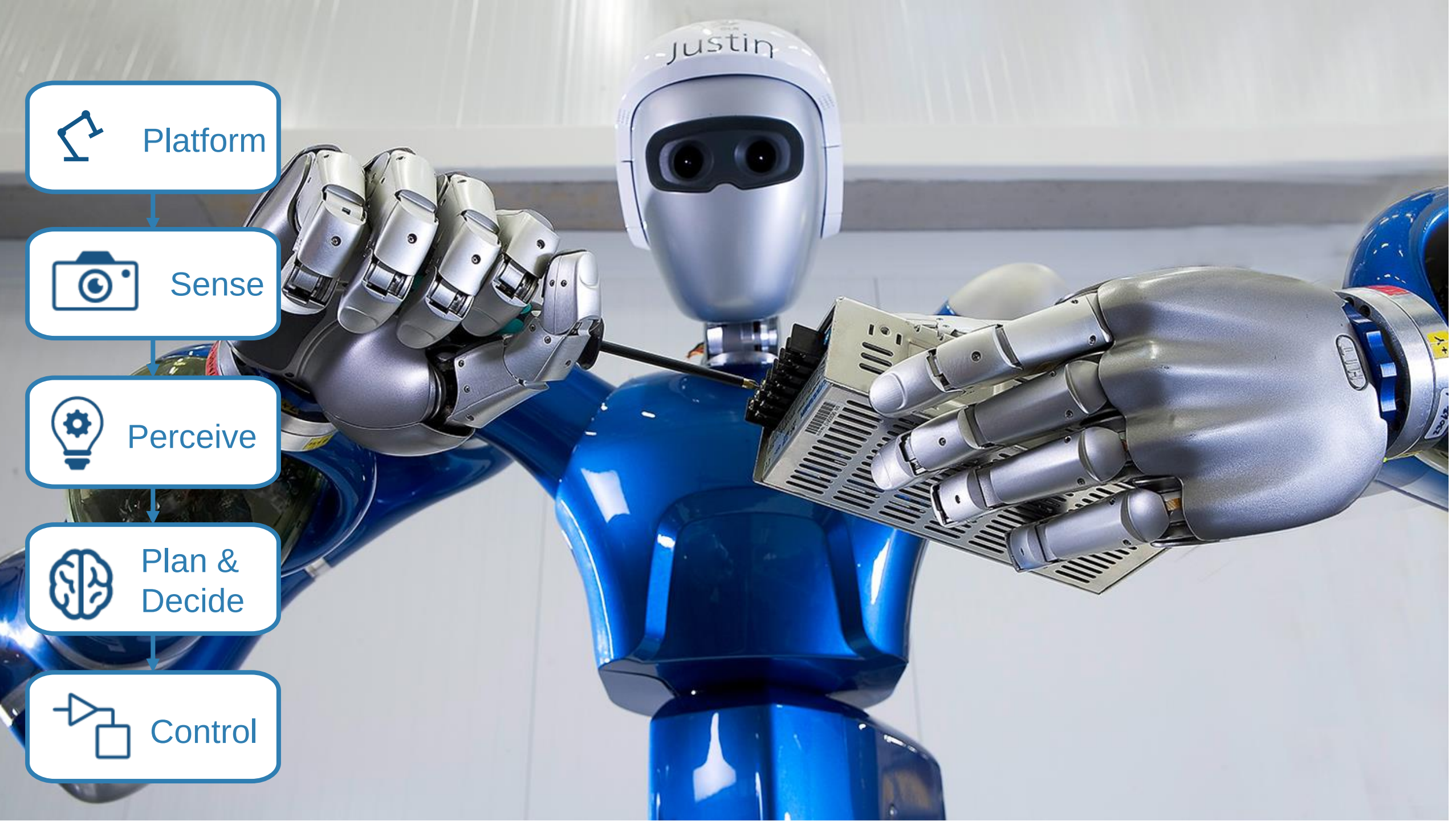
1. 多域仿真
2. 信任能够使得复杂的工作流简化并能与其他工具集成的工具
3. 基于模型设计

自主机器人系统的挑战

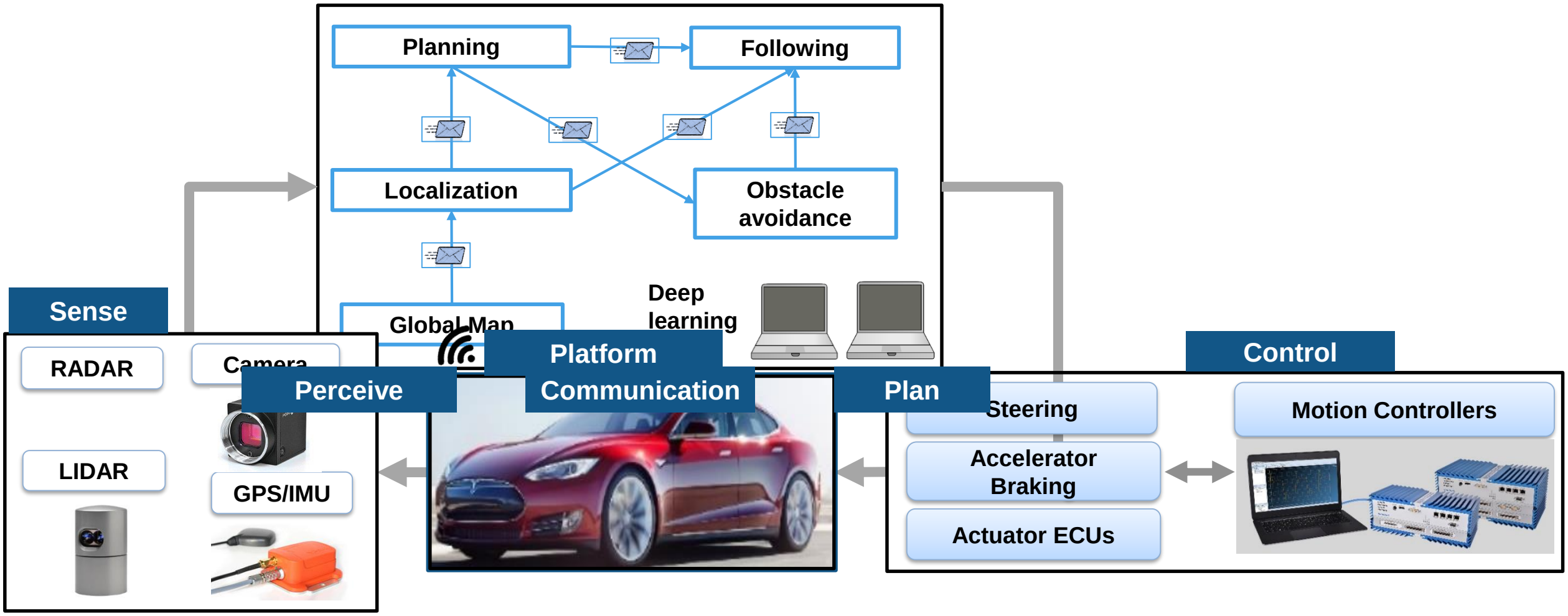
- 应用多域专业知识
- 算法的复杂性
- 端到端工作流
- 技术深度和系统稳定性
- 知识产权保护

成功应该是什么样的？


Justin

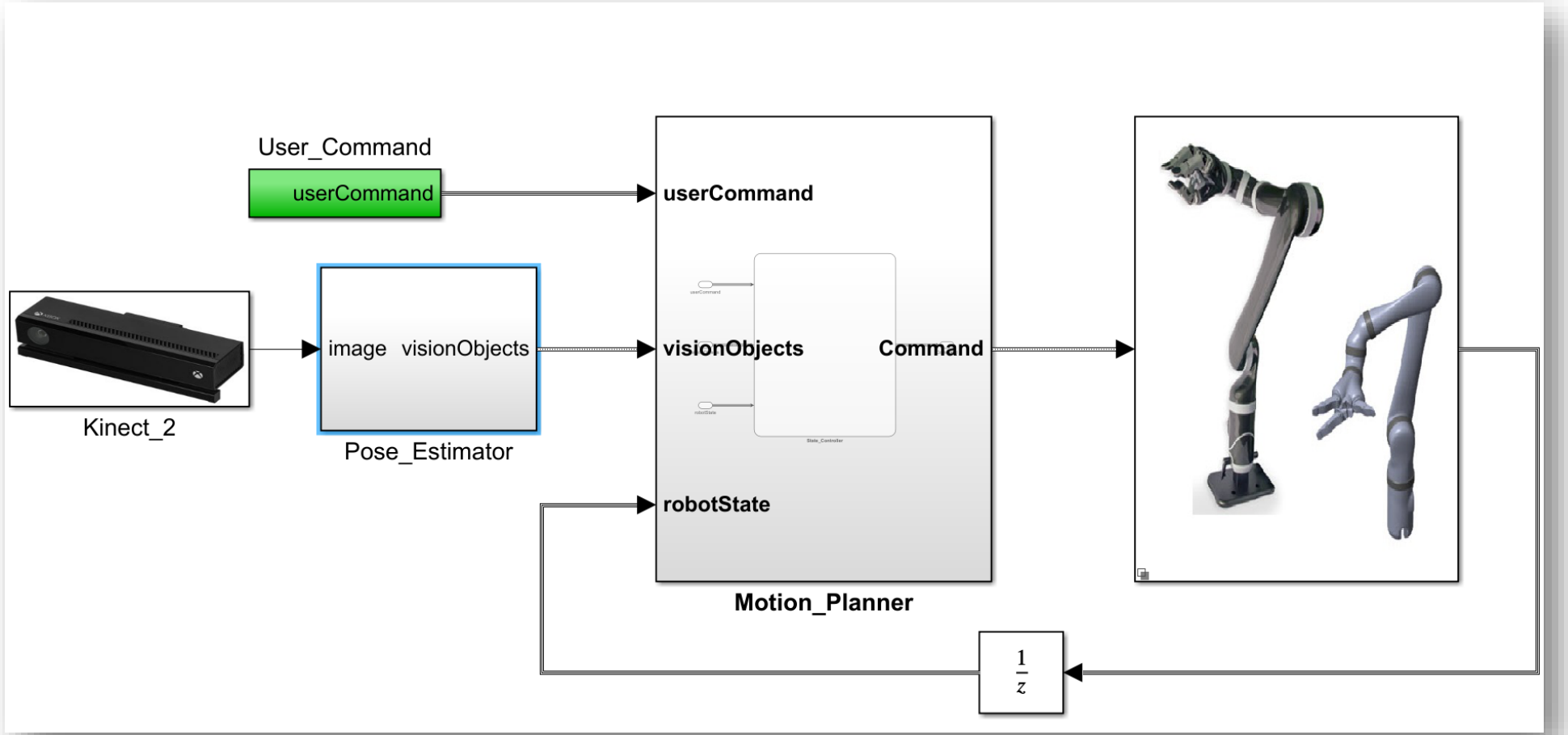

Justin
Platform
Sense
Perceive
Plan &
Decide
Control

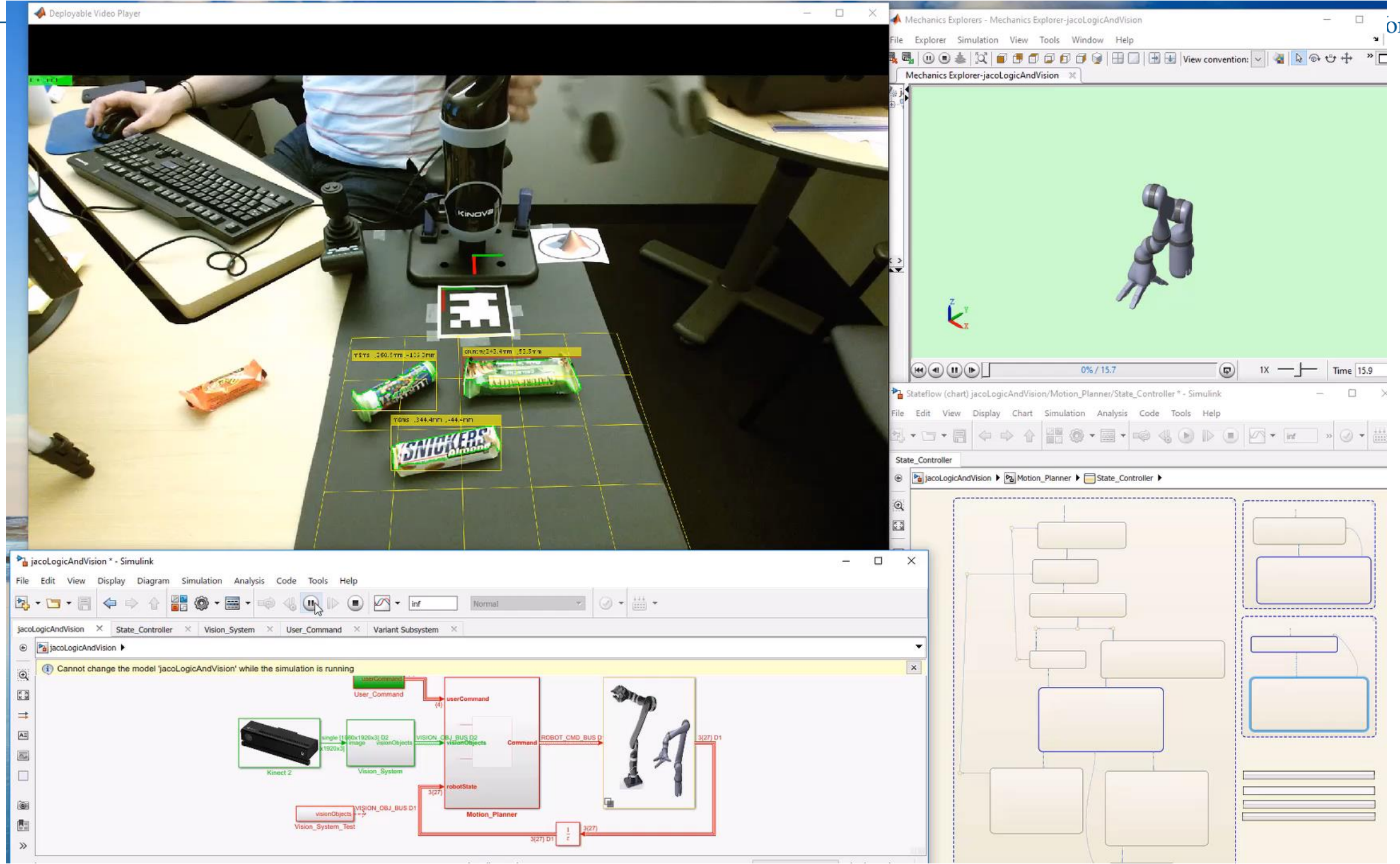
另一个例子：自主驾驶汽车



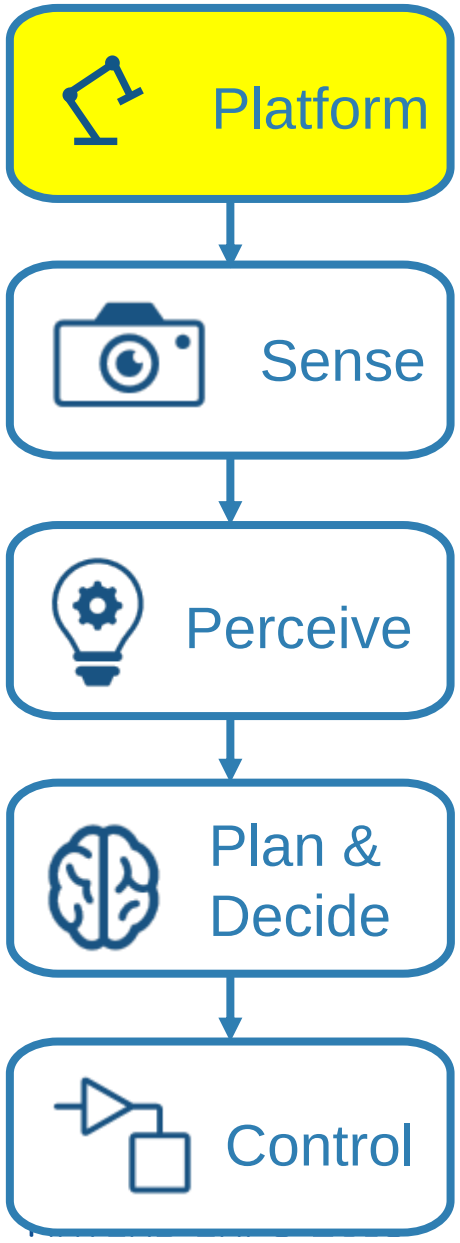
今天：设计抓取放置应用

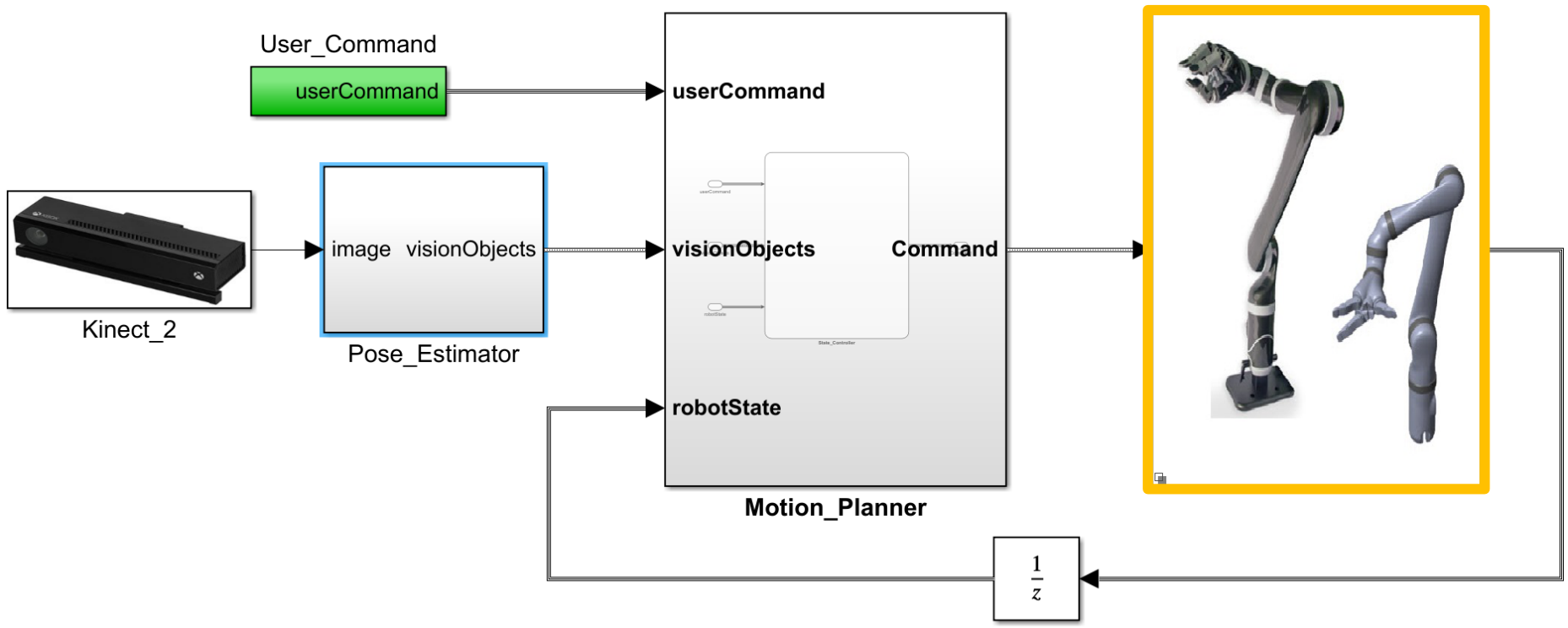
- Platform
- Sense
- Perceive
- Plan & Decide
- Control




Deployable Video Player
Mechanics Explorers - Mechanics Explorer-jacoLogicAndVision
File Explorer Simulation View Tools Window Help
Mechanics Explorer-jacoLogicAndVision
View convention:
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
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
VISION_OBJ_BUS.D2
visionObjects
Command
ROBOT_CMD_BUS.D1
3(27) D1
robotState
3(27)
Motion_Planner
visionObjects
VISION_OBJ_BUS.D1
Vision_System_Test

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

如何建立一个满足我要求的系统的模型？

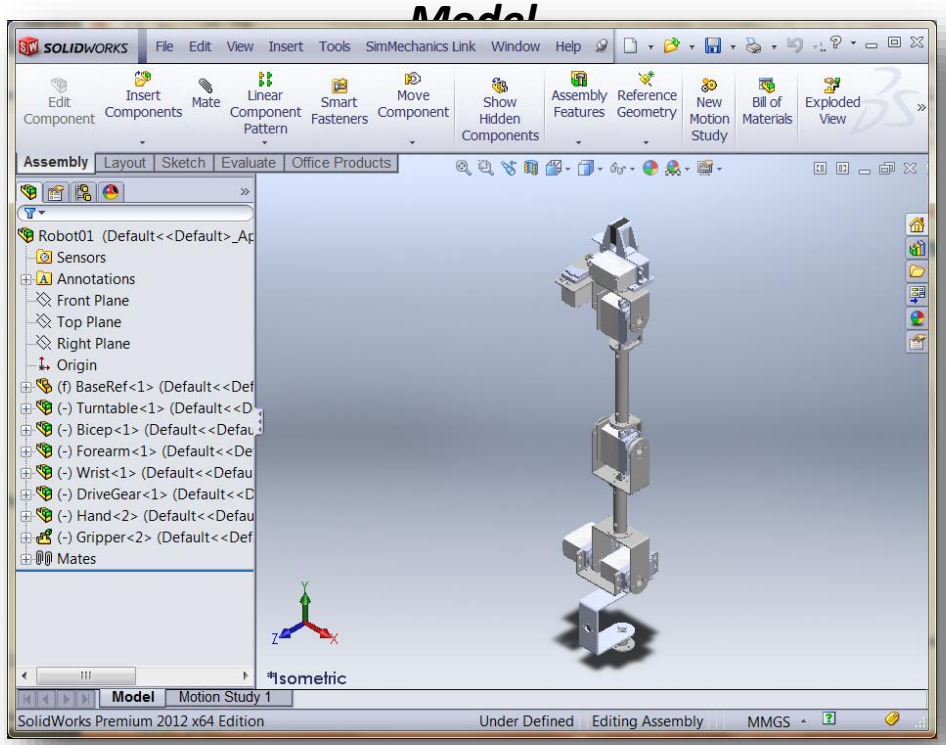
Mechanics

Actuators

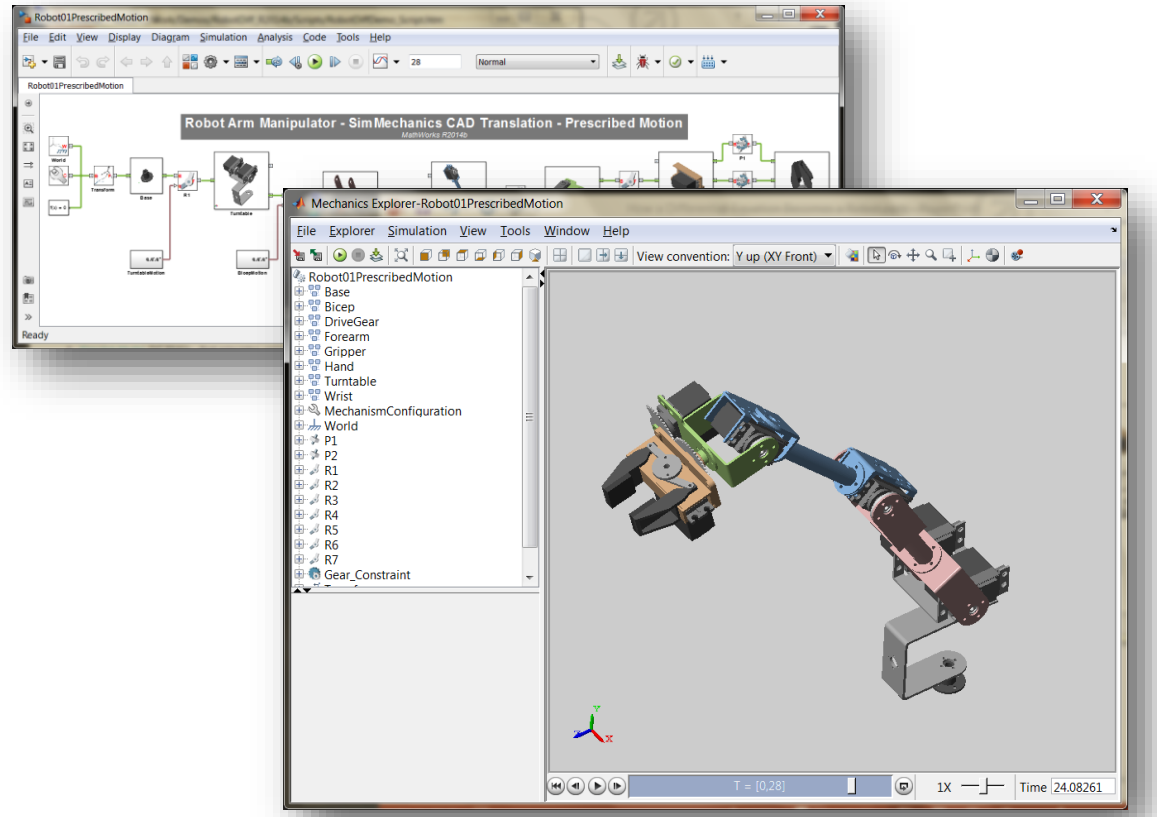
Environment

力学：从通用CAD工具导入模型

SolidWorks Model

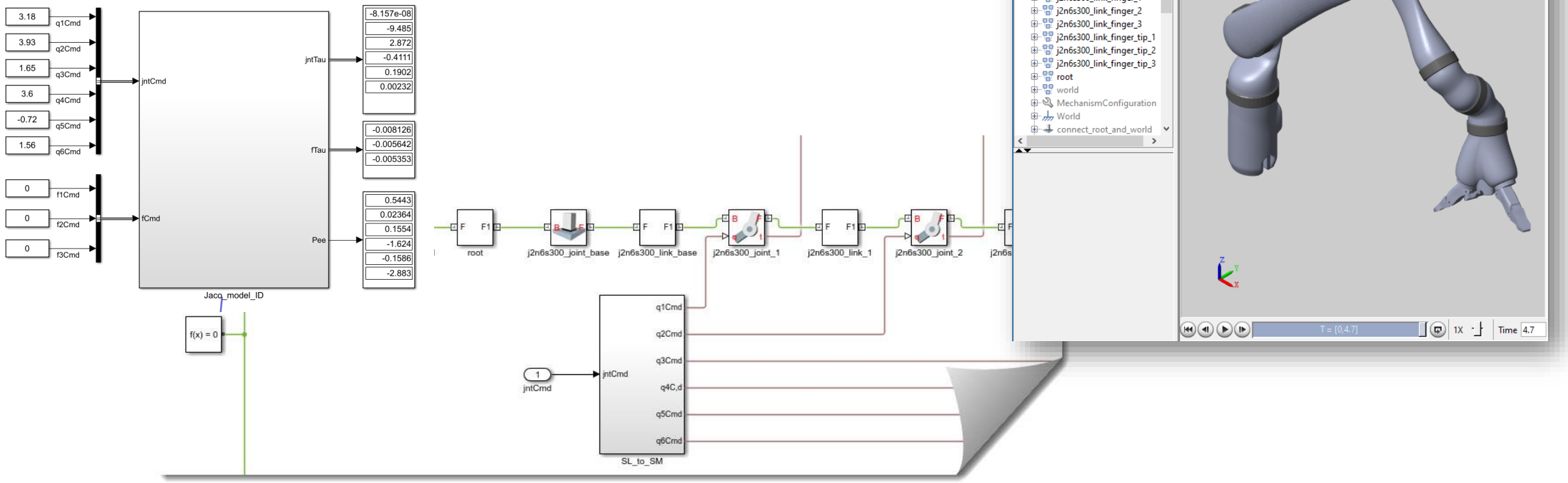


Simscape Multibody Model



力学：一条命令从URDF文件导入

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

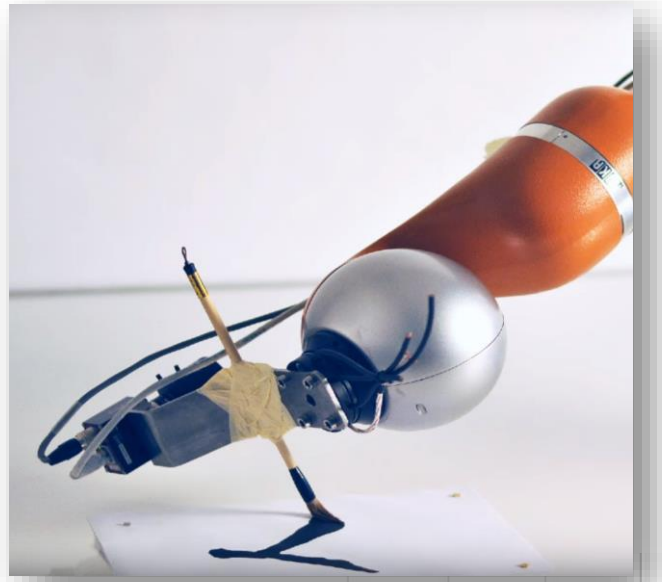


Demo

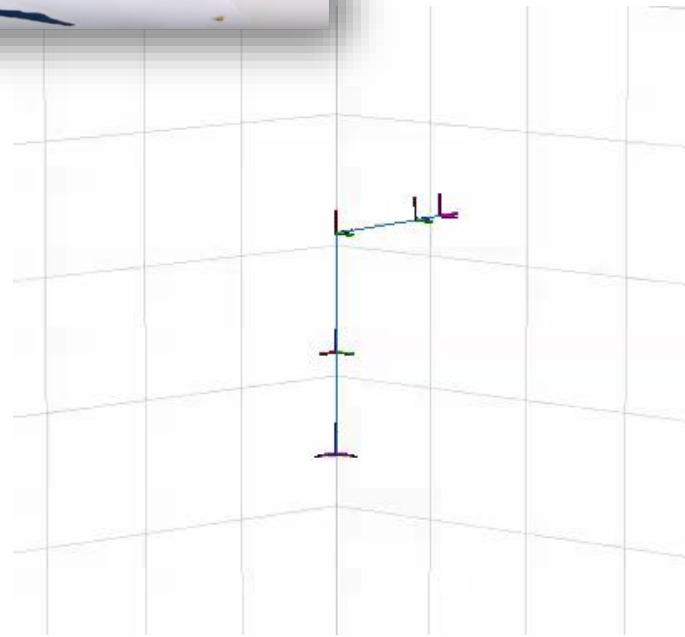
刚体树动力学

计算刚体树动力学

- 定义刚体惯性属性
- 计算刚体树
  - 前向动力学
  - 反向动力学
  - 质量矩阵
  - 速度积
  - 重力力矩
  - 重心位置和雅可比阵

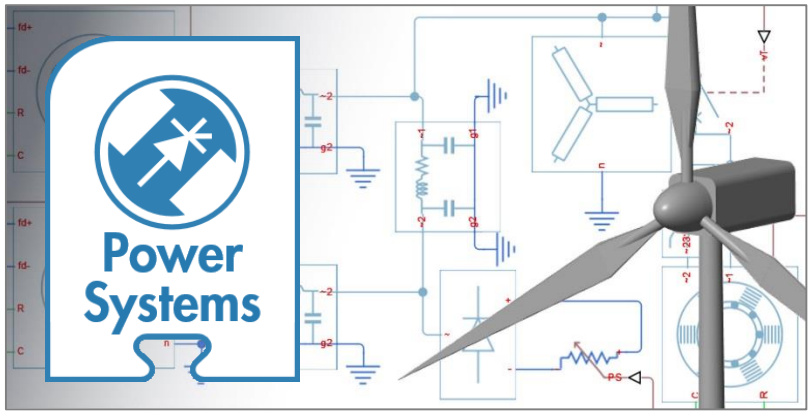


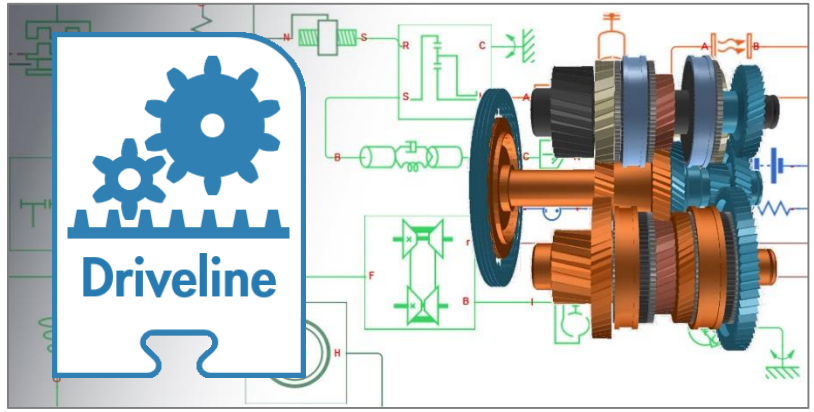
R2017a

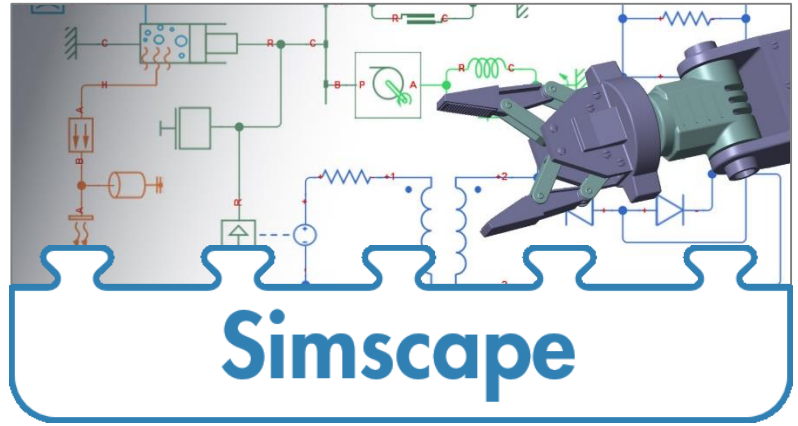


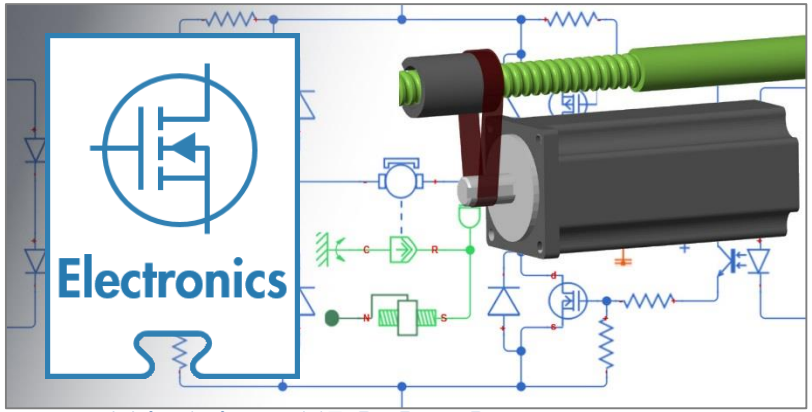
```
» load exampleRobots.mat
» lbr.DataFormat = 'column';
» q = lbr.randomConfiguration;
» tau = inverseDynamics(lbr, q);
```

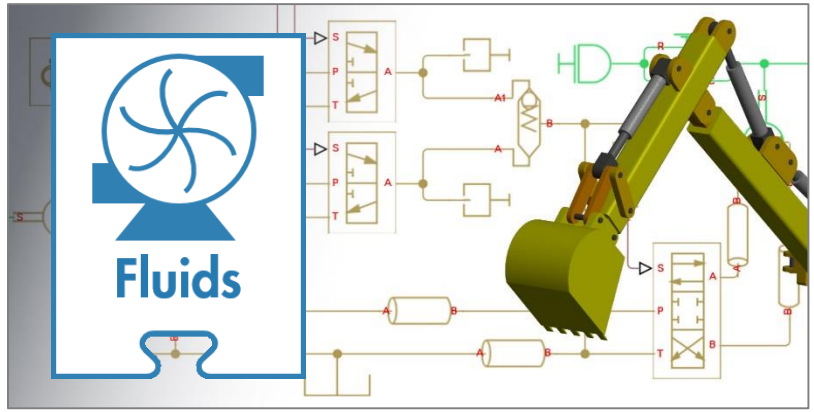
作动器：对其他域建模

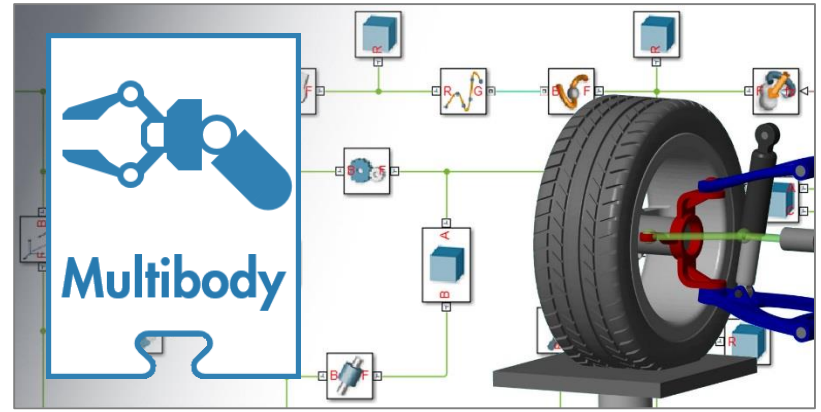

Power
Systems


Driveline

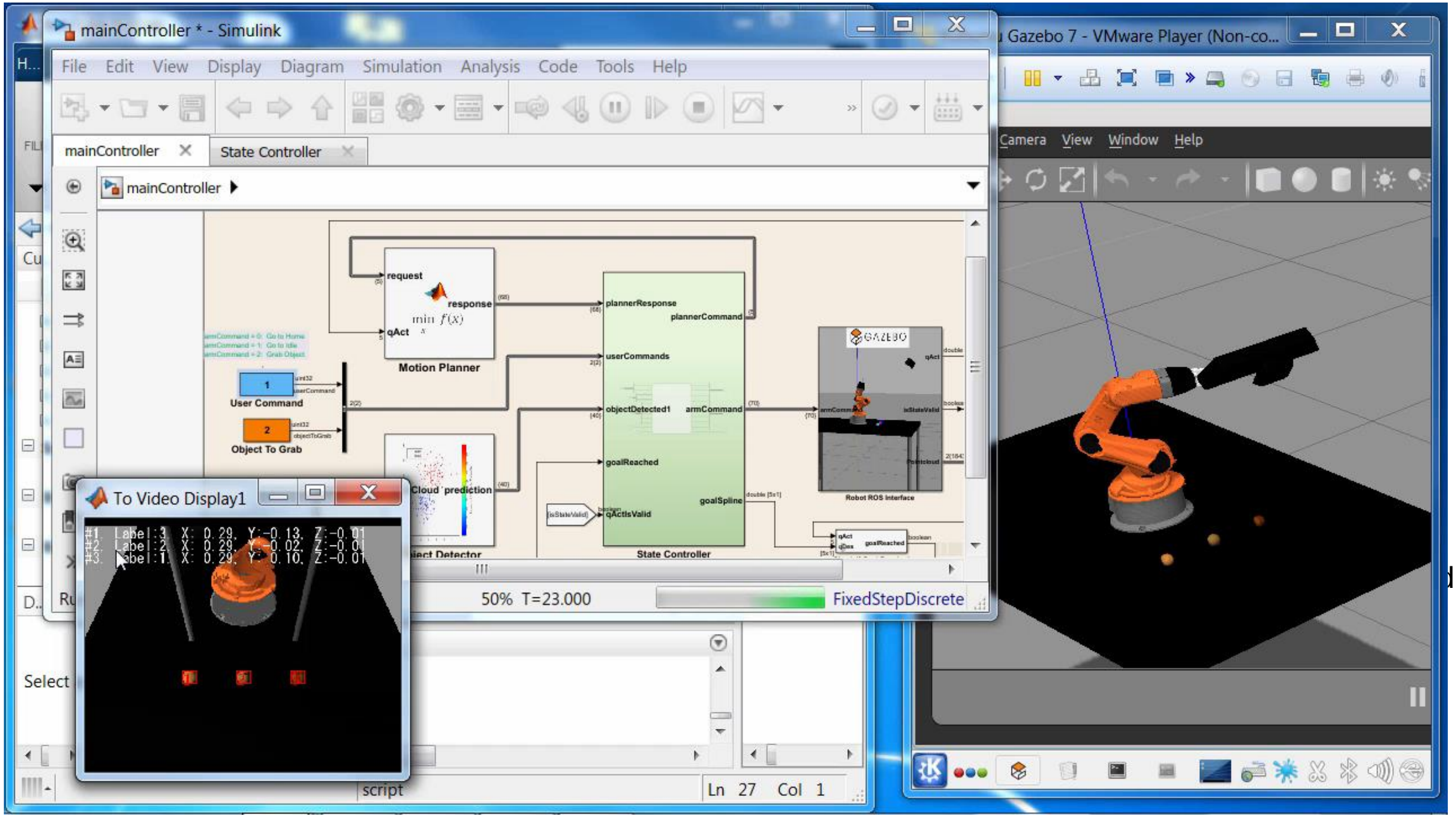

Simscape


Electronics

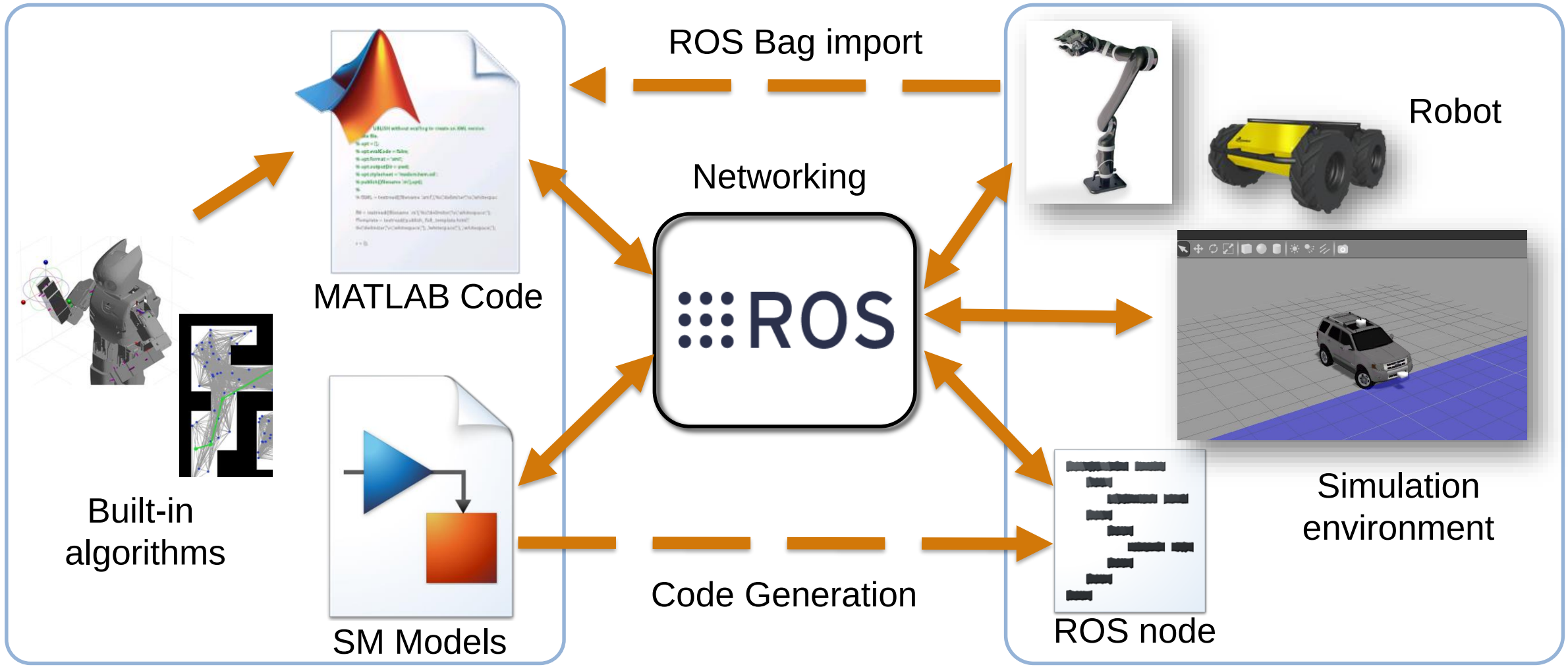

Fluids


Multibody

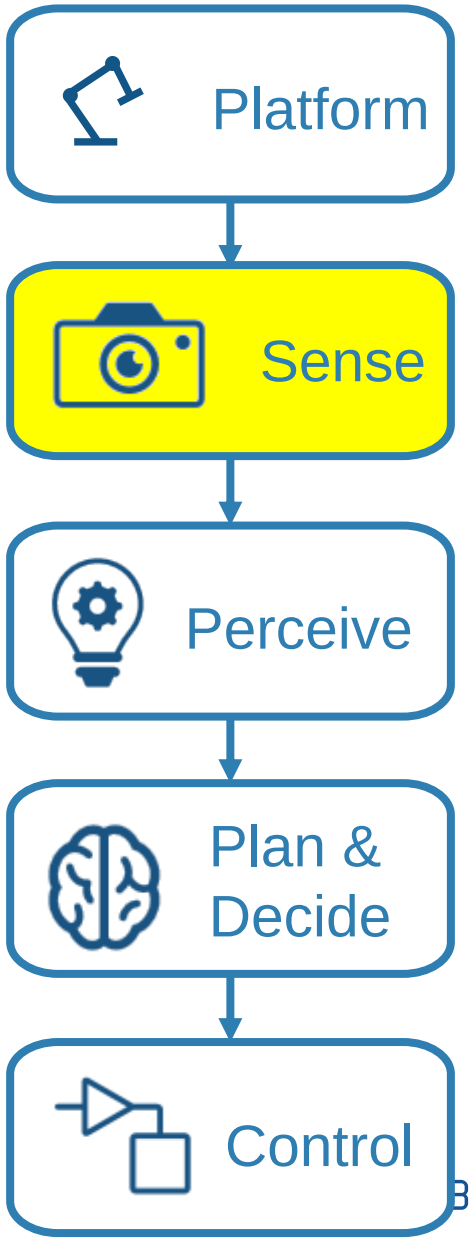
环境：与外部机器人仿真环境连接

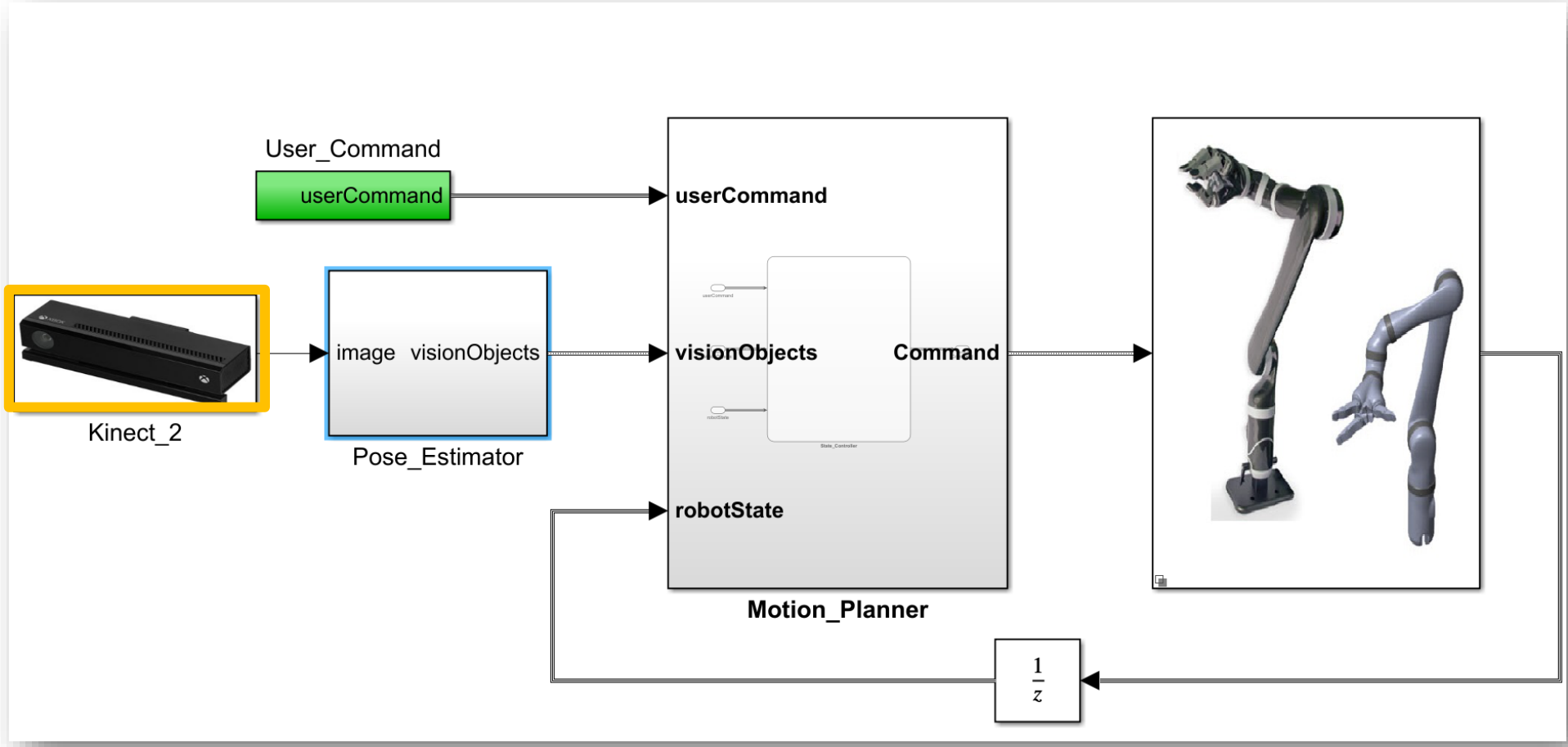


环境：通过ROS将MATLAB和Simulink连接



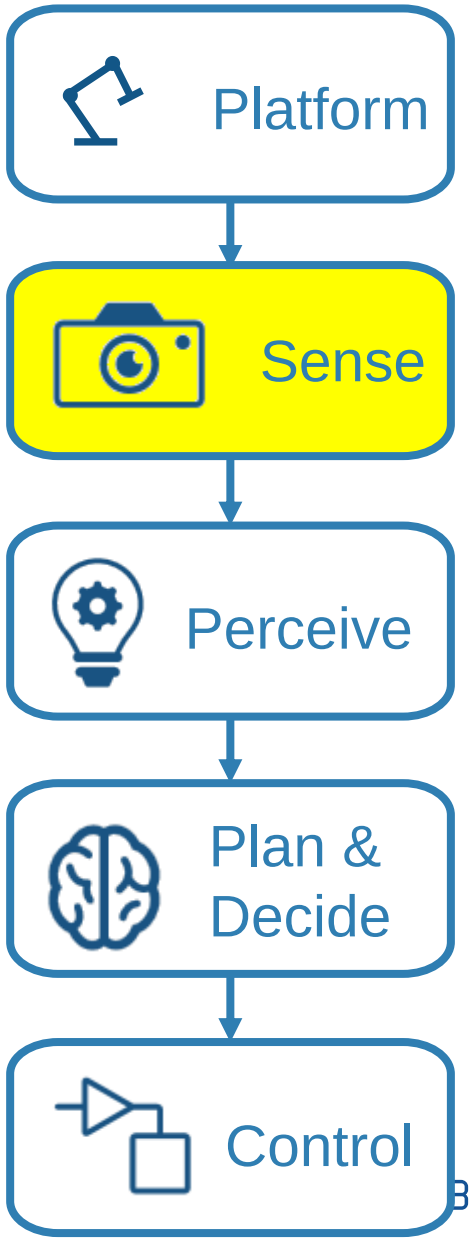
设计抓取放置应用



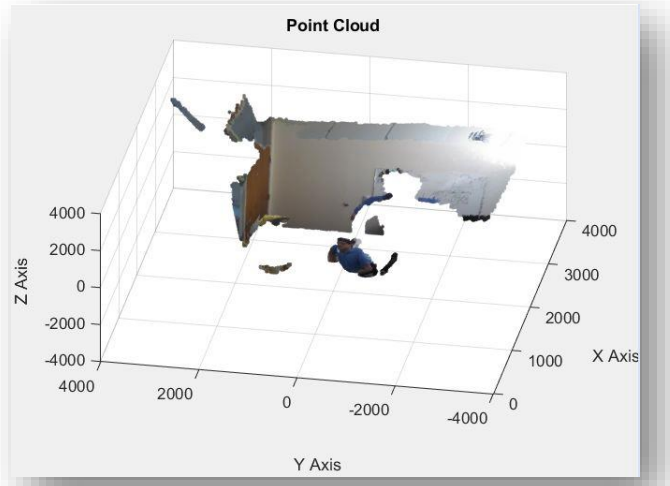


Demo

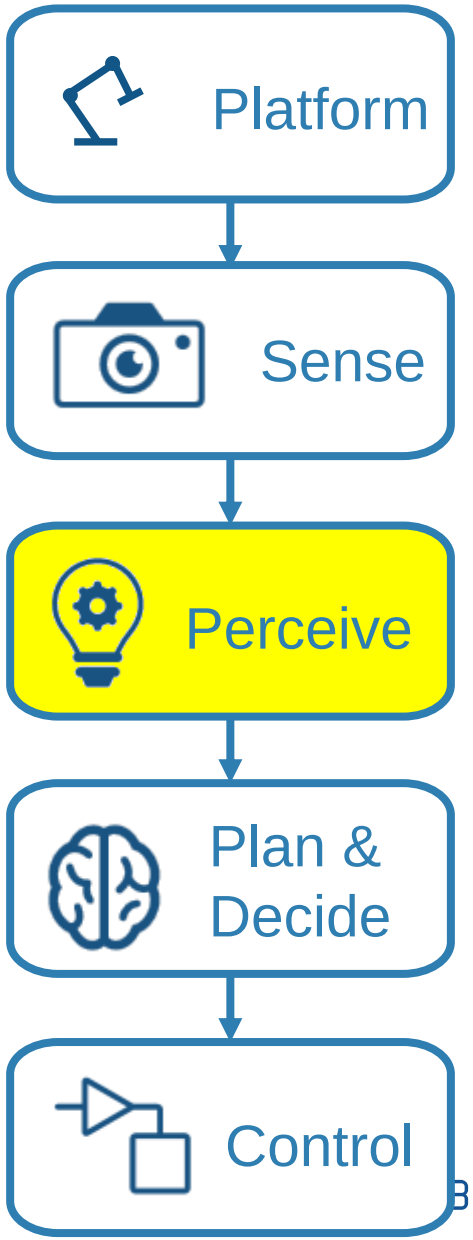
设计抓取放置应用

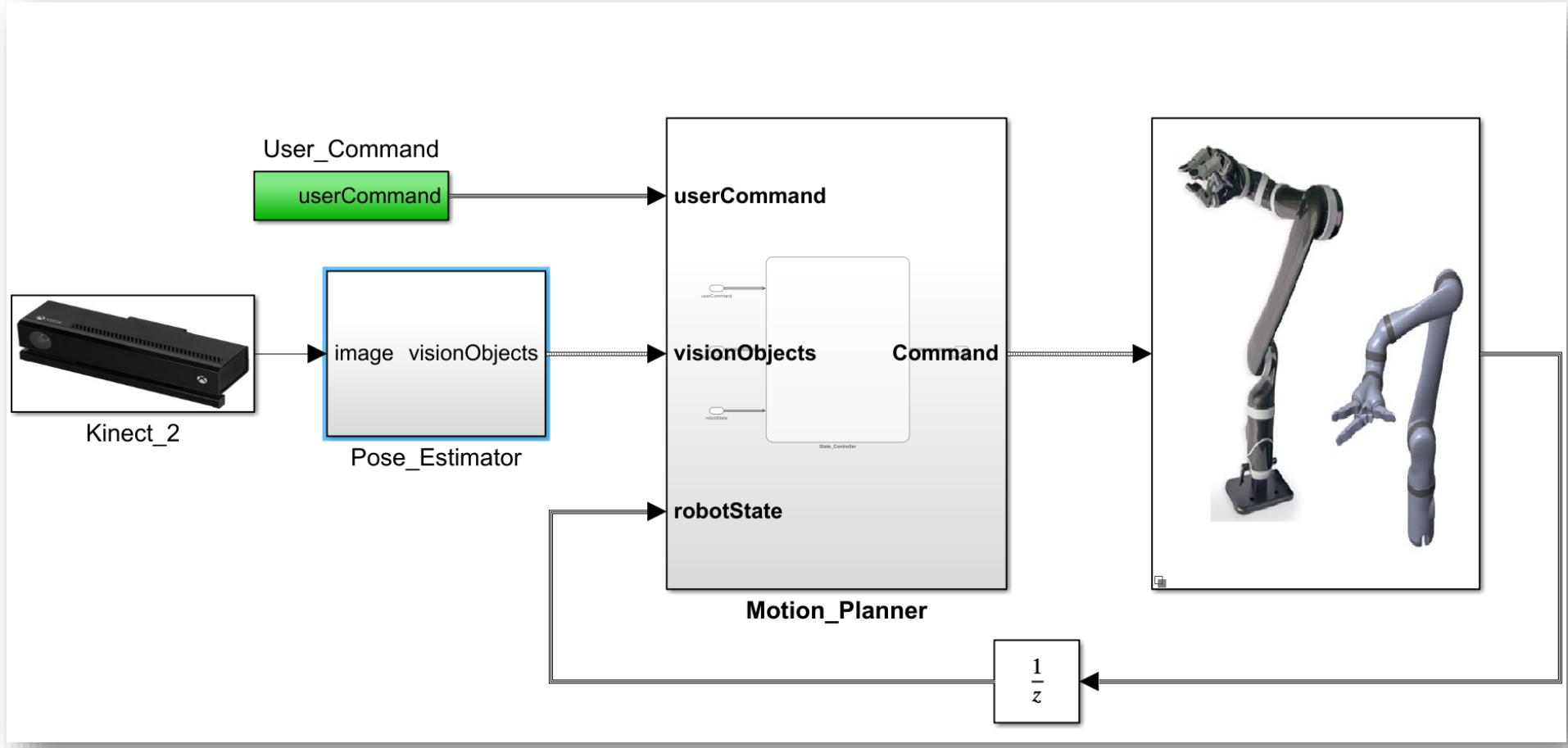


- 支持通用传感器
- 图像分析
- Apps应用
- 图像增强
- 可视化点云



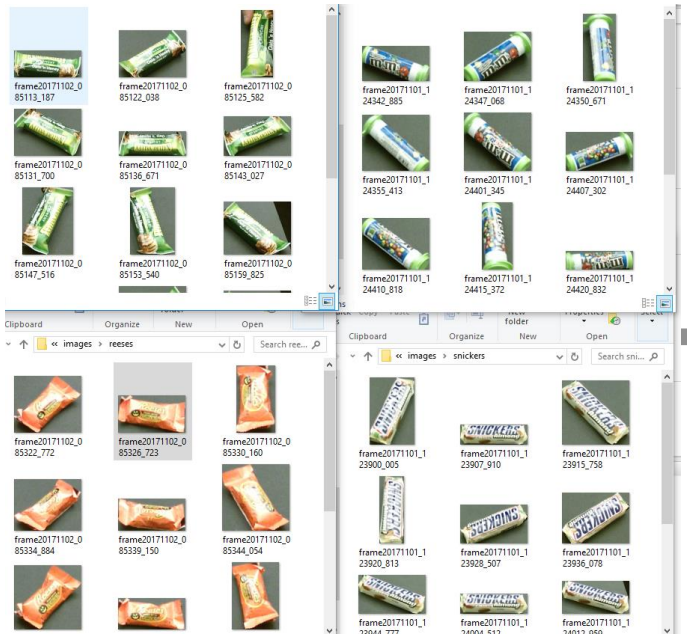
今天：设计抓取放置应用


Platform
Sense
Perceive
Plan &
Decide
Control


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

目标分类器和位姿估计

Images

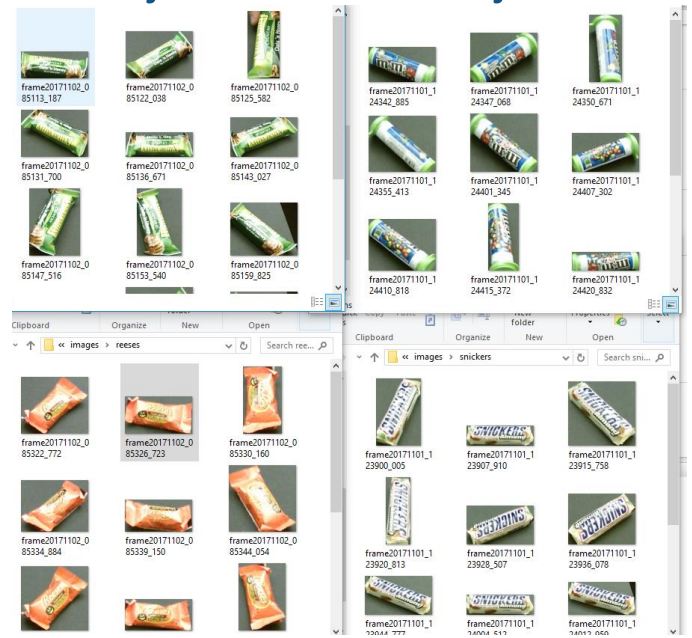


Pose Estimator

Labels and Poses

Object 1

Object 2

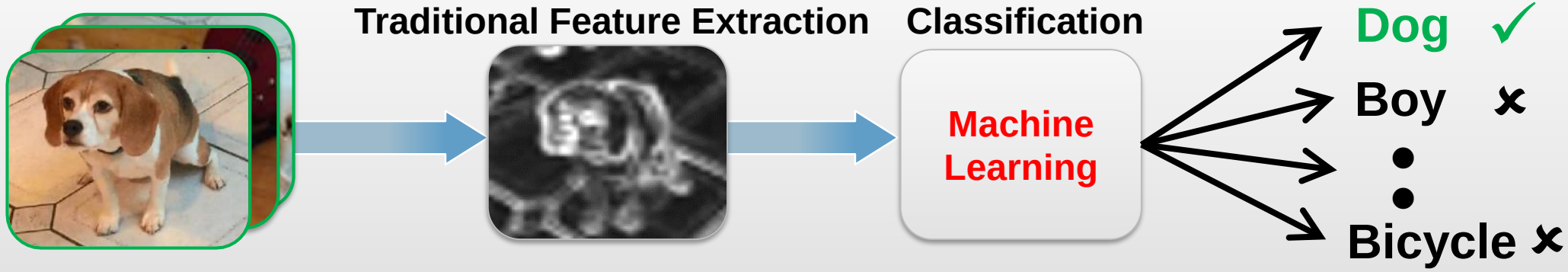


Object 3

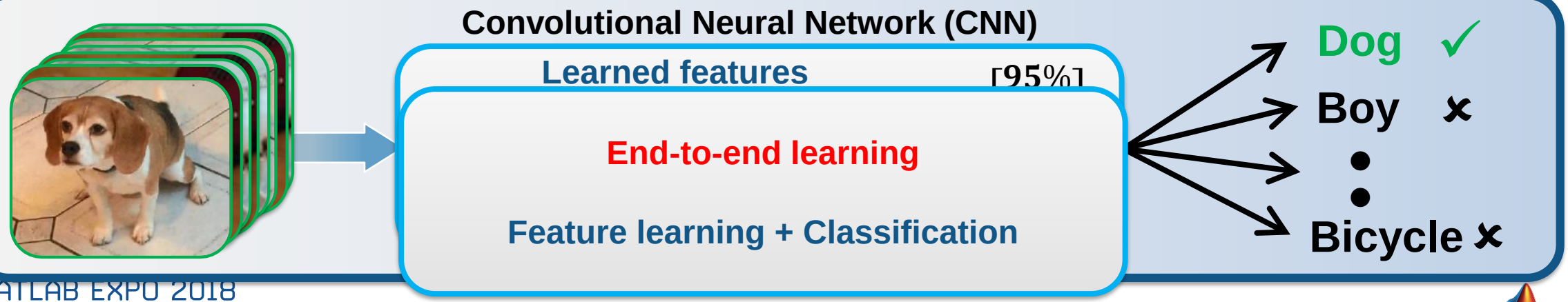
Object 4

MATLAB让机器学习简单和易用

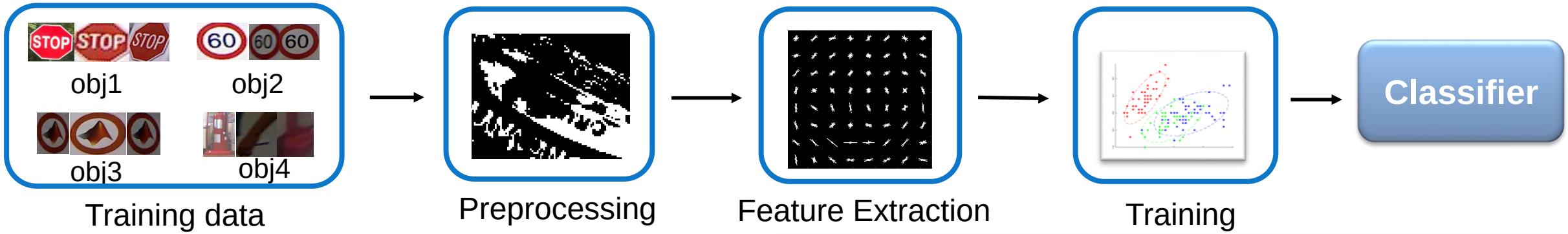
Traditional Machine Learning approach

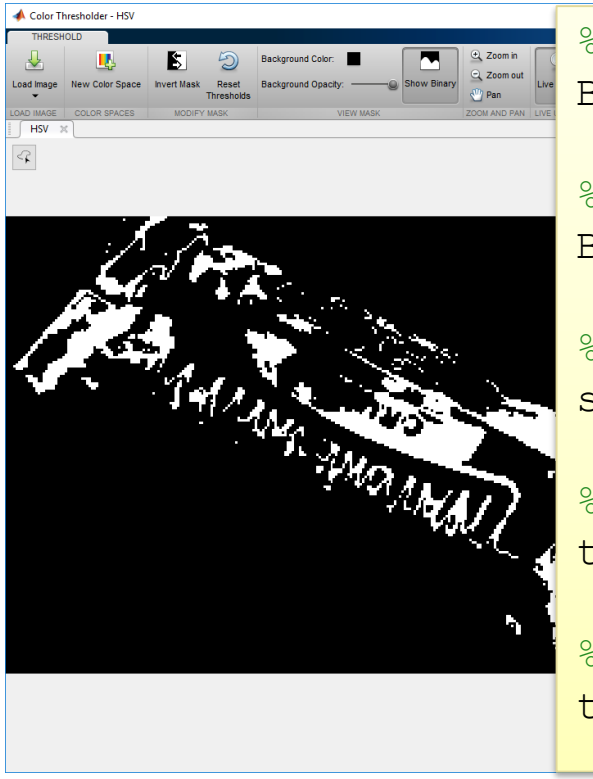


Deep Learning approach

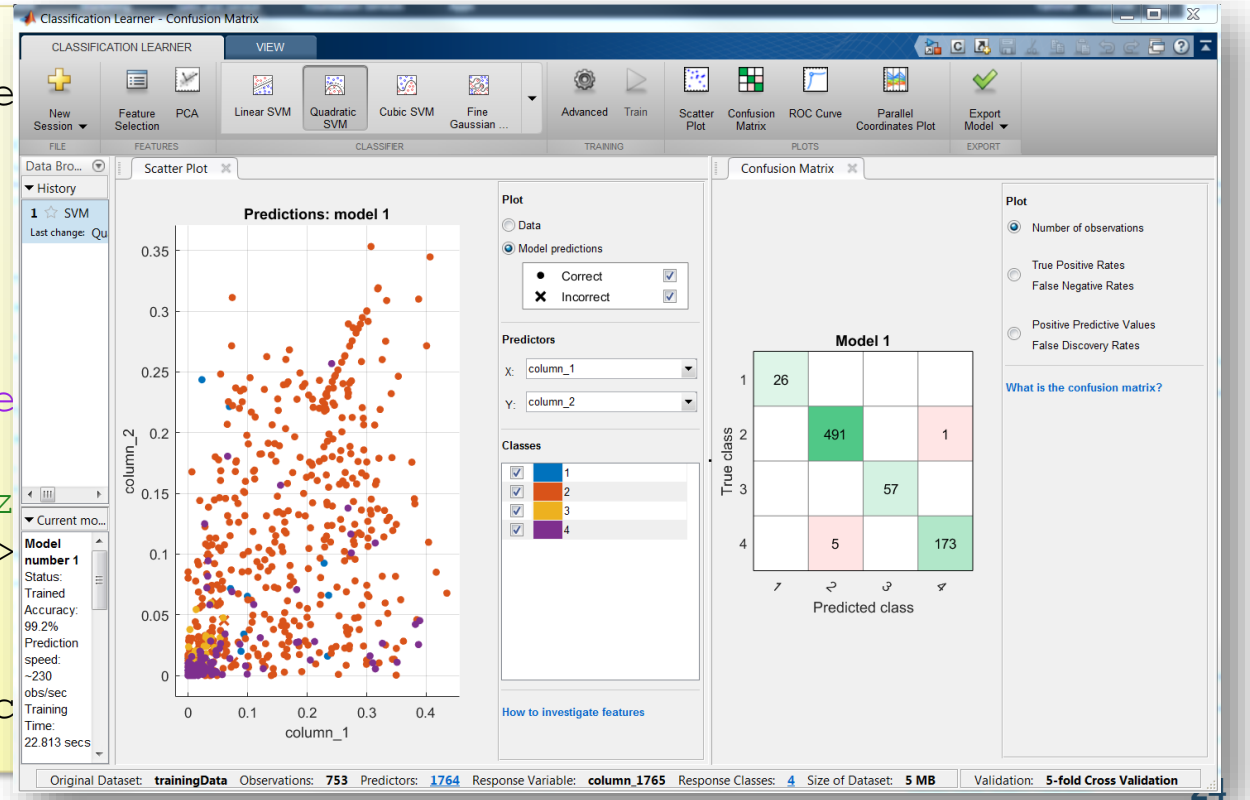


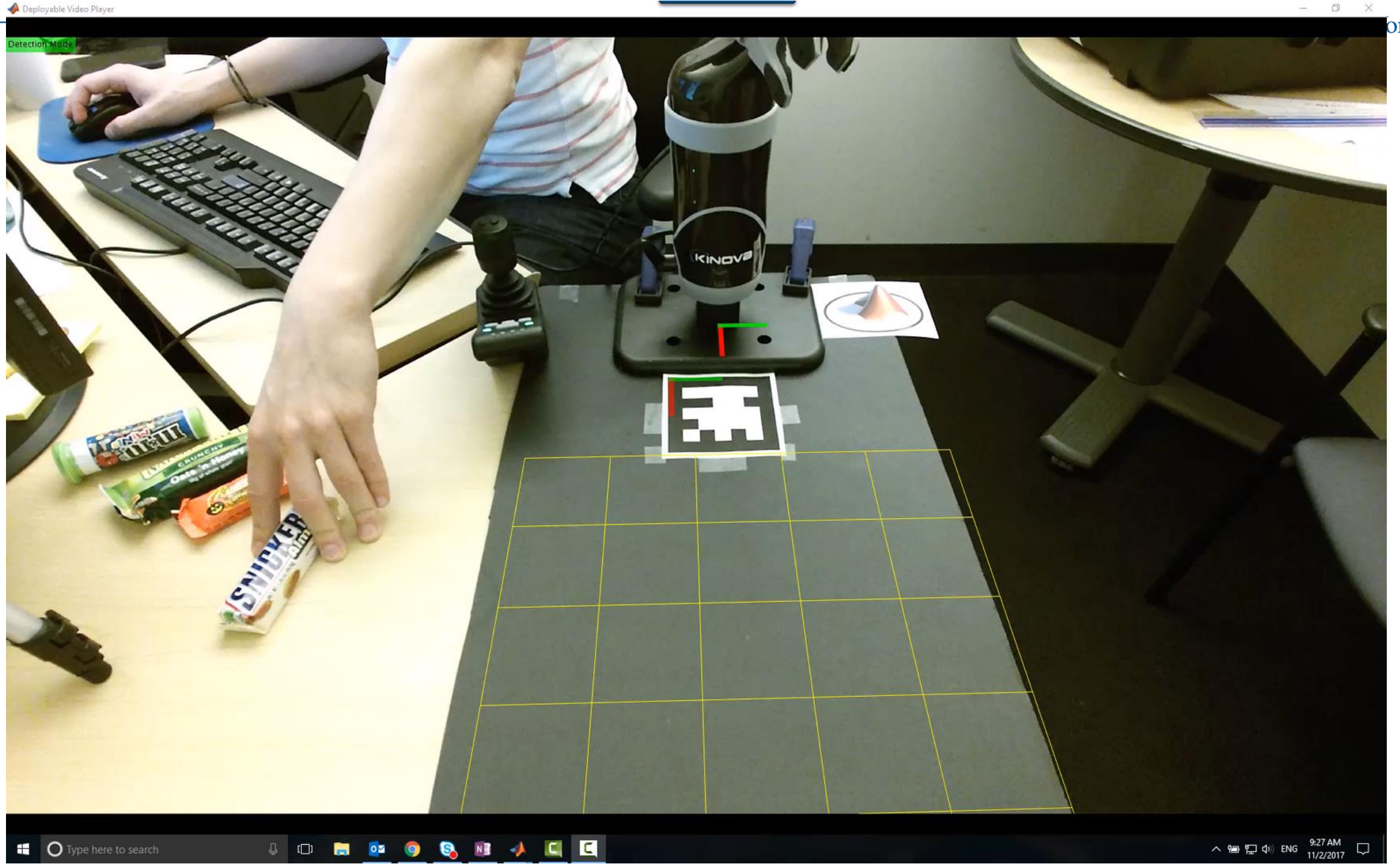
用MATLAB使复杂工作流简单



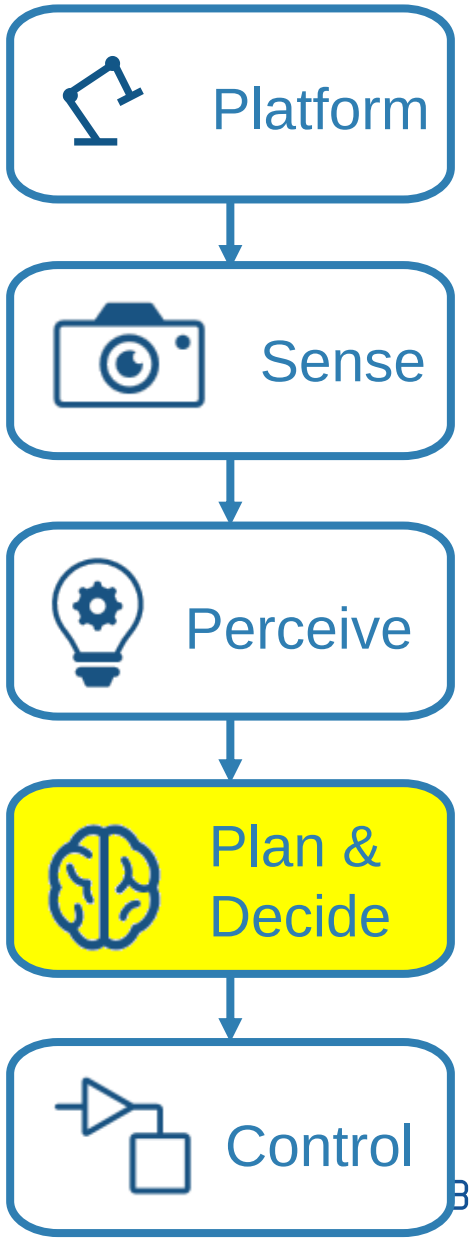


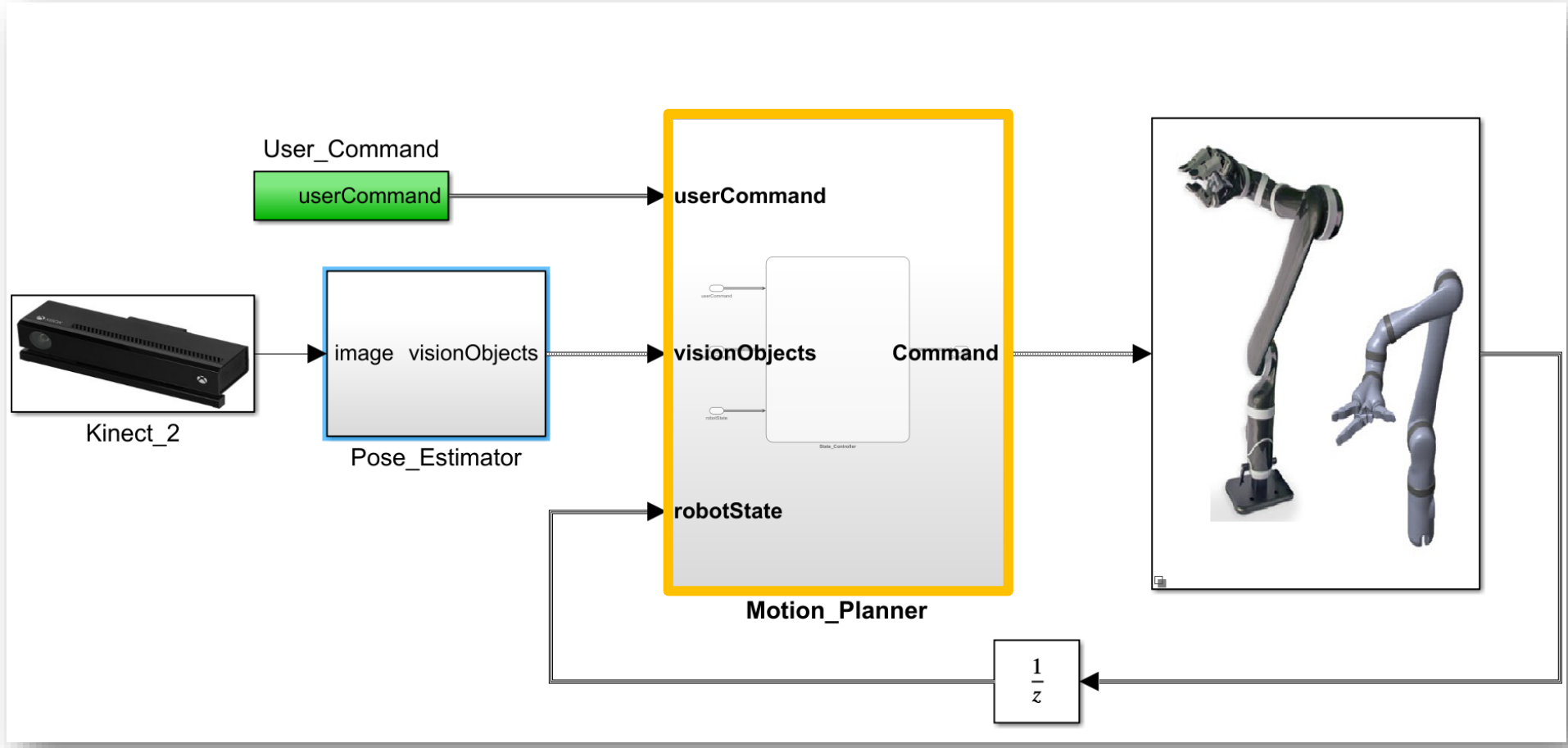

```
% Detect regions
BW = createMask(videoFrame

% 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
```



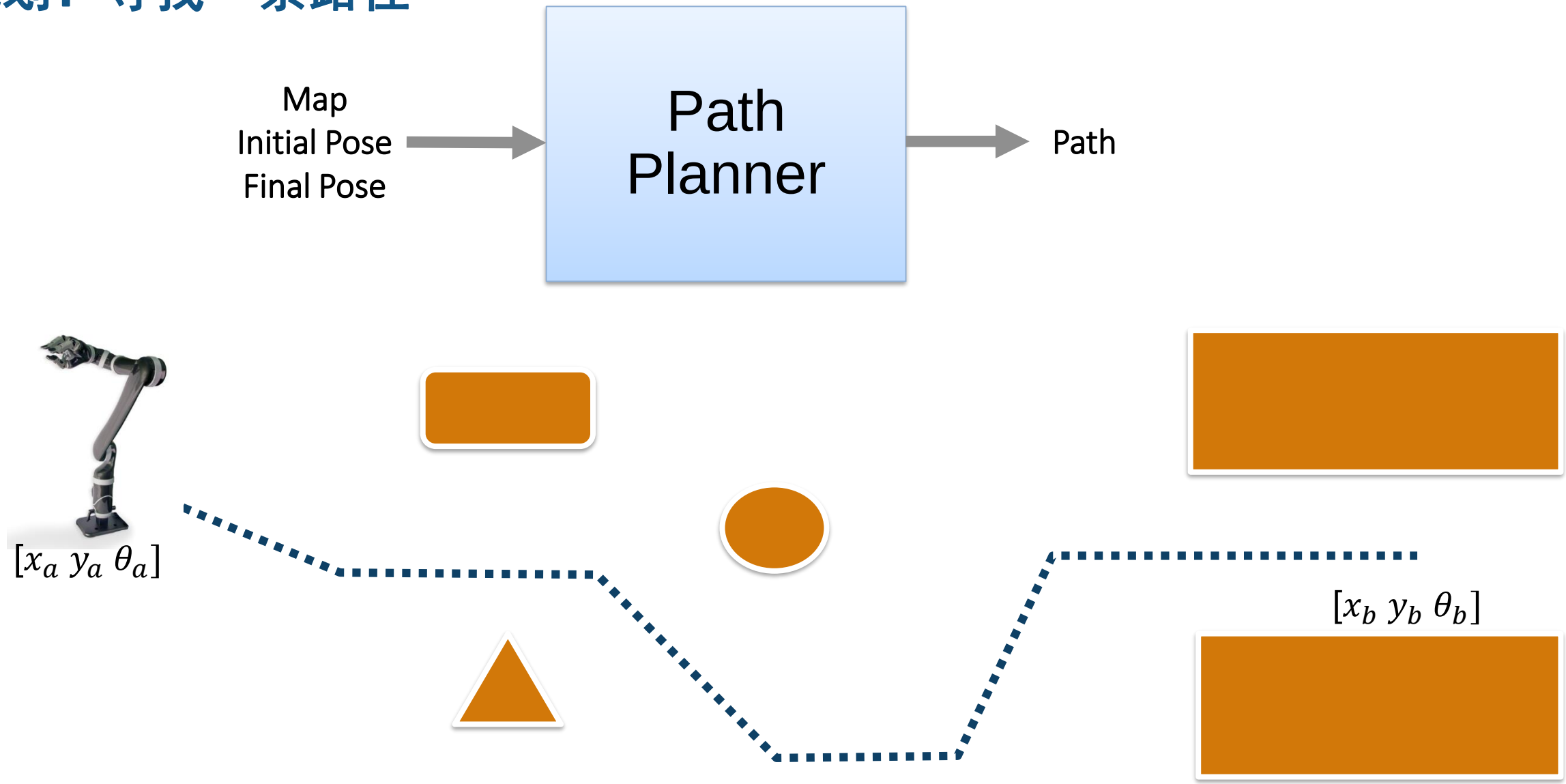

Deployable Video Player
Detection Mode
KINOVA
Type here to search
ENG
9:27 AM
11/2/2017

设计抓取放置应用

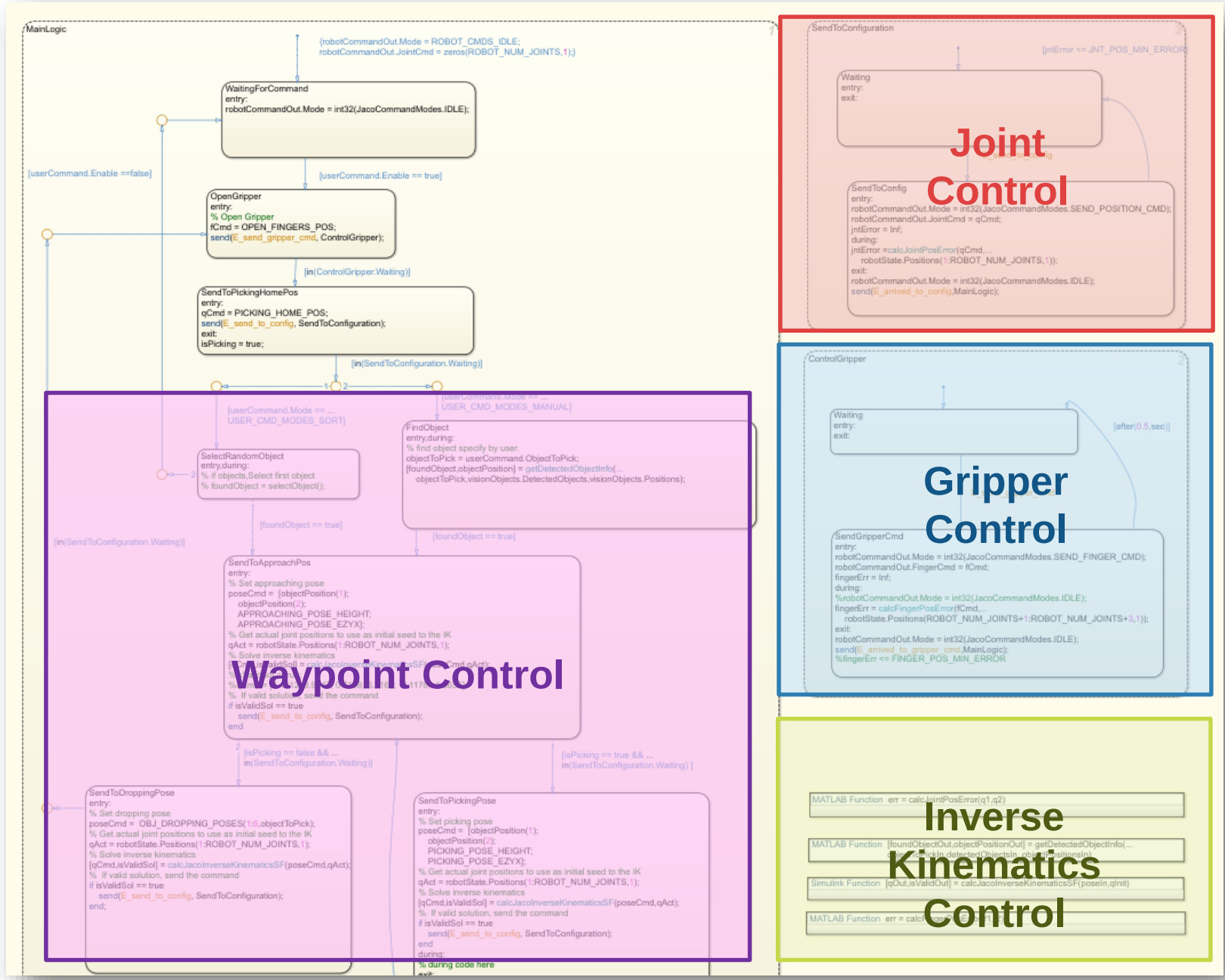

Platform
Sense
Perceive
Plan &
Decide
Control


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

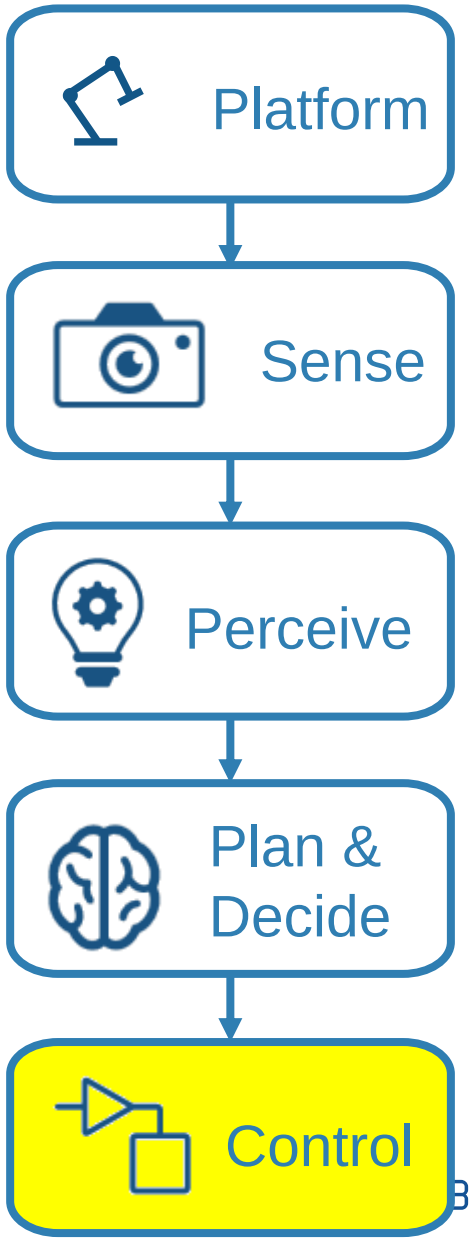
规划：寻找一条路径

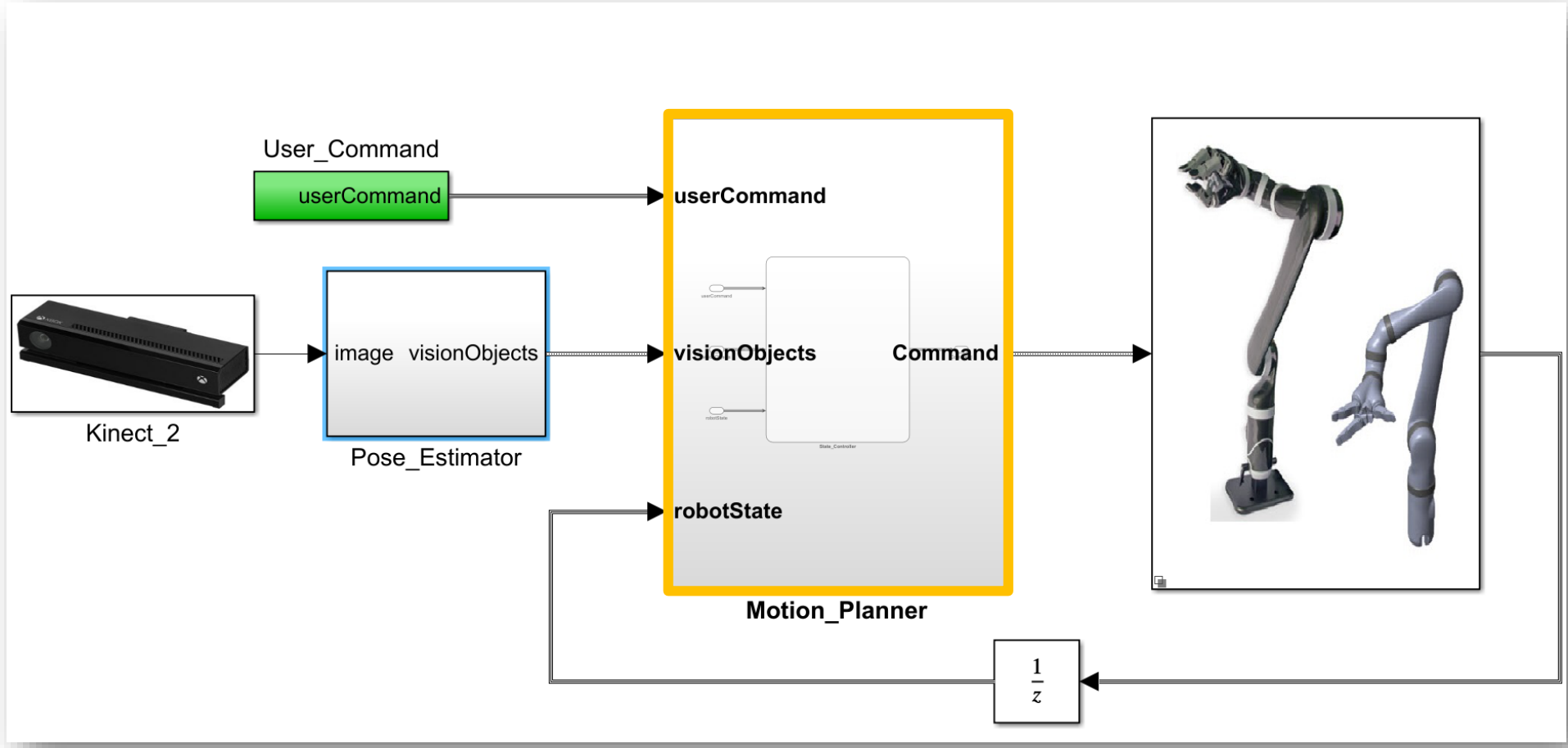

Map
Initial Pose
Final Pose
Path Planner
Path
$[x_a\ y_a\ \theta_a]$
$[x_b\ y_b\ \theta_b]$

使用Stateflow规划



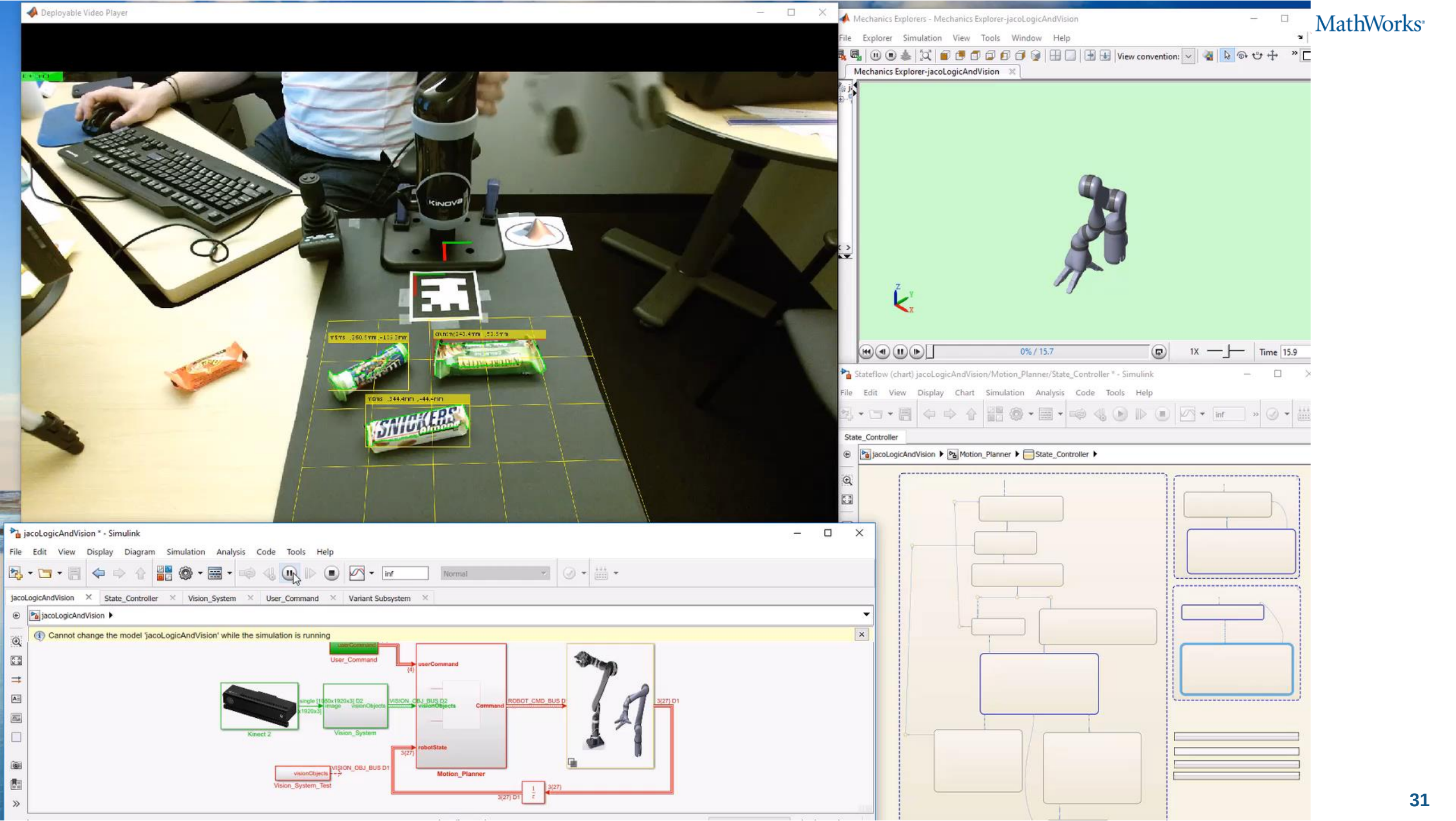
设计抓取放置应用


Platform
Sense
Perceive
Plan &
Decide
Control


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

探索内置函数：反向运动学

```
% Create ik solver object
ik = robotics.InverseKinematics('RigidBodyTree',jaco2n6s300)
% 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);
```


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
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
VISION_OBJ_BUS D2
visionObjects
Command
ROBOT_CMD_BUS D
3(27) D1
robotState
3(27)
visionObjects
VISION_OBJ_BUS D1
Vision_System_Test
Motion_Planner
3(27) D1
3(27)

本次演讲亮点

成功开发一个自主机器人系统需要：

- 多域仿真
- 信任能够使得复杂的工作流简化并能与其他工具集成的工具
- 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

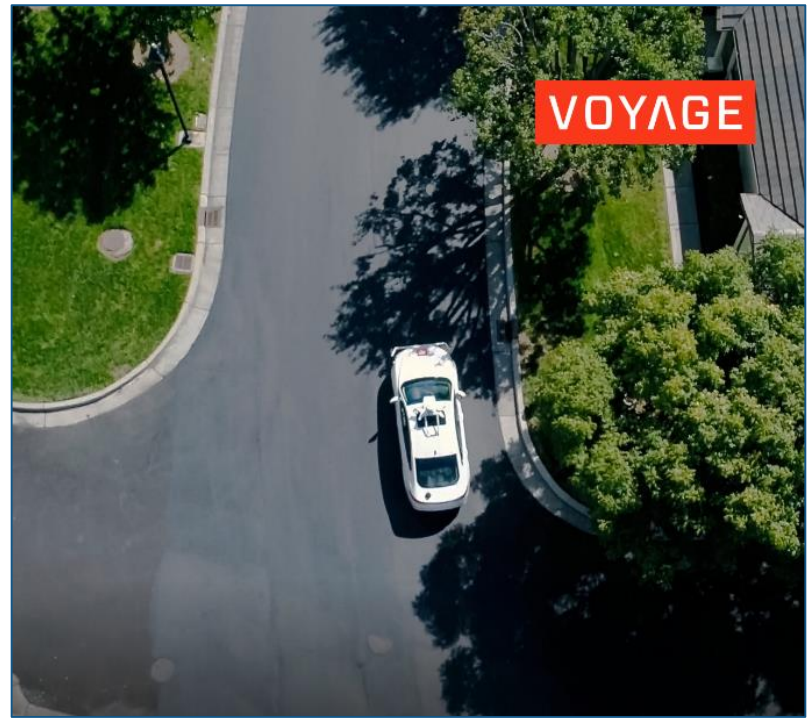
An OTTO self-driving vehicle from Clearpath Robotics.

"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



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

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 MATLAB Coder into a Docker Container.

Results

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

"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

用户案例：双足机器人

The Challenge

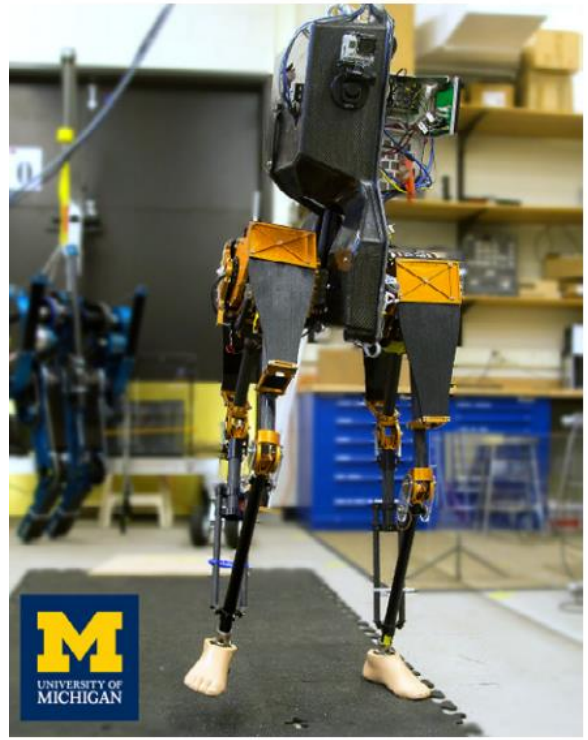
Develop a control system for an underactuated bipedal robot with 13 degrees of freedom

The Solution

Use Model-Based Design with MATLAB and Simulink to model the legs and torso, develop and simulate the control algorithms, and generate code for the real-time implementation

The Results

- Controller development accelerated
- Focus on high-level objectives maintained
- Approach adopted at other institutions



"When other researchers see that we've gone directly from controllers developed in MATLAB and Simulink to a real-time implementation with Simulink Real-Time, they get pretty excited. The approach we took is now being used in other departments at the University of Michigan and by robotics researchers at other universities, including MIT and Oregon State University."

- Prof. Jesse Grizzle, University of Michigan

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

Preceyes采用基于模型设计加速世界第一个眼科手术机器人的开发

Challenge

Develop a real-time control system for robot-assisted surgical procedures performed within the human eye

Solution

Use Model-Based Design with MATLAB and Simulink to model and simulate the control system and use Simulink Coder and Simulink Real-Time to deploy it to a real-time target

Results

- Development Core controller developed by one engineer
- Patient safety assured
- Road map to industrialization set

Link to user story

The PRECEYES Surgical System. Image copyright and courtesy Preceyes.

"MATLAB and Simulink provided a single platform that supported our complete workflow and all the components and protocols we needed for our robotic system. That enabled us to quickly develop a safe, real-time device, ready for clinical investigation."

- Maarten Beelen, Preceyes

```
% Thank you
```