MATLAB EXPO 2021

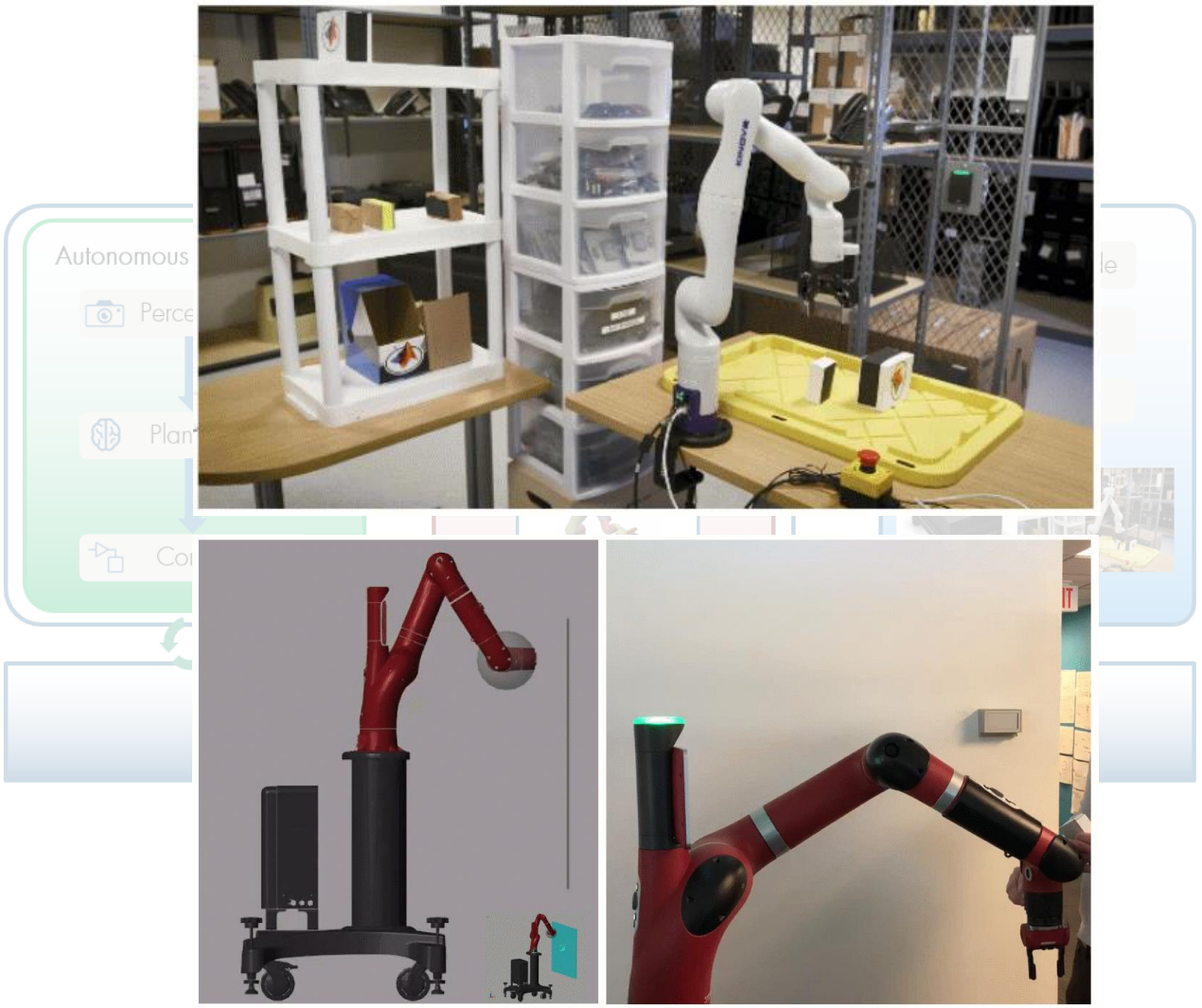
# 智慧工厂：从感知到运动规划的自主工业机器人

*周玲*

MathWorks®

# 使用 MATLAB 和 Simulink开发自主机器人

- 工业机器人的平台设计工具
- 开发自主工业机器人应用的工具
- 用 MATLAB 和 Simulink实现软件部署及硬件接口
- 使用 MATLAB 和 Simulink的集成工作流



# 行业趋势 – 先进机器人

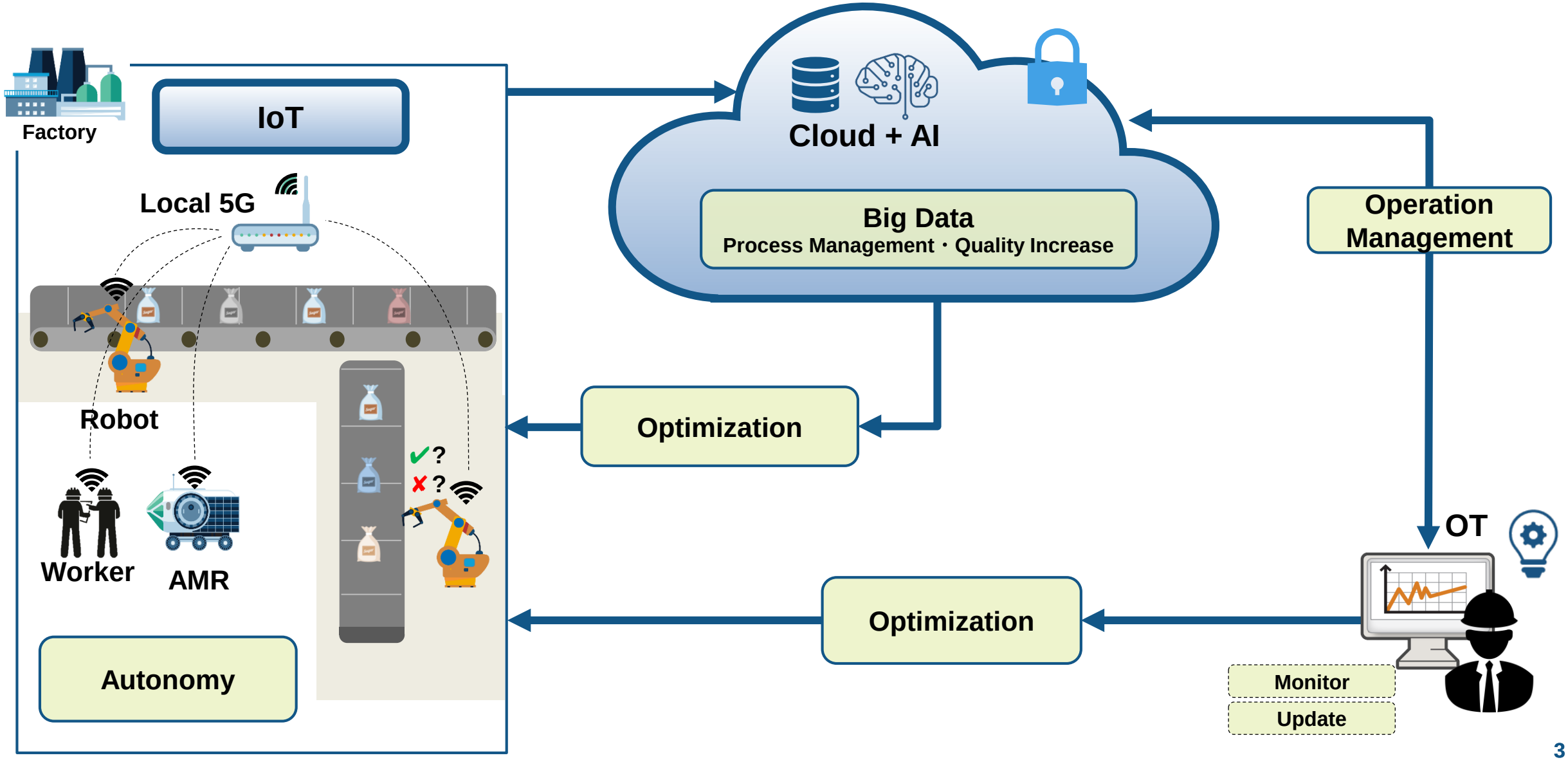
智慧工厂是:

- 包含更先进的系统
- 利用新技术
- 增加组件之间的连接性
- 加强基础设施，实现工业 IoT

柔性制造，自主生产

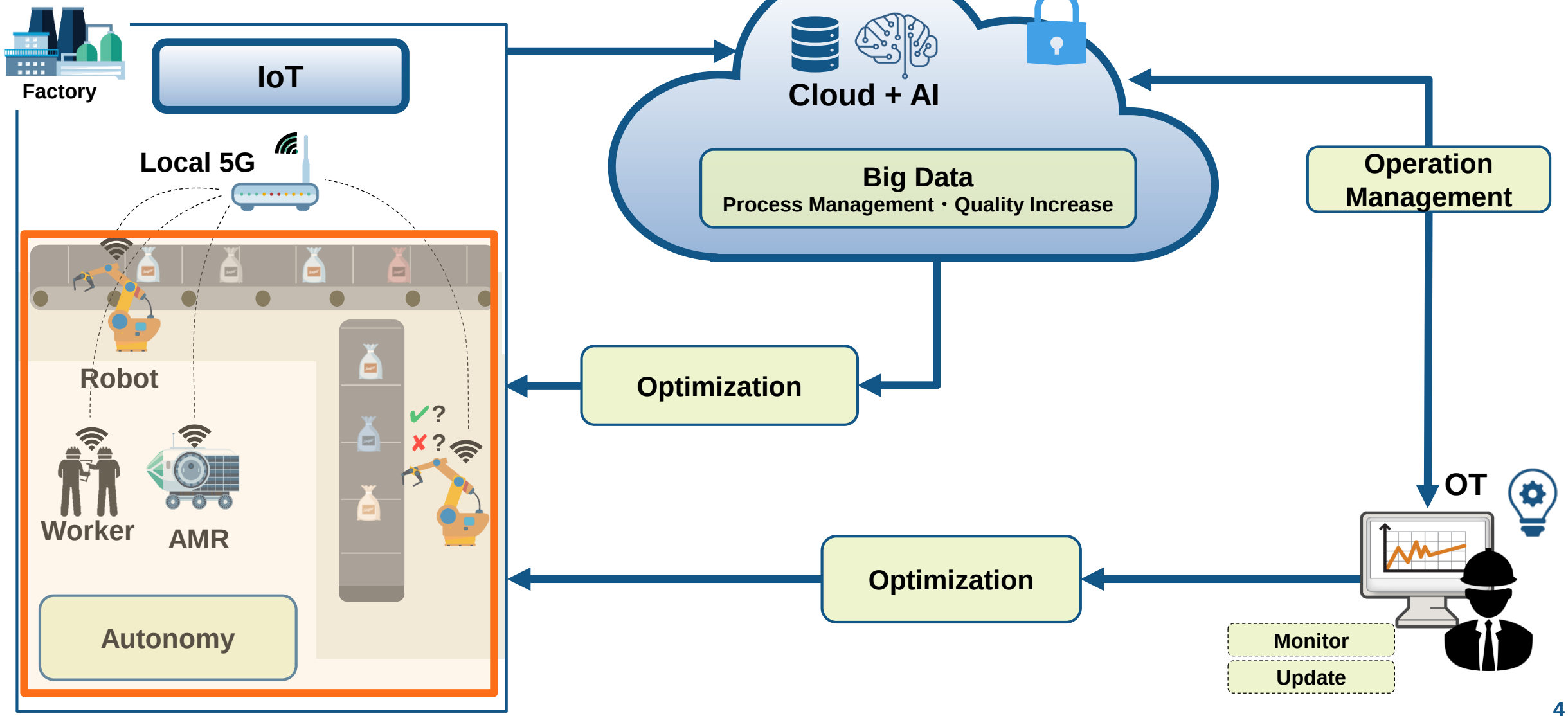
协作机器人
基于 AI 的机器人

# 智慧工厂概念：资产、运营和劳动力优化



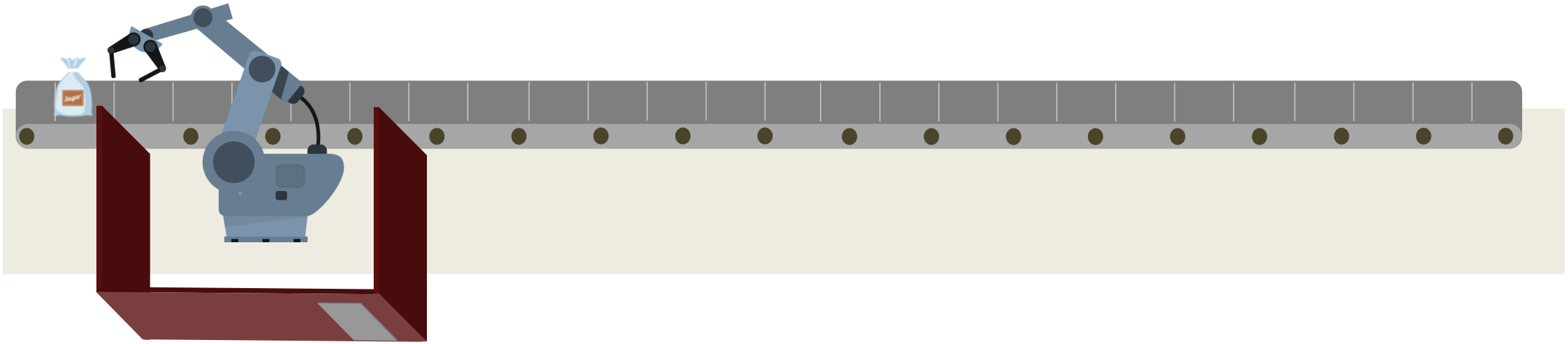
# 智慧工厂概念：资产、运营和劳动力优化

我们今天谈论的



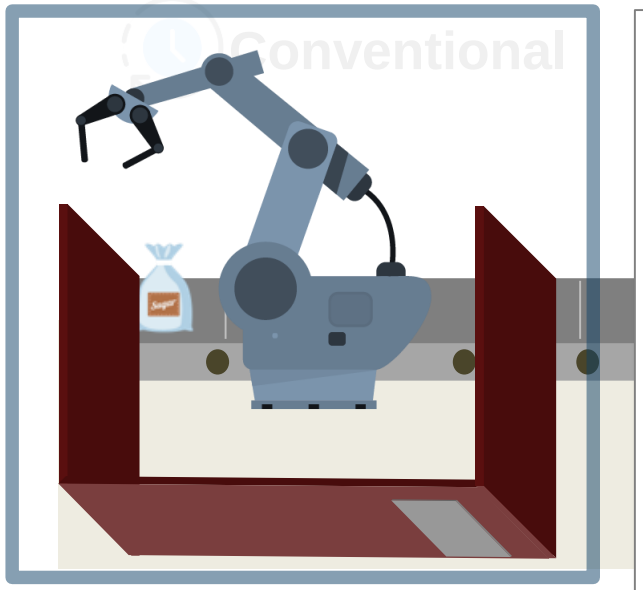
# 智慧工厂中的机器人:趋势与方向

**传统的**



- **自动化系统**
  - 重复任务
  - 手动编程
  - 需要安全框架

# 智慧工厂中的机器人：趋势与方向




- **自动化系统**
  - 重复任务
  - 手动编程
  - 需要安全框架

MathWorks

## Mitsubishi Heavy Industries Develops Robotic Arm for Removing Nuclear Fuel Debris

### Challenge

Design a multi-axis robot for removing molten fuel debris from the Fukushima Daiichi nuclear power station

### Solution

Use MATLAB and Simulink to perform hardware measurement tests and to model and simulate individual robot axes and controllers

### Results

- Development time halved
- Positioning accuracy requirement exceeded
- Shared platform for interorganizational collaboration established

**Rendering of Mitsubishi Heavy Industries' seven-meter-long robotic arm capable of withstanding up to 2000 kg of processing reaction force**

*"Model-Based Design with MATLAB and Simulink supports a wide range of options, from classic to modern control, which made it possible to respond easily to any changes in design constraints and to meet the demanding accuracy requirement for this robot."*

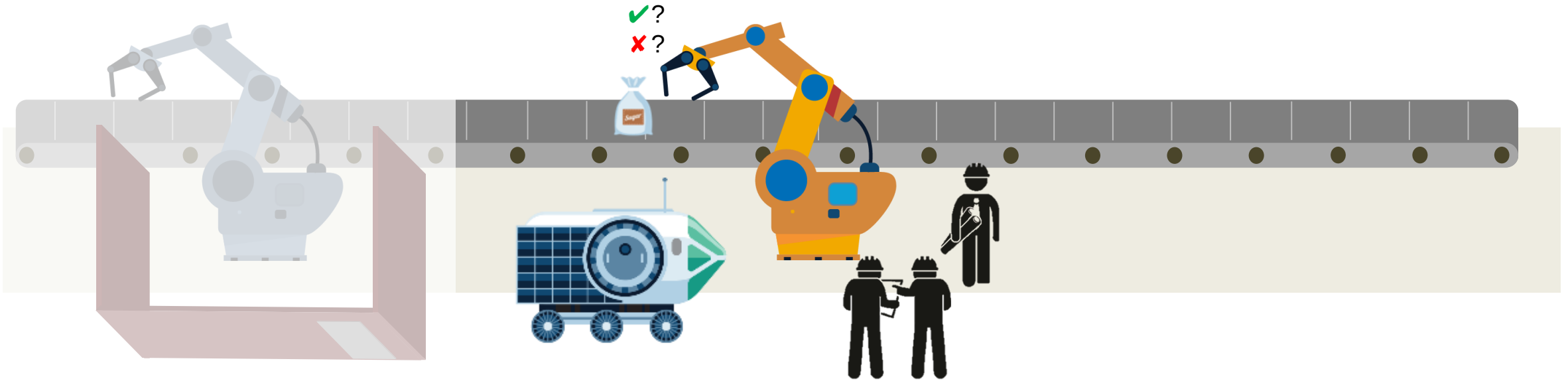
*- Tadashi Murata, Mitsubishi Heavy Industries*

Link to user story

# 智慧工厂中的机器人：趋势与方向

Conventional

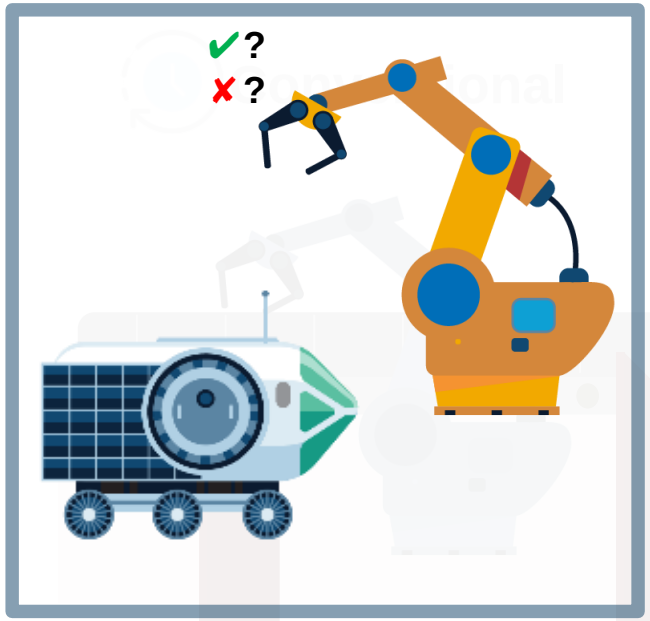
当前




- 自动化系统
  - 重复任务
  - 手动编程
  - 需要安全框架

- **柔性自动化**
  - 协作机器人
  - 高级算法
  - 安全

# 智慧工厂中的机器人:趋势与方向

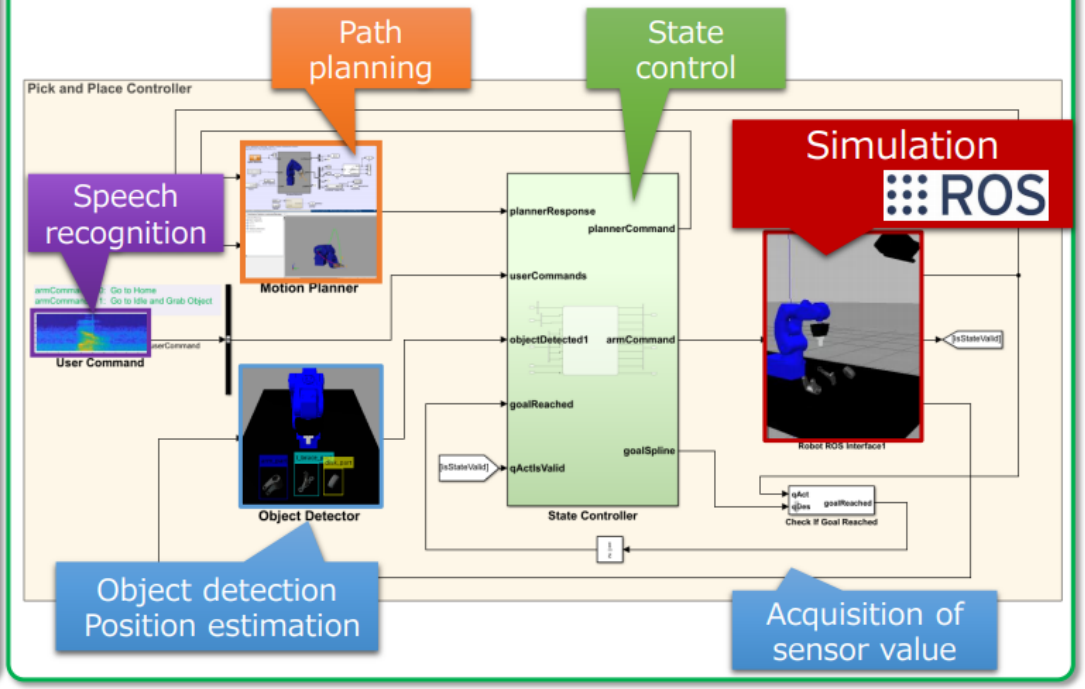



## Sample app overview

**Flow of explanation**

1. Object detection Position estimation
2. Path planning
3. Speech recognition
4. State machine

**Block diagram (Simulink model)**



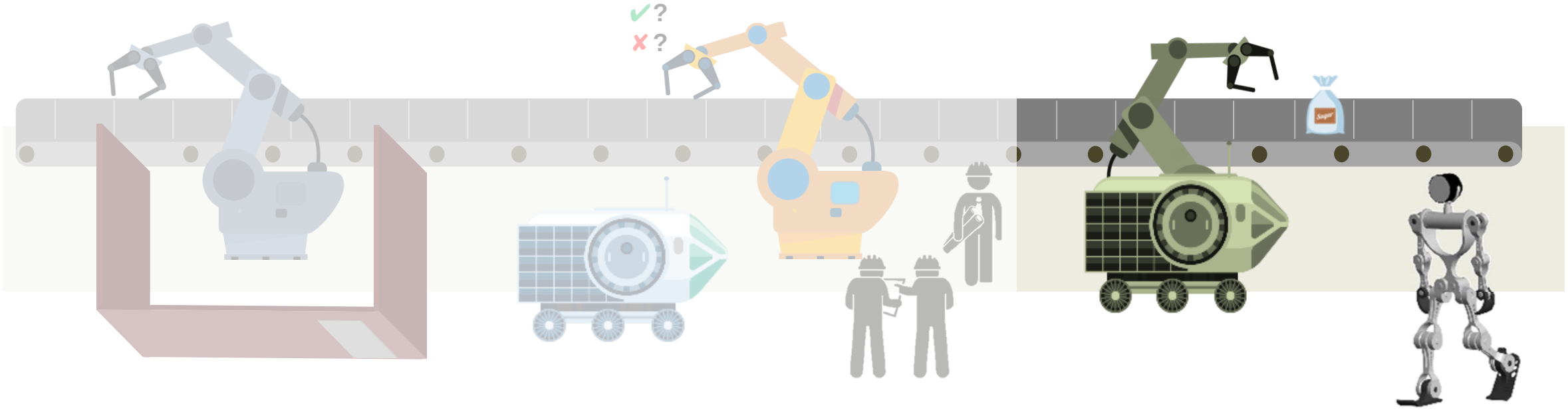





MATLAB EXPO 2019 in Japan
Customer Presentation

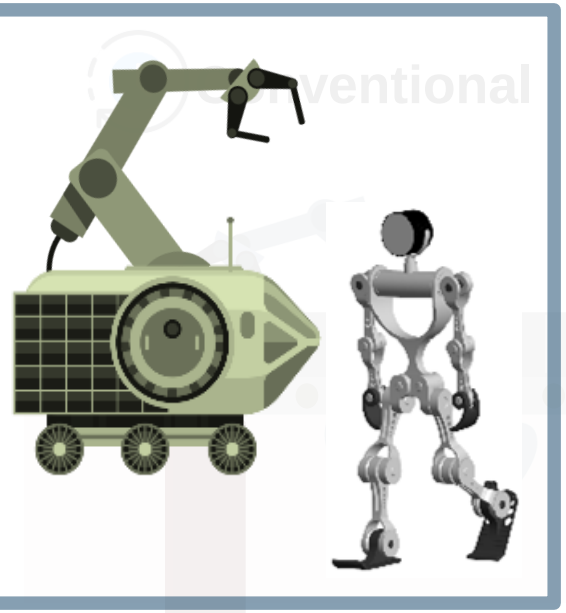
# 智慧工厂中的机器人：趋势与方向

传统的　　当前　　未来



- 自动化系统
  - 重复任务
  - 手动编程
  - 需要安全框架
- 柔性自动化
  - 协作机器人
  - 高级算法
  - 安全
- **自主系统**
  - 任务合作
  - 信息共享
  - 适应环境

# 智慧工厂中的机器人:趋势与方向



MathWorks

## German Aerospace Center (DLR) Robotics and Mechatronics Center Develops Autonomous Humanoid Robot with Model-Based Design

### 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

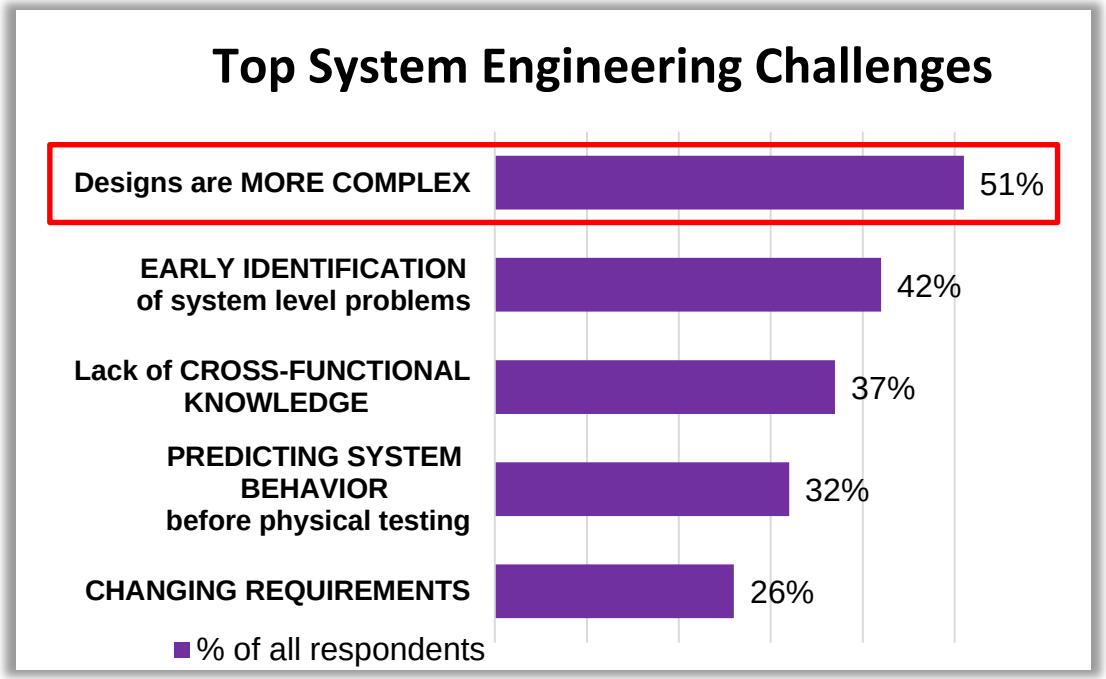
- Programming defects eliminated
- Complex functionality implemented in hours
- Advanced control development by students enabled

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

*"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*

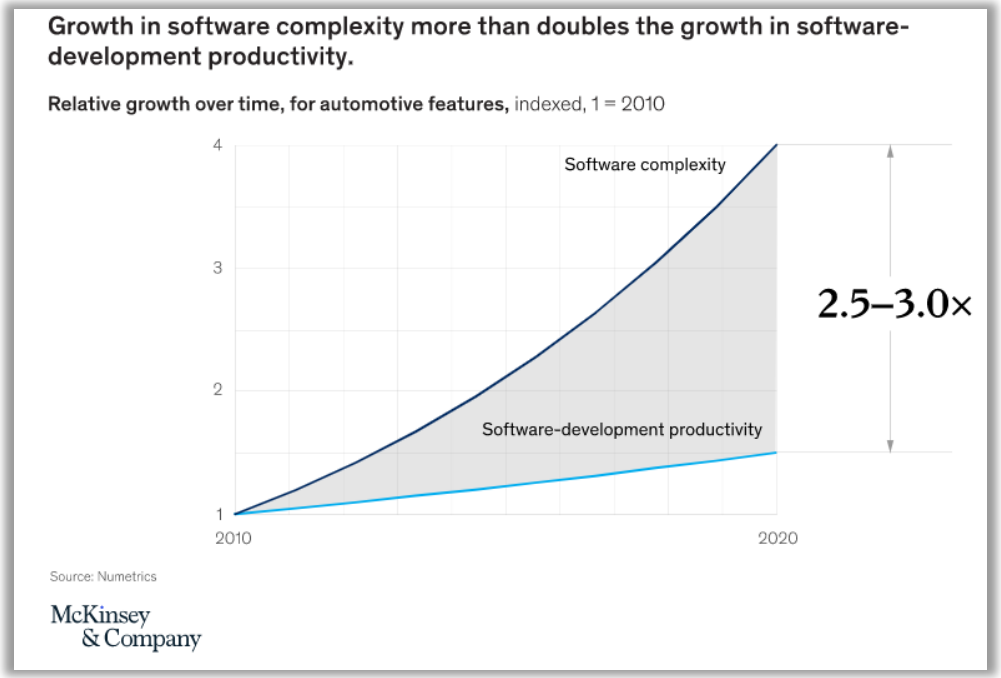
Link to user story

# 挑战：复杂度的增加




***Design Complexity*** is the most commonly challenge in the Aberdeen survey. Cited by 51% in 2014, up from 27% in 2009

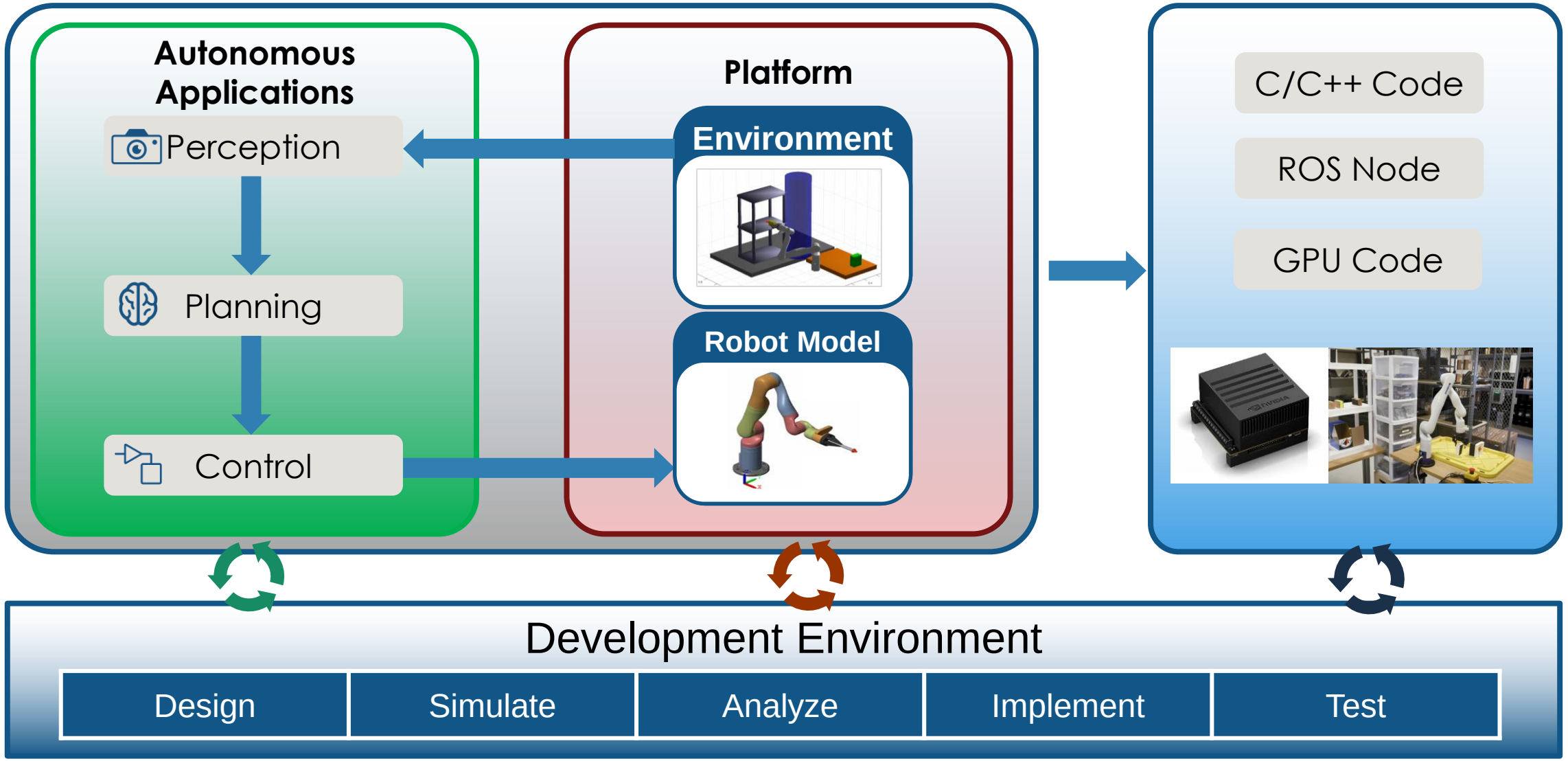
*Source: Aberdeen Group, April 2014*



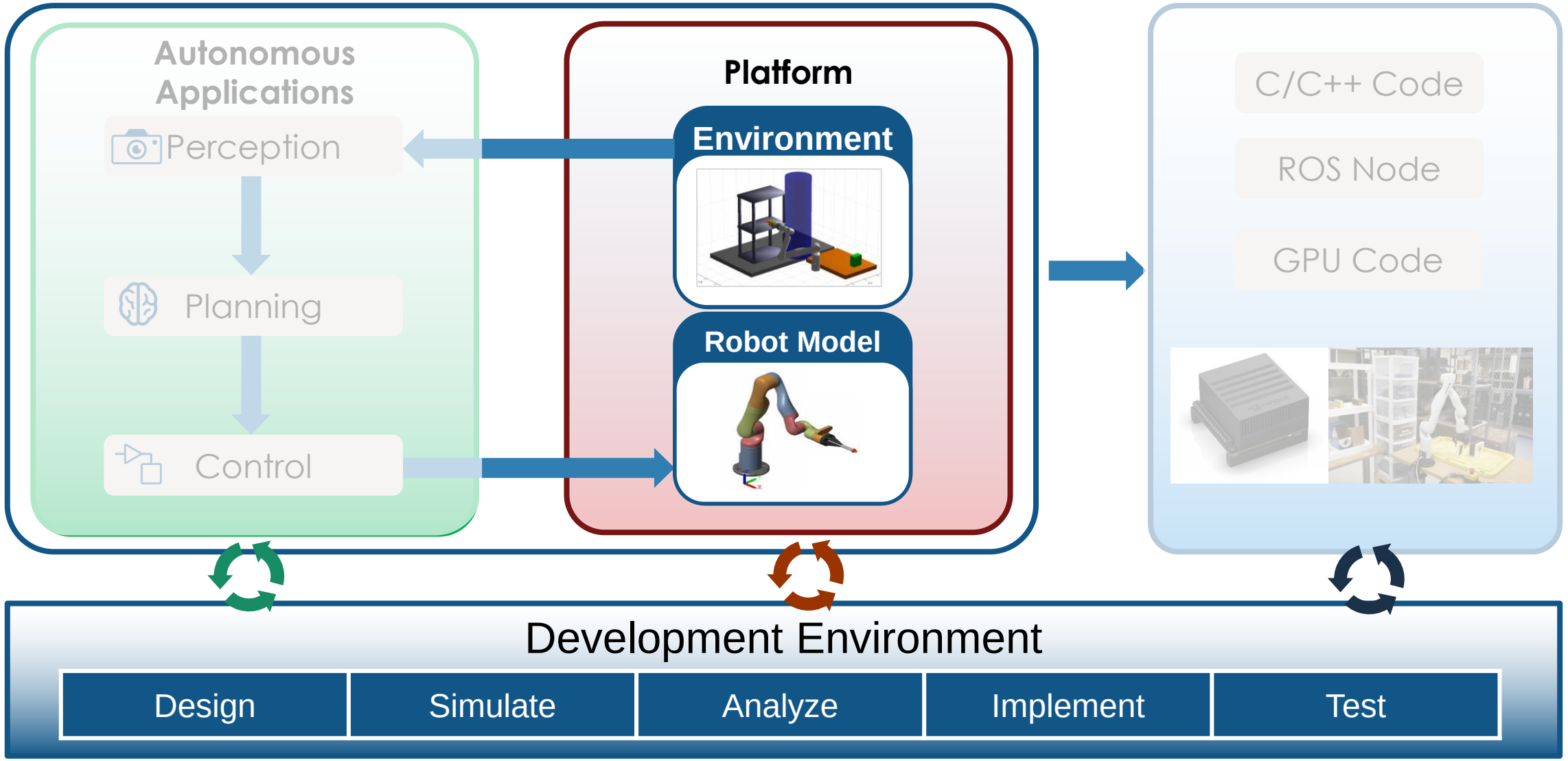

***Complexity for software features*** is currently growing at double to triple the speed of software-development productivity

*Source: McKinsey & Company, Feb 2020*

# 使用 MATLAB 和 Simulink开发自主机器人

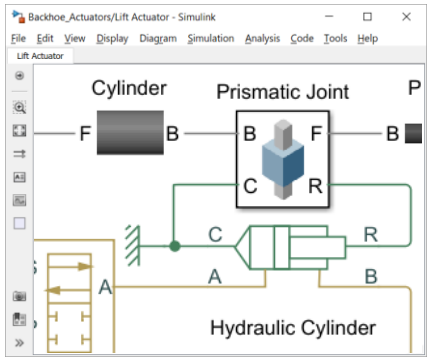


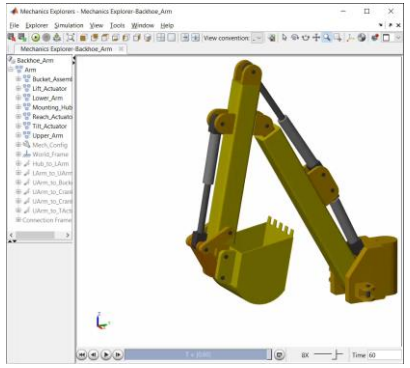
# 使用 MATLAB 和 Simulink开发自主机器人

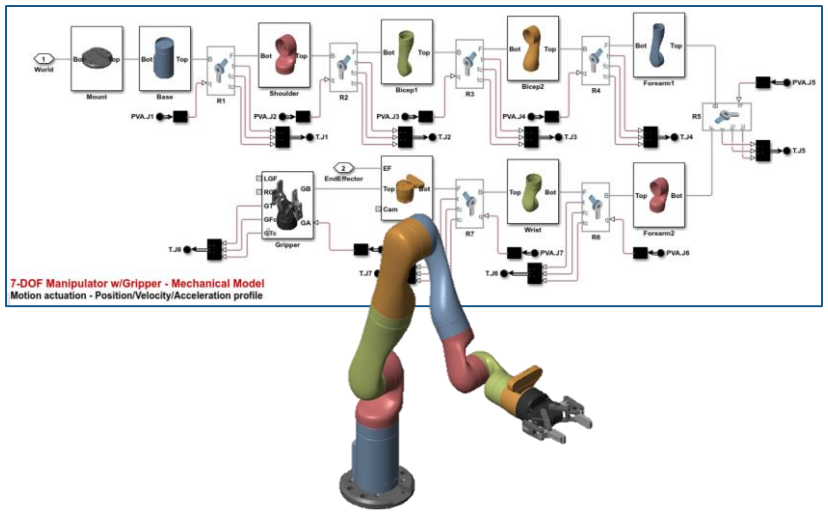


# 物理系统和运动学建模

## 创建物理模型

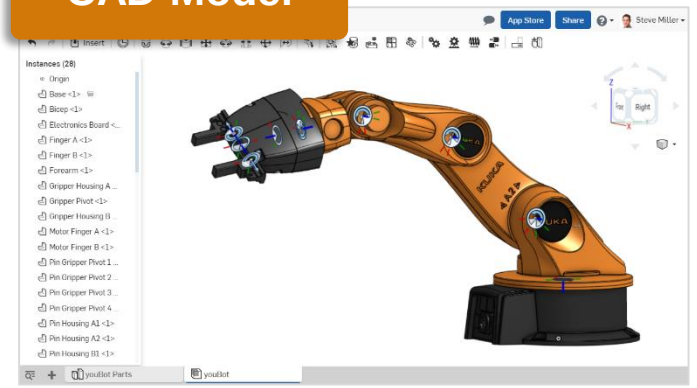


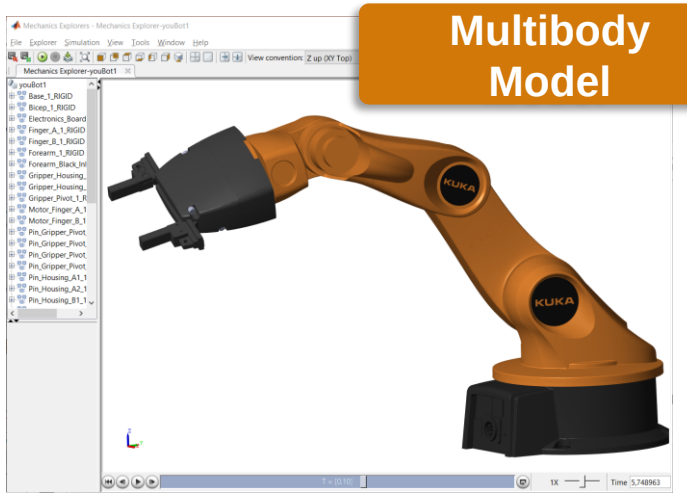



## 从CAD 工具自动导入

CAD Model

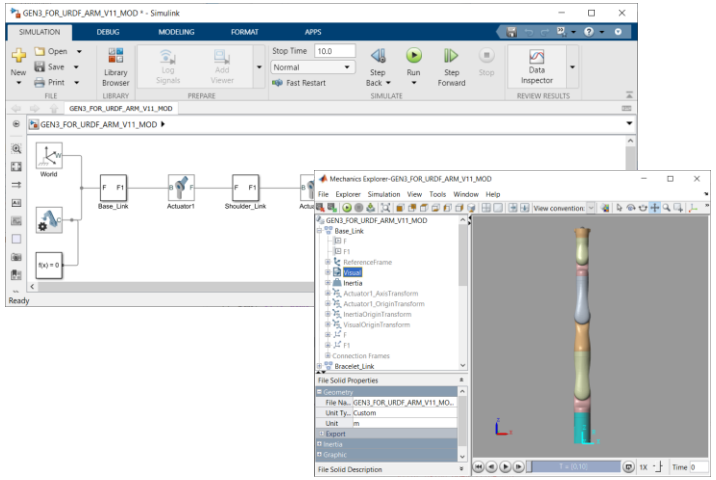


Multibody Model

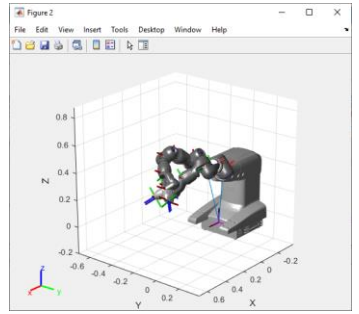


## URDF / Robot Library

```
%% Import robot from URDF
smimport('GEN3_FOR_URDF_ARM_V11_MOD.urdf');
```



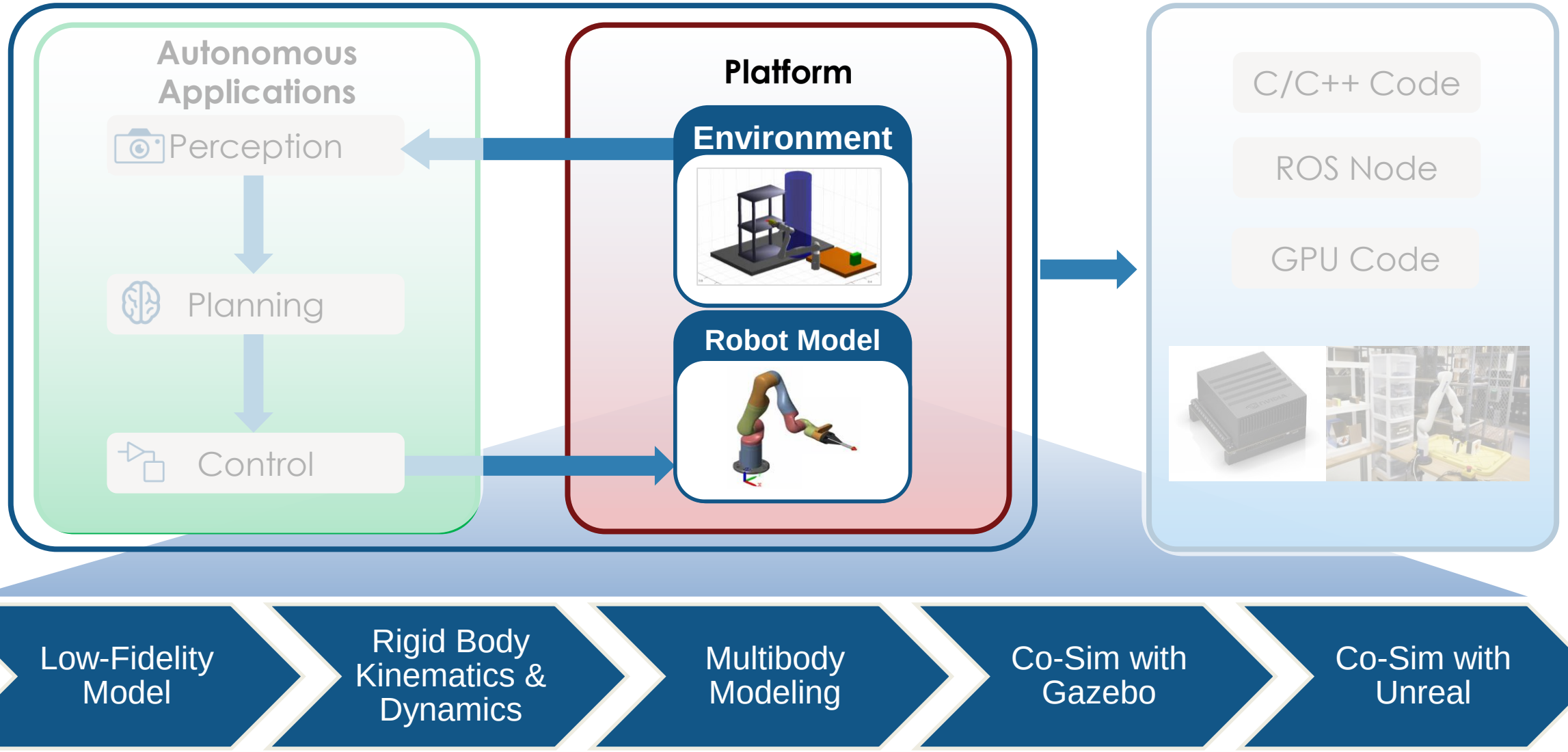
```
%% Use robot library
Robot = loadrobot('abbYuMi');
Show(robot);
```



*Simscape Multibody, Robotics System Toolbox*

# 机器人仿真

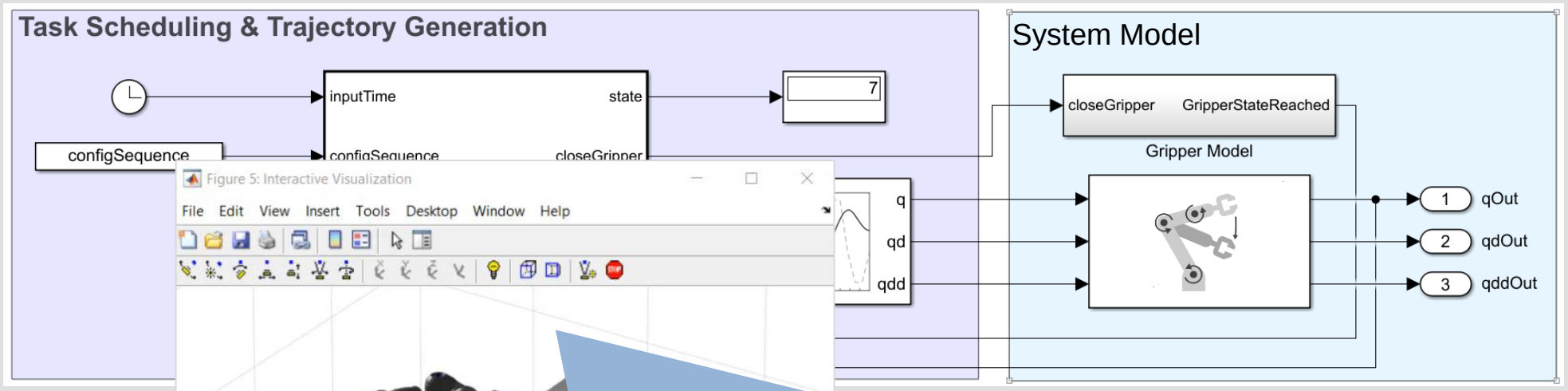
## Simulate First and Simulate Often!

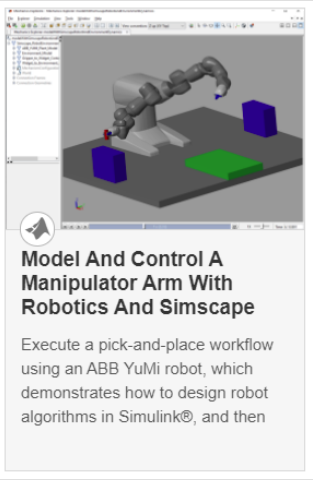


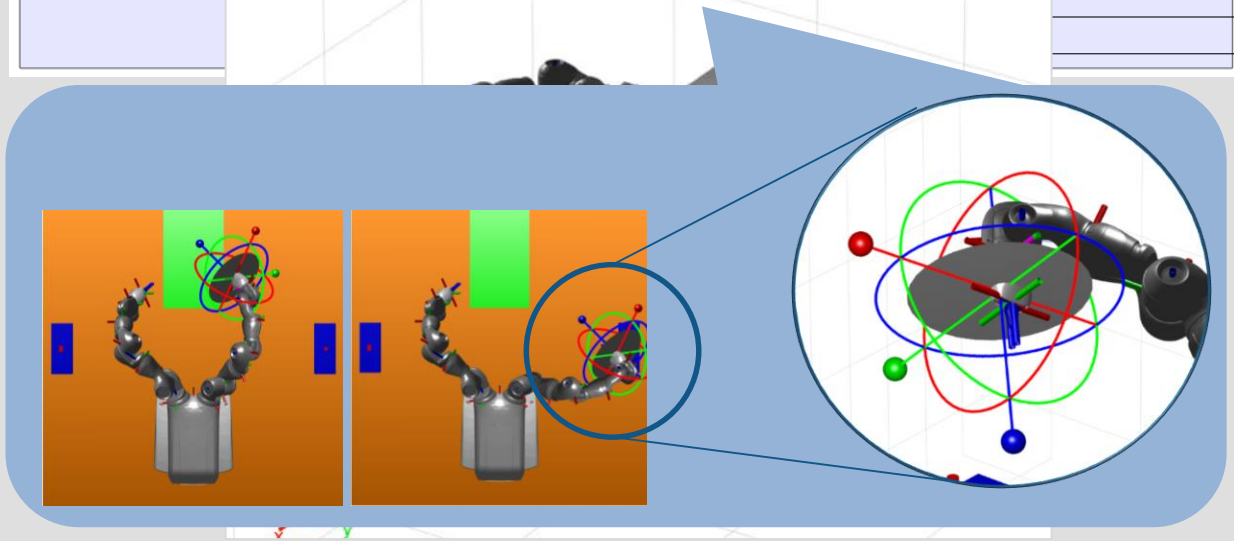
# 快速和低保真度仿真



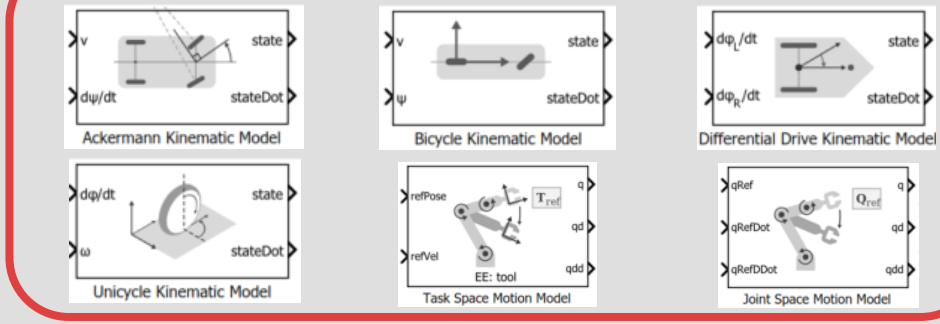
## 带关节空间运动模型的简化系统模型



## Motion Models

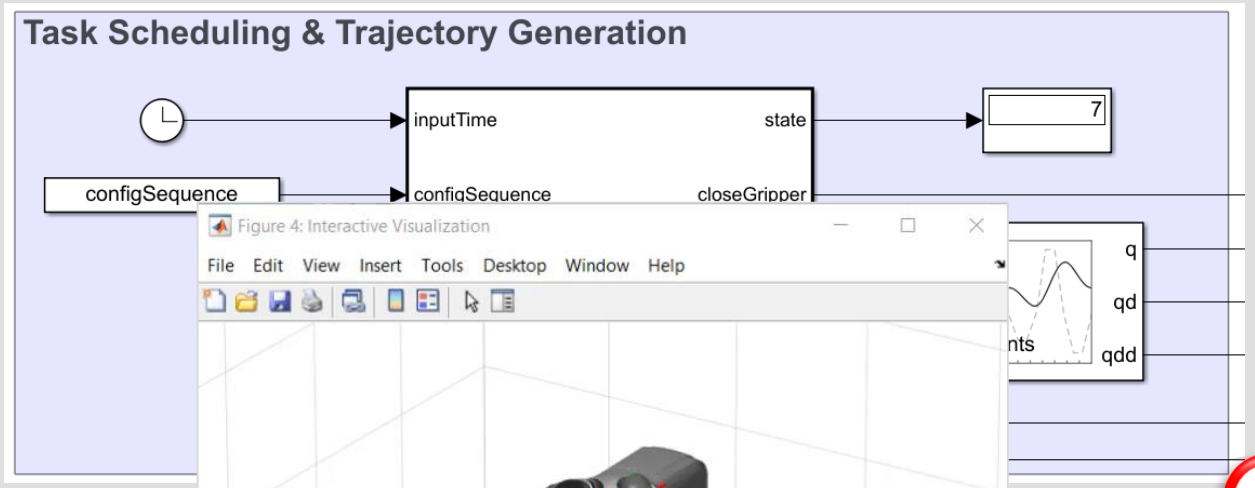


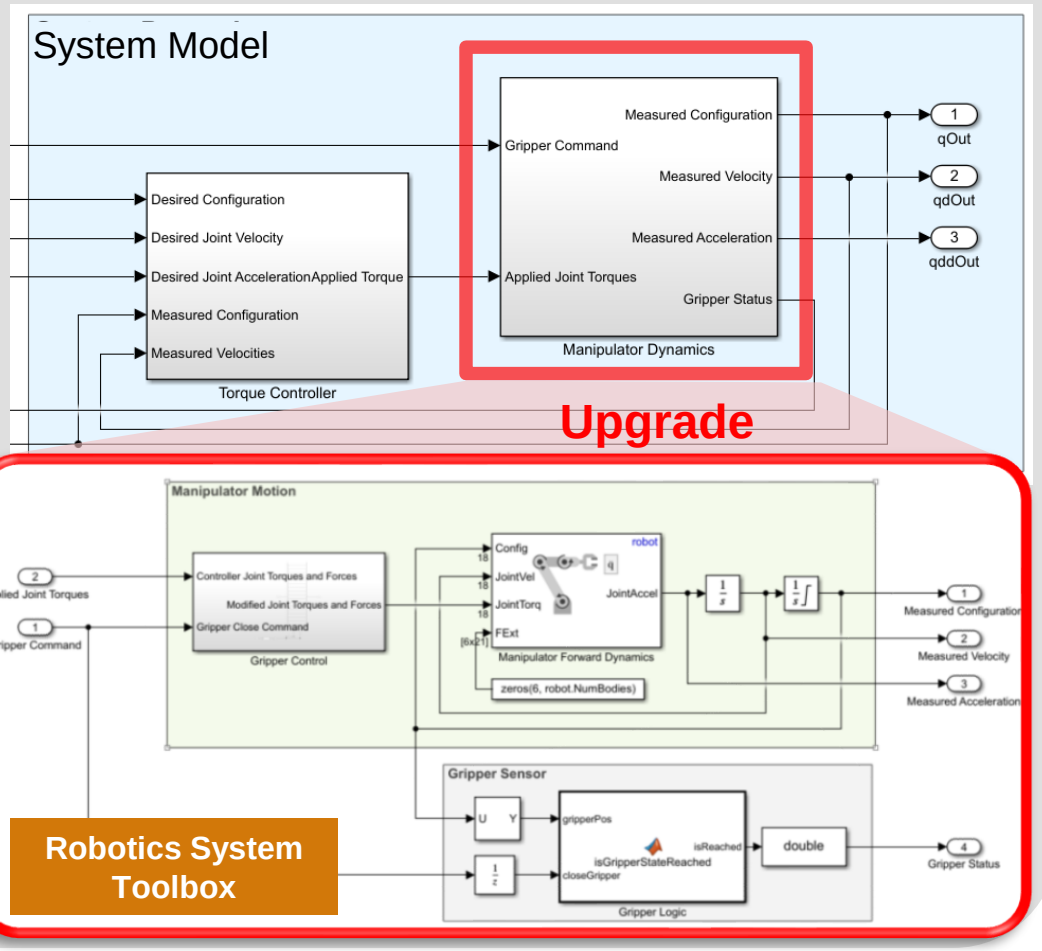

*Simscape Multibody, Robotics System Toolbox*

# 动态仿真



## 接受关节扭矩和抓手指令的机械臂动力学模型



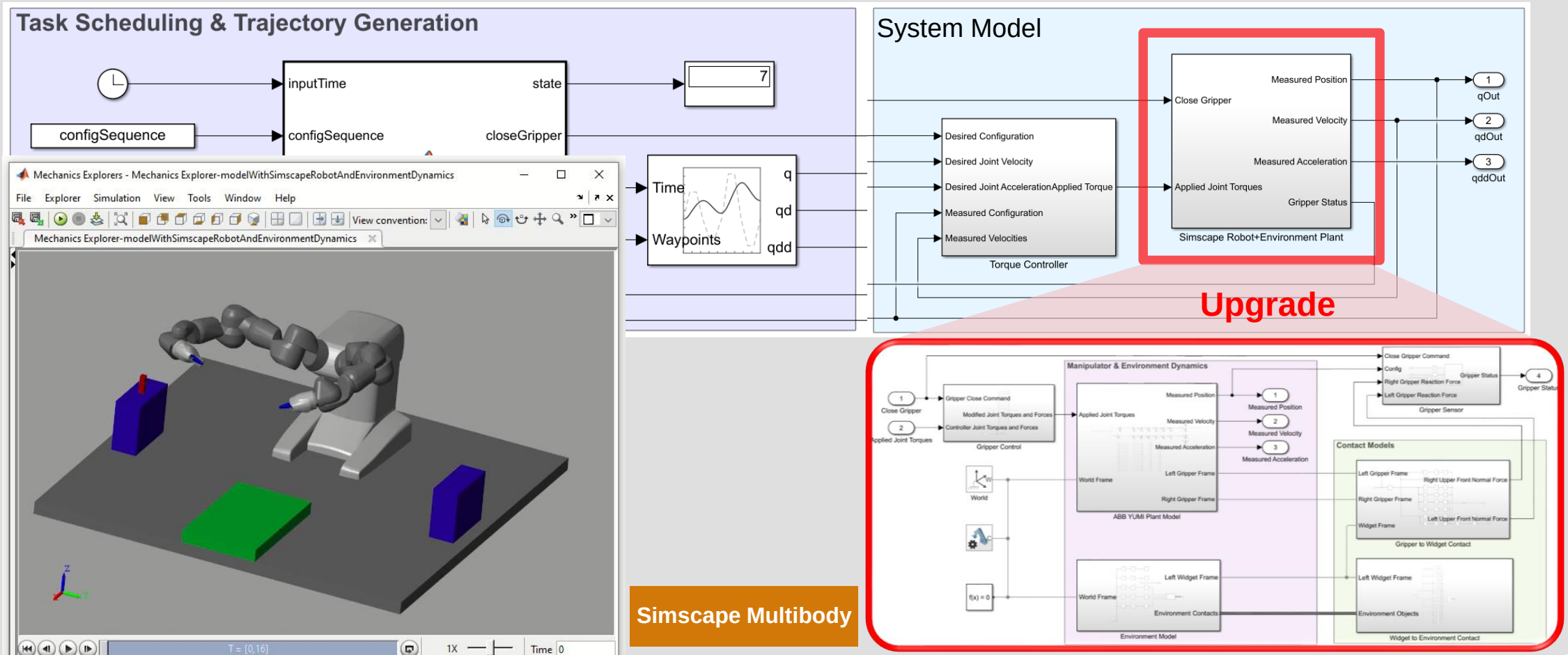




*Simscape Multibody, Robotics System Toolbox*

# 多体仿真



## 内置关节约束和接触模型的物理动力学系统

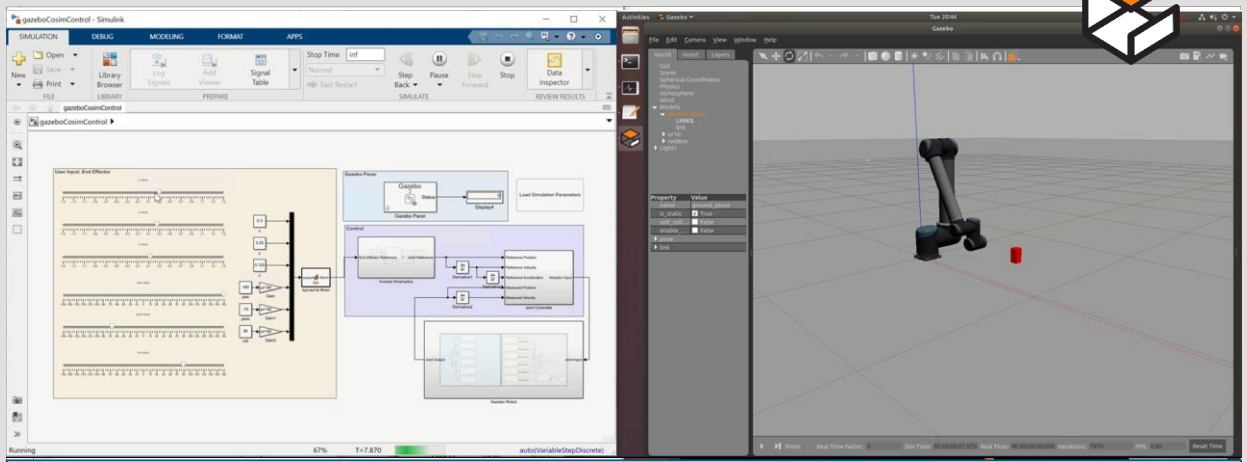



*Simscape Multibody, Robotics System Toolbox*

# 环境建模



## 与第三方模拟器联合仿真传感器和环境模型



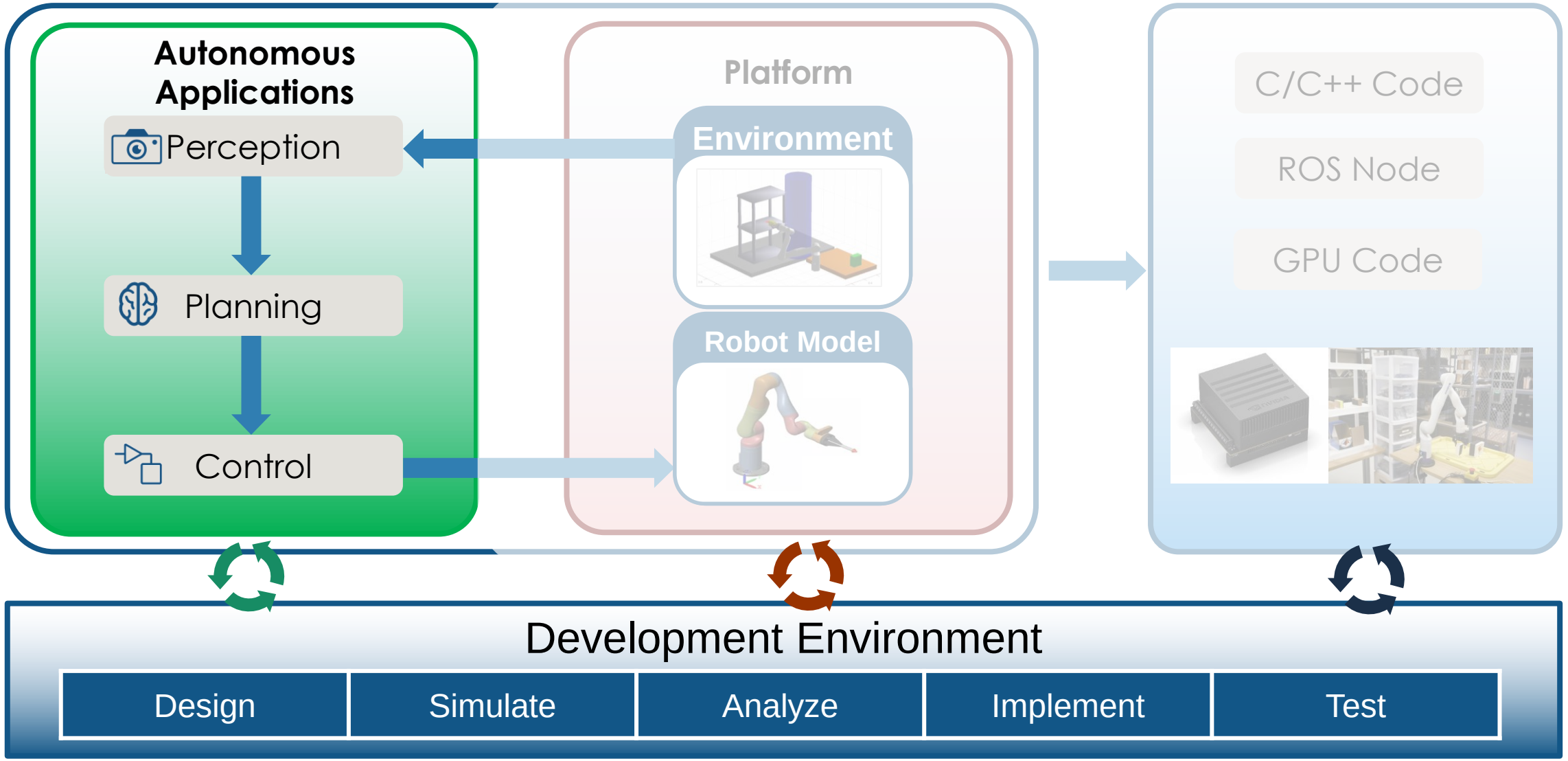
Gazebo Co-simulation with Time Synchronization


### Control Manipulator Robot with Co-Simulation in Simulink and Gazebo

Simulate control of a robotic manipulator using co-simulation between Simulink and Gazebo. The example uses Simulink™ to model the robot behavior, generate control commands, send these commands to Gazebo, and control the pace of the Gazebo simulation.

*Simscape Multibody, Robotics System Toolbox, ROS Toolbox*

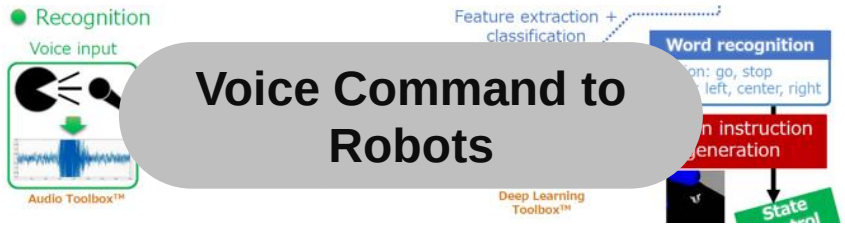
# 使用 MATLAB 和 Simulink开发自主机器人



# 如何将深度学习用于机器人?

## ❑语音识别

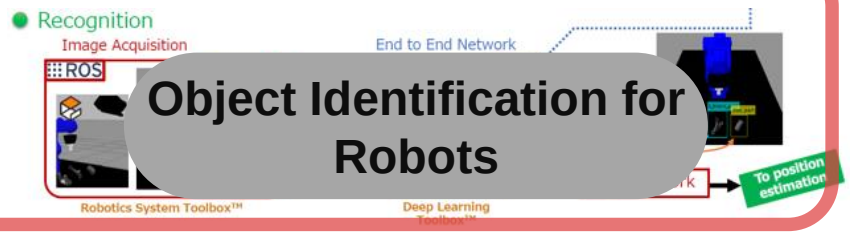
Audio → Deep Learning → Text




Voice Command to Robots

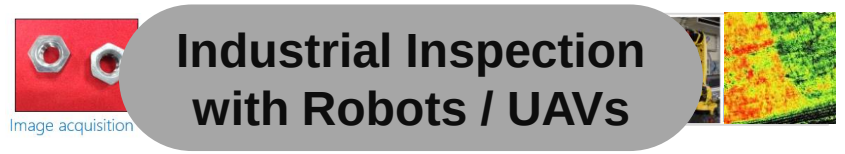
## ❑计算机视觉

Pixels → Deep Learning → Labels




Object Identification for Robots

Pixels → Deep Learning → Image Abnormal

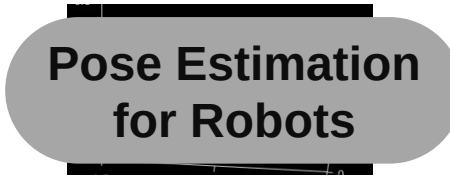



Industrial Inspection with Robots / UAVs

## ❑点云处理

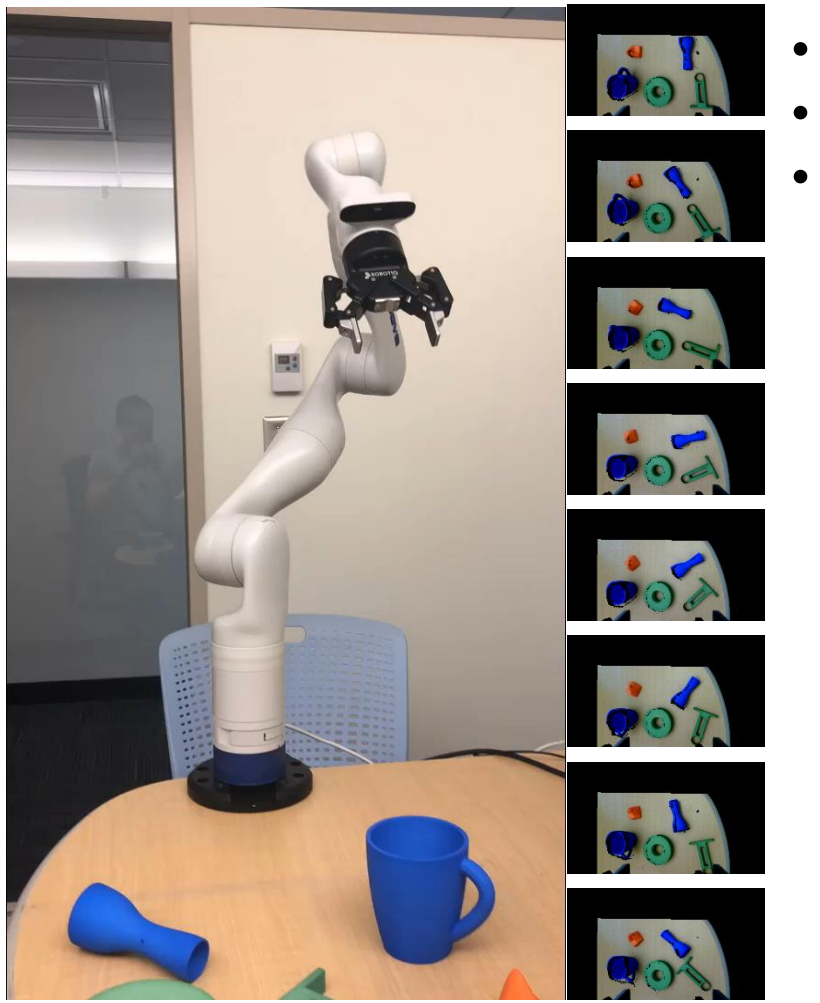
Point Clouds → Deep Learning → Structure

Localization

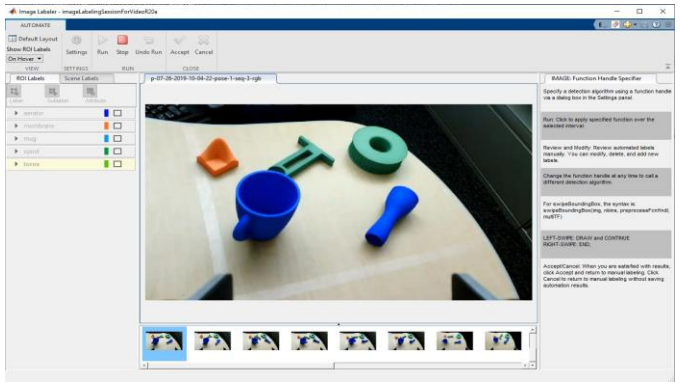


Pose Estimation for Robots

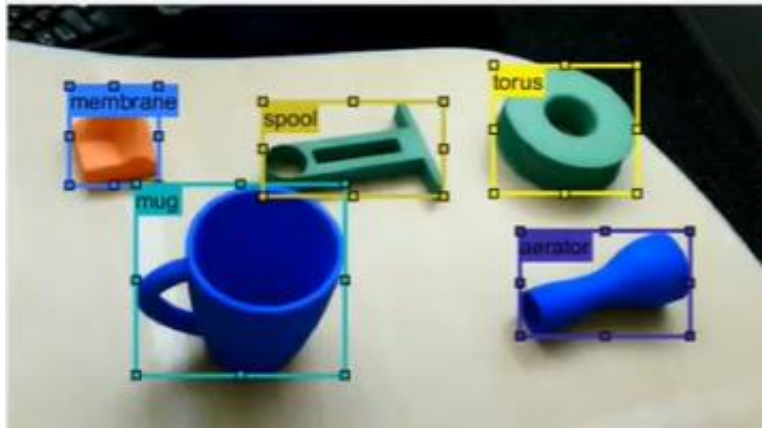
# 感知
## 目标分类的深度学习



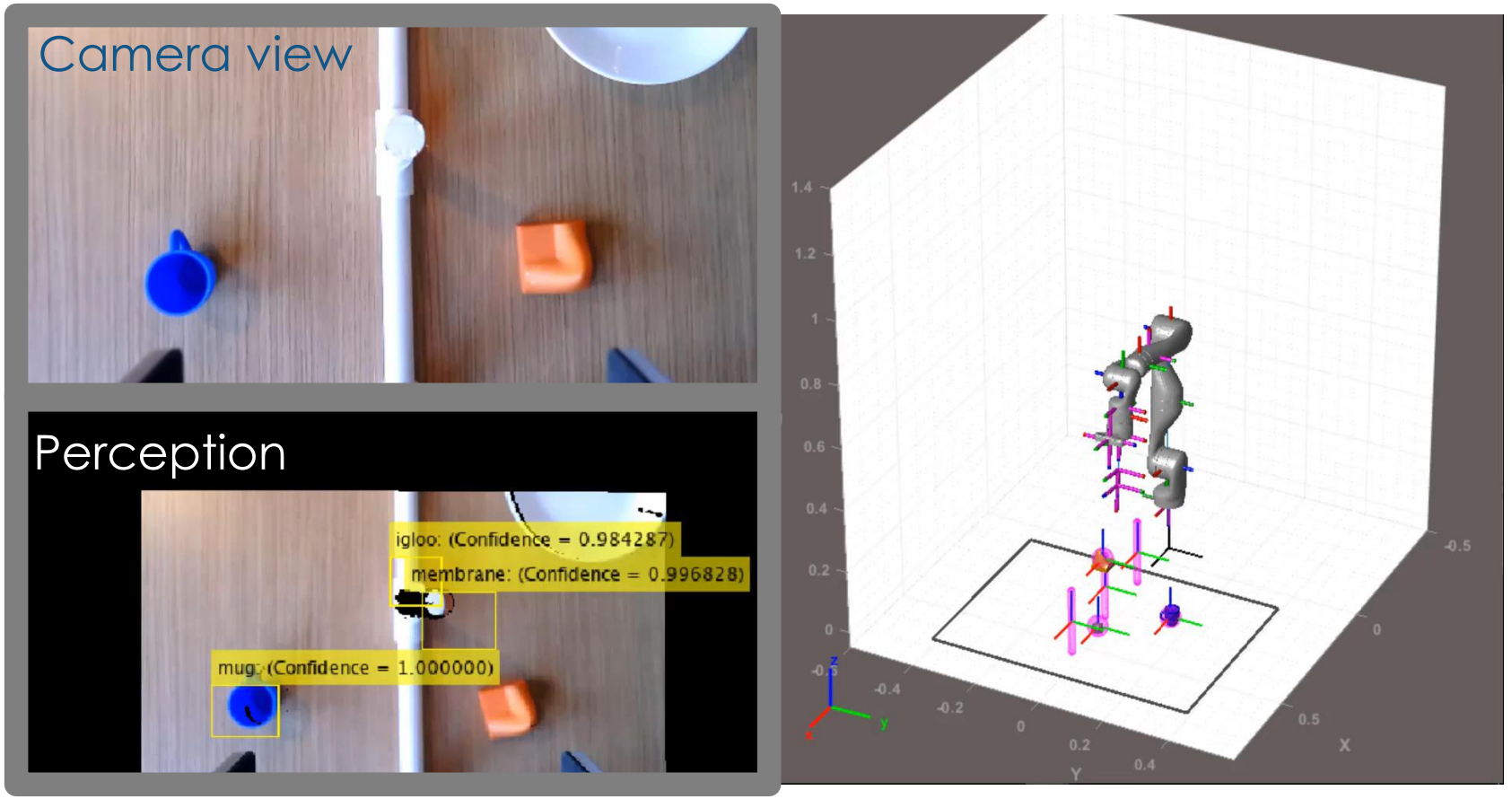
自动标注
迭代学习



使用深度学习的对象探测器
(YOLO v2)

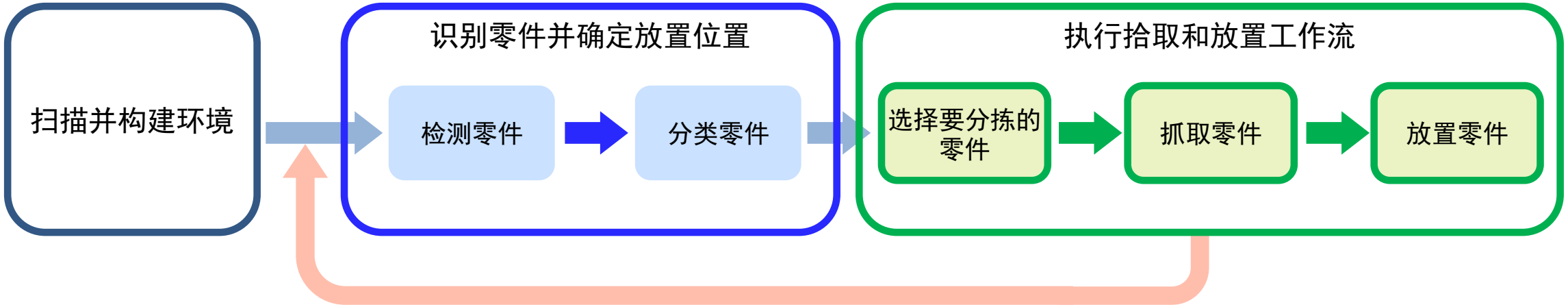


# 感知
## 目标分类

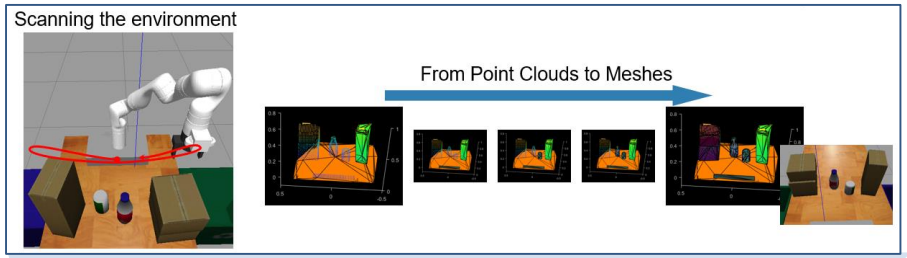



*Computer Vision Toolbox, Deep Learning Toolbox, Robotics System Toolbox, Stateflow*

# 分拣机器人的完整工作流程

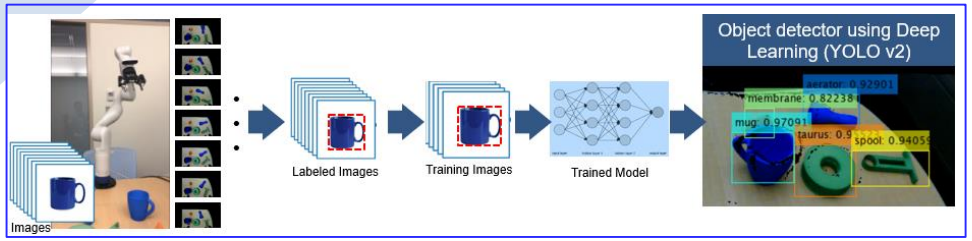


# 分拣机器人的完整工作流程

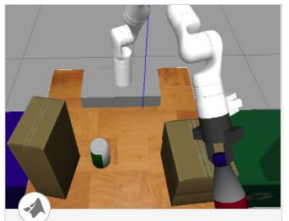



扫描并构建环境

- 动态环境
- 灵活操作

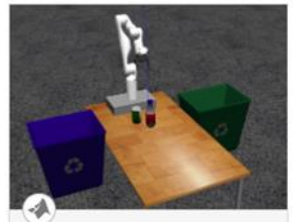



**Shipping examples**



**Pick-and-Place Workflow in Gazebo using Point-Cloud Processing and RRT Pat…**

Setup an end-to-end pick and place workflow for a robotic manipulator like the KINOVA® Gen3.

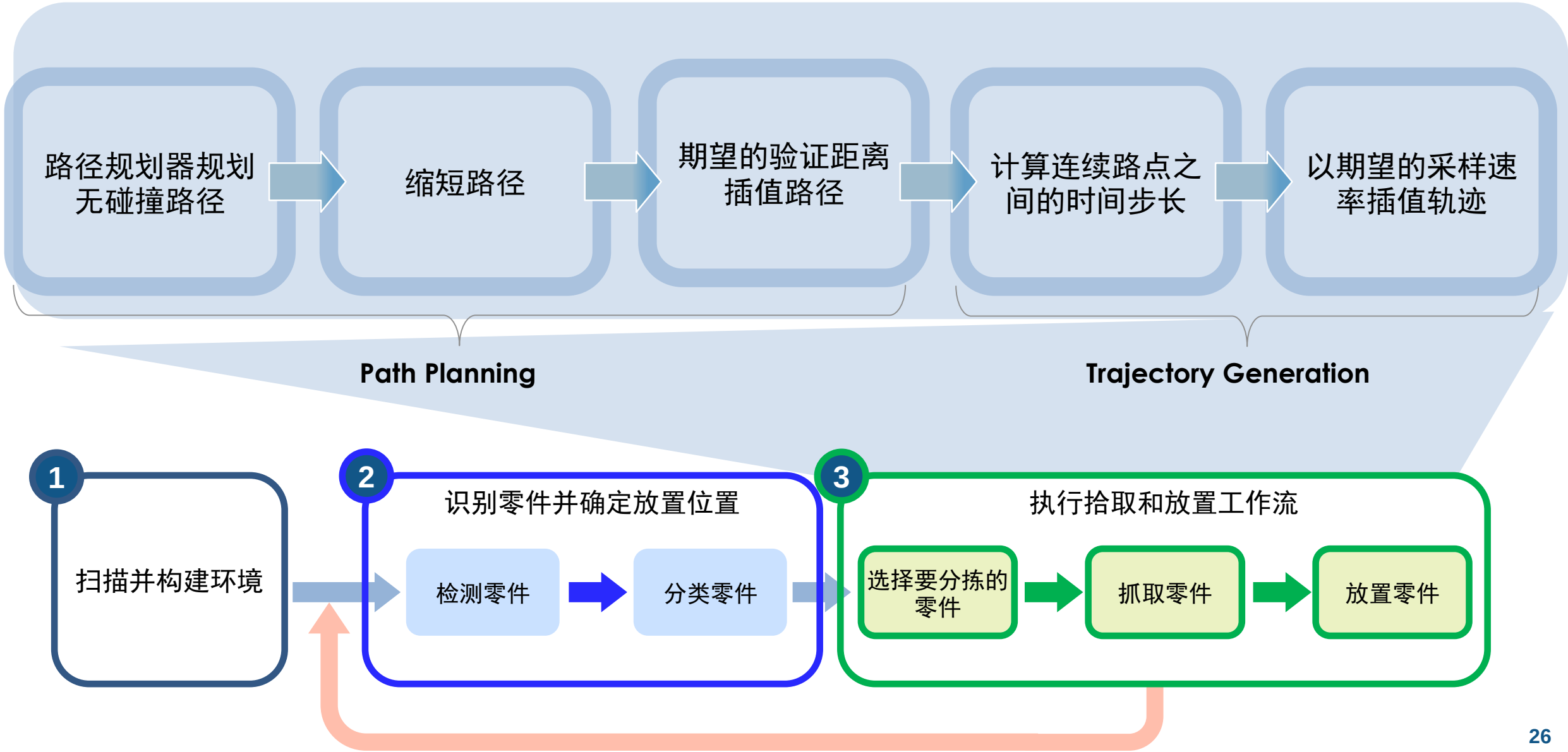


**Pick-and-Place Workflow in Gazebo using ROS**

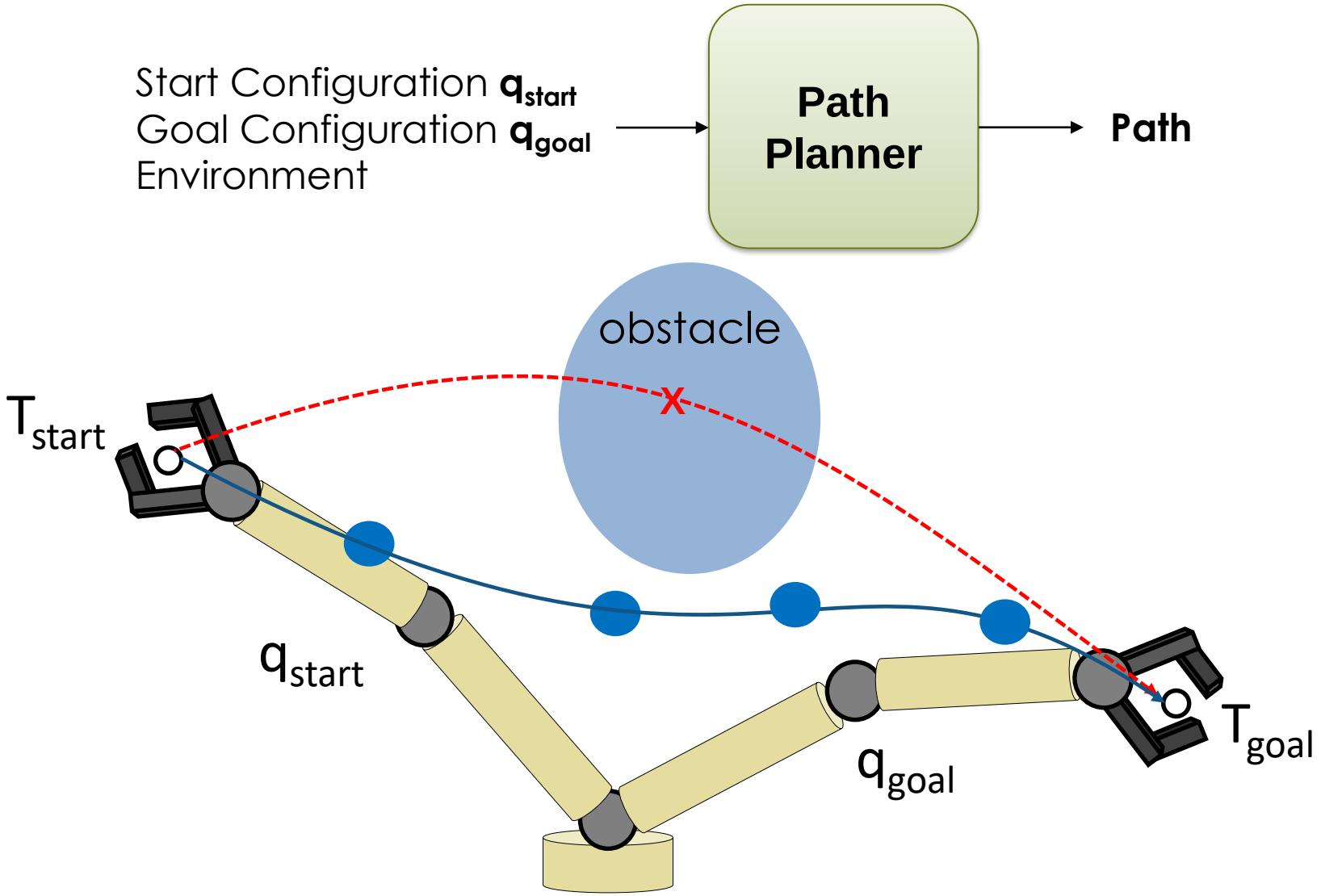
Setup an end-to-end pick and place workflow for a robotic manipulator like the KINOVA® Gen3 and simulate the robot in a physics

1. 扫描并构建环境
2. 识别零件并确定放置位置
   - 检测零件 → 分类零件
3. 执行拾取和放置工作流
   - 选择要分拣的零件 → 抓取零件 → 放置零件

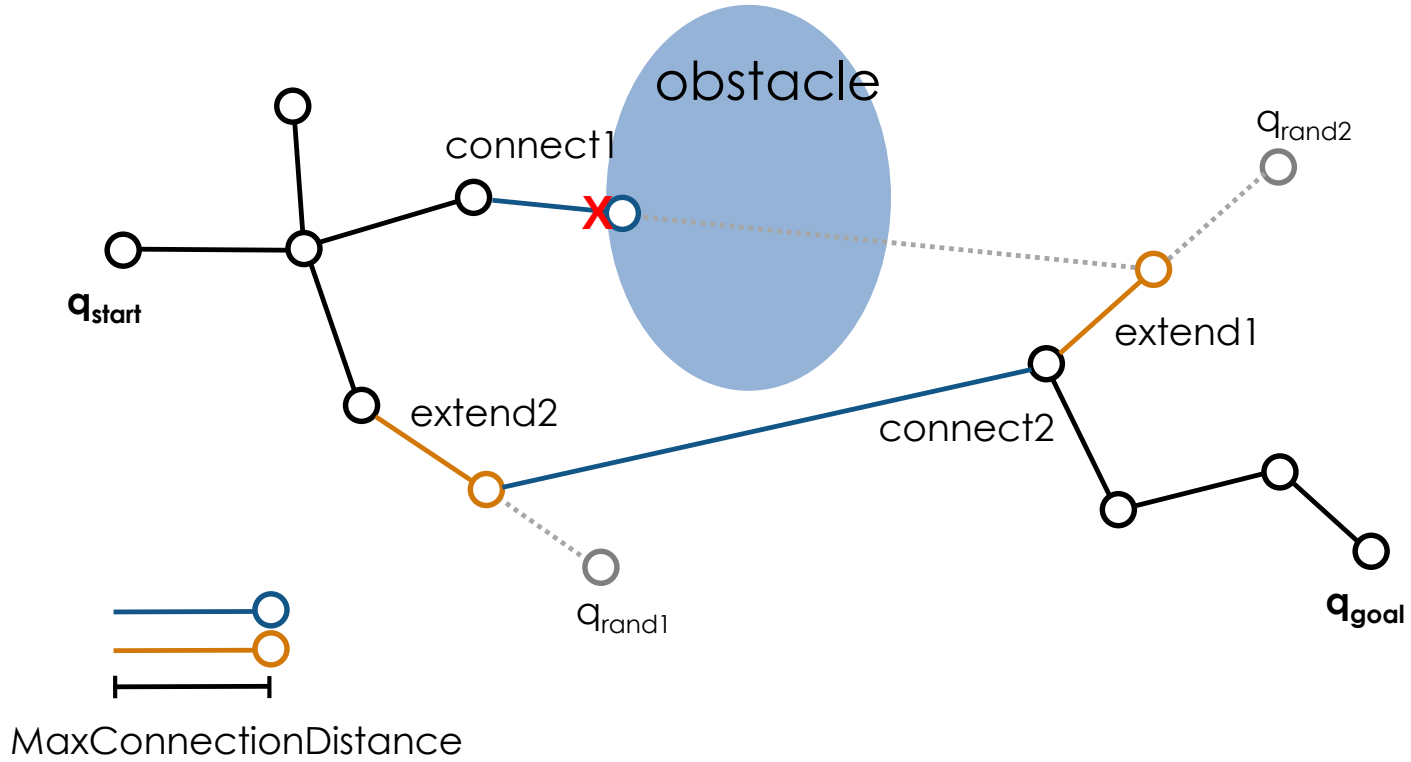
# 分拣机器人的完整工作流程



# 路径规划

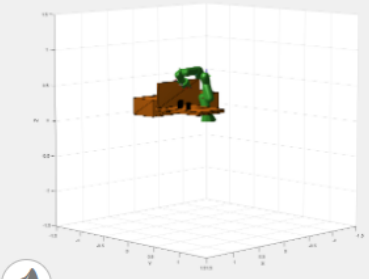

Start Configuration $q_{start}$
Goal Configuration $q_{goal}$
Environment
Path Planner
Path
obstacle
$T_{start}$
$q_{start}$
$q_{goal}$
$T_{goal}$

# 机械臂的双向 RRT算法




EnableConnectHeuristic = true

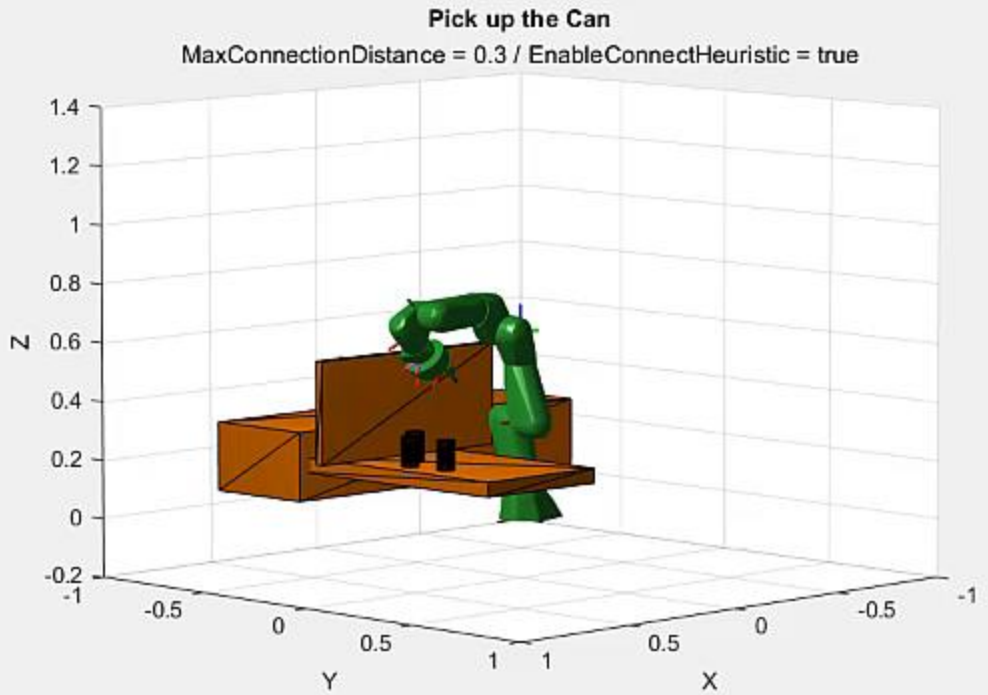
# 示例



**Pick and Place Using RRT for Manipulators**

Using manipulators to pick and place objects in an environment may require path planning algorithms like the rapidly-exploring random tree

https://www.mathworks.com/help/robotics/ug/pick-and-place-using-rrt-for-manipulators.html



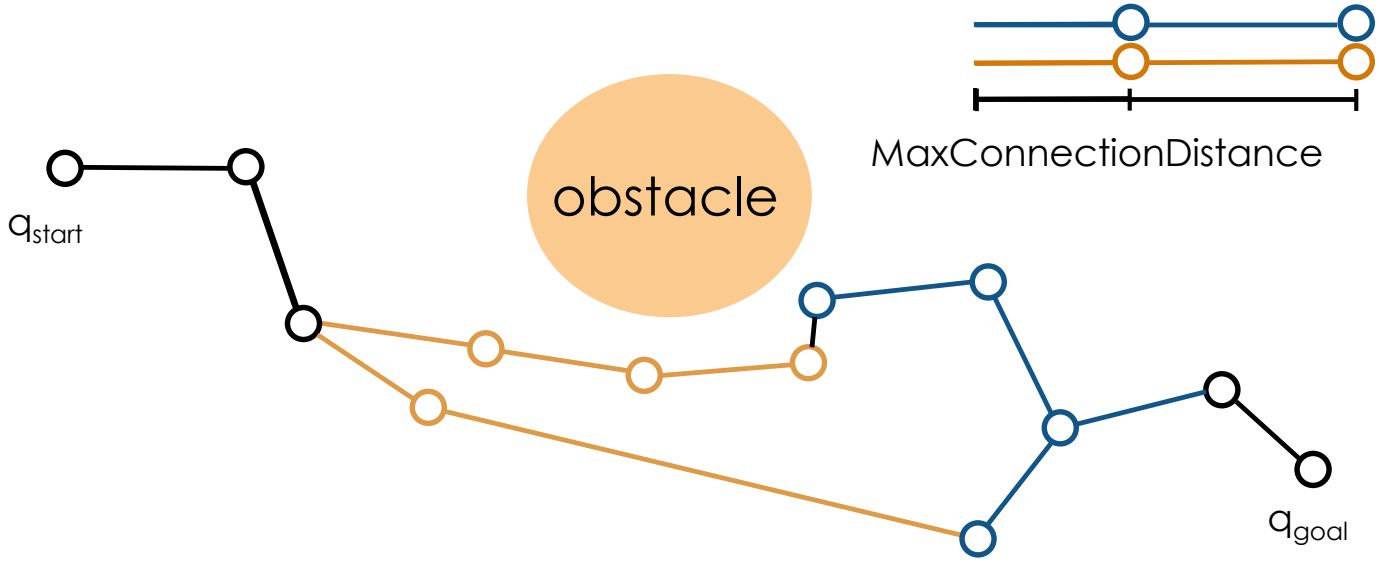
# 调整规划器

可调属性

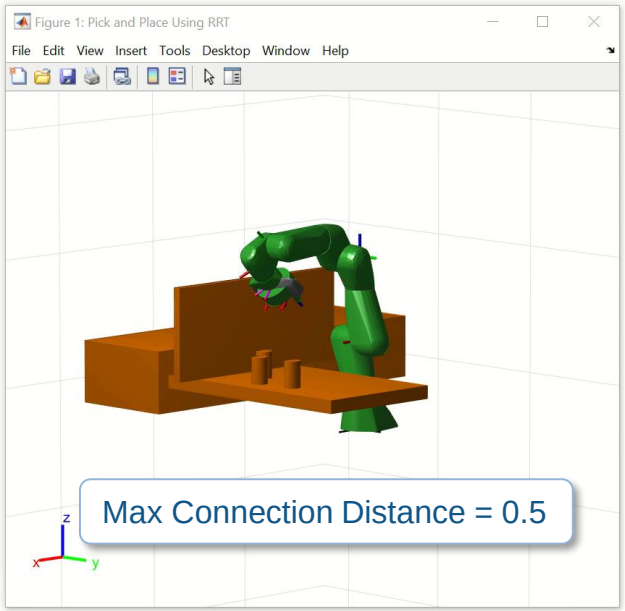
**最大连接距离**

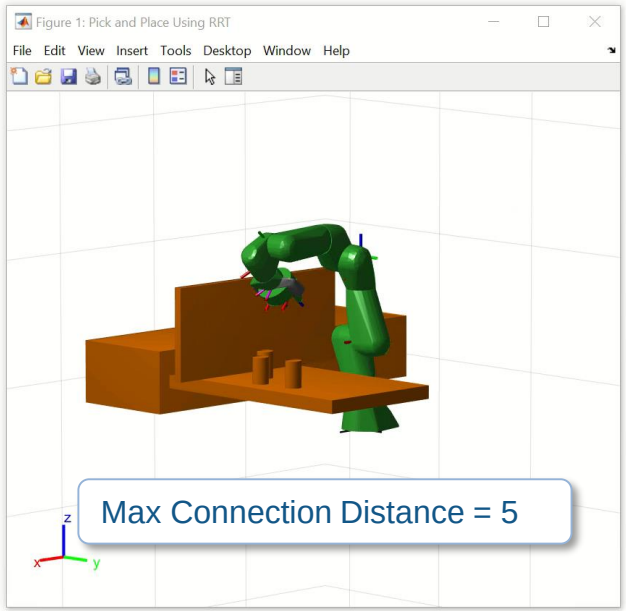
验证距离

使能连接启发式




EnableConnectHeuristic = false



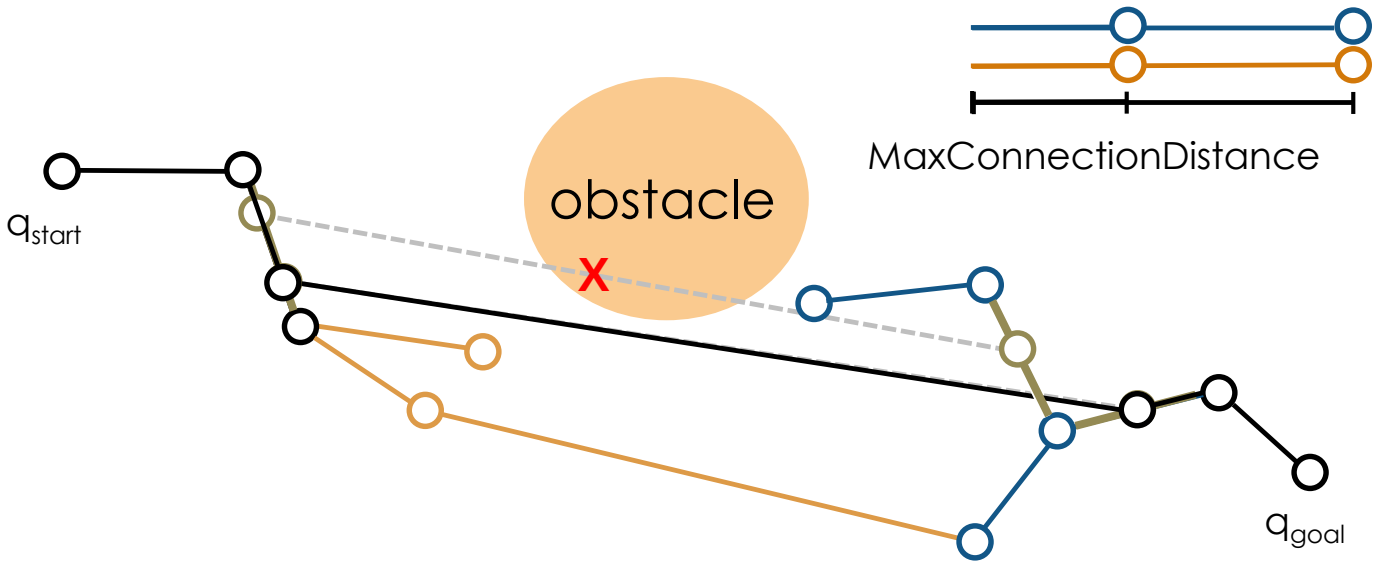
# 调整规划器

可调属性

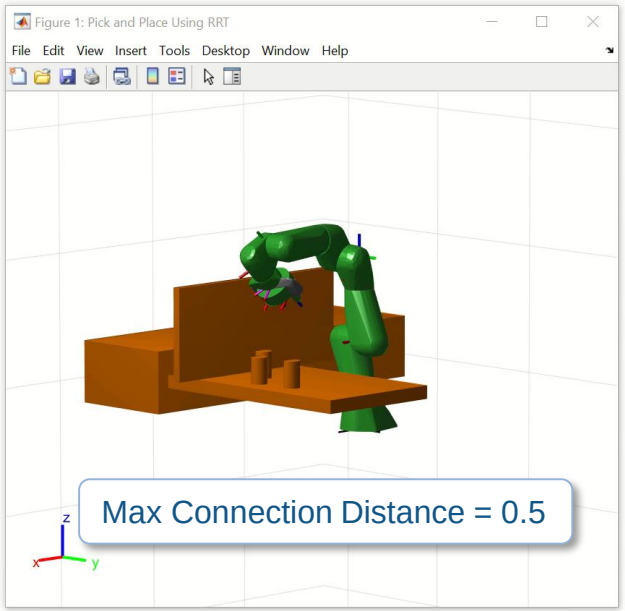
**最大连接距离**

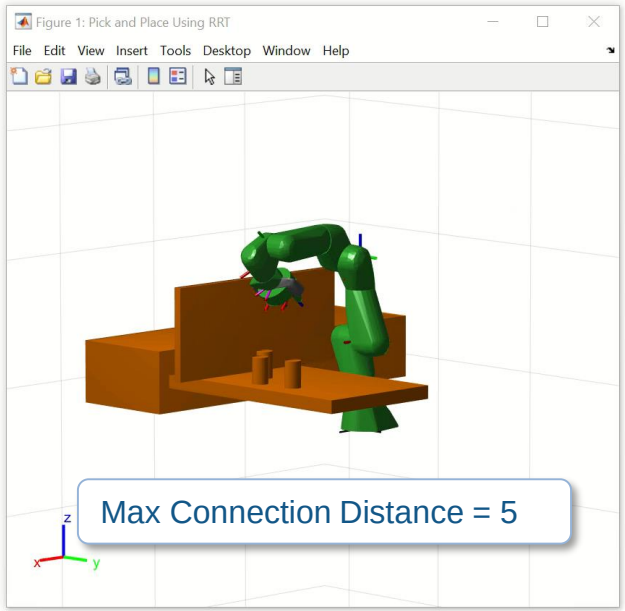
验证距离

使能连接启发式

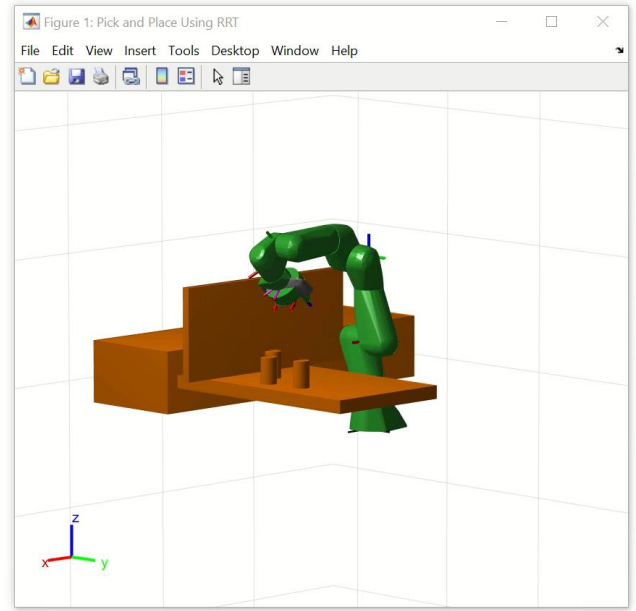



EnableConnectHeuristic = false




Shorten



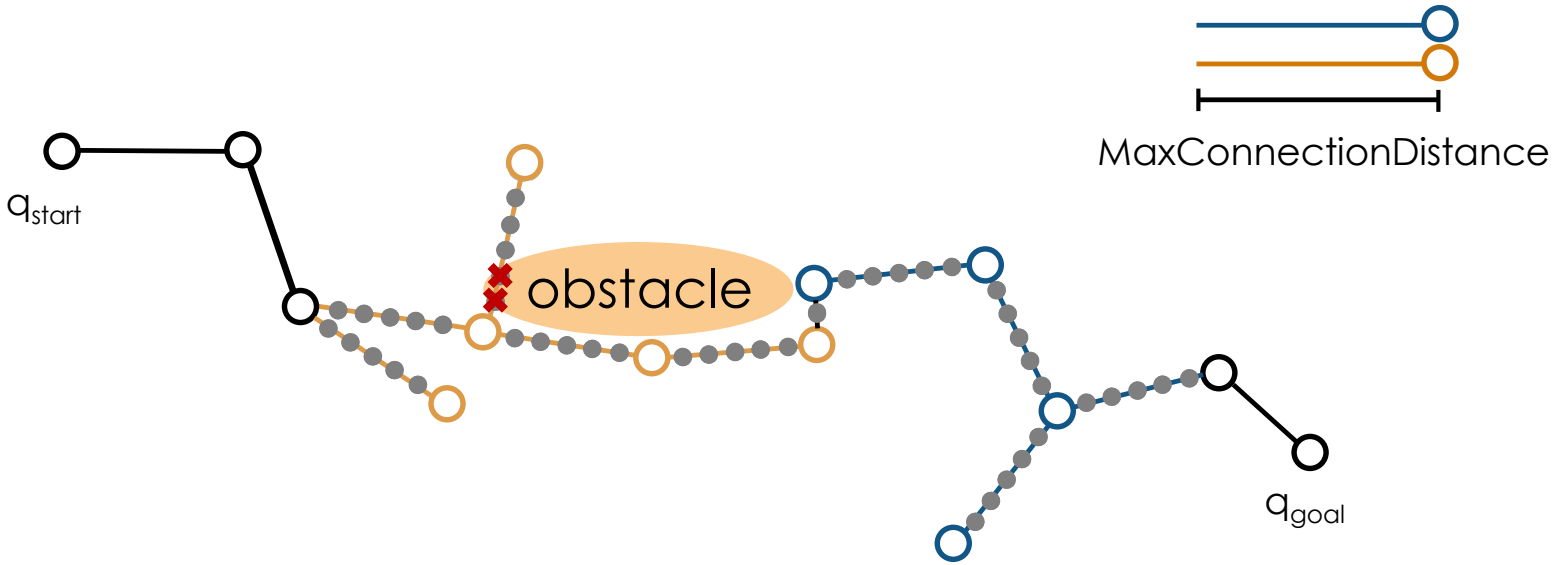
# 调整规划器

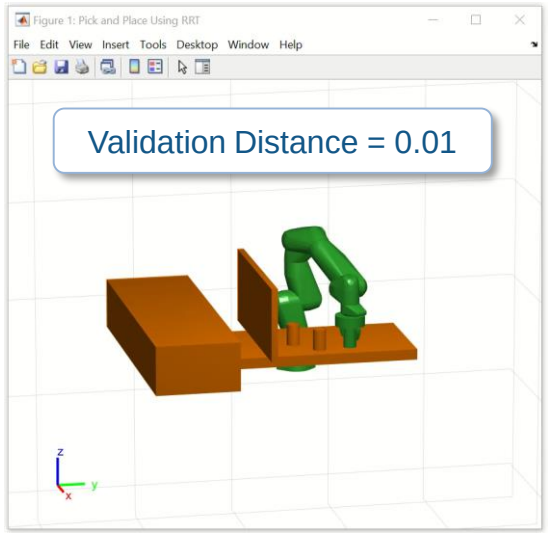
可调属性

最大连接距离

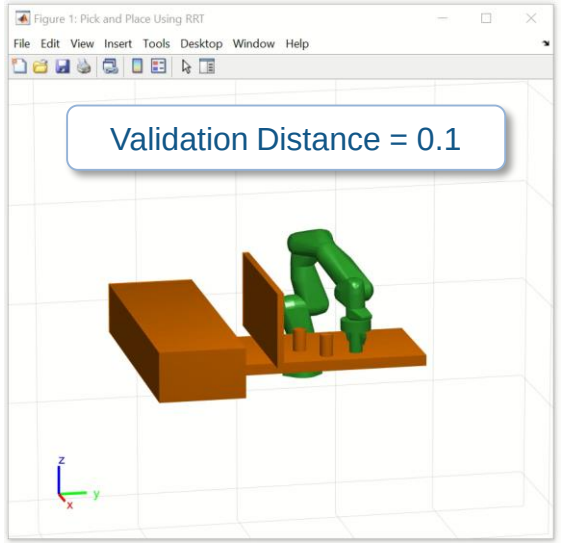
**验证距离**

使能连接启发式

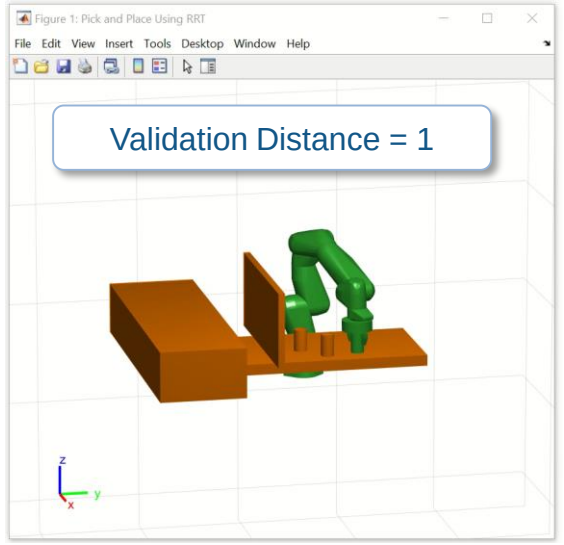



Average Plan Time: 16 s




Average Plan Time: 9 s




Average Plan Time: 0.5 s

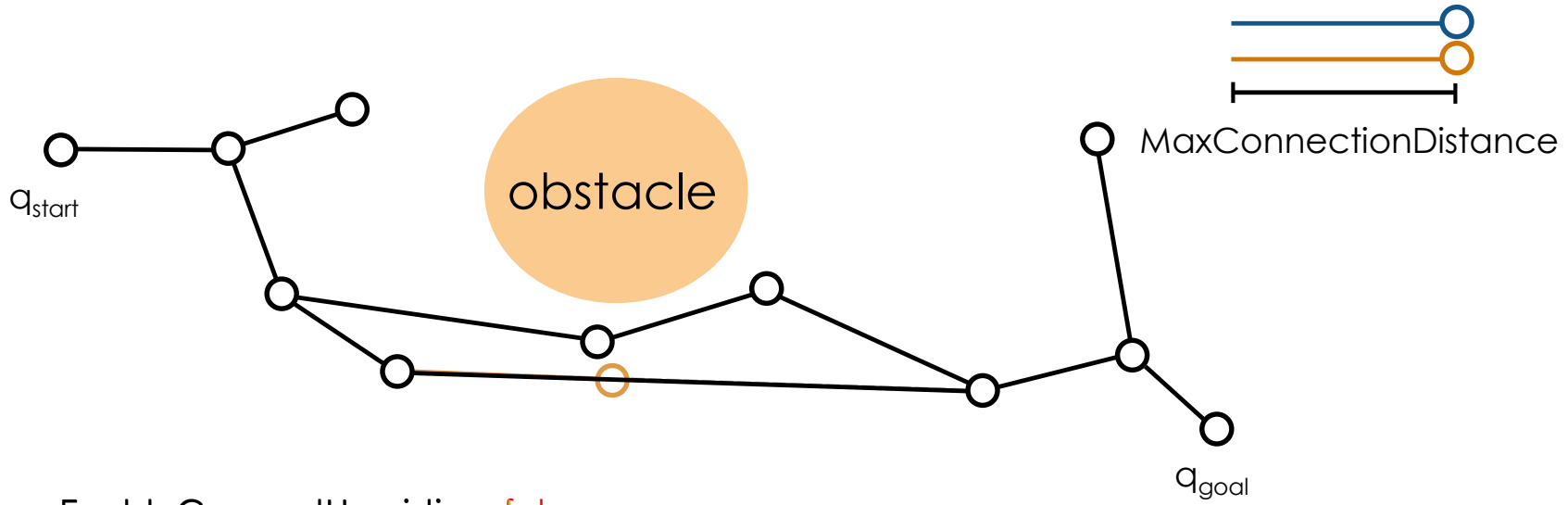
# 调整规划器

可调属性

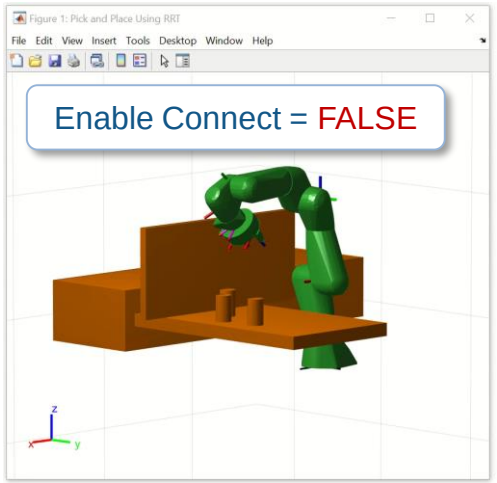
最大连接距离

验证距离

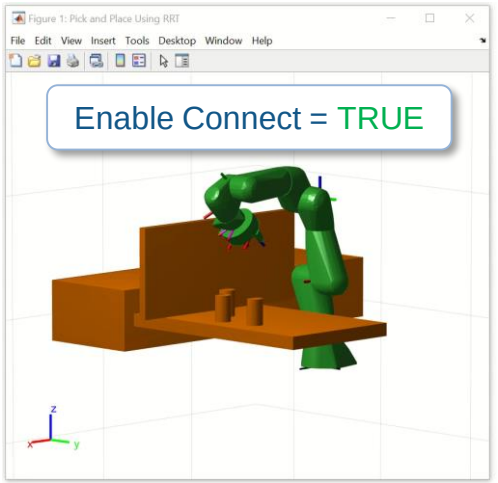
**使能连接启发式**




EnableConnectHeuristic = false




Average Plan Time: 3.0 s




Average Plan Time: 1.2 s

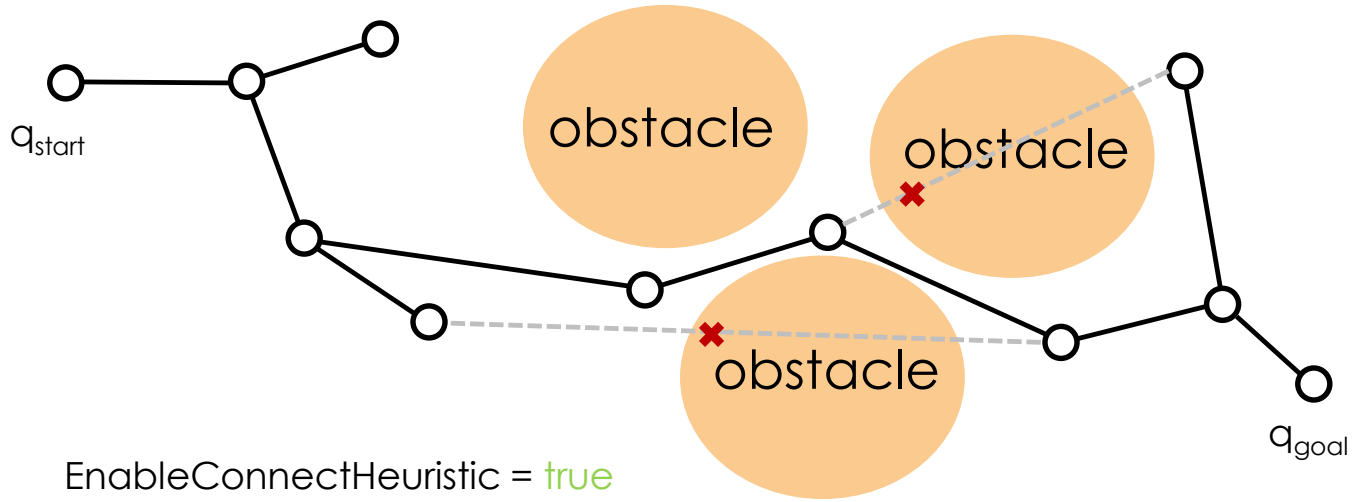
# 调整规划器

可调属性

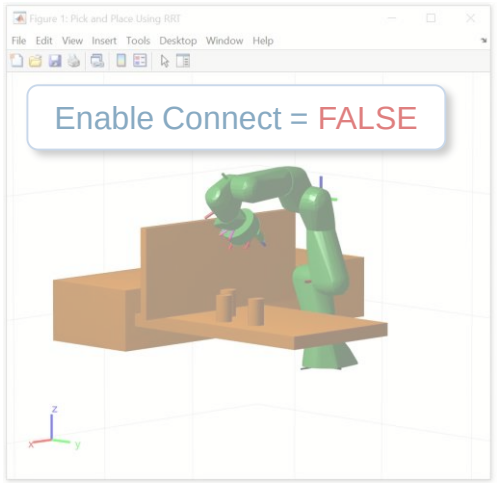
最大连接距离

验证距离

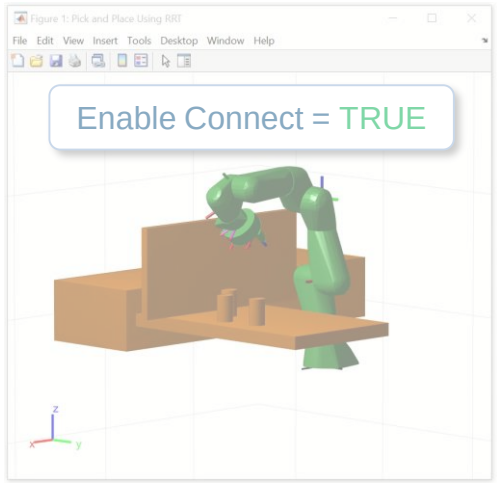
**使能连接启发式**



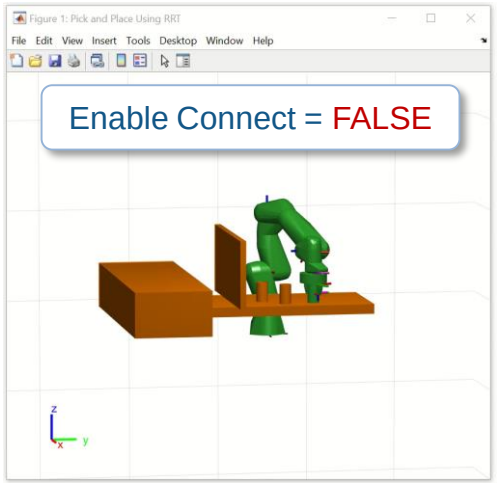

EnableConnectHeuristic = true



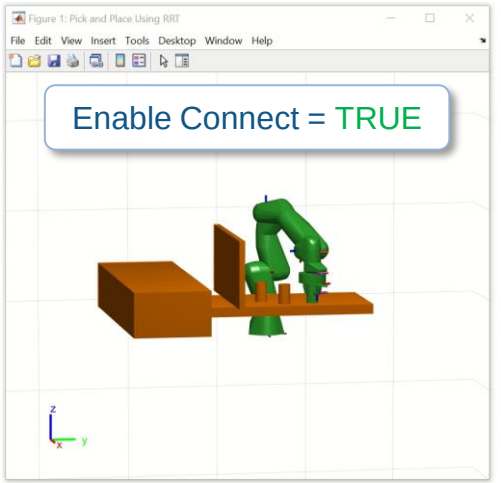

Average Plan Time: 3.0 s




Average Plan Time: 1.2 s

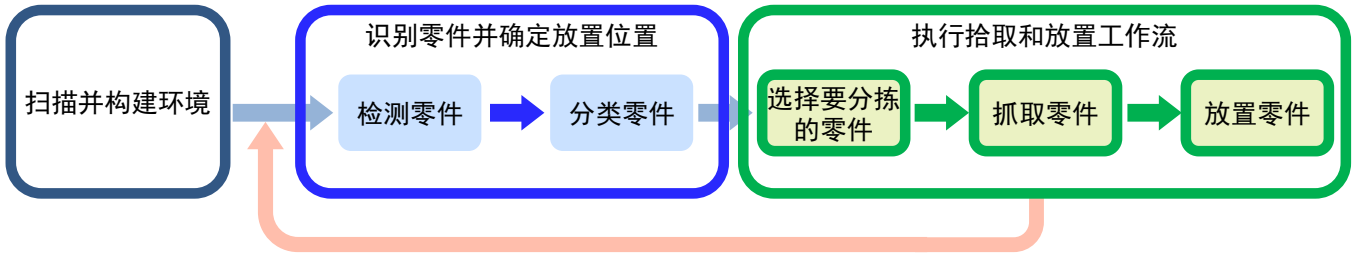



Average Plan Time: 5.6 s

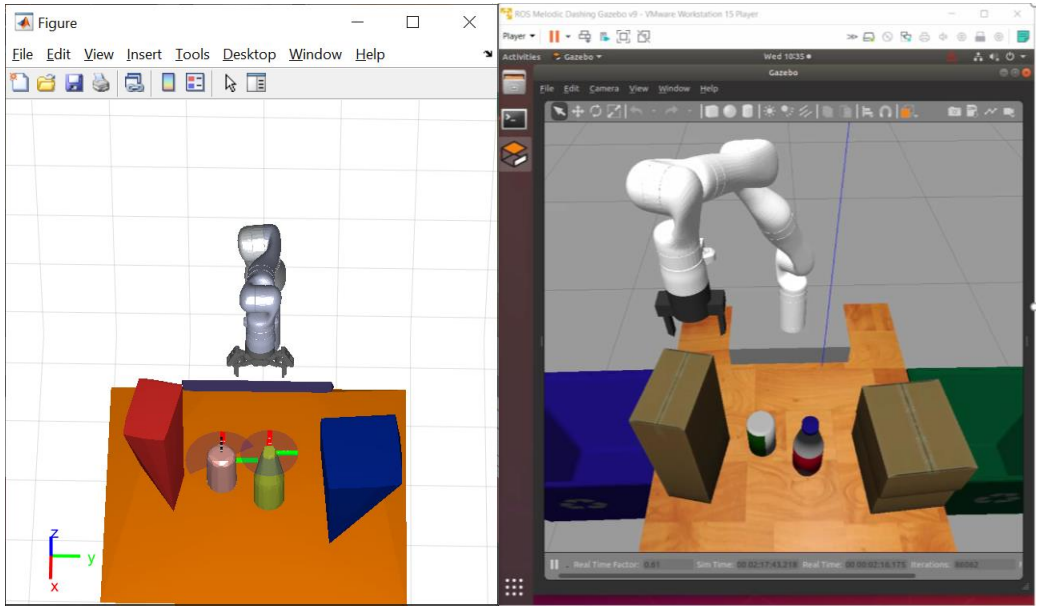


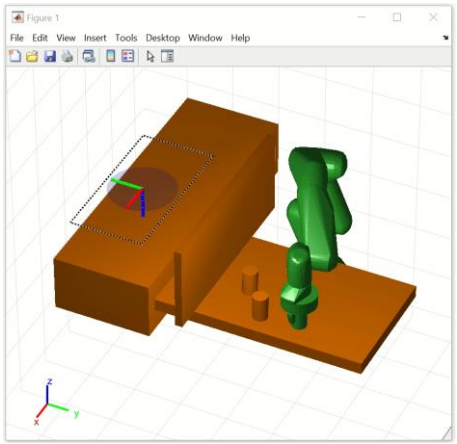

Average Plan Time: 4.5 s

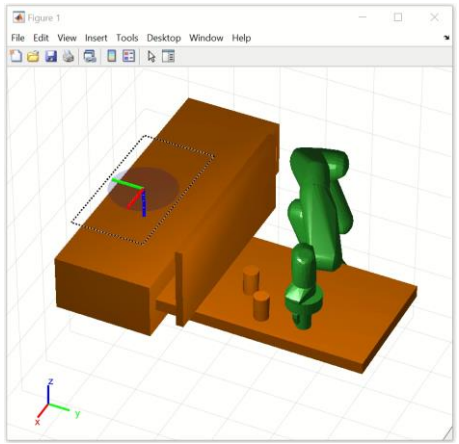
# 扩大运动规划器的范围

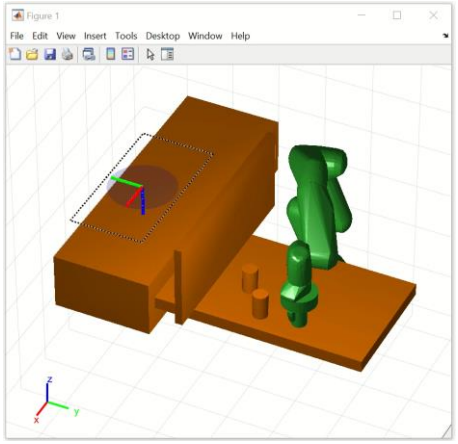



使用**workspace goal region** 规划区域而不是构型



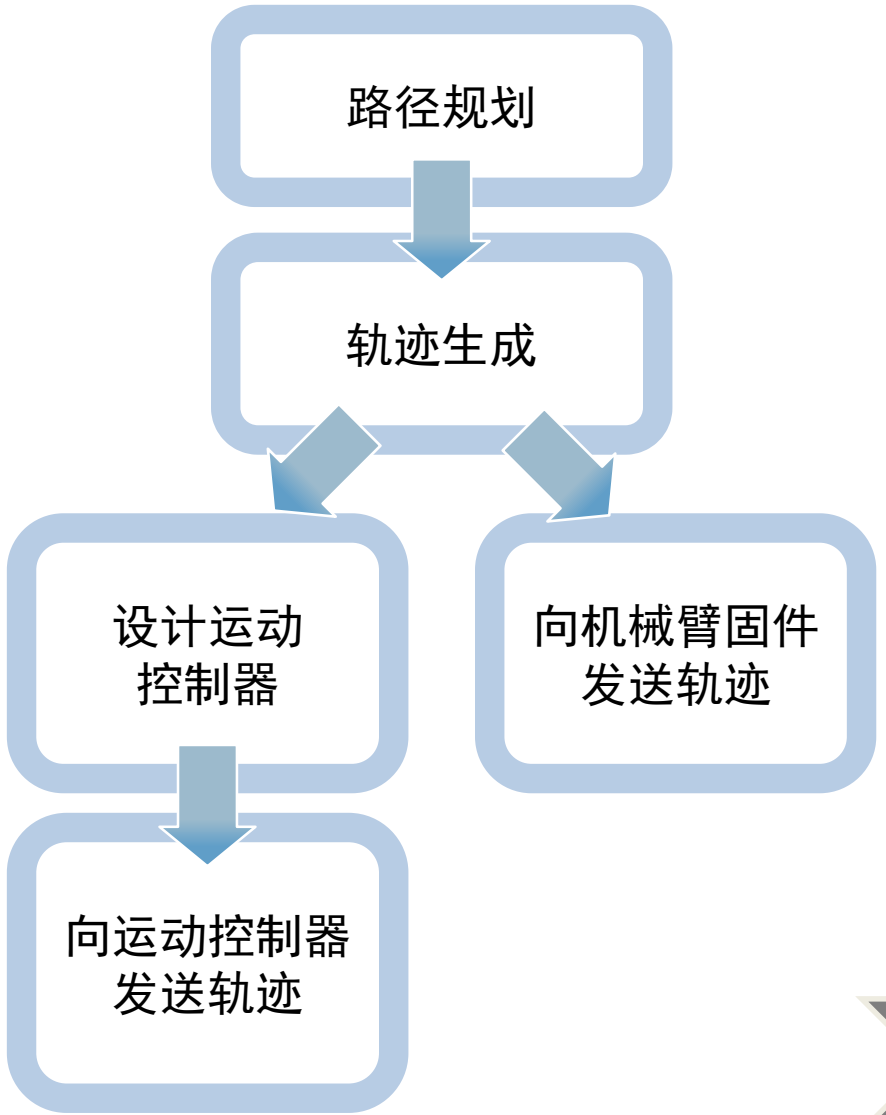


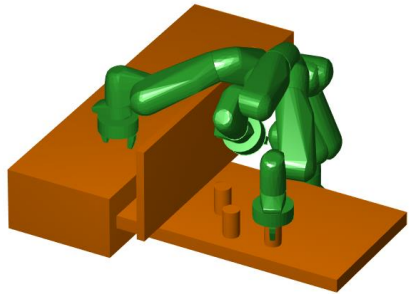


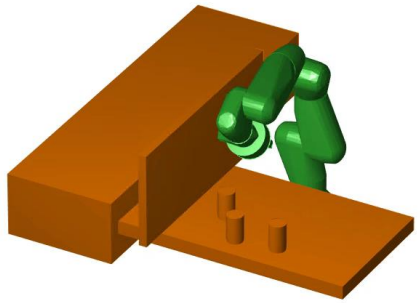
```
>> pickRegion = workspaceGoalRegion("sampleEE")
>> pickRegion.ReferencePose = bottlePose;
>> pickRegion.Bounds(4,:) = [-pi pi];


>> placeRegion = workspaceGoalRegion("sampleEE")
...
>> planner = manipulatorRRT(robot, env);
>> path = plan(planner, startConfig, placeRegion)
```

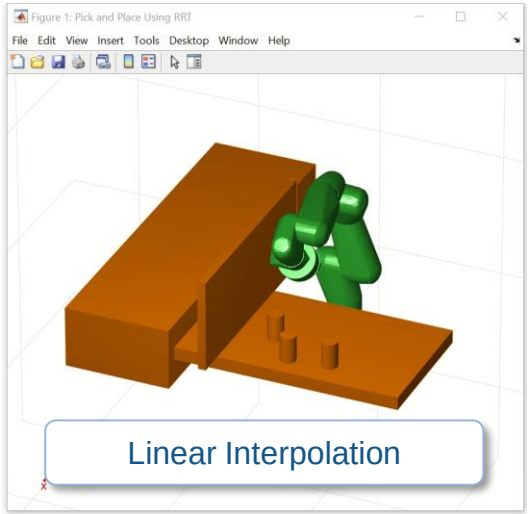
# 轨迹生成

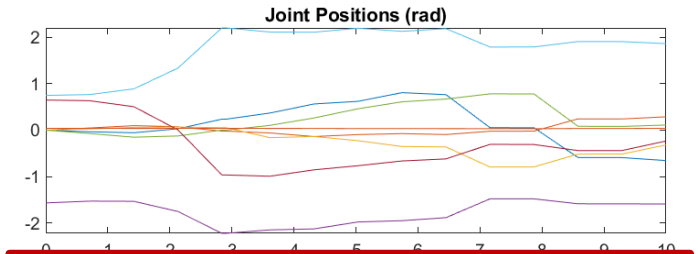


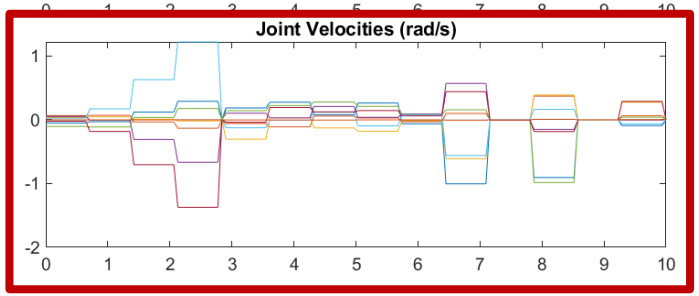





输入构型 → Path Planner → 路径构型的特定序列 → Trajectory Generator → 关节轨迹（随着时间的移动）



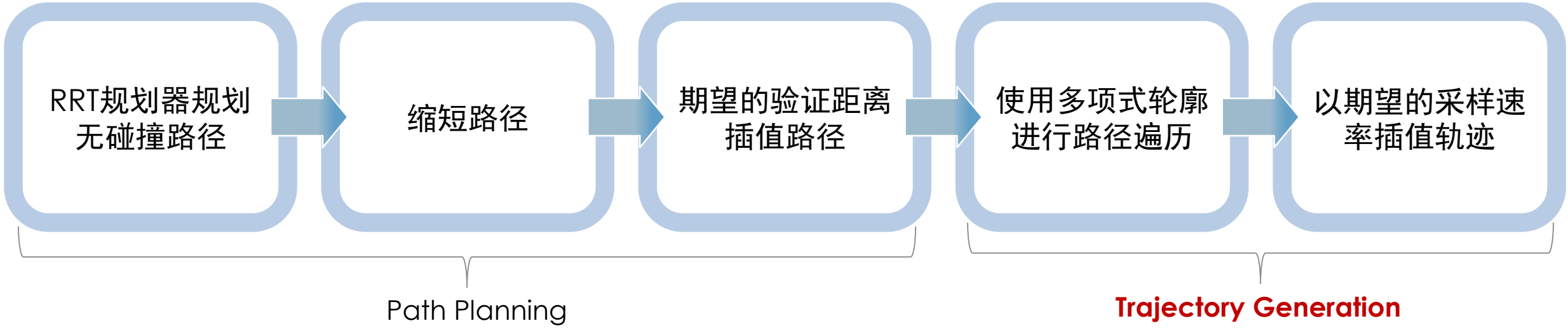



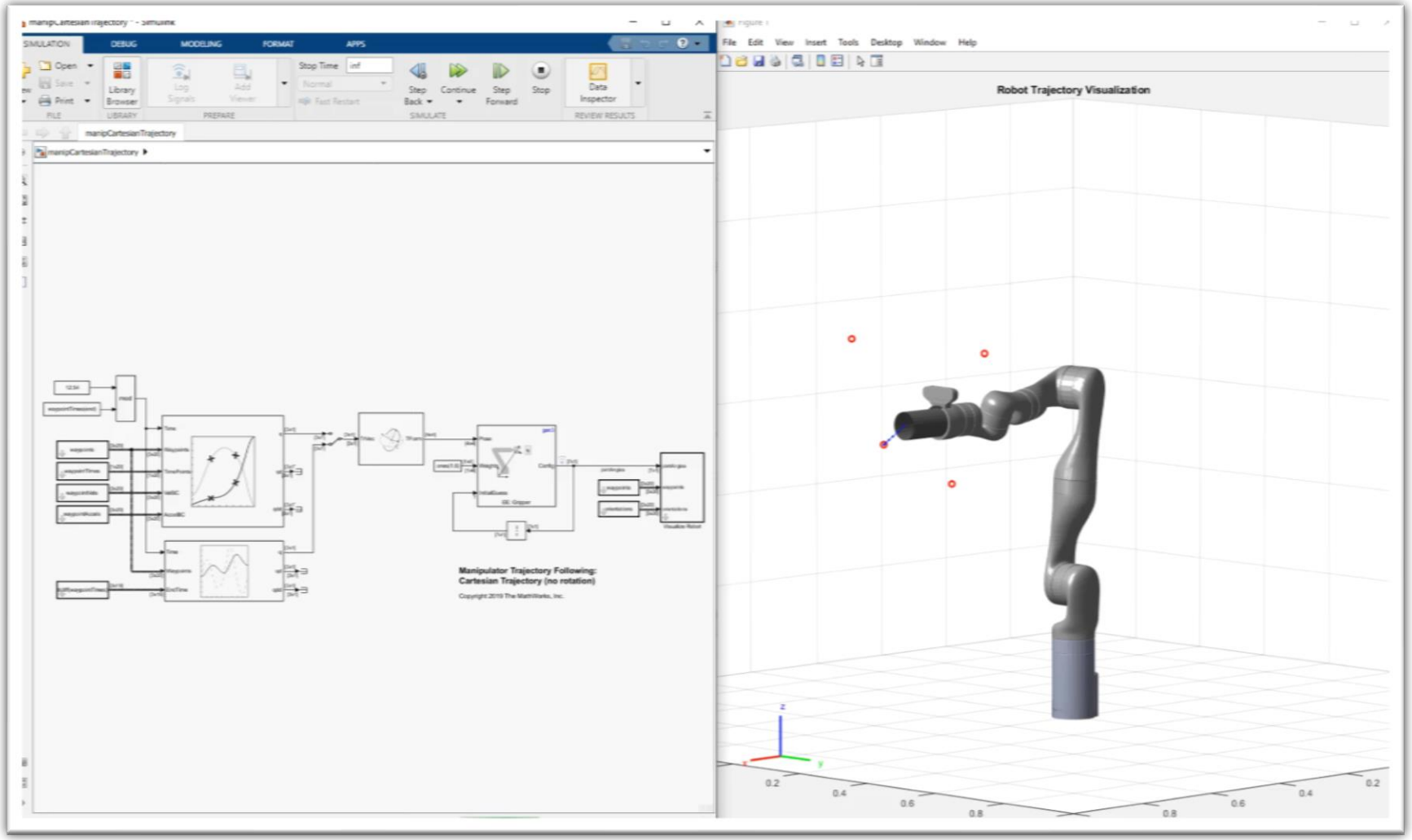

向控制器发送轨迹 → 验证轨迹 → 执行路径

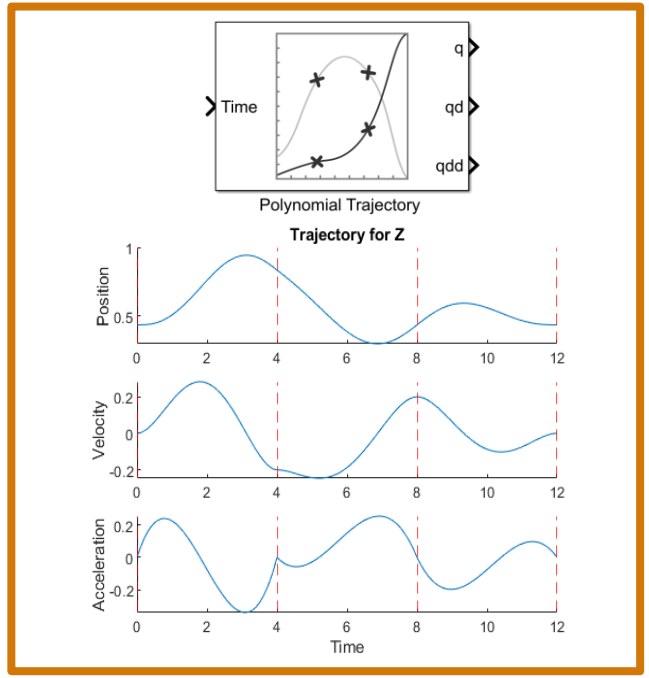
# 轨迹生成

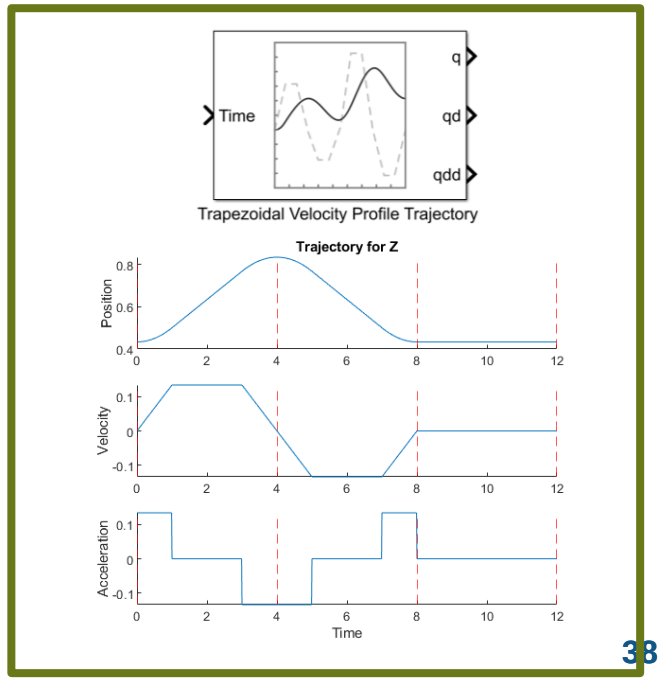
- 任务 –生成一个基于时间的控制序列，将机械手从初始构型移动到目的地
- 局部使用一类多项式函数连接路径点



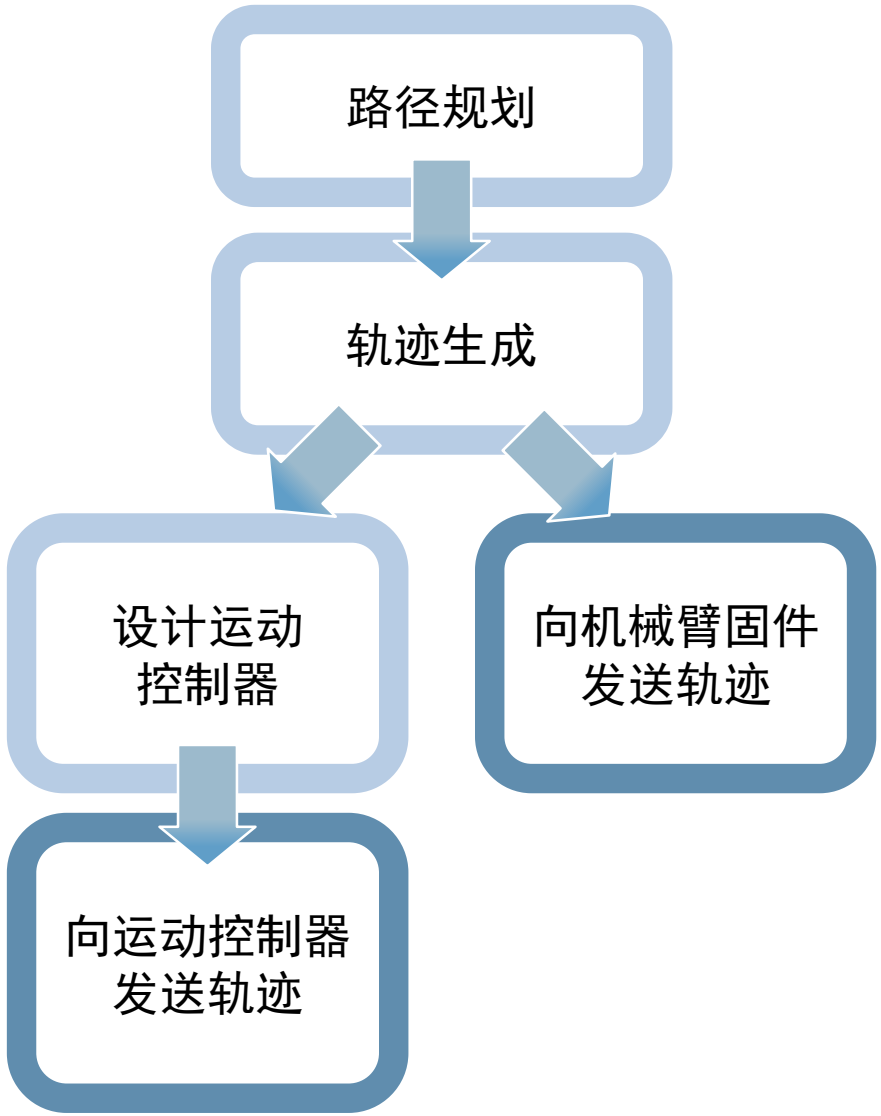
# 轨迹生成


Robot Trajectory Visualization
Manipulator Trajectory Following:
Cartesian Trajectory (no rotation)

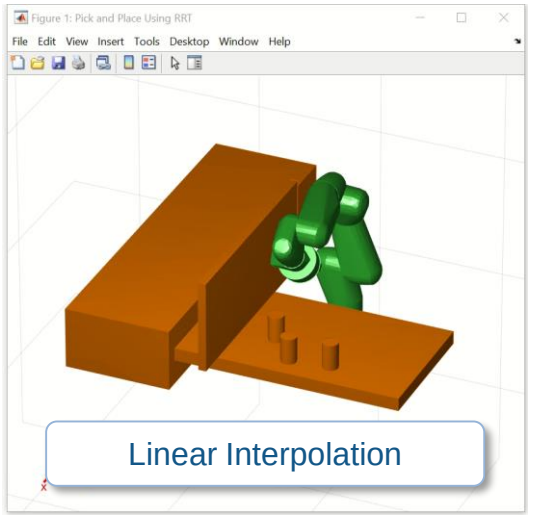

Time
q
qd
qdd
Polynomial Trajectory
Trajectory for Z
Position
Velocity
Acceleration
Time

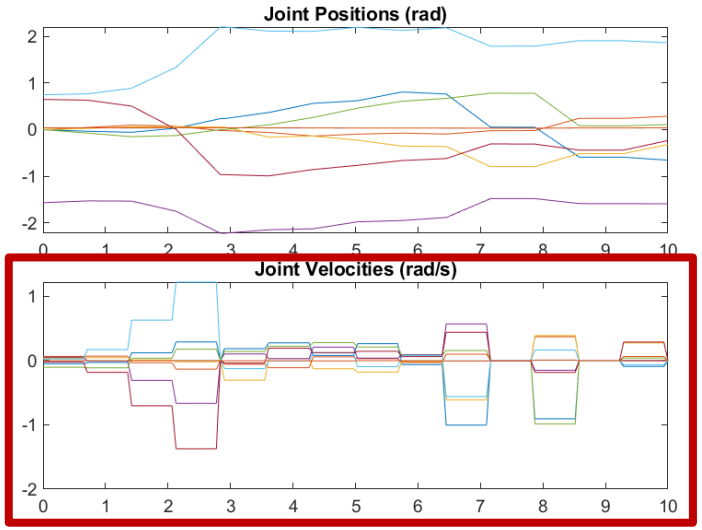

Time
q
qd
qdd
Trapezoidal Velocity Profile Trajectory
Trajectory for Z
Position
Velocity
Acceleration
Time

# 轨迹生成

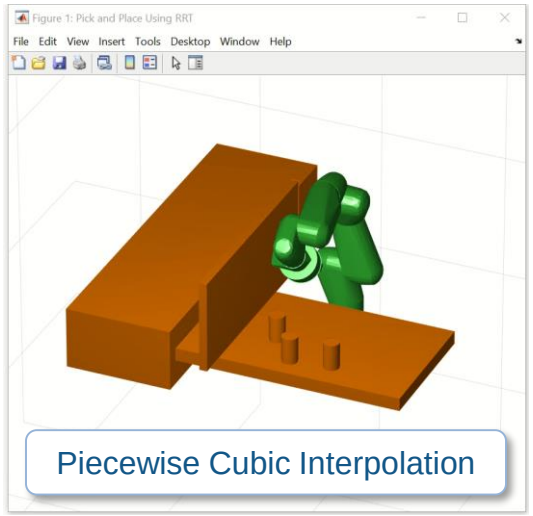


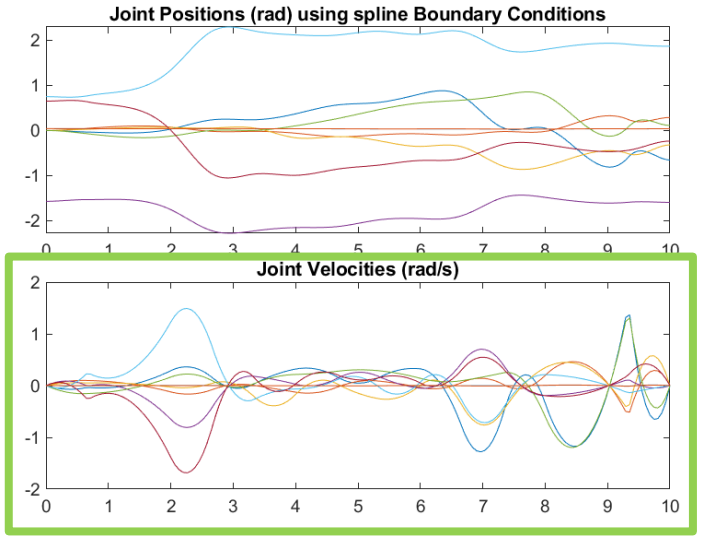

向控制器发送轨迹 → 验证轨迹 → 执行路径




```
>> [q, qd, qdd] = cubicpolytraj(wpts, tpts, tvec, ...
    'VelocityBoundaryCondition', splineVelBounds)
```



# 运动控制

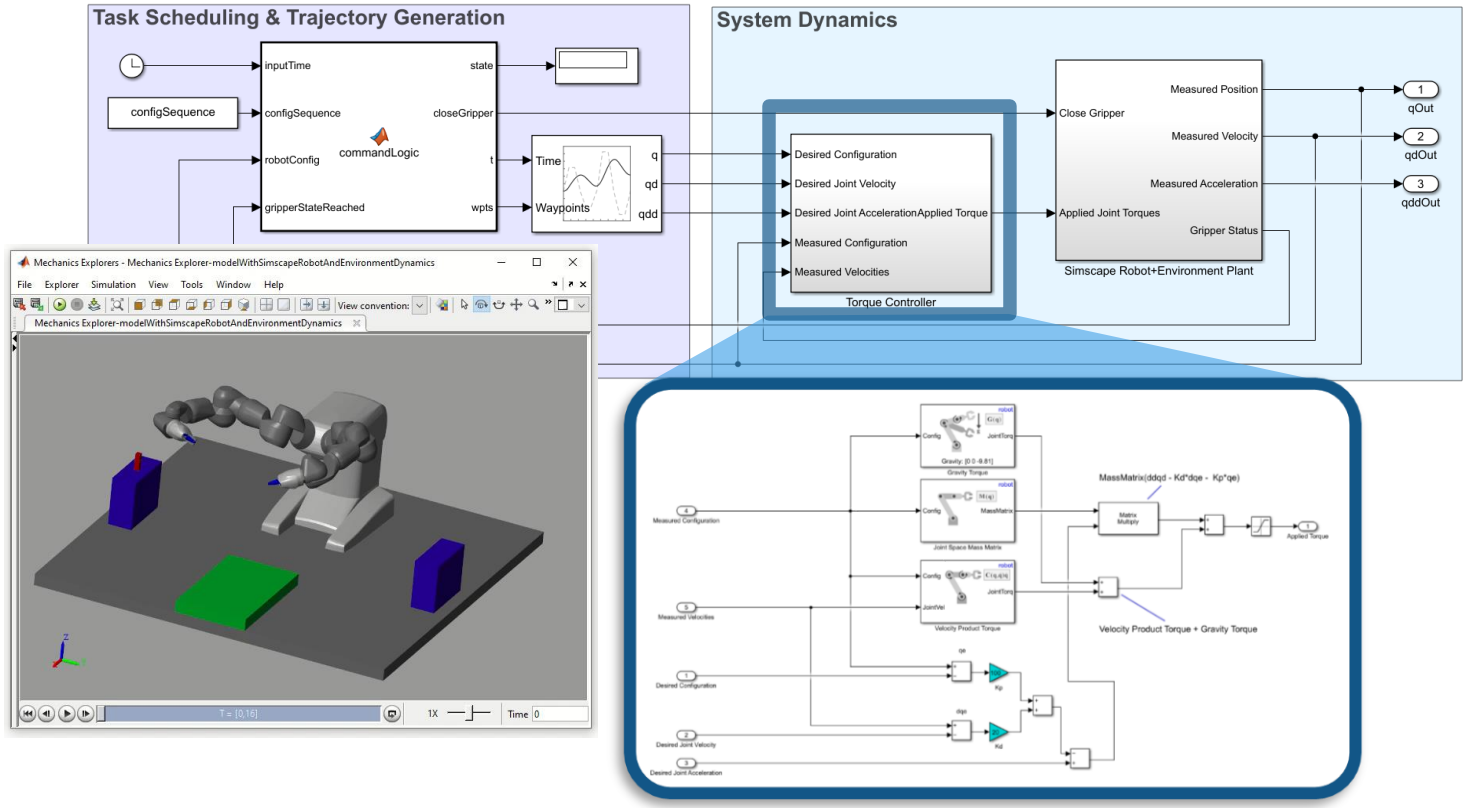
路径规划

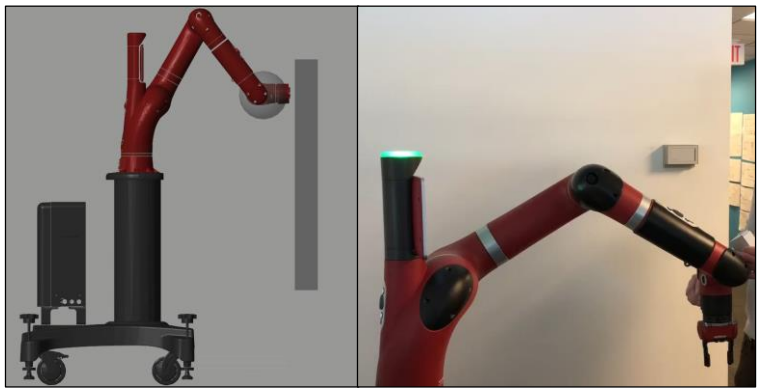
轨迹生成

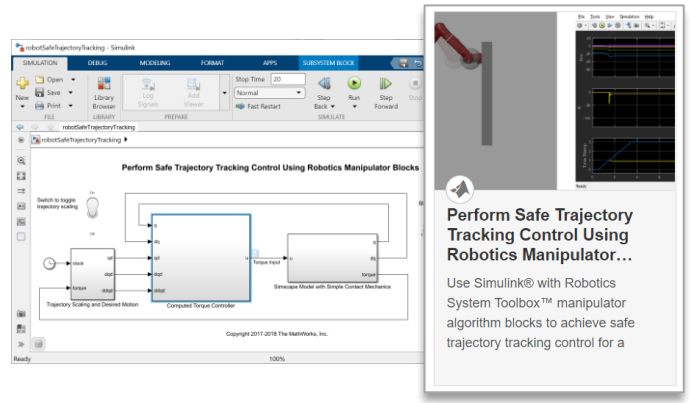
设计运动控制器

向机械臂固件发送轨迹

向运动控制器发送轨迹





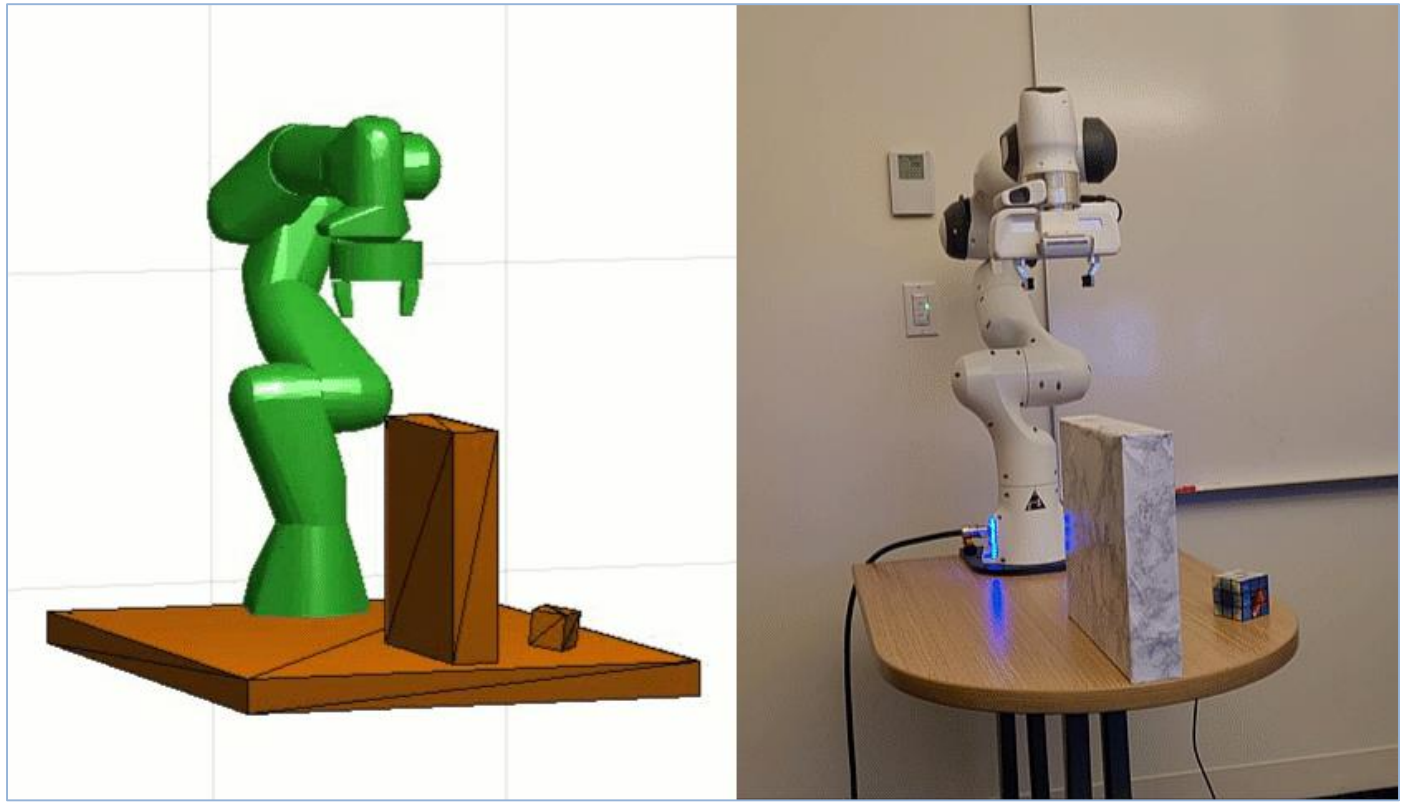
# 示例演示

问题定义

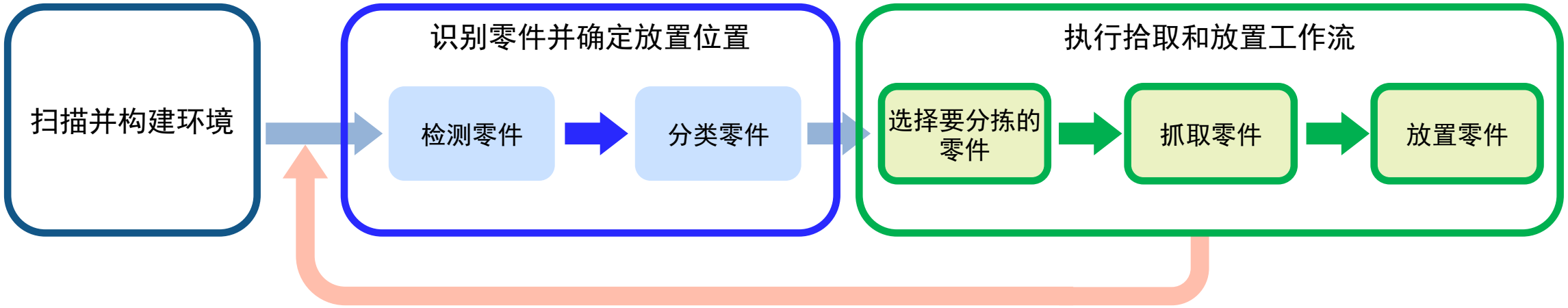
路径规划

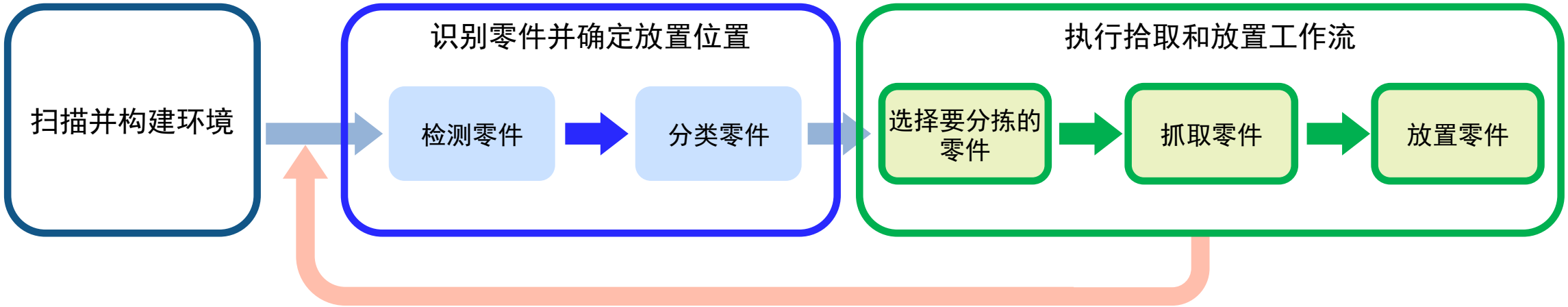
轨迹生成

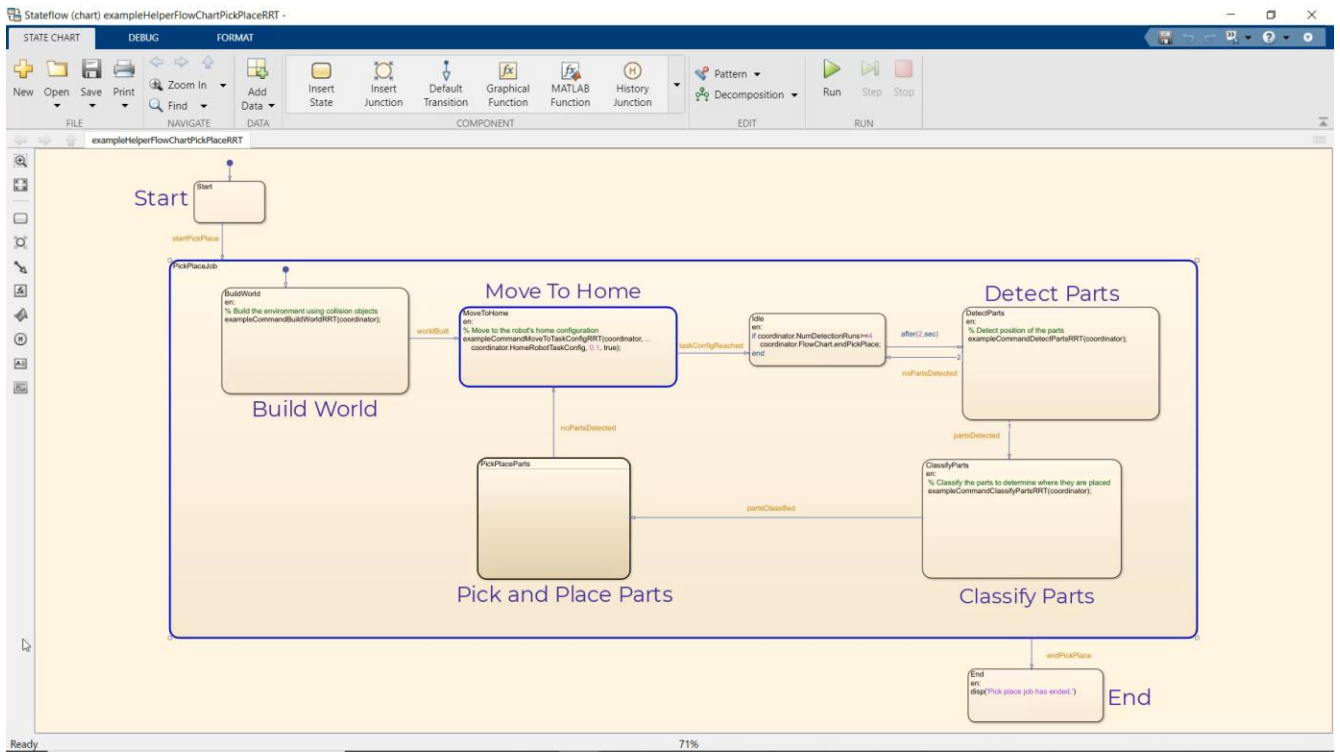
运动控制



# 分拣机器人的完整工作流程

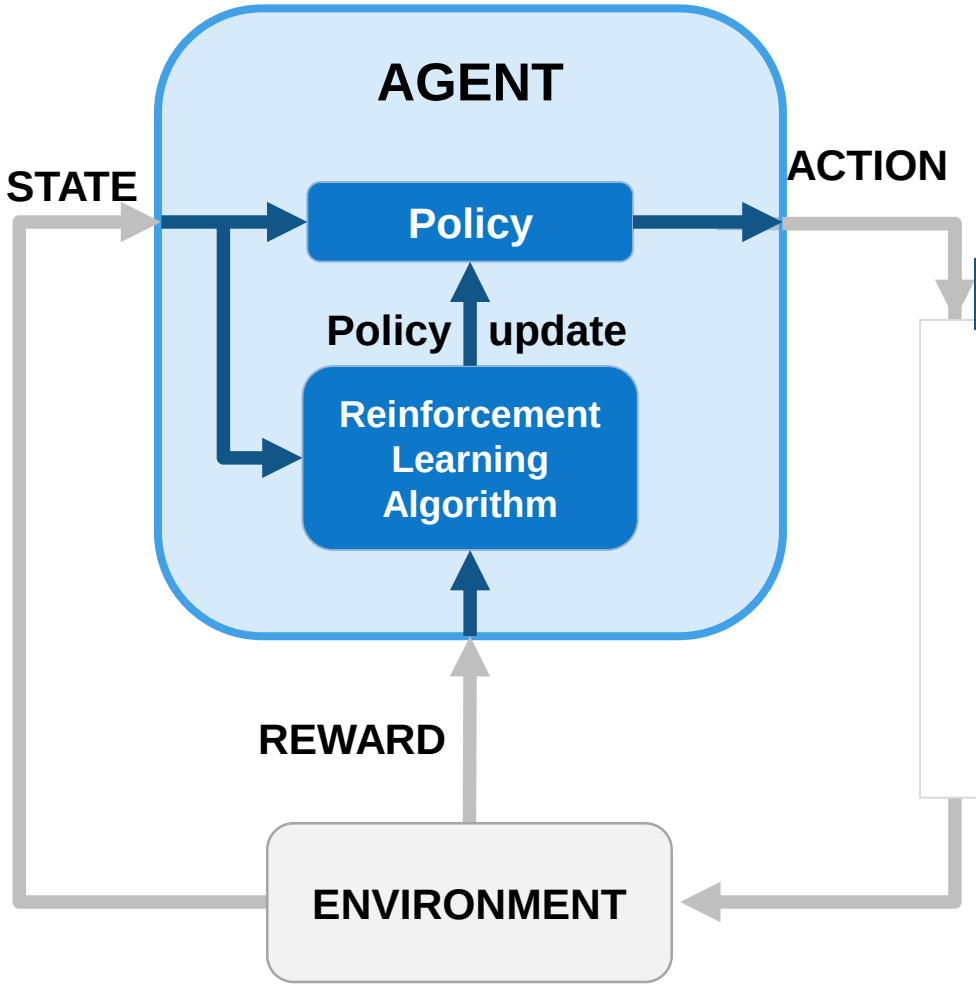



扫描并构建环境
识别零件并确定放置位置
检测零件
分类零件
执行拾取和放置工作流
选择要分拣的零件
抓取零件
放置零件

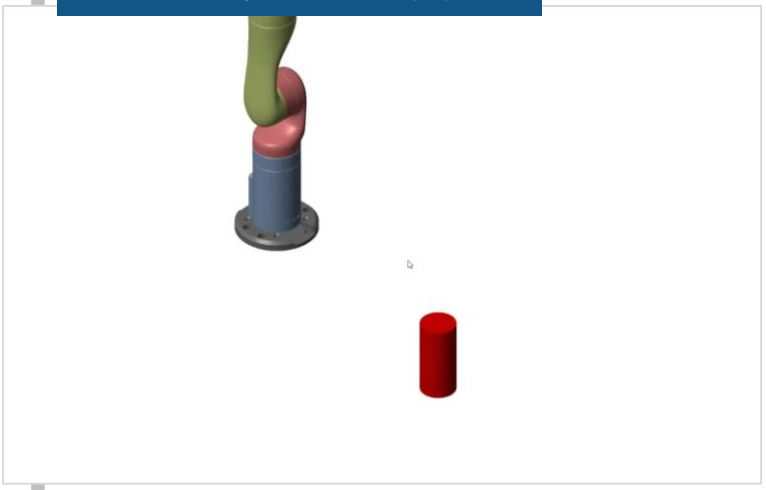

Stateflow (chart) exampleHelperFlowChartPickPlaceRRT
STATE CHART
DEBUG
FORMAT
exampleHelperFlowChartPickPlaceRRT
Start
Move To Home
Detect Parts
Build World
Pick and Place Parts
Classify Parts
End
Ready
71%

# 先进控制:强化学习

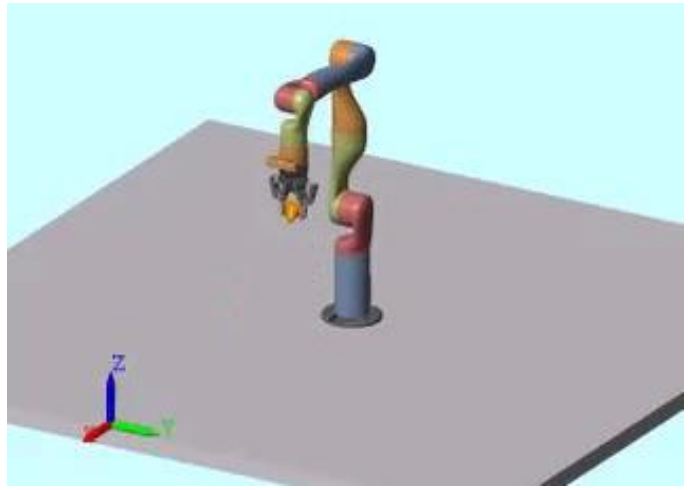
## 训练机器人到达目标对象




训练到达目标对象

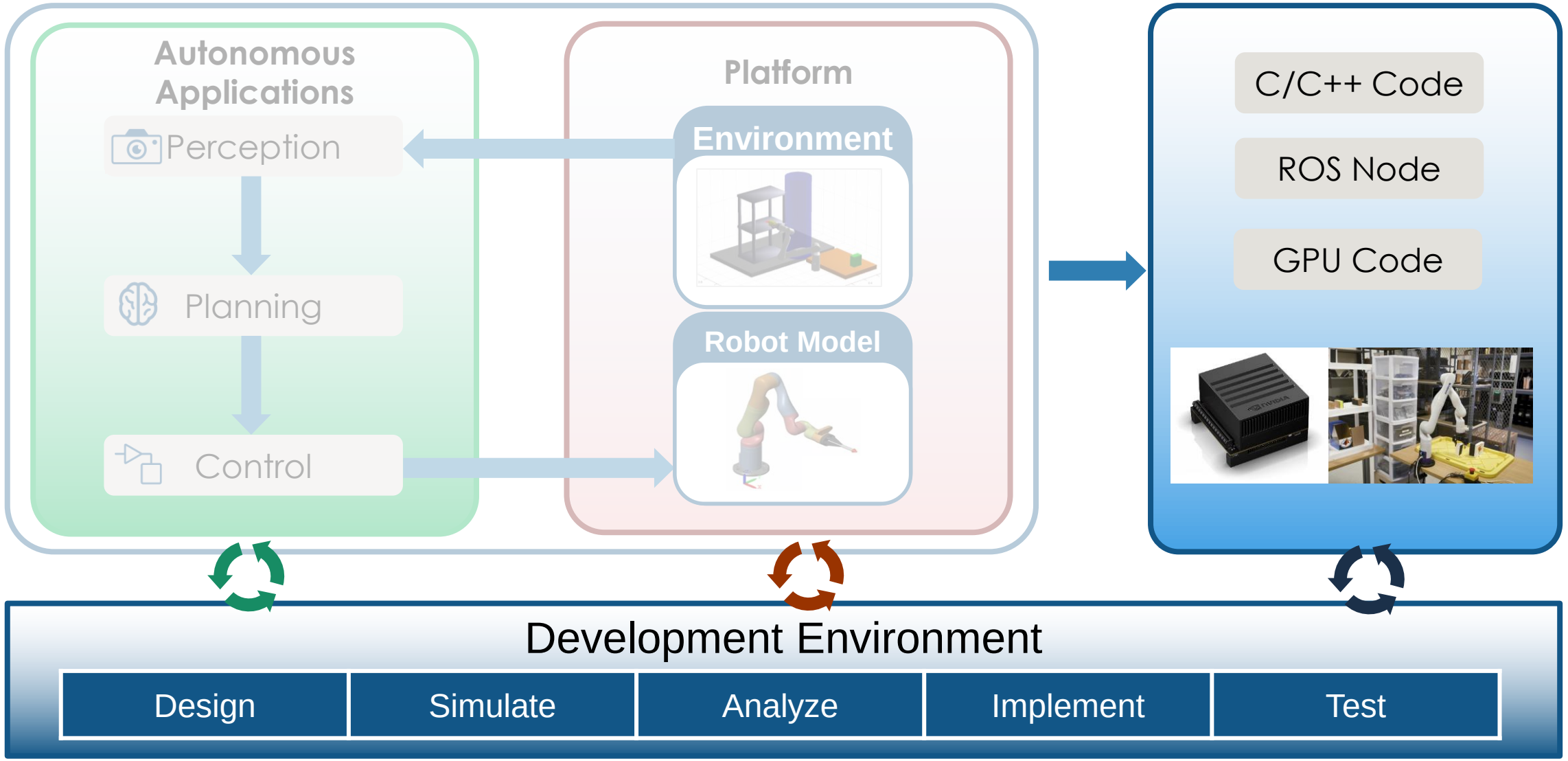


训练抓住目标对象



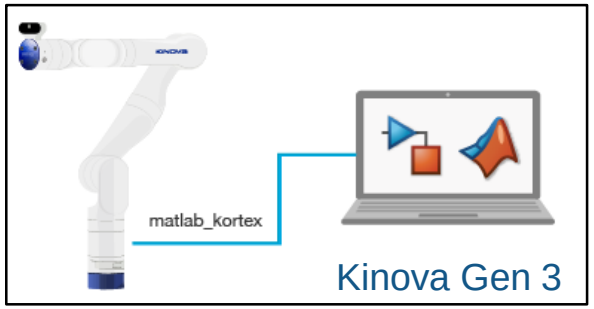
*Reinforcement Learning Toolbox, Simscape Multibody*

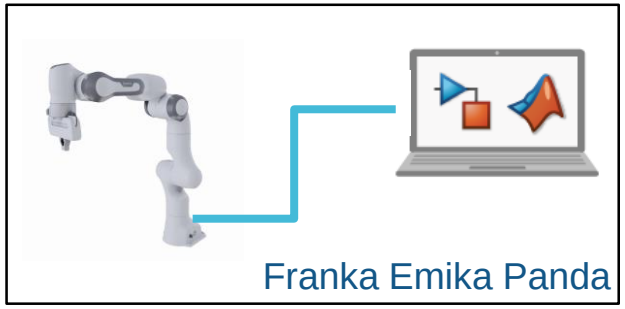
# 使用 MATLAB 和 Simulink开发自主机器人



# 硬件连接和部署

1



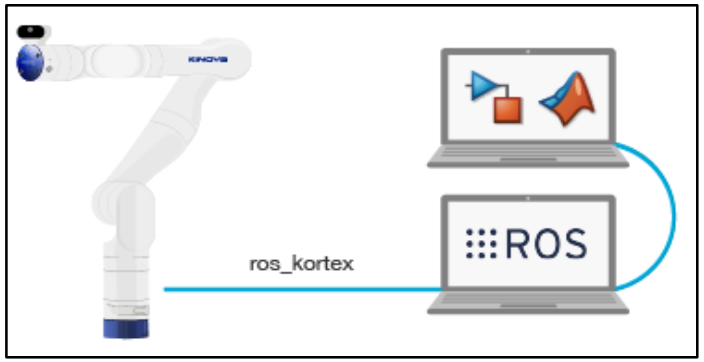



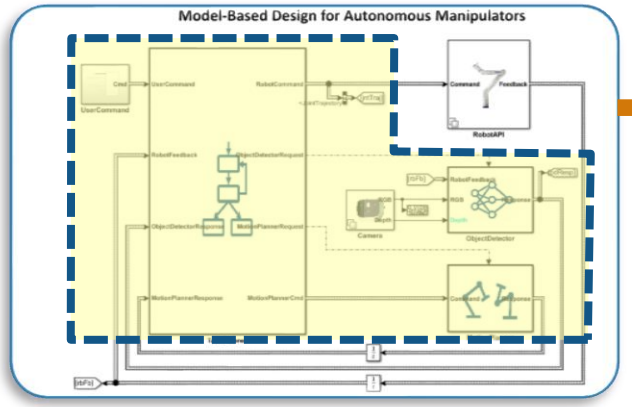

MATLAB APIs

2

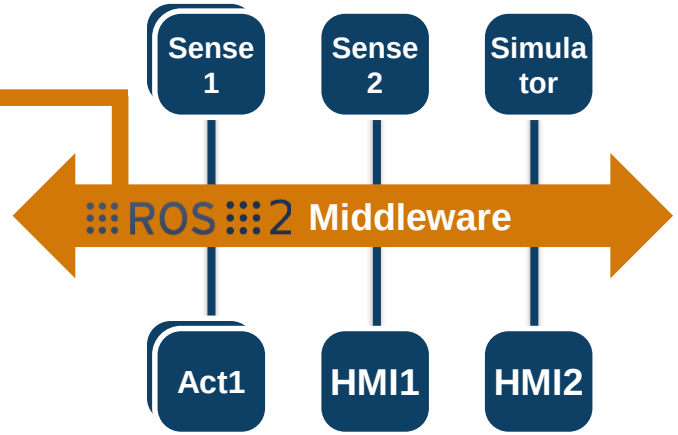
在MATLAB和 Simulink中设计算法 → 生成代码 → 使用packNGo打包 → 与现有C/C++ APIs集成

3



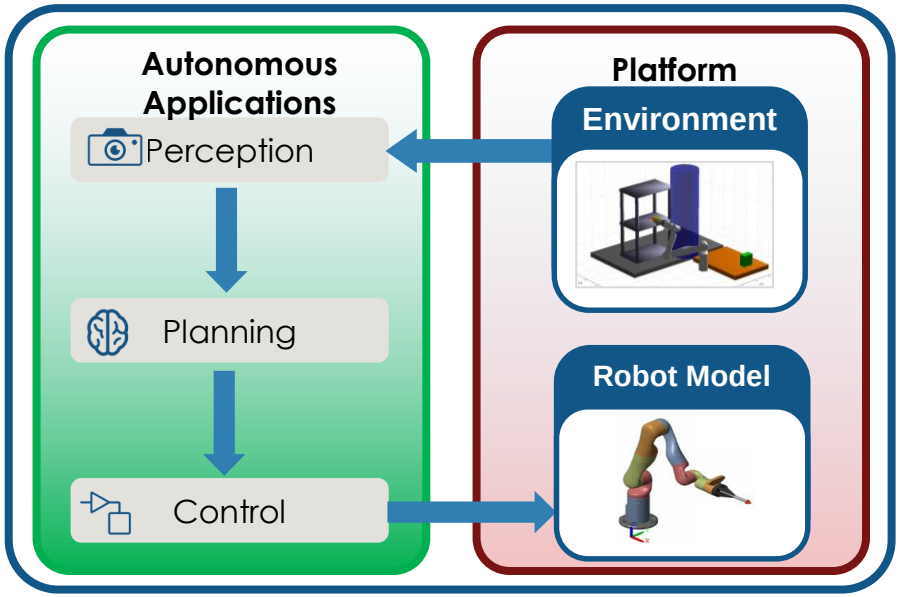



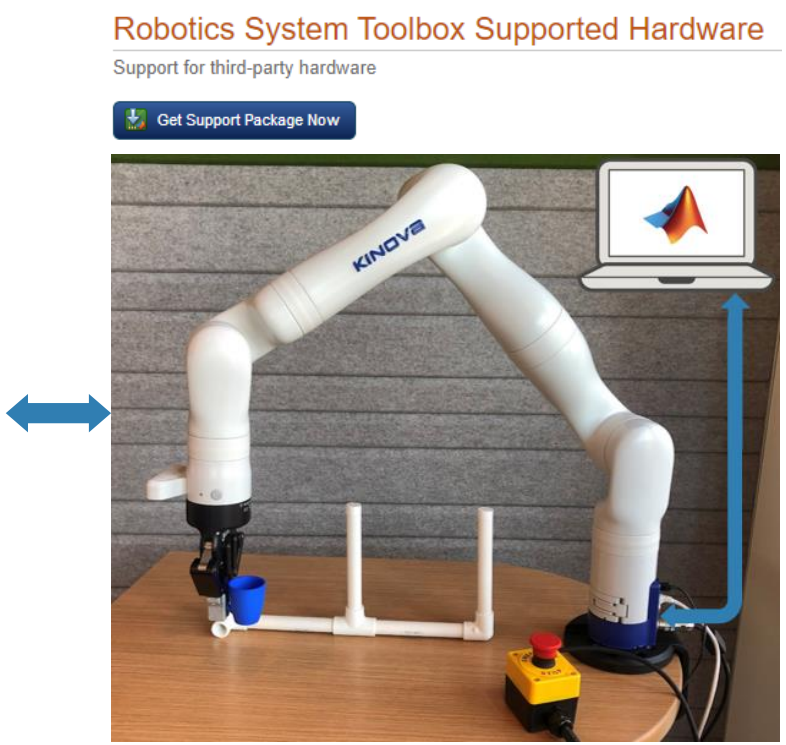

Application




*Robotics System Toolbox, ROS Toolbox, Coders*

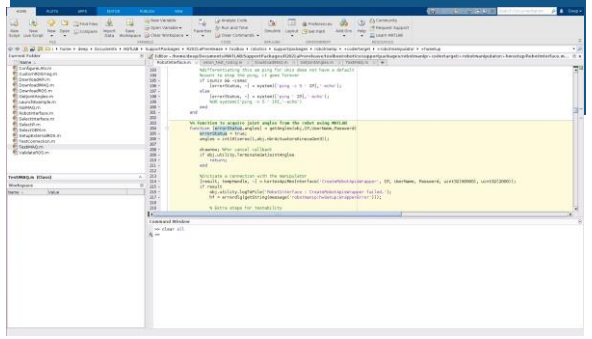
# 硬件连接



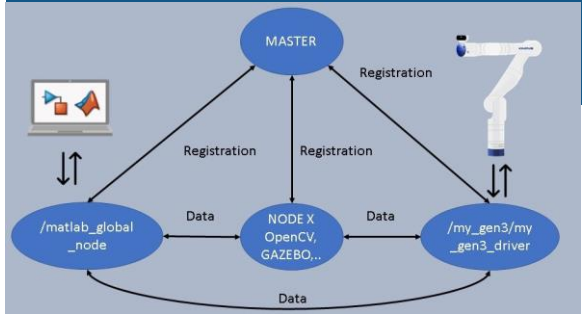

R2020b

MATLAB API



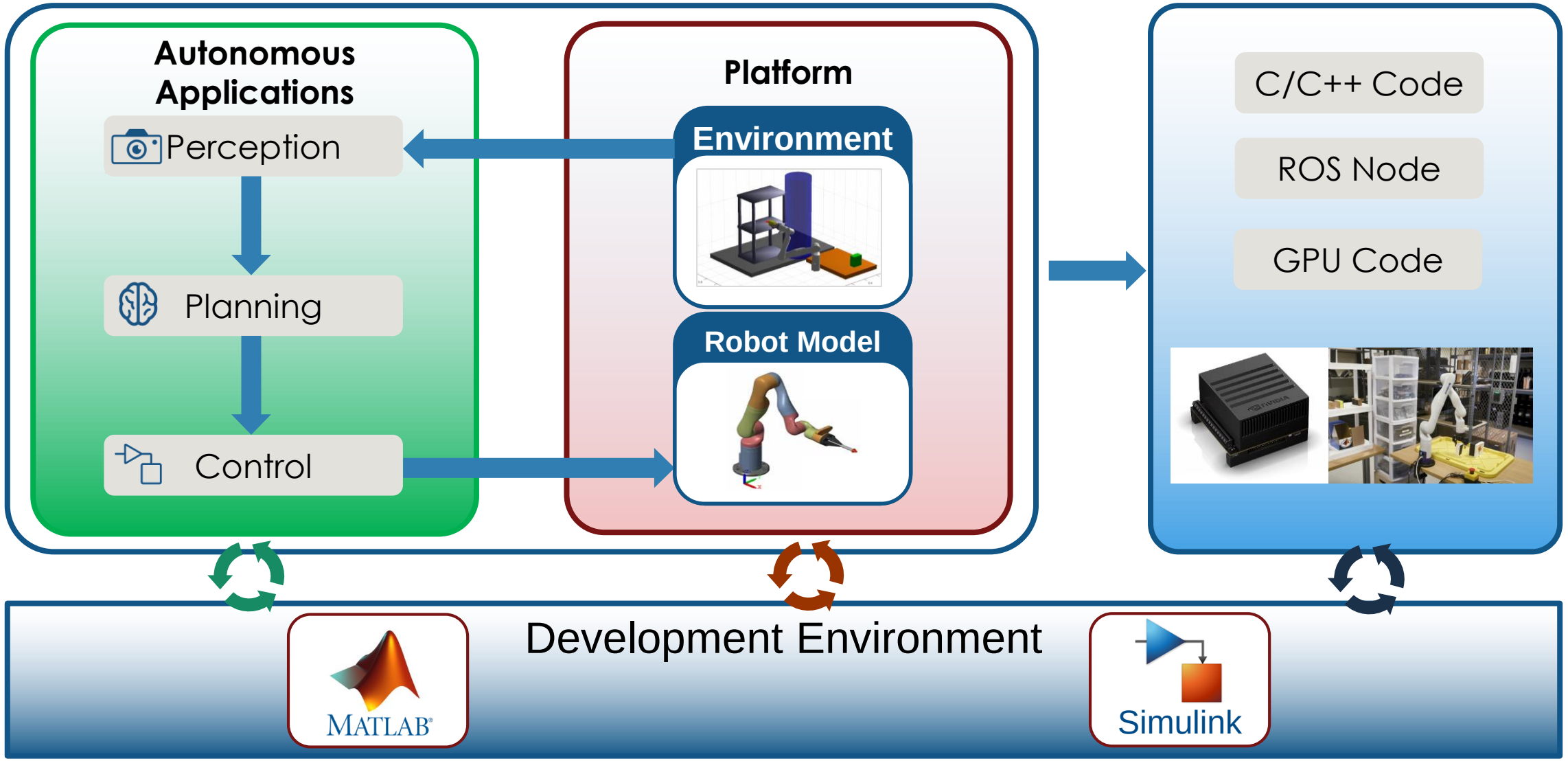
R2021a

ROS Driver

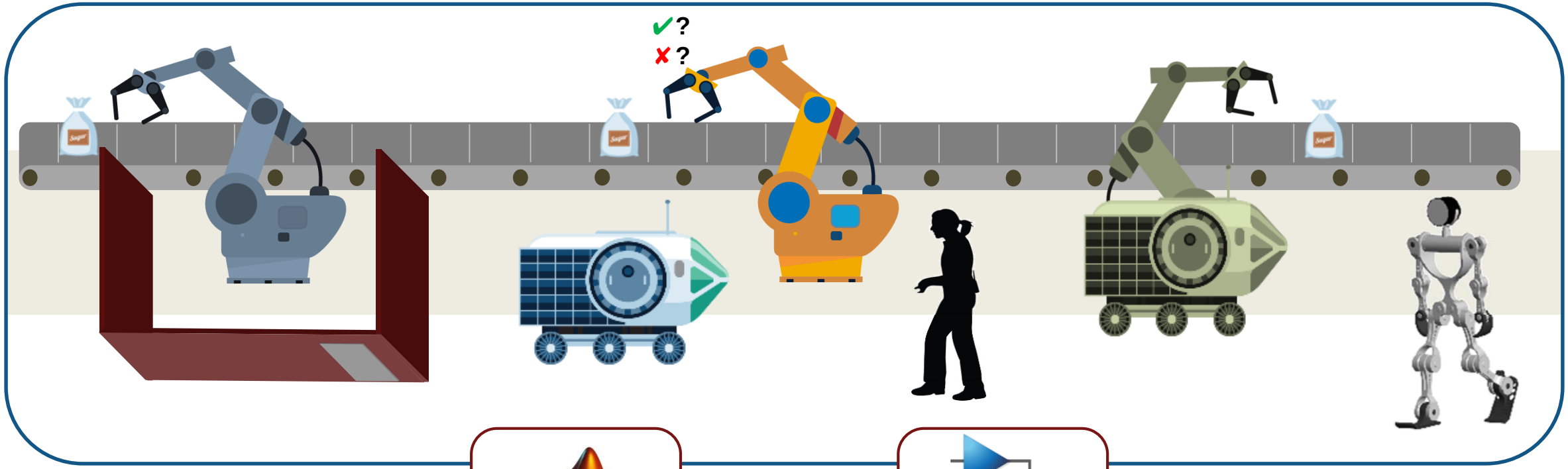



*Robotics System Toolbox, ROS Toolbox*

# 使用 MATLAB 和 Simulink开发自主机器人



# 智慧工厂自主机器人开发



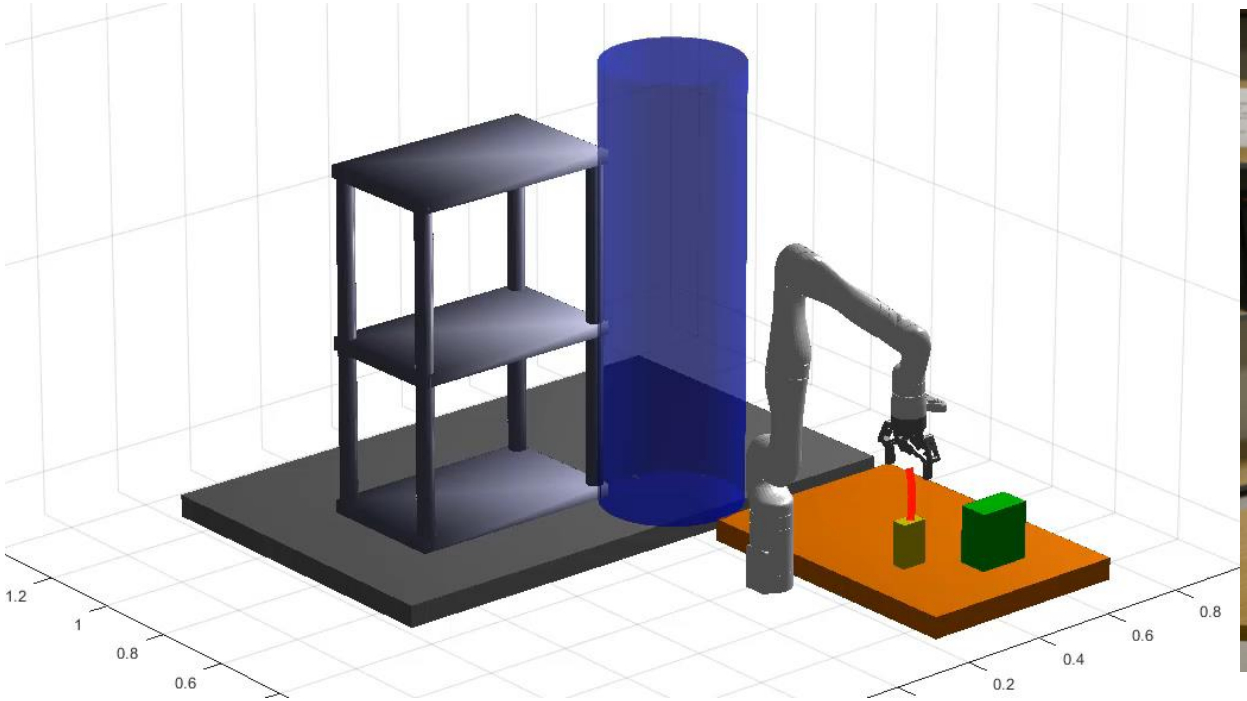

MATLAB

Simulink

**开发从感知到运动规划的自主应用**
**优化系统级行为**

**支持MATLAB/Simulink 统一环境**

# 使用案例：仓库分拣机器人



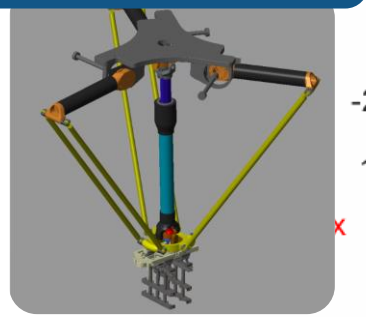

# 使用案例: Delta 机器人用于自动零件分拣

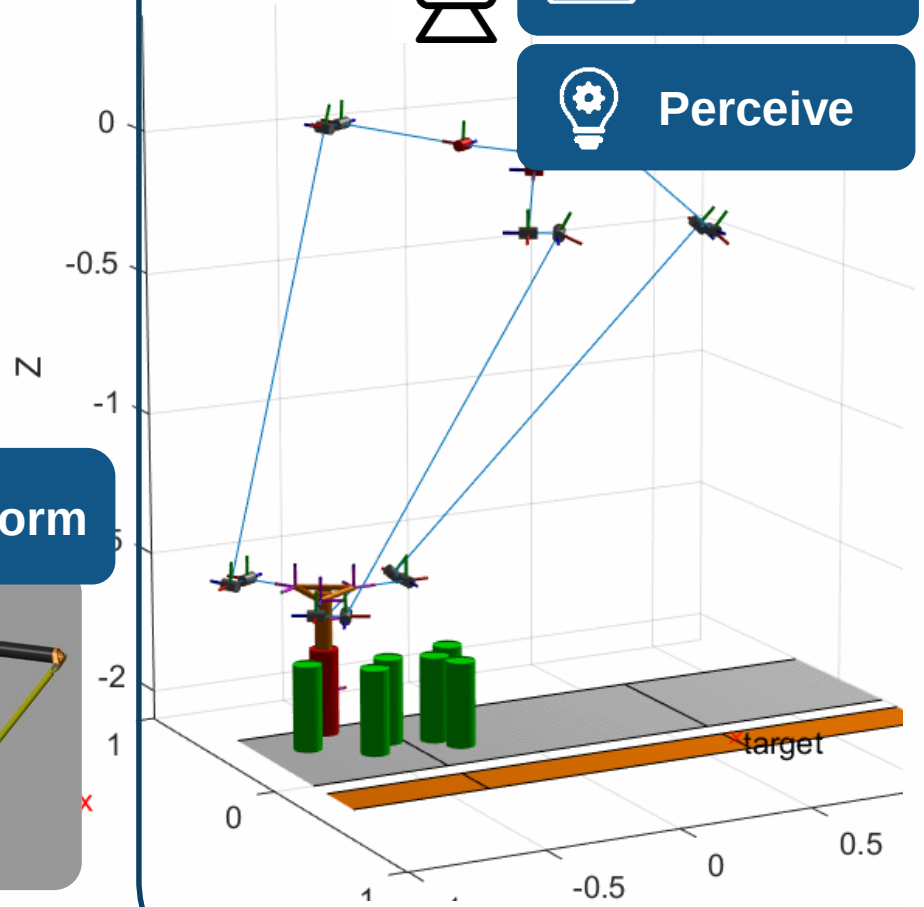
Connect

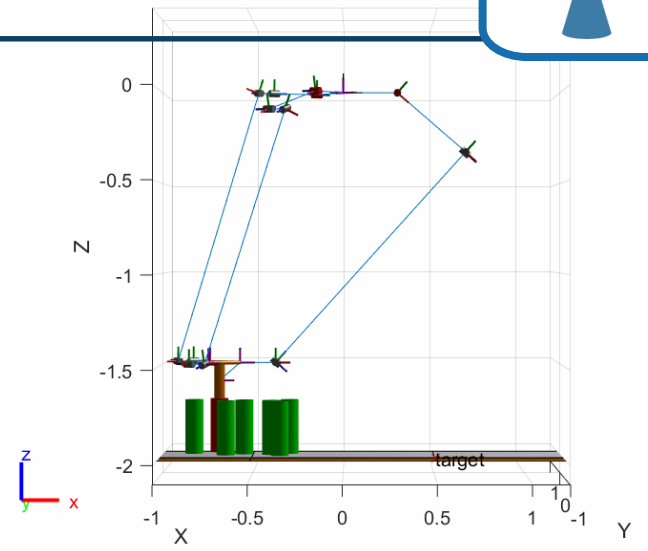
Sense

Perceive

Platform

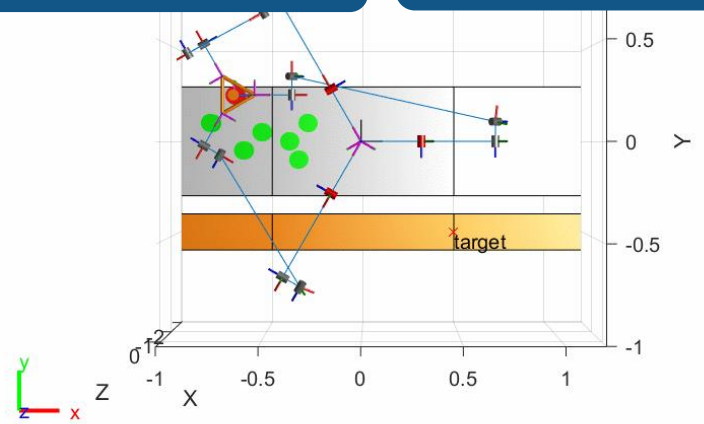





Plan & Decide

Control



# 总结

## 未来工厂中先进的机器人系统

- 协作机器人
- 支持 AI 的机器人

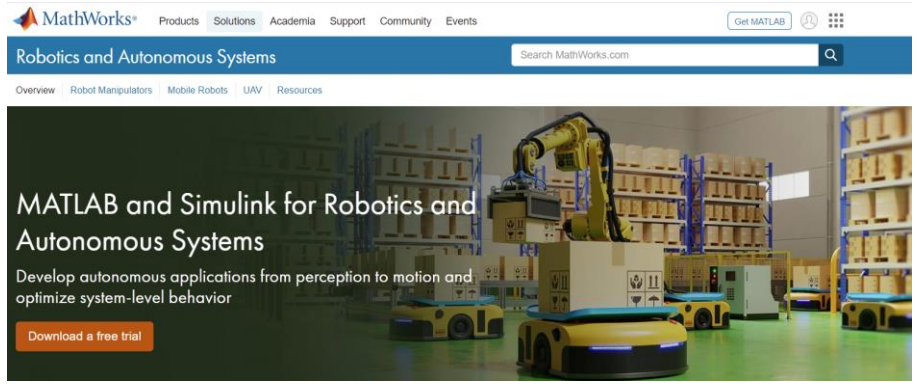
## 开发自主机器人系统的三大支柱

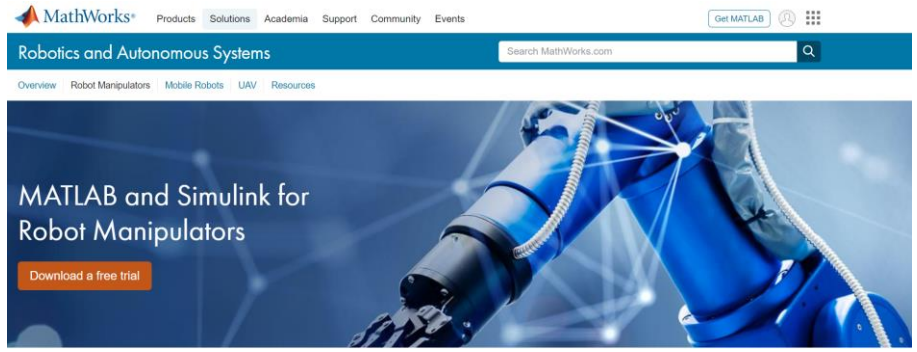
- 带环境模型的平台设计
- 自主应用开发
- 部署

## MATLAB 和 Simulink是一个统一的开发环境

- 开发从感知到运动规划的自主应用
- 仿真和优化系统级行为

# 了解更多信息

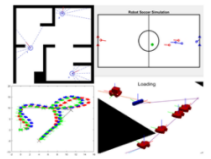



**MathWorks Robotics Solution Page**
mathworks.com/robotics

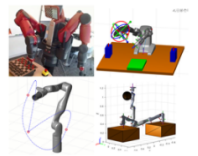
### Ground Vehicles and Mobile Robotics

- Kinematic motion models for simulation
- Control and simulation of warehouse robots
- Programming of soccer robot behavior (Video)
- Simulation and programming of robot swarm (Video)
- Mapping, Localization and SLAM (See Section Below)
- Motion Planning and Path Planning (See Section Below)
- Mobile Robotics Simulation Toolbox (Video)
- Robotics Playground (Robotics Education - Video)



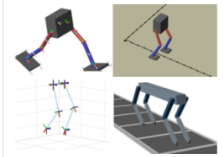
### Manipulation

- Tools for rigid body tree dynamics and analysis
- Inverse Kinematics (Blog and GitHub Repo)
- Inverse kinematics with spatial constraints
- Interactive Inverse Kinematics
- Collision checking (Self-Collisions, Environment Collisions)
- Trajectory Generation (Blog, GitHub Repo)
- Safe trajectory planning (Impedance based control)
- Pick and place workflows (Using Gazebo)



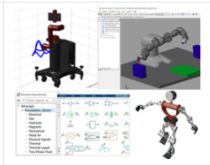
### Legged Locomotion

- Modeling and simulation of walking robots (GitHub Repo)
- Pattern Generation for Walking Robots (Video)
- Linear Inverted Pendulum Model (LIPM)for humanoid walking (Video)
- Deep Reinforcement Learning for Walking Robots (Video)
- Modeling of quadruped robot running (Files)
- Quadruped Robot Locomotion Using DDPG Agent

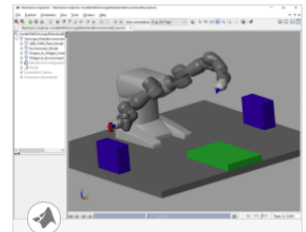


### Robot Modeling

- Simscape Tools for Modeling and Simulation of Physical Systems
- Simulate Manipulator Actuators and Tune Control Parameters
- Algorithm Verification Using Robot Models
- Import Robots to MATLAB from URDF Files
- Import Robots from CAD and URDF Files

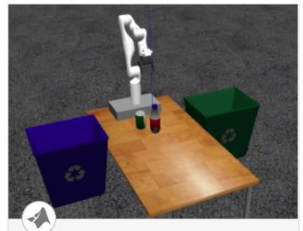


**Awesome-MATLAB-Robotics GitHub Repo**



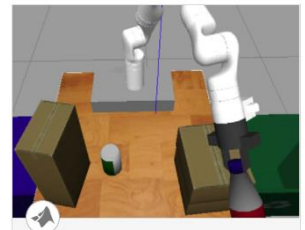
**Model And Control A Manipulator Arm With Robotics And Simscape**

Execute a pick-and-place workflow using an ABB YuMi robot, which demonstrates how to design robot algorithms in Simulink®, and then



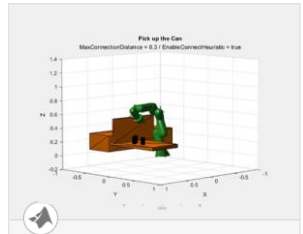
**Pick-and-Place Workflow in Gazebo using ROS**

Setup an end-to-end pick and place workflow for a robotic manipulator like the KINOVA® Gen3 and simulate the robot in a physics



**Pick-and-Place Workflow in Gazebo using Point-Cloud Processing and RRT Pat…**

Setup an end-to-end pick and place workflow for a robotic manipulator like the KINOVA® Gen3.



**Pick and Place Using RRT for Manipulators**

Using manipulators to pick and place objects in an environment may require path planning algorithms like the rapidly-exploring random tree

**Robotics Examples**

# MATLAB EXPO 2021

# 谢谢！