

# MATLAB EXPO

## 2021

### Smart Factory: Advanced Industrial Robots from Perception to Motion

*YJ Lim*



*Ronal George*



# Autonomous Robot Development with MATLAB & Simulink



Platform design tools for industrial robots



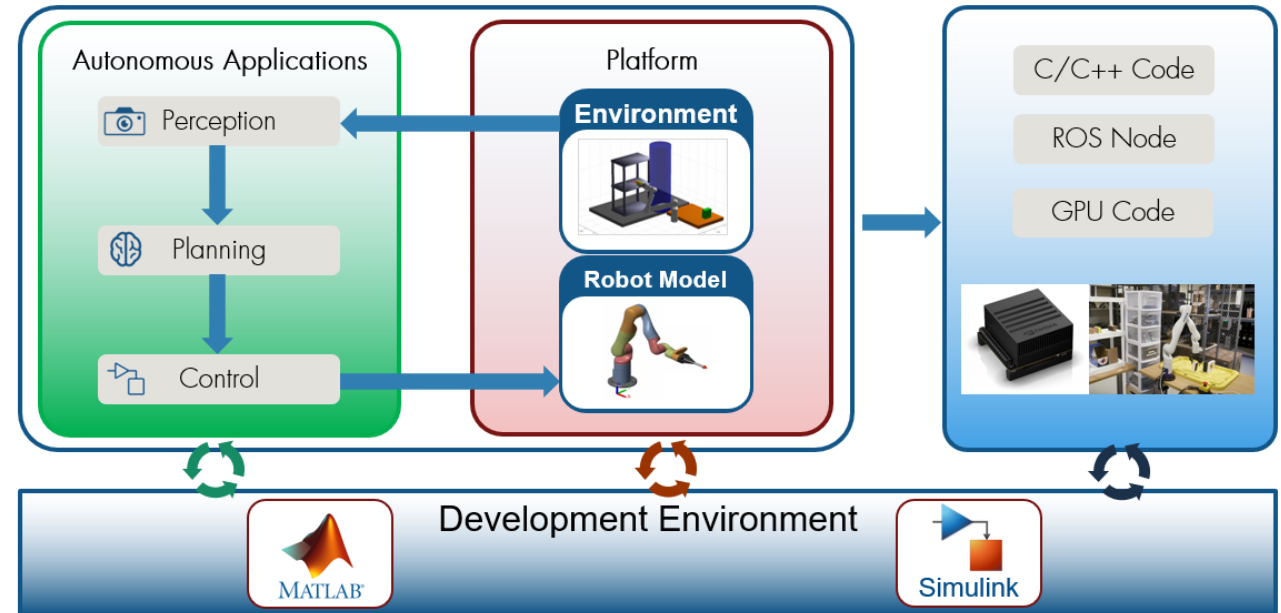
Tools for autonomous industrial robot applications



Deployment and hardware interface with MATLAB and Simulink



Integrated workflows enabled by MATLAB and Simulink



# Autonomous Robot Development with MATLAB & Simulink



Platform design tools for industrial robots



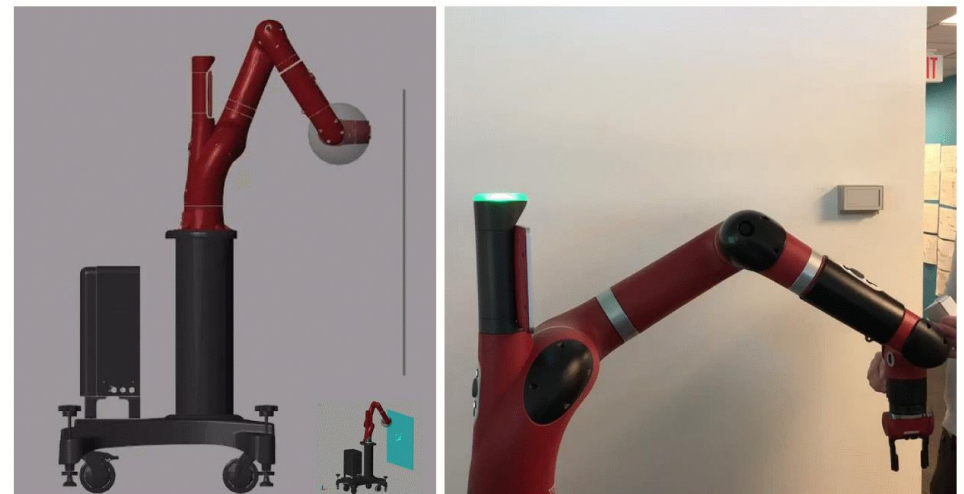
Tools for autonomous industrial robot applications



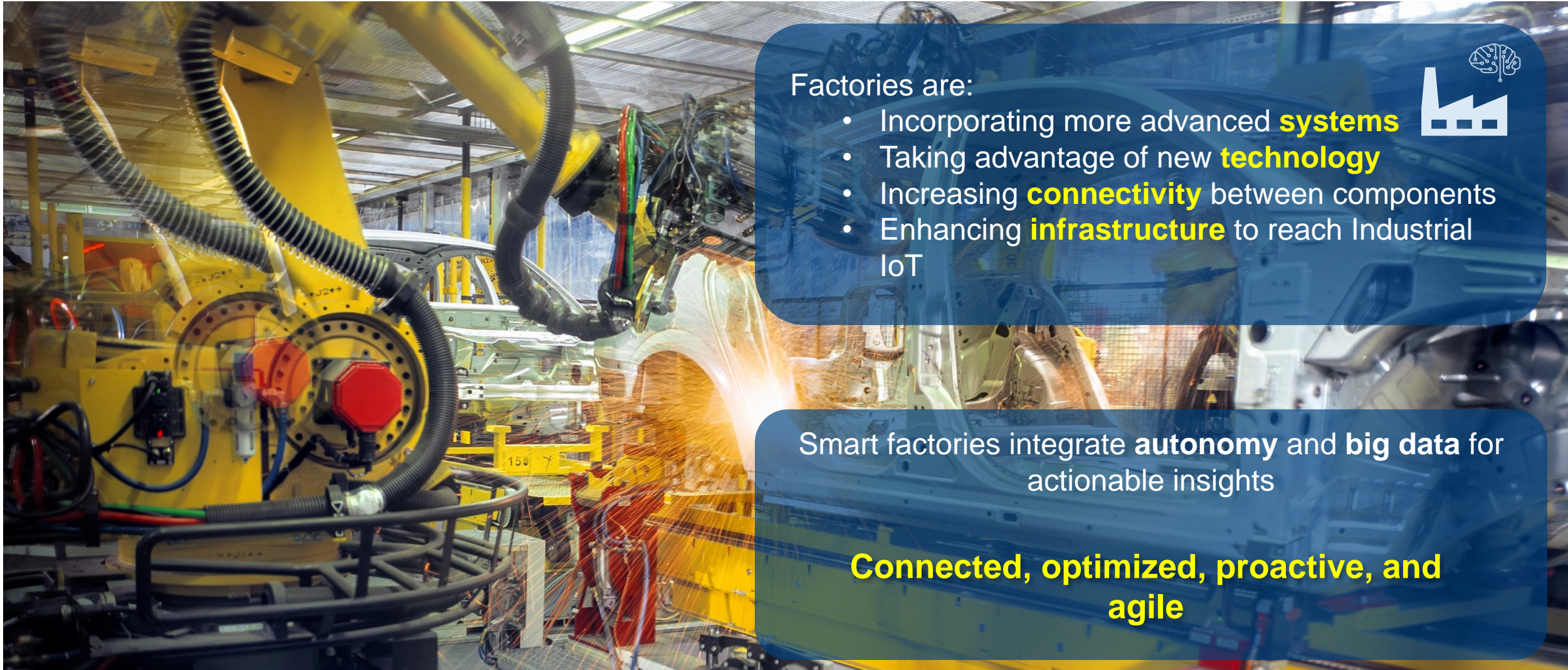
Deployment and hardware interface with MATLAB and Simulink



Integrated workflows enabled by MATLAB and Simulink



# Industry Trends – Advanced Robotics



Factories are:

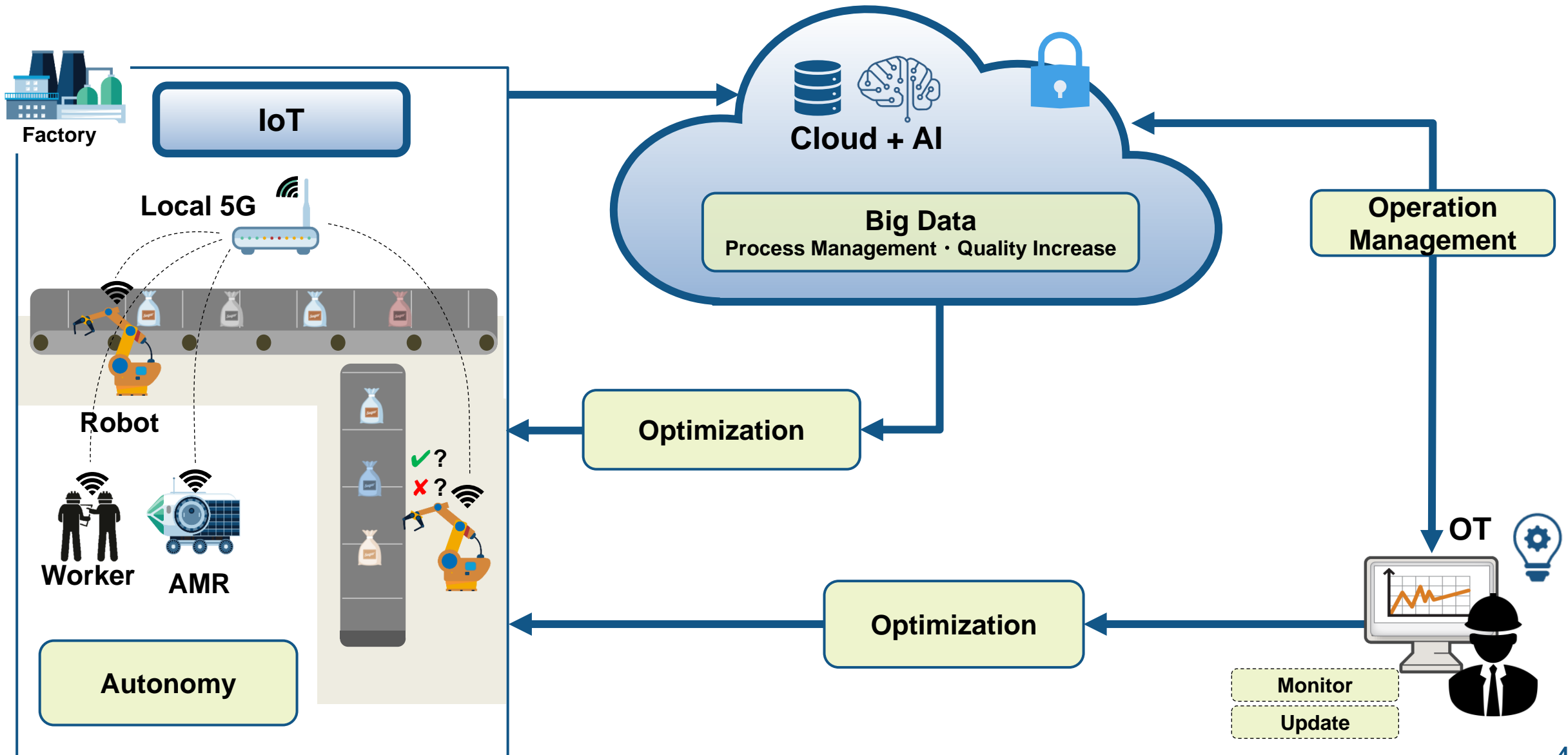
- Incorporating more advanced **systems**
- Taking advantage of new **technology**
- Increasing **connectivity** between components
- Enhancing **infrastructure** to reach Industrial IoT



Smart factories integrate **autonomy** and **big data** for actionable insights

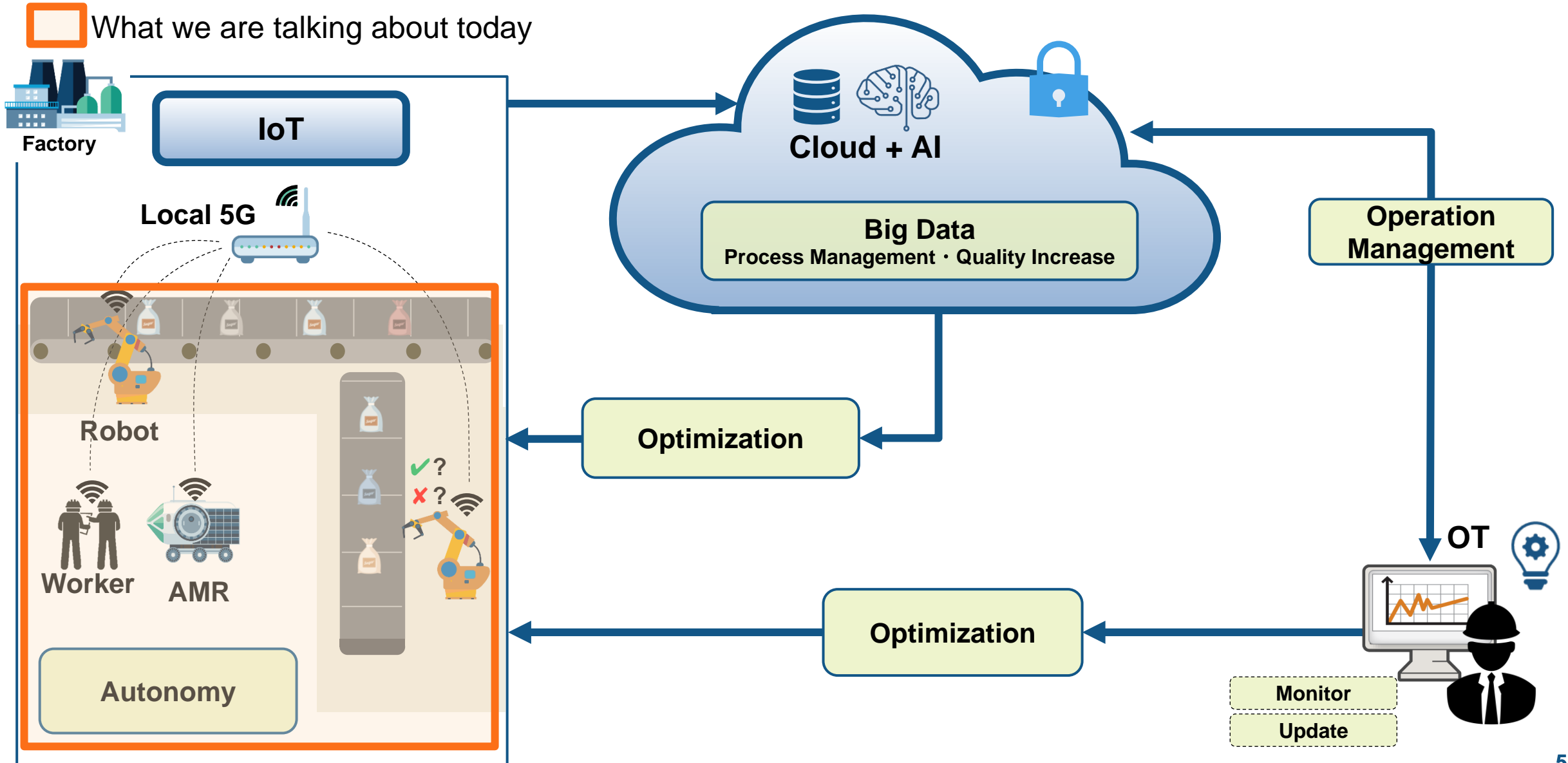
**Connected, optimized, proactive, and agile**

# Smart Factory Concept: Asset, Operation, & Workforce Optimization



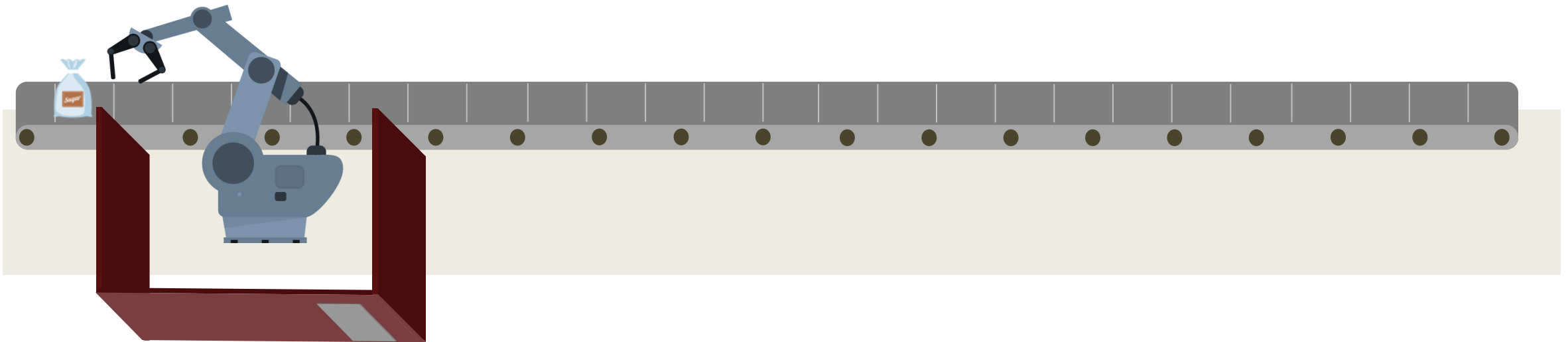
# Smart Factory Concept: Asset, Operation, & Workforce Optimization

 What we are talking about today



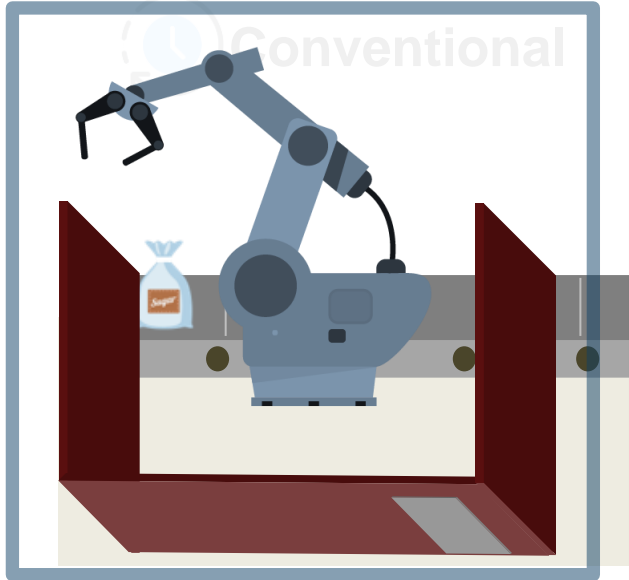
# Robotics in Smart Factory: Trends and Directions

## Conventional



- **Automated Systems**
  - Repetitive tasks
  - Manual programming
  - Safety fence needed

# Robotics in Smart Factory: Trends and Directions



- **Automated System**
  - Repetitive tasks
  - Manual programming
  - Safety fence needed



## 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

[Link to user story](#)



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

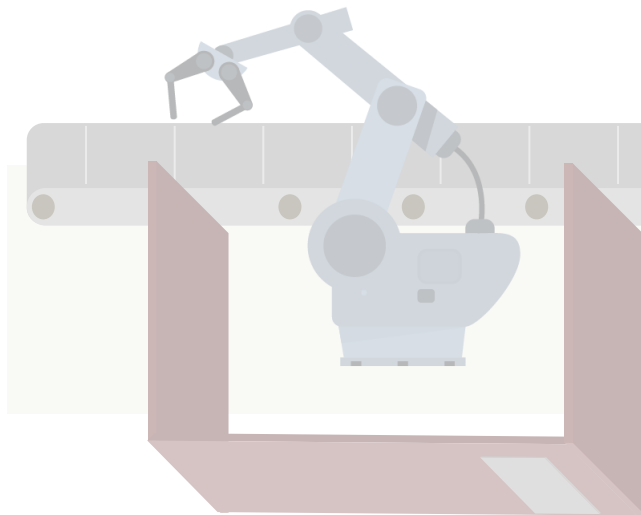
*"Model-Based Design with MATLAB and Simulink supports a wide range of options, from classic to modern control, which made it possible to respond easily to any changes in design constraints and to meet the demanding accuracy requirement for this robot."*

- Tadashi Murata, Mitsubishi Heavy Industries

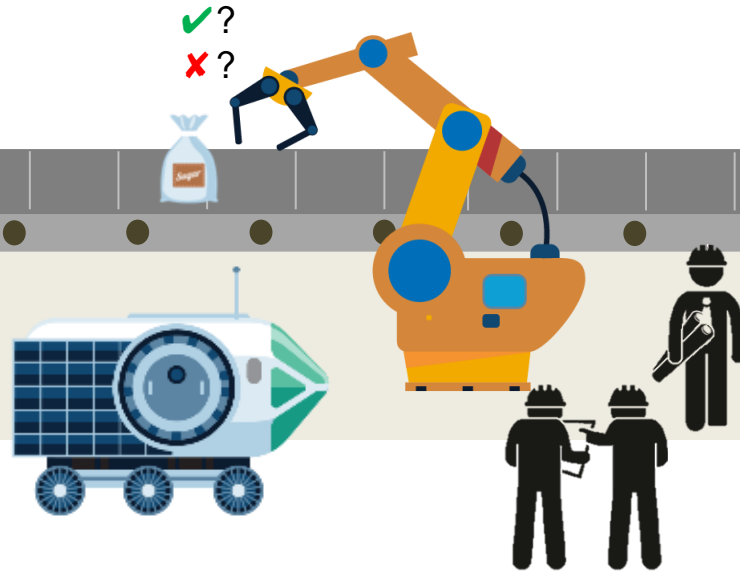


# Robotics in Smart Factory: Trends and Directions

 Conventional



 Current



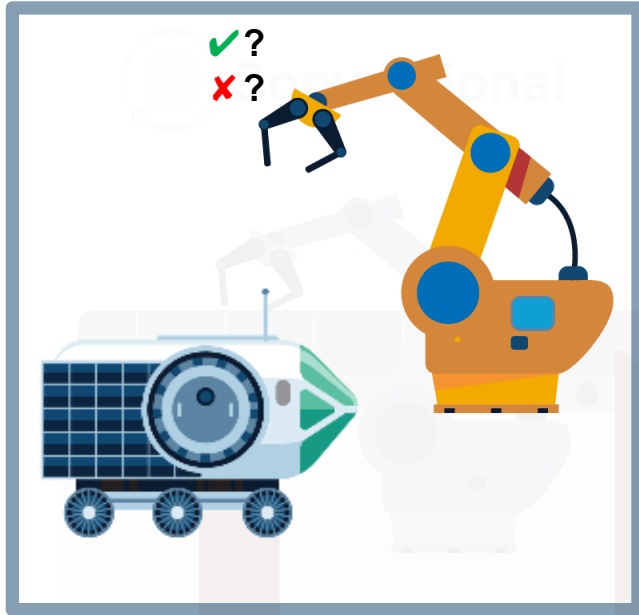
- **Automated Systems**

- Repetitive tasks
- Manual programming
- Safety fence needed

- **Flexible Automation**

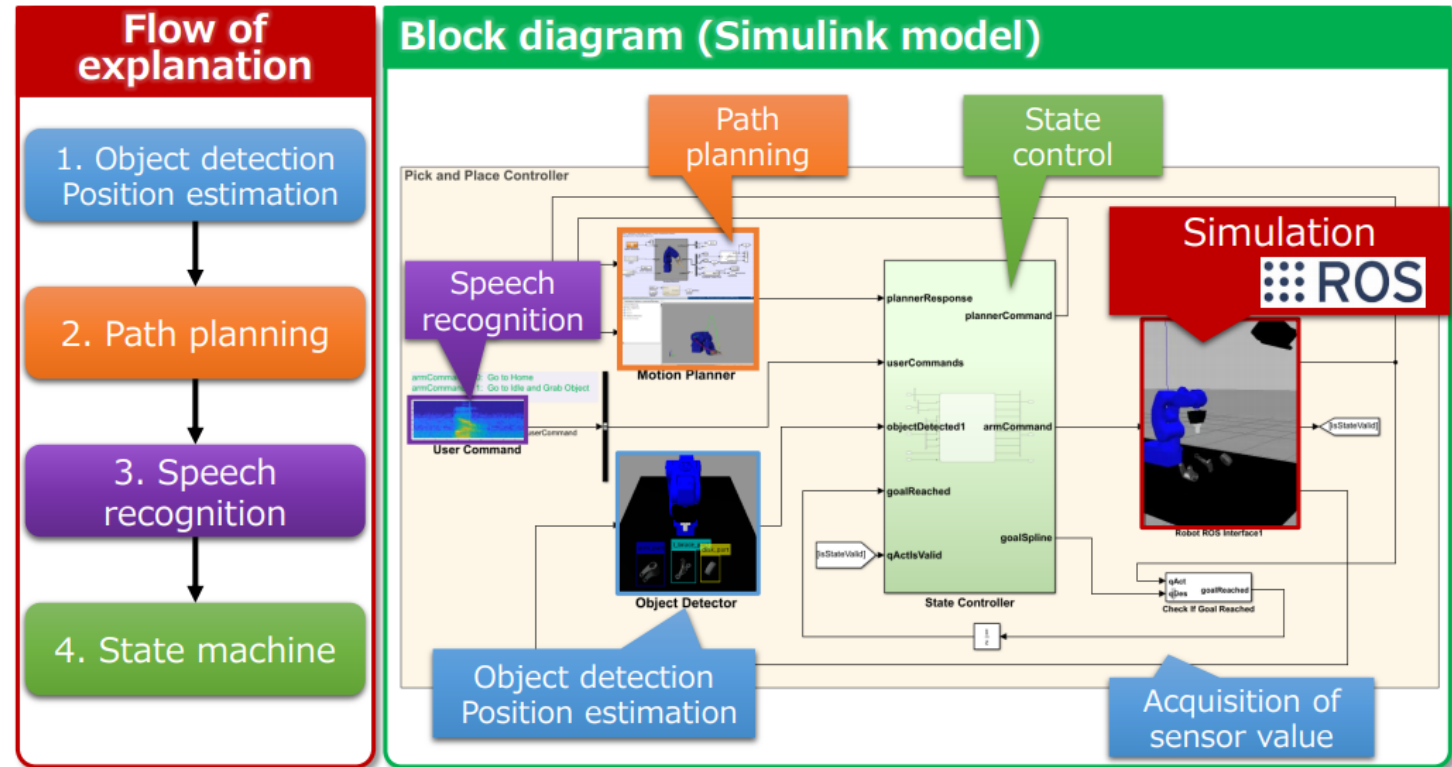
- Collaborative robots
- Advanced algorithm
- Safety

# Robotics in Smart Factory: Trends and Directions



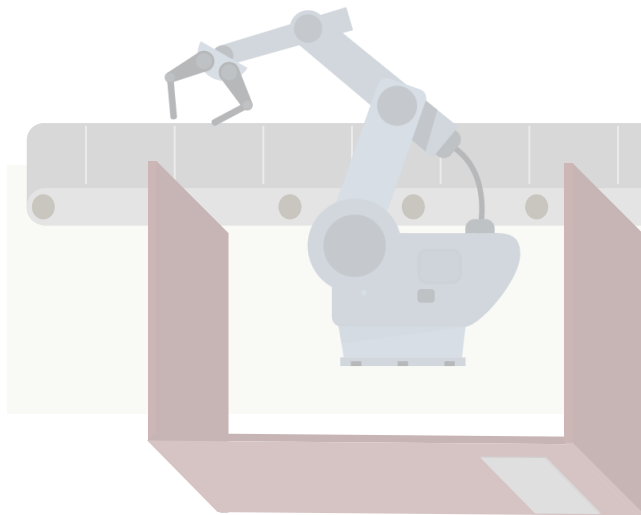
- Automated Systems
  - Repetitive tasks
  - Manual programming
  - Safety fence needed

## Sample app overview



# Robotics in Smart Factory: Trends and Directions

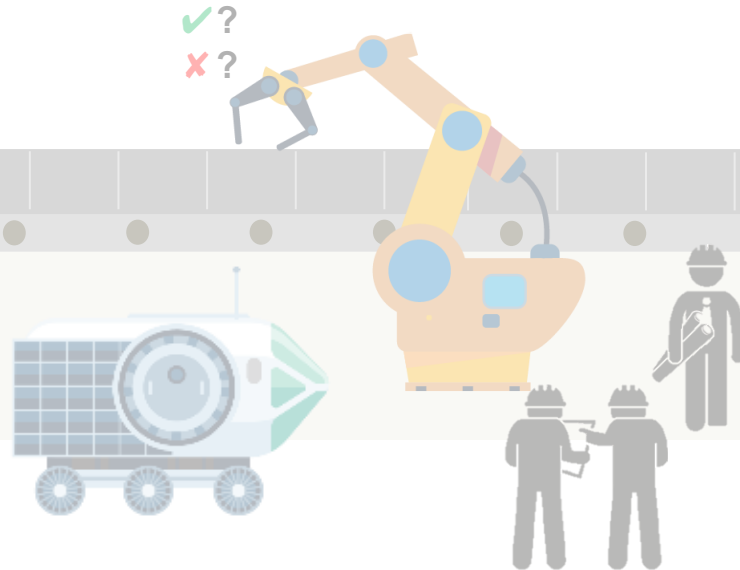
 Conventional



## Automated Systems

- Repetitive tasks
- Manual programming
- Safety fence needed

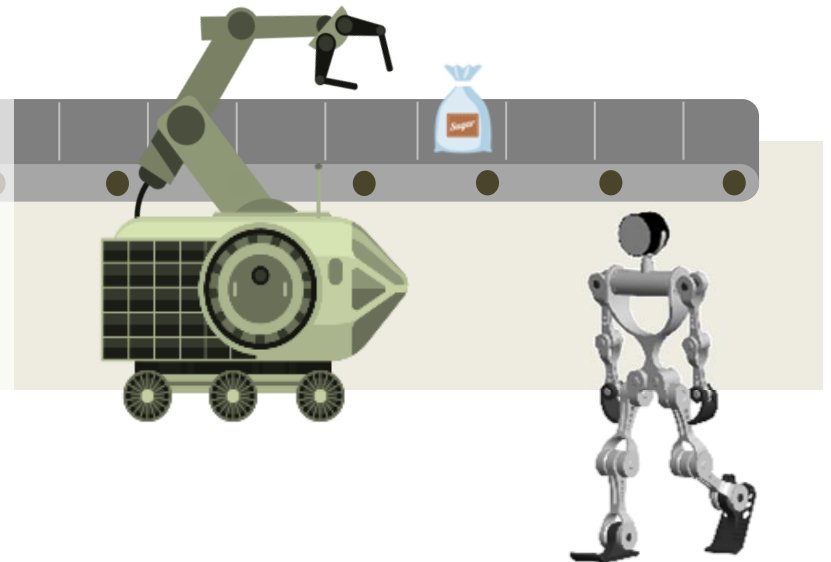
 Current



## Flexible Automation

- Collaborative robot
- Advanced algorithm
- Safety

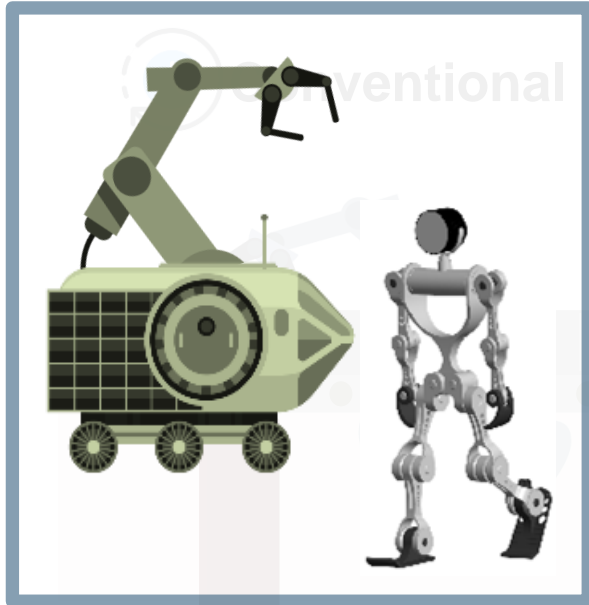
 Future



## Autonomous Systems

- Task cooperation
- Intelligence sharing
- Adapting to the environment

# Robotics in Smart Factory: Trends and Directions



- Automated System
  - Repetitive tasks
  - Manual programming
  - Safety fence needed



## 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

[Link to user story](#)



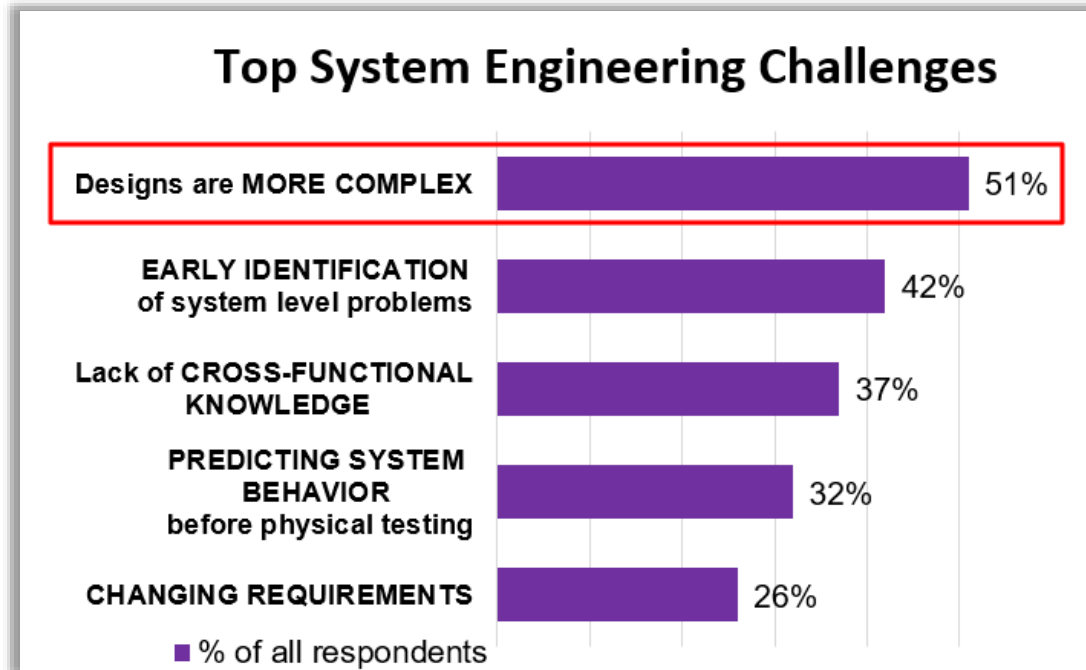
DLR's humanoid robot Agile Justin autonomously performing a complex construction task.

*"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

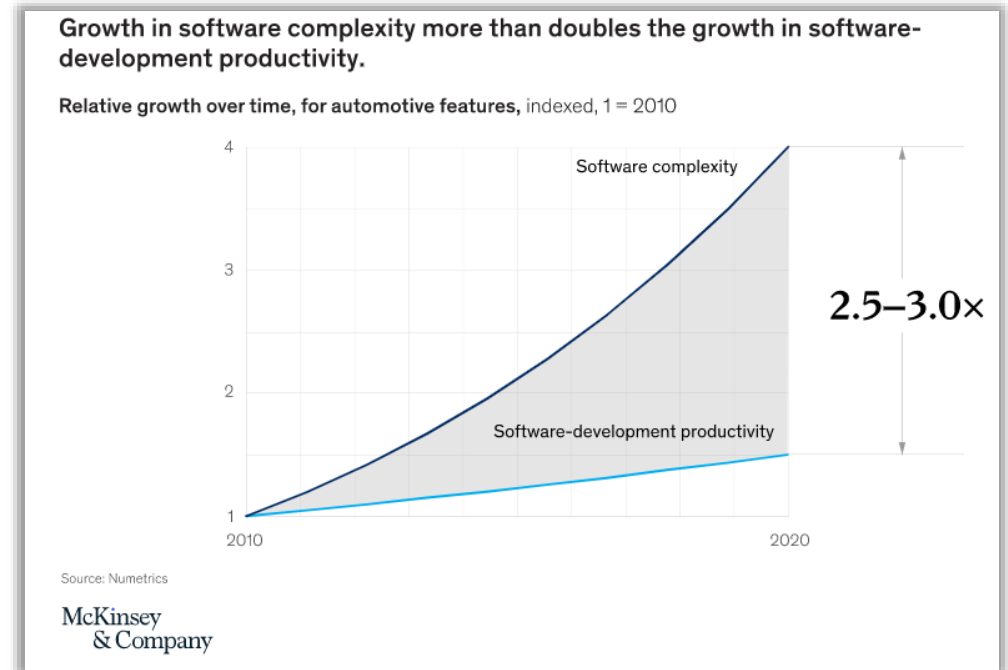
CONFIDENTIAL | 2

# Challenges: Growing Complexity



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

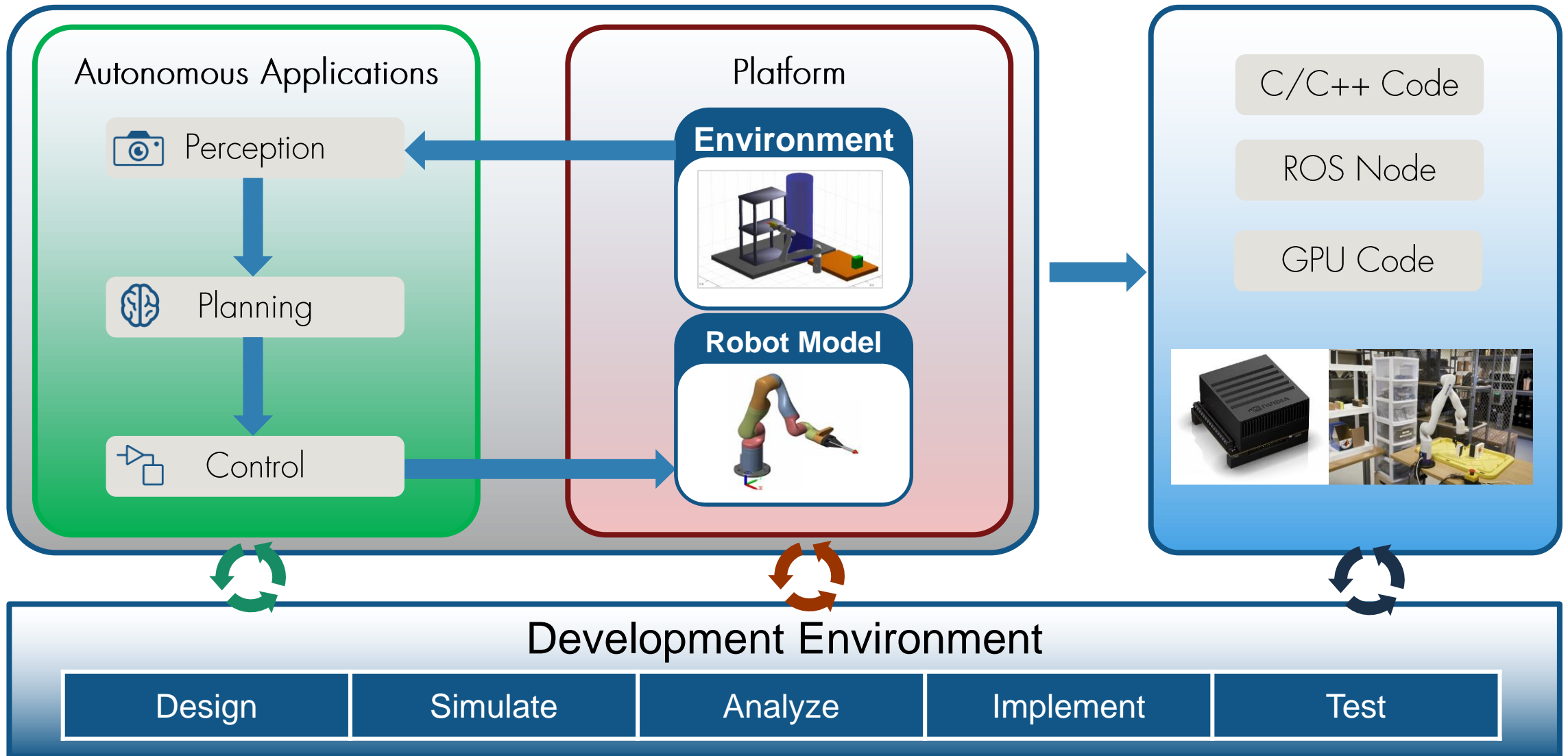
Source: Aberdeen Group, April 2014



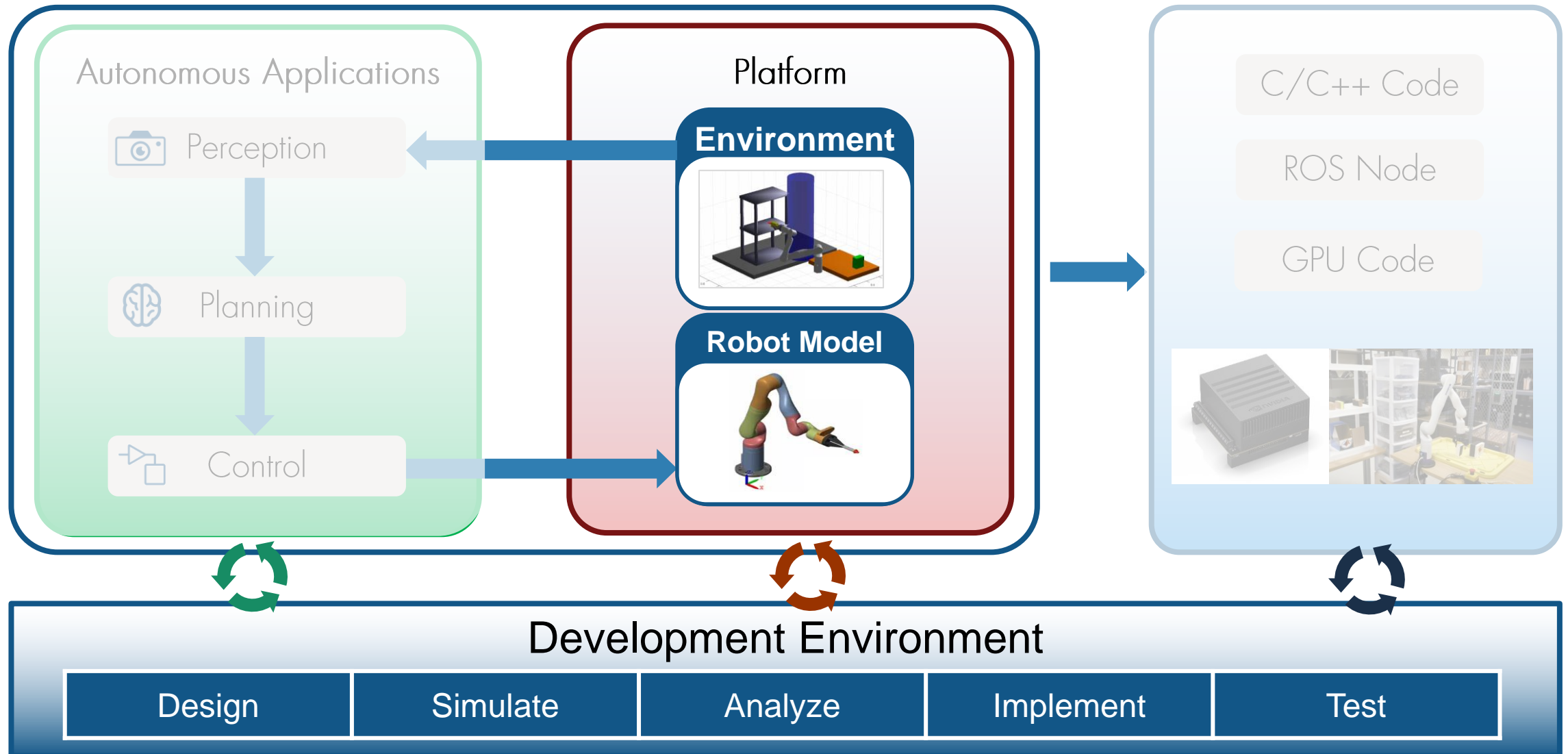
**Complexity for software features** is currently growing at double to triple the speed of software-development productivity

Source: McKinsey & Company, Feb 2020

# Autonomous Robot Development with MATLAB & Simulink

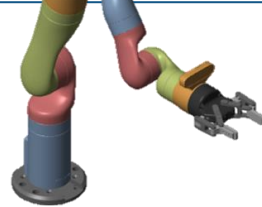
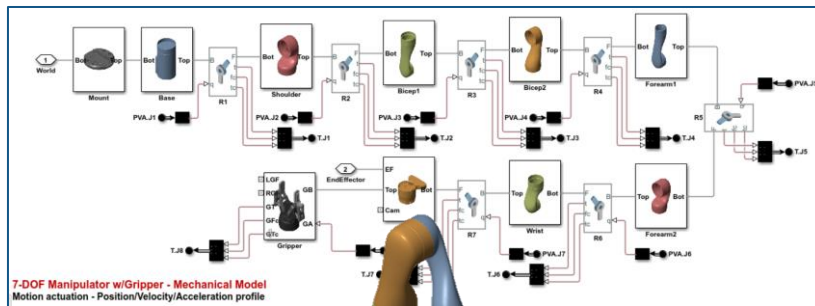
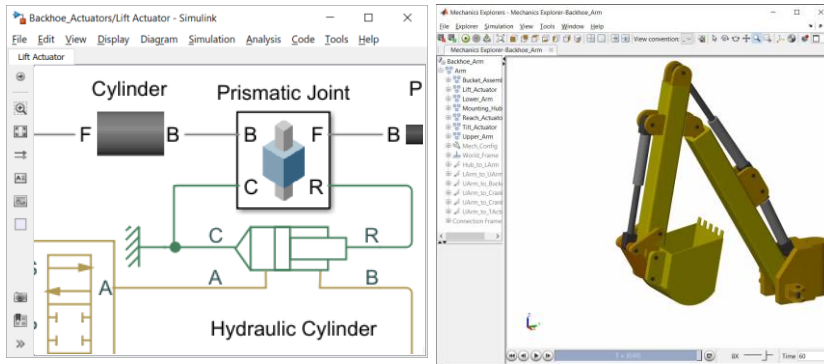


# Autonomous Robot Development with MATLAB & Simulink



# Physical & Kinematic Modeling

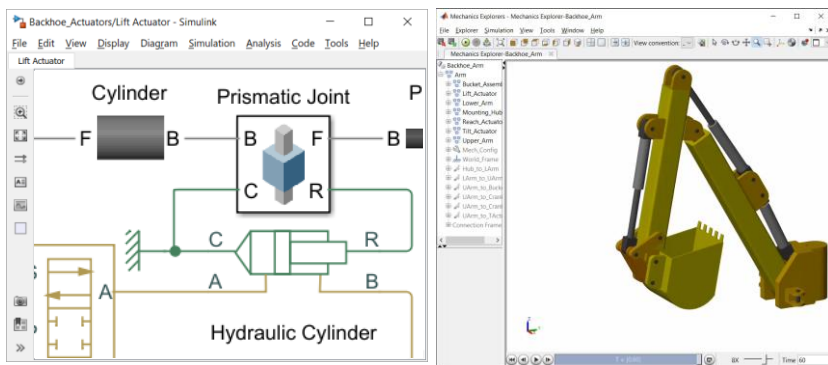
## Creating a physical model



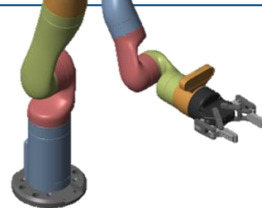
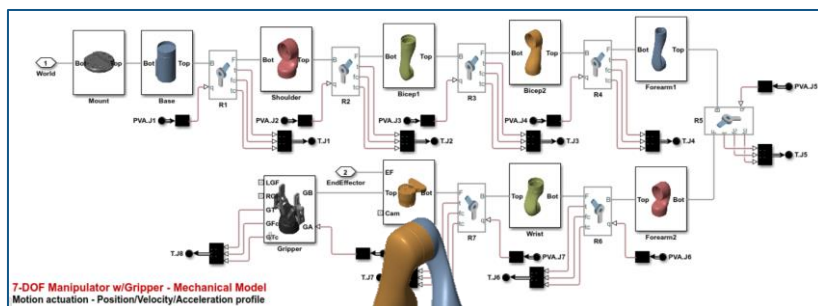
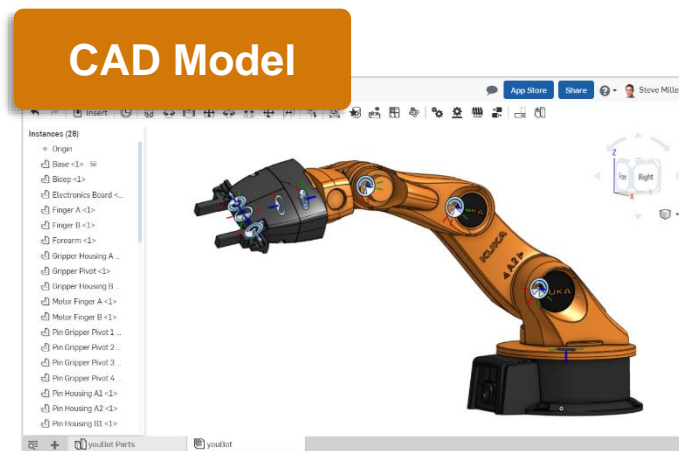


# Physical & Kinematic Modeling

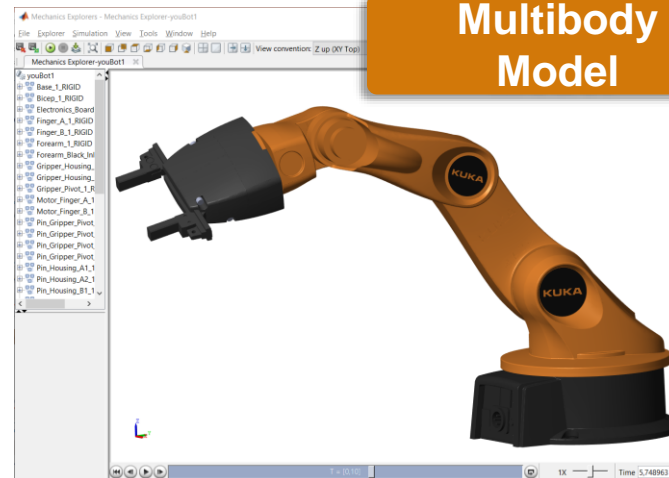
## Creating a physical model



## Automatic import from CAD Tools

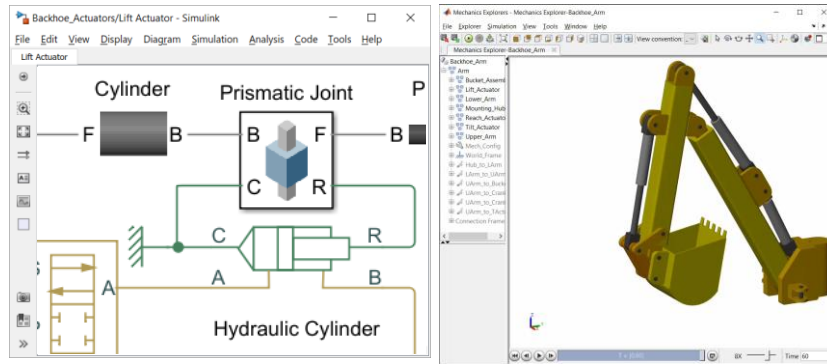


## Multibody Model

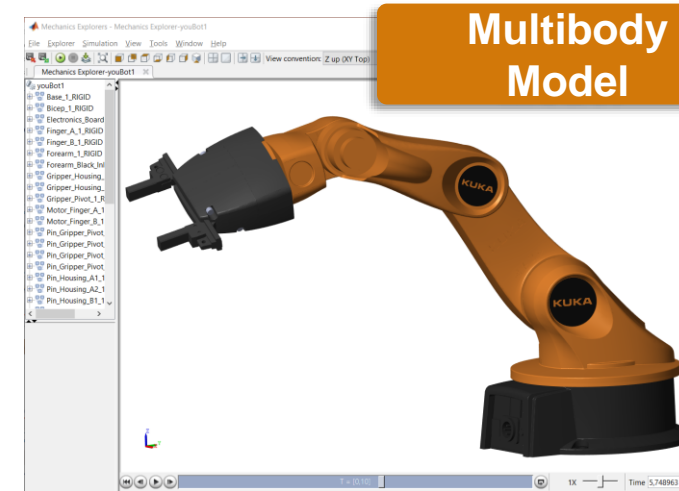
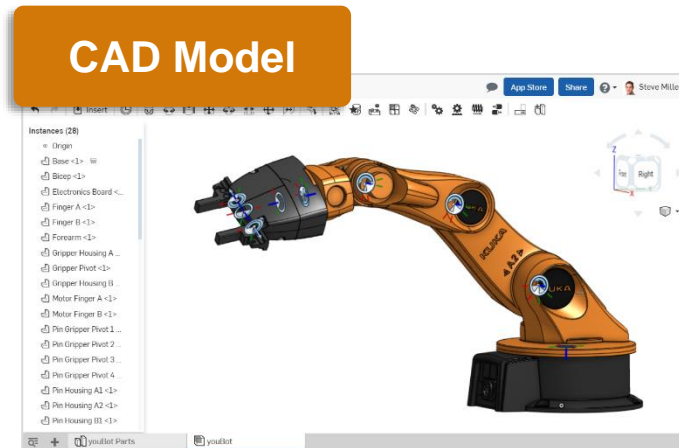


# Physical & Kinematic Modeling

## Creating a physical model

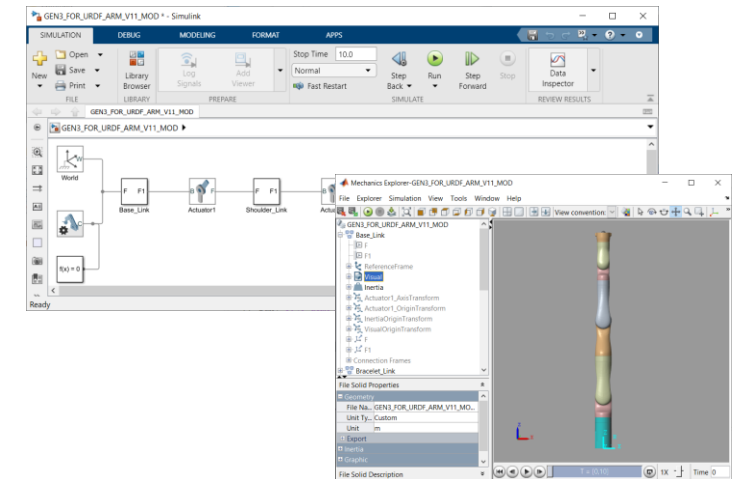


## Automatic import from CAD Tools

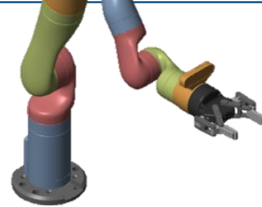
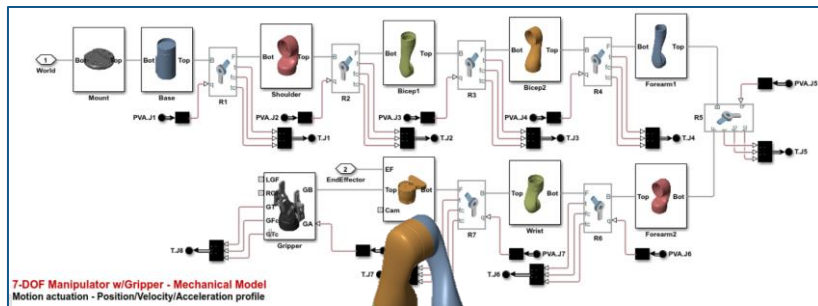
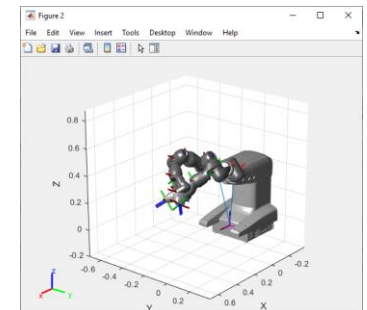


## URDF / Robot Library

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

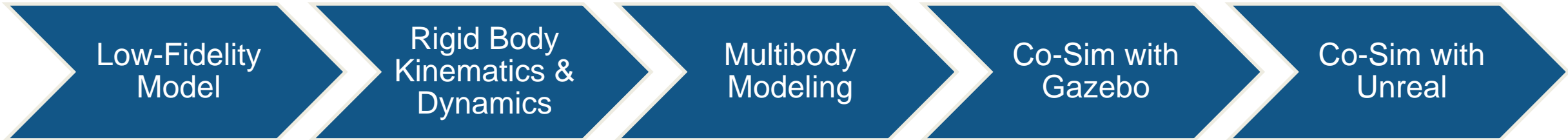
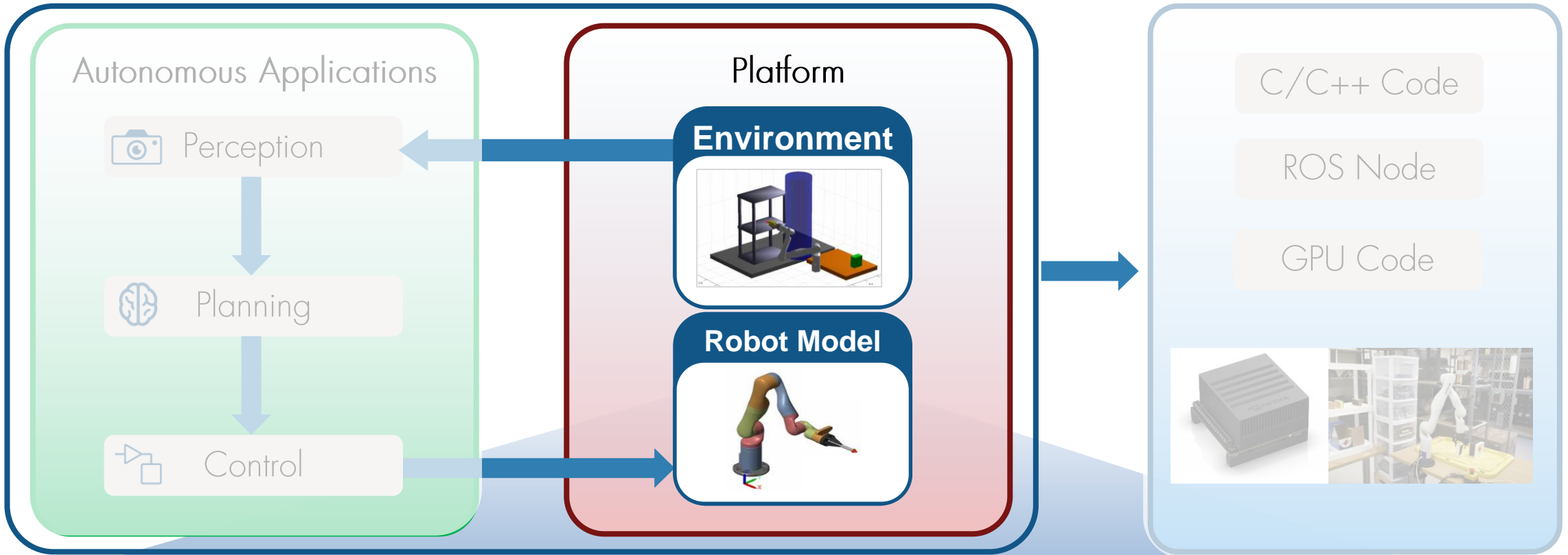


```
%% Use robot library
Robot = loadrobot('abbYuMi');
Show(robot);
```

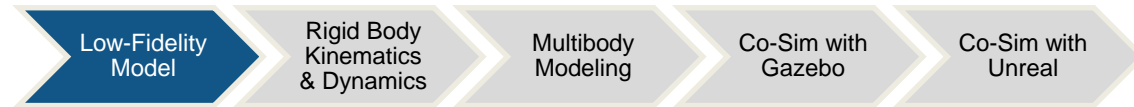


# Robot Simulation

Simulate First and Simulate Often!

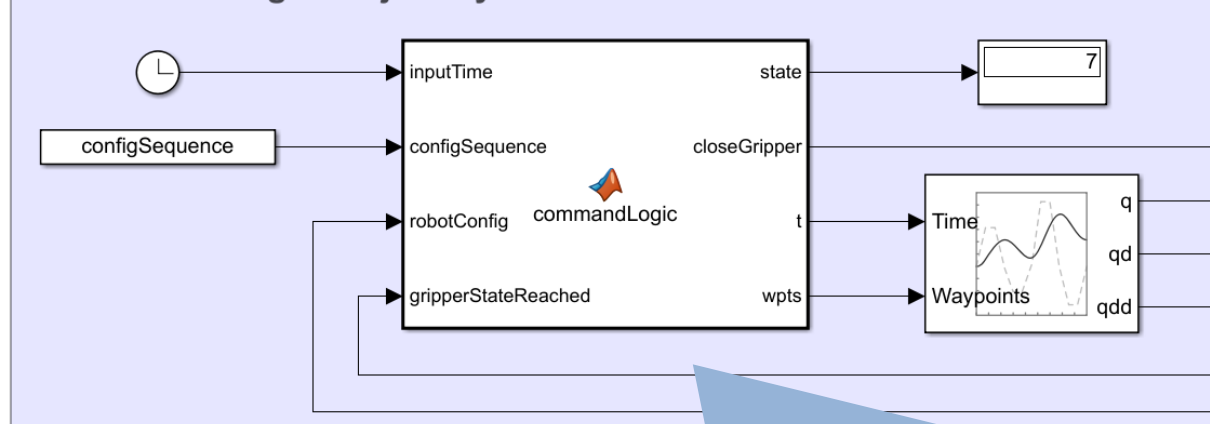


# Fast & Low-Fidelity Simulation

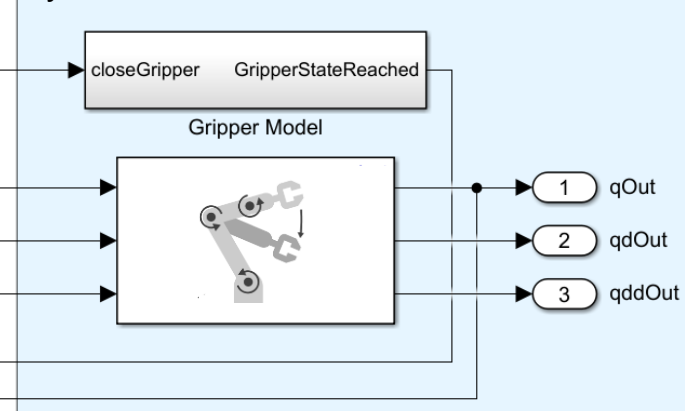


## Simplified system model with a joint space motion model

### Task Scheduling & Trajectory Generation



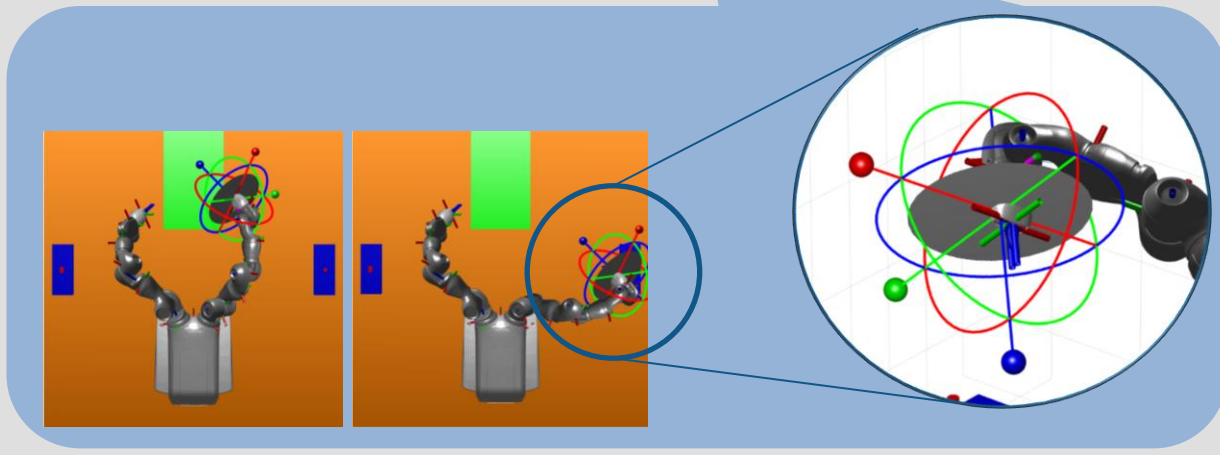
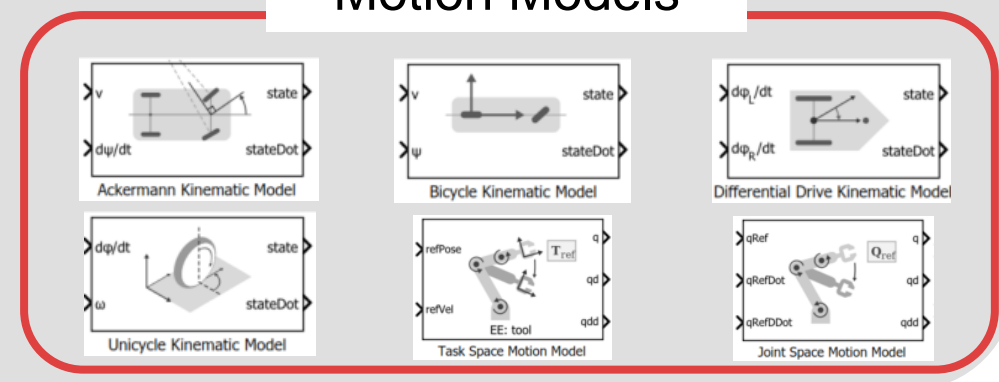
### System Model



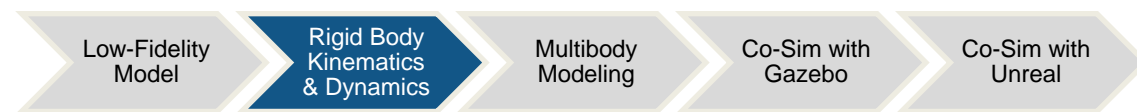
**Model And Control A Manipulator Arm With Robotics And Simscape**

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

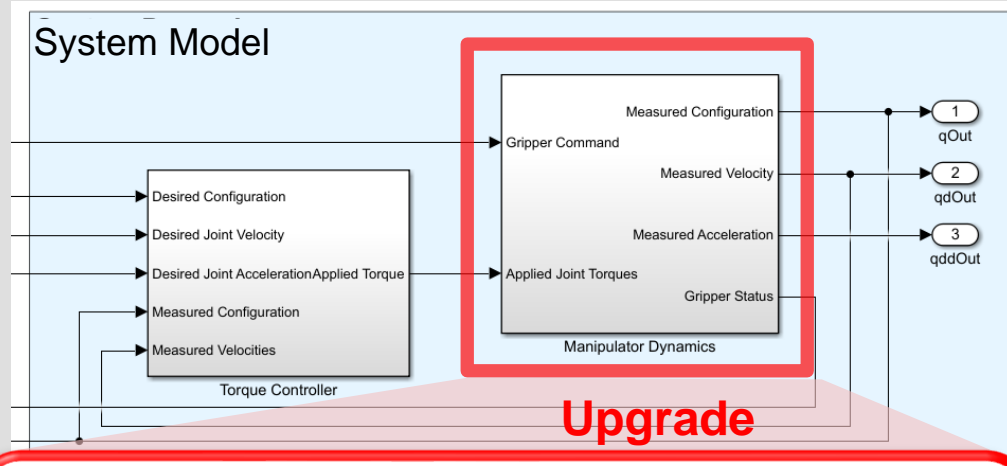
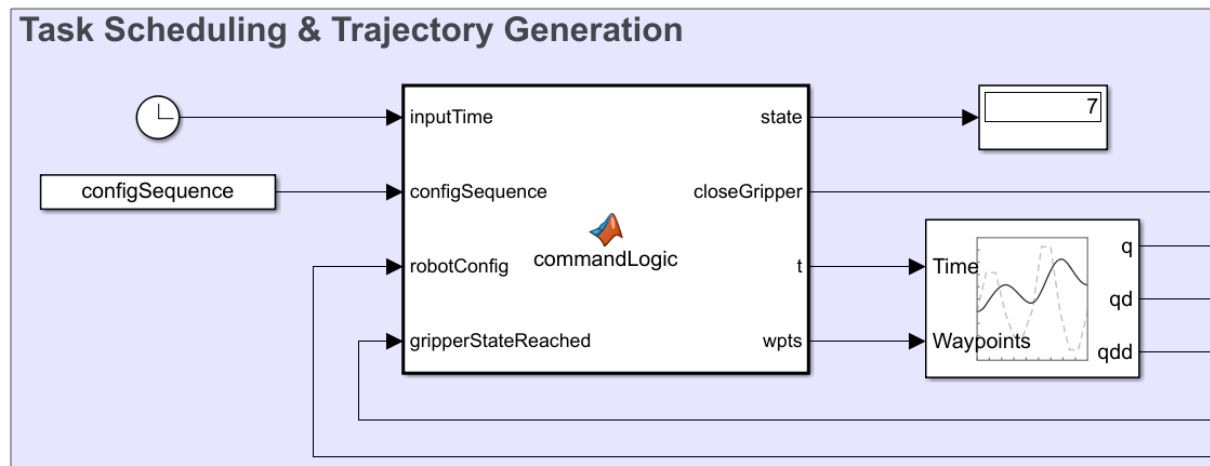
### Motion Models



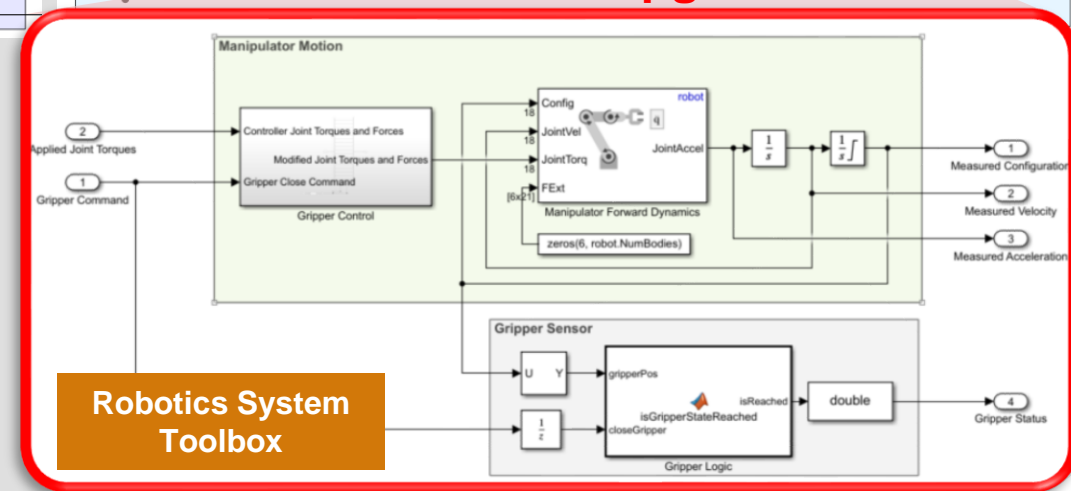
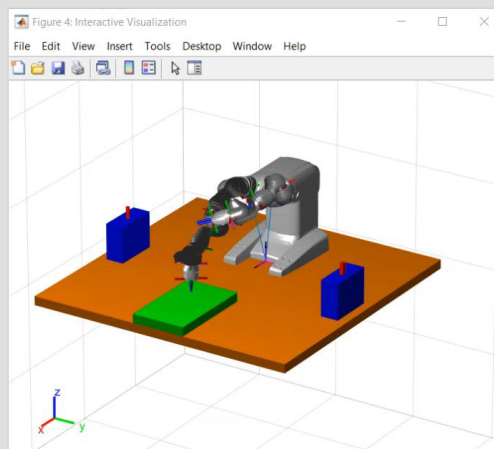
# Dynamic Simulation



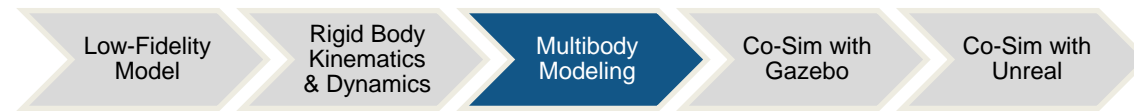
## Manipulator dynamics model that accepts joint torques and gripper commands



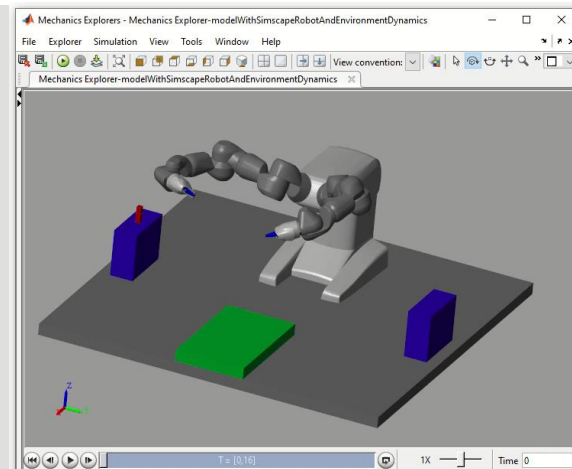
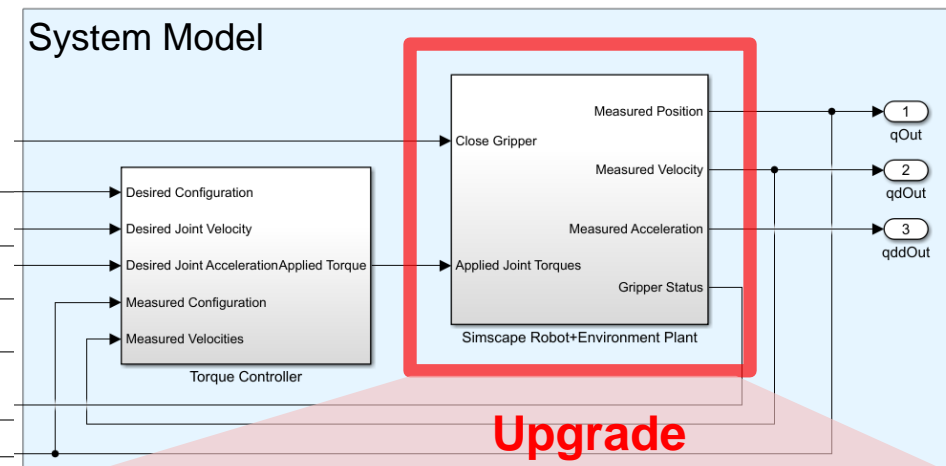
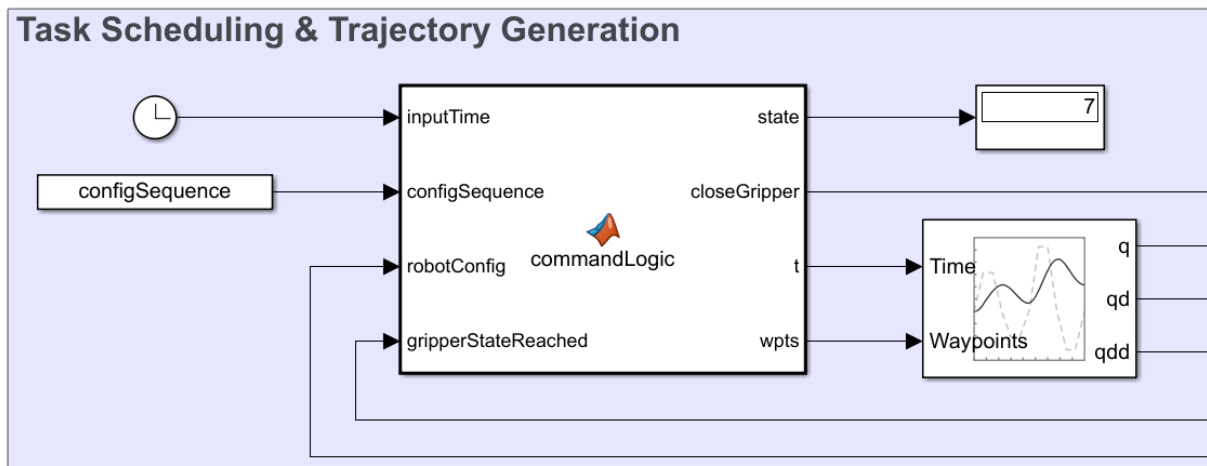
**Upgrade**



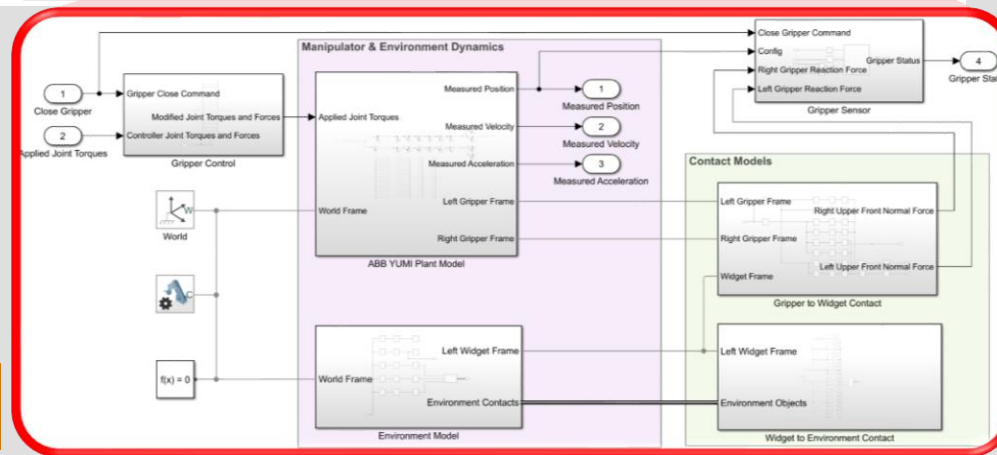
# Multibody Simulation



## Physical dynamics system with built-in joint limits and contact modeling



Simscape Multibody



# Environment Modeling

Low-Fidelity Model

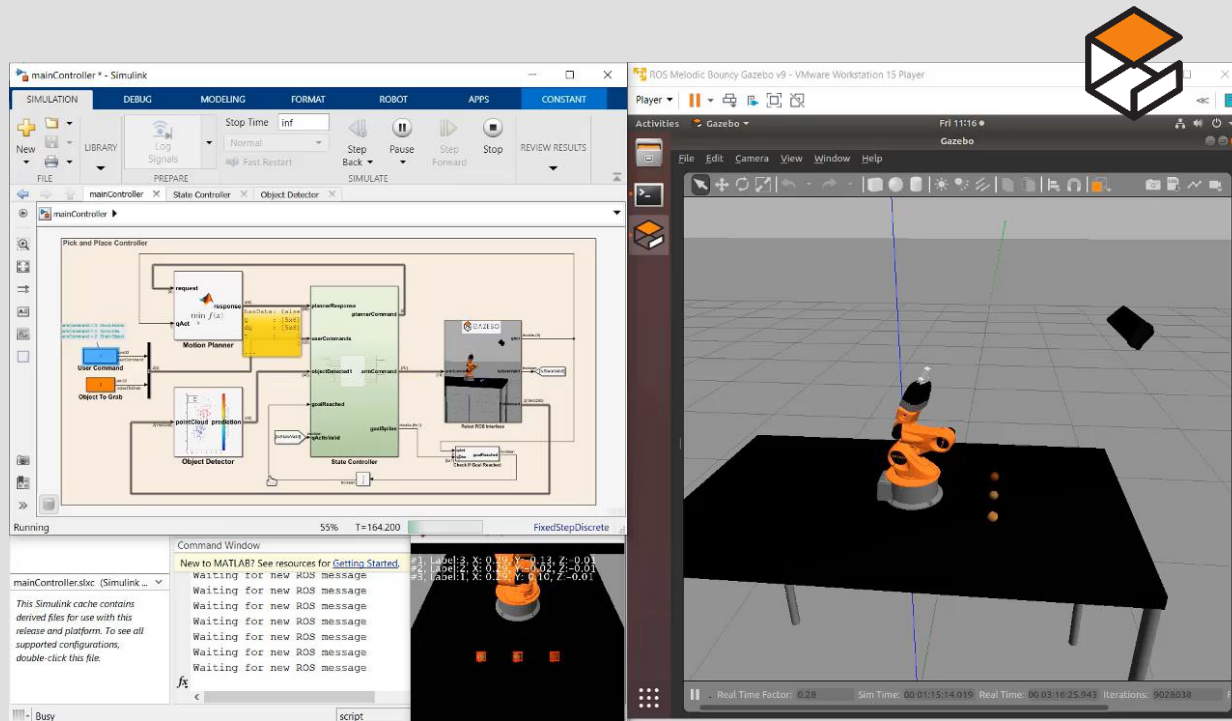
Rigid Body Kinematics & Dynamics

Multibody Modeling

Co-Sim with Gazebo

Co-Sim with Unreal

## Co-simulation with third-party simulators for sensors and environment models



# Environment Modeling

Low-Fidelity Model

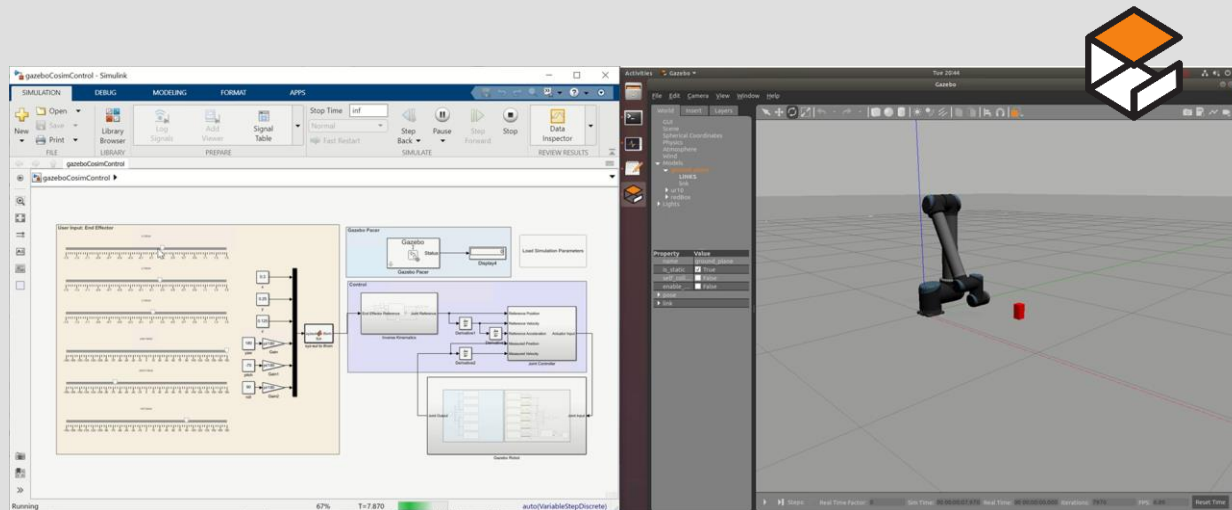
Rigid Body Kinematics & Dynamics

Multibody Modeling

Co-Sim with Gazebo

Co-Sim with Unreal

## Co-simulation with third-party simulators for sensors and environment models



### 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.



# Environment Modeling

Low-Fidelity Model

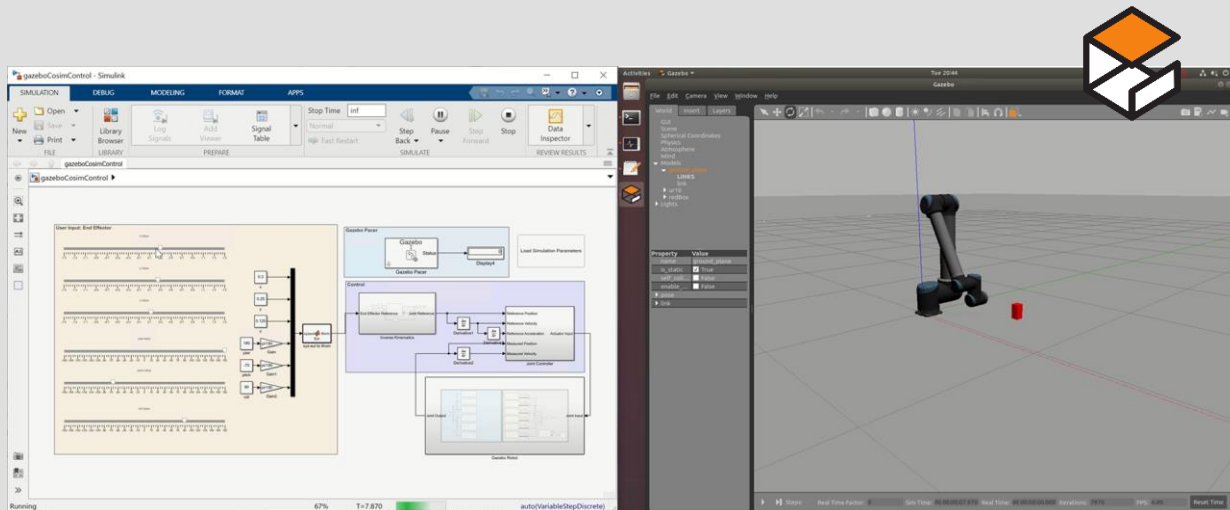
Rigid Body Kinematics & Dynamics

Multibody Modeling

Co-Sim with Gazebo

Co-Sim with Unreal

## Co-simulation with third-party simulators for sensors and environment models



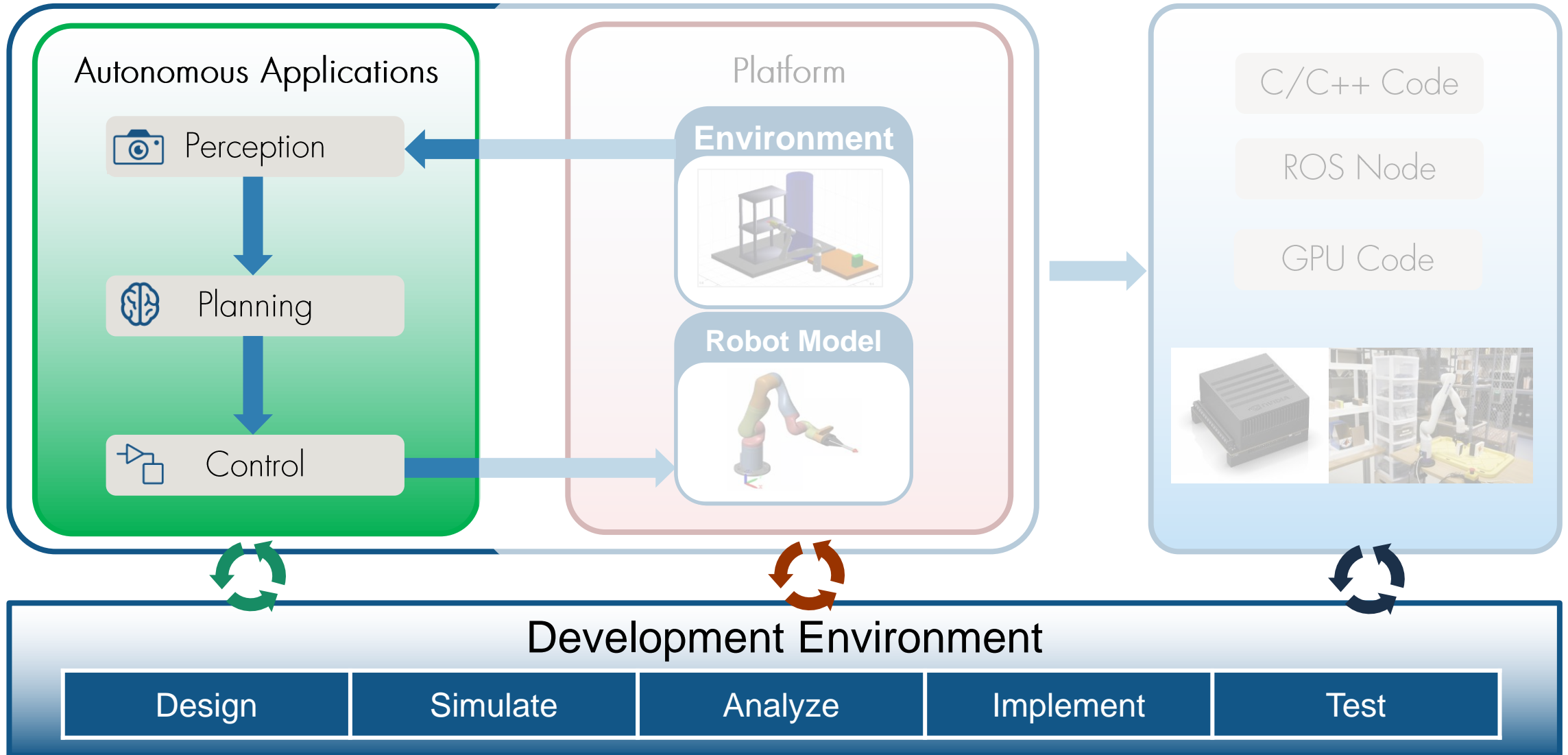
### Gazebo Co-simulation with Time Synchronization

#### 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.

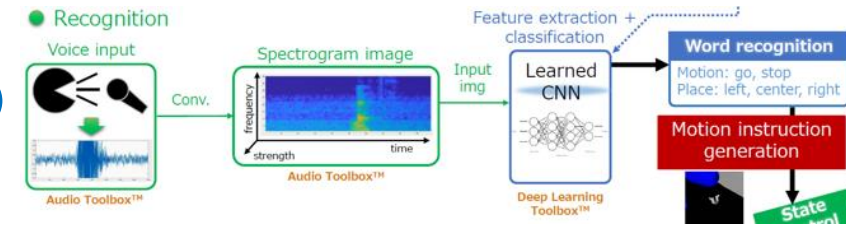


# Autonomous Robot Development with MATLAB & Simulink



# How is deep learning used for robotics?

## □ Speech Recognition



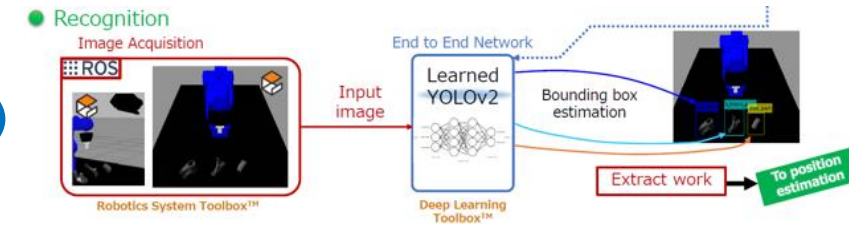
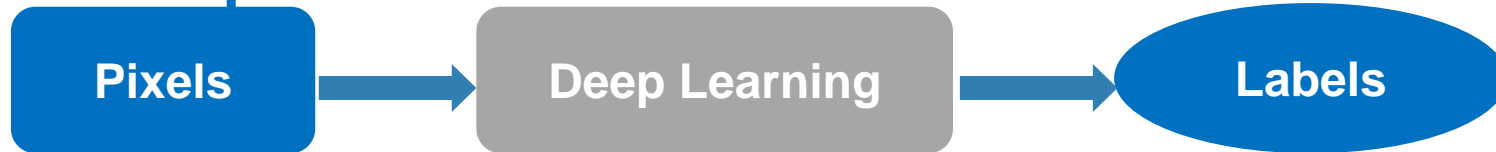
Voice Command to Robots

# How is deep learning used for robotics?

## Speech Recognition



## Computer Vision

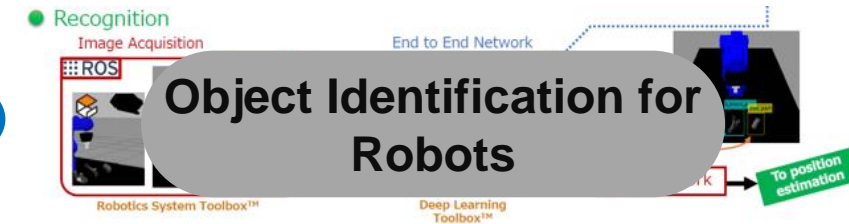


# How is deep learning used for robotics?

## Speech Recognition



## Computer Vision

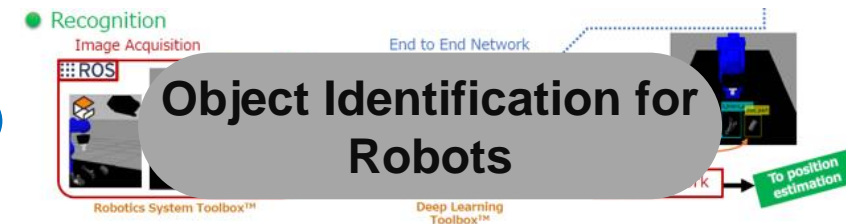
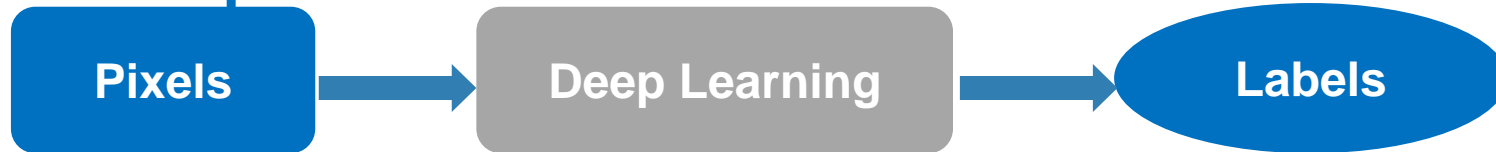


# How is deep learning used for robotics?

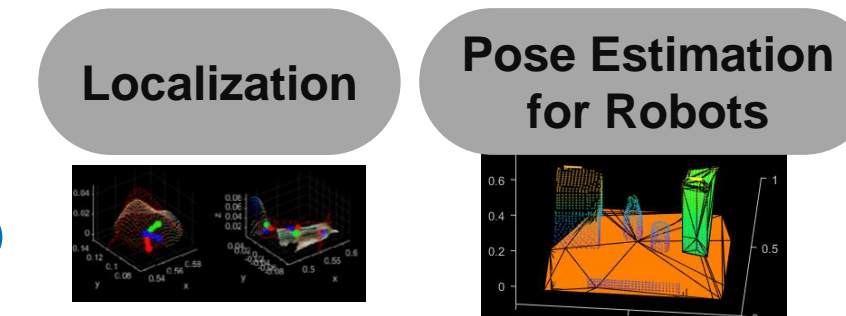
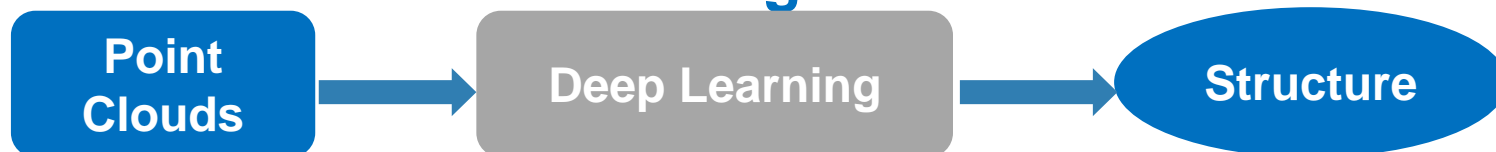
## Speech Recognition



## Computer Vision



## Point Cloud Processing



# How is deep learning used for robotics?

## Speech Recognition



Recognition  
Voice input  
Audio Toolbox™

Feature extraction + classification  
Word recognition  
Instruction generation  
State

**Voice Command to Robots**

## Computer Vision



Recognition  
Image Acquisition  
ROS  
End to End Network  
Robotics System Toolbox™  
Deep Learning Toolbox™

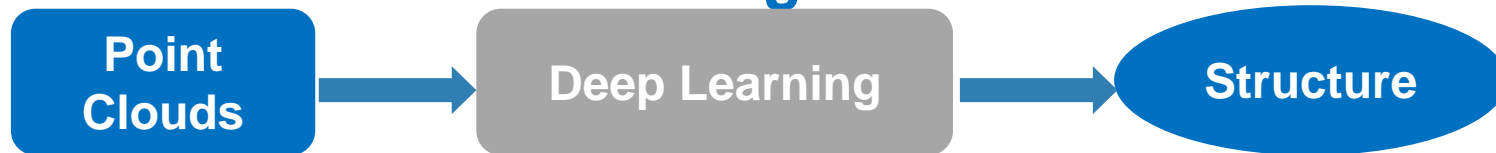
**Object Identification for Robots**



Image acquisition

**Industrial Inspection with Robots / UAVs**

## Point Cloud Processing

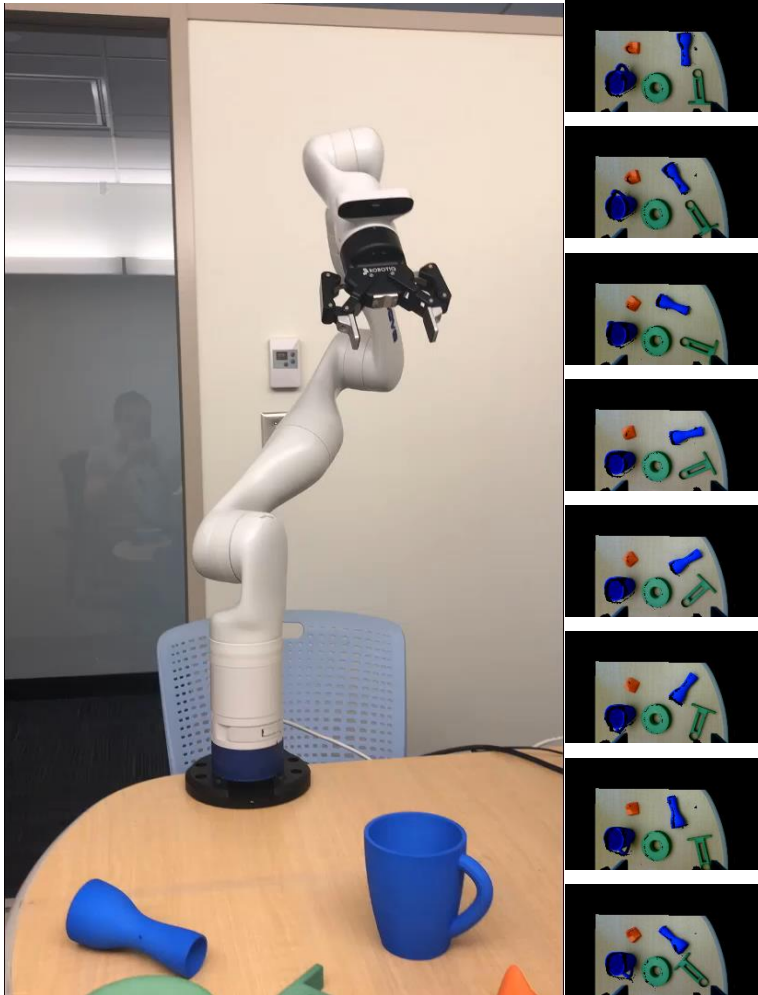


**Localization**

**Pose Estimation for Robots**

# Perception

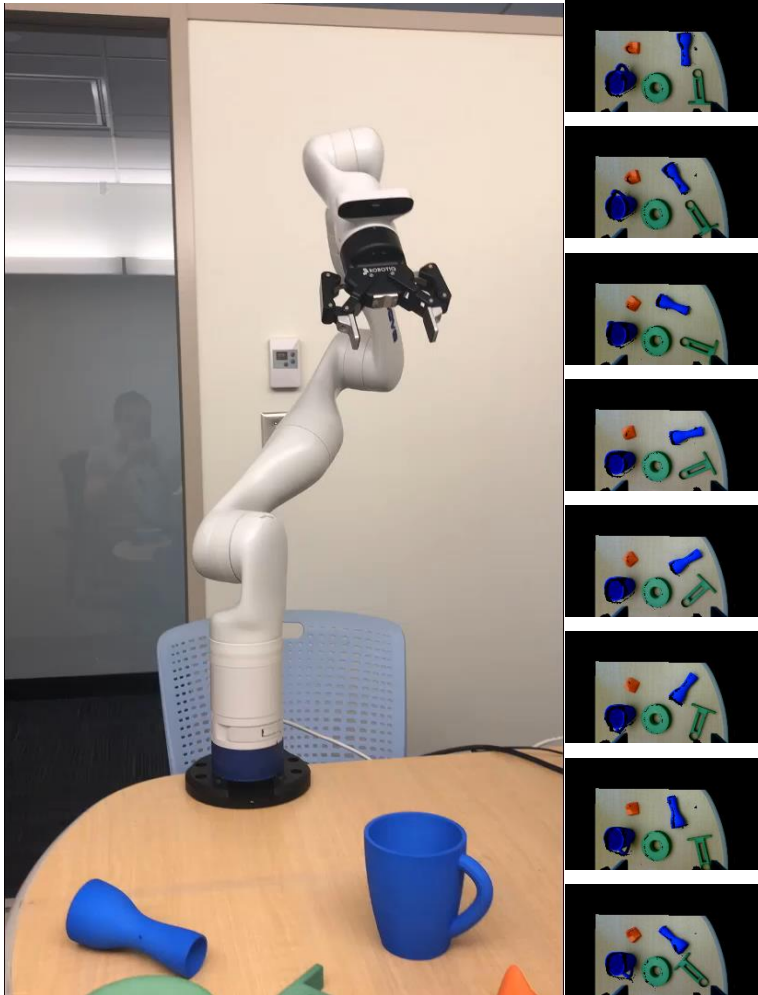
## Deep learning for object classification



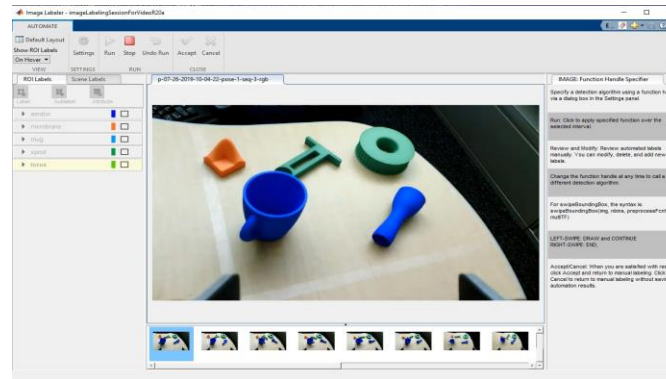


# Perception

## Deep learning for object classification

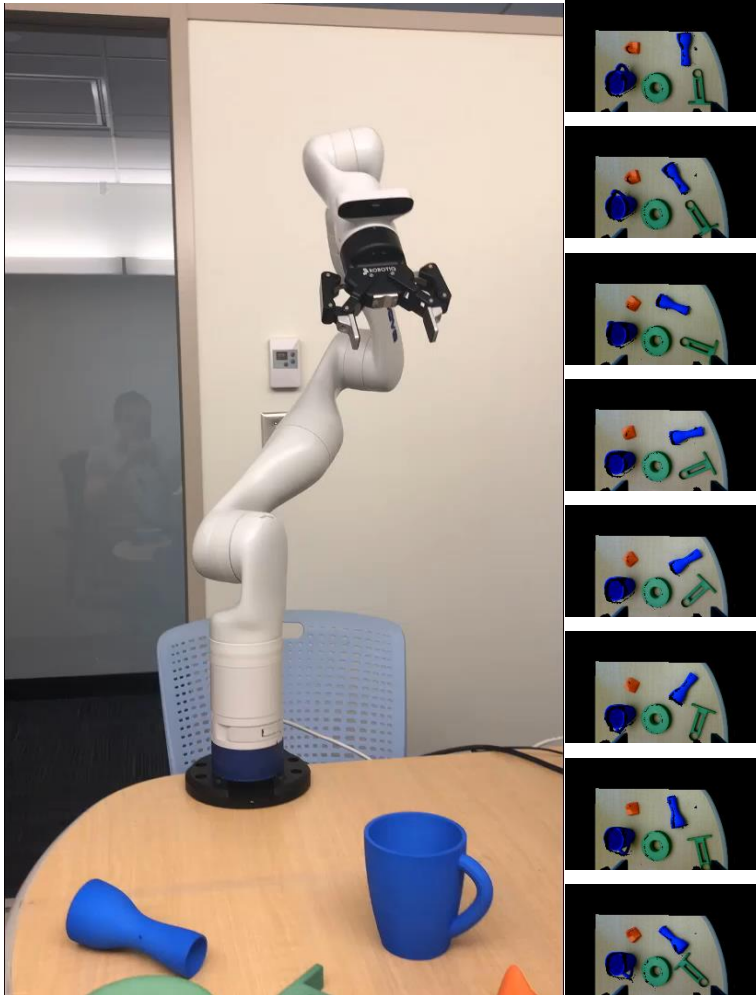


Automated Labeling & Iterative Learning



# Perception

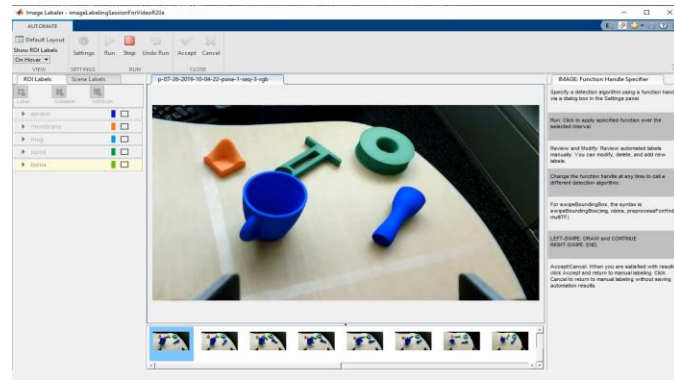
## Deep learning for object classification



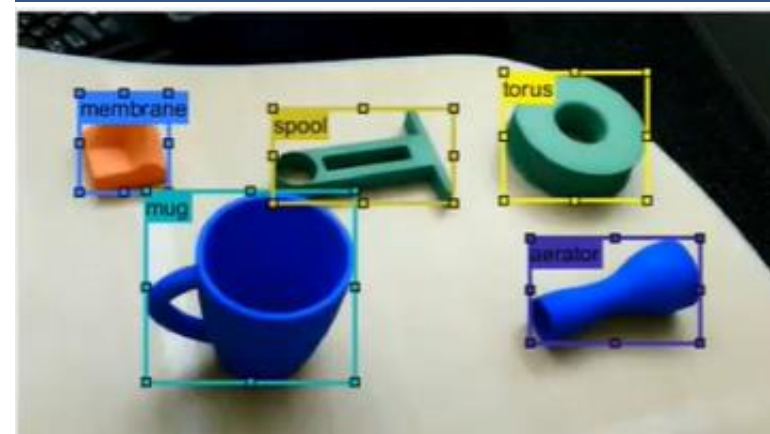
- 
- 
- 



Automated Labeling & Iterative Learning

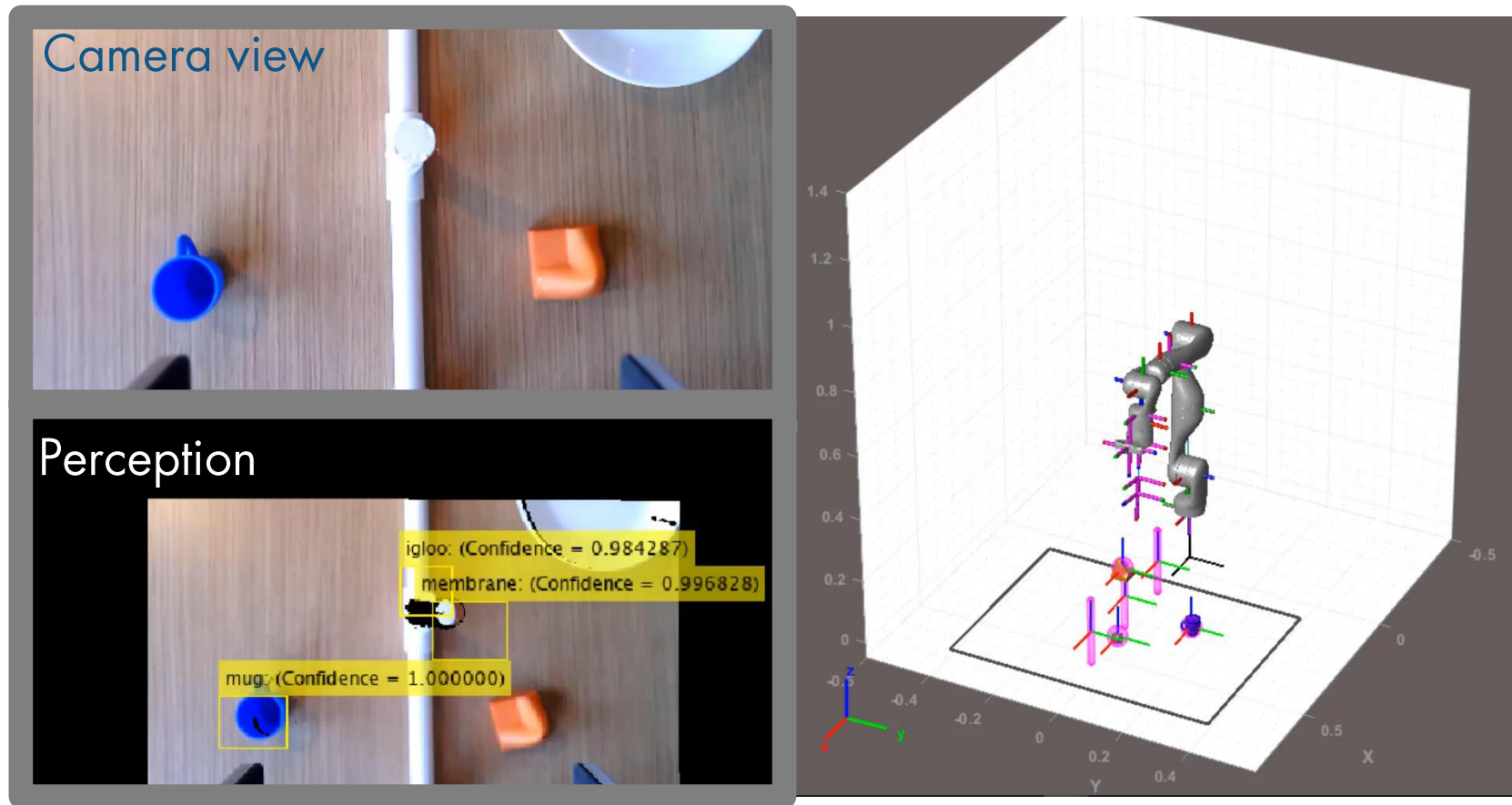


Object detector using Deep Learning (YOLO v2)

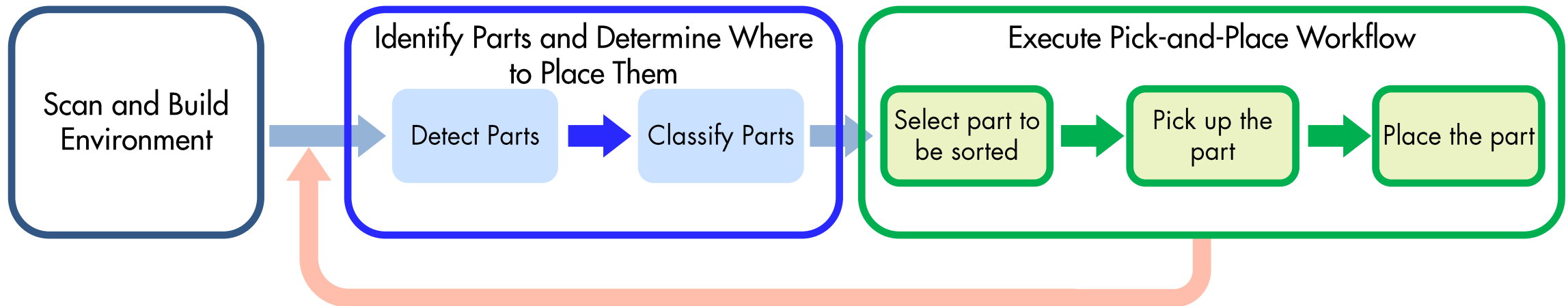


# Perception

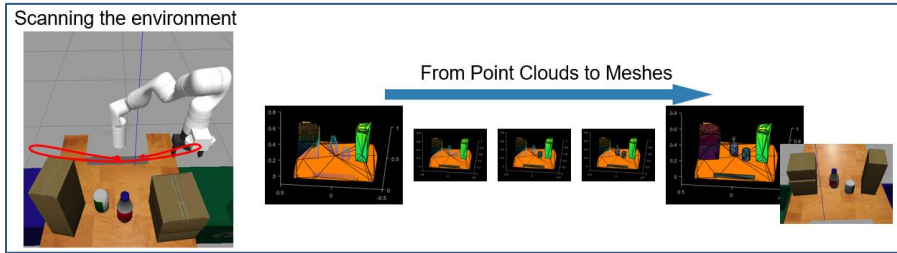
## Object Classification



# Full Workflow of Pick-and-Place Robots

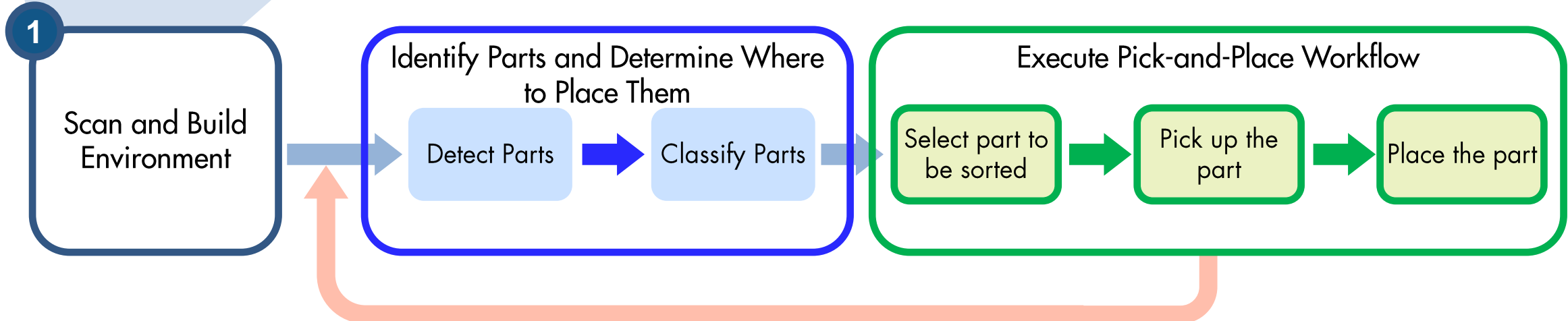


# Full Workflow of Pick-and-Place Robots

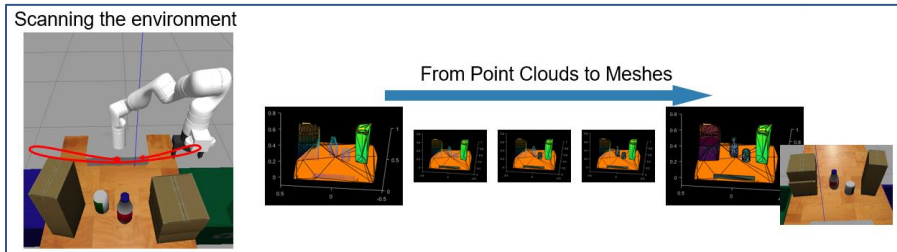


Scan-and-Build Environment

- Dynamic environment
- Flexible operations

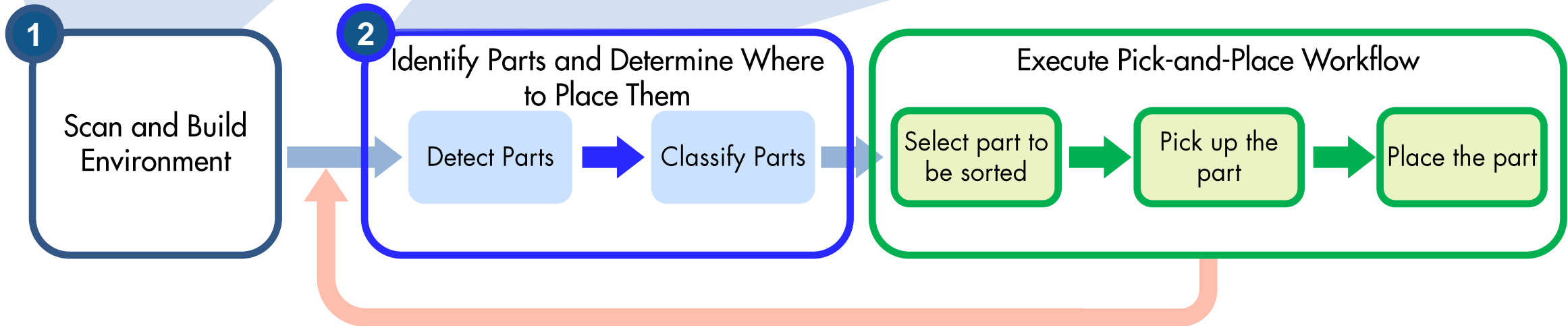
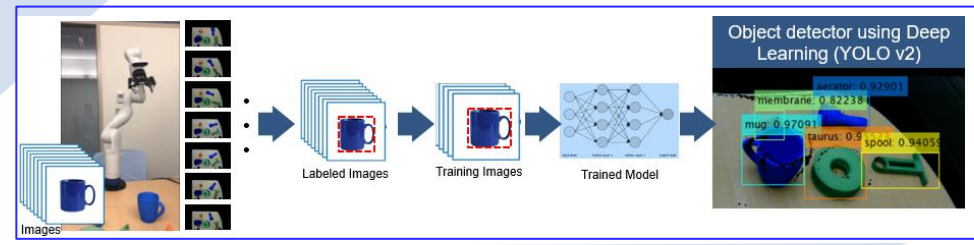


# Full Workflow of Pick-and-Place Robots

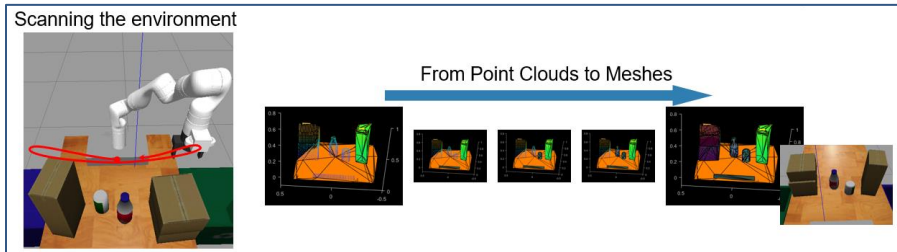


- Dynamic environment
- Flexible operations

## Scan-and-Build Environment

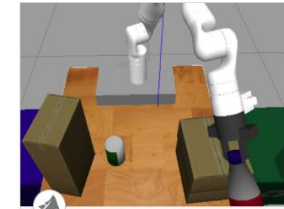


# Full Workflow of Pick-and-Place Robots



- Dynamic environment
- Flexible operations

## Shipping examples



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

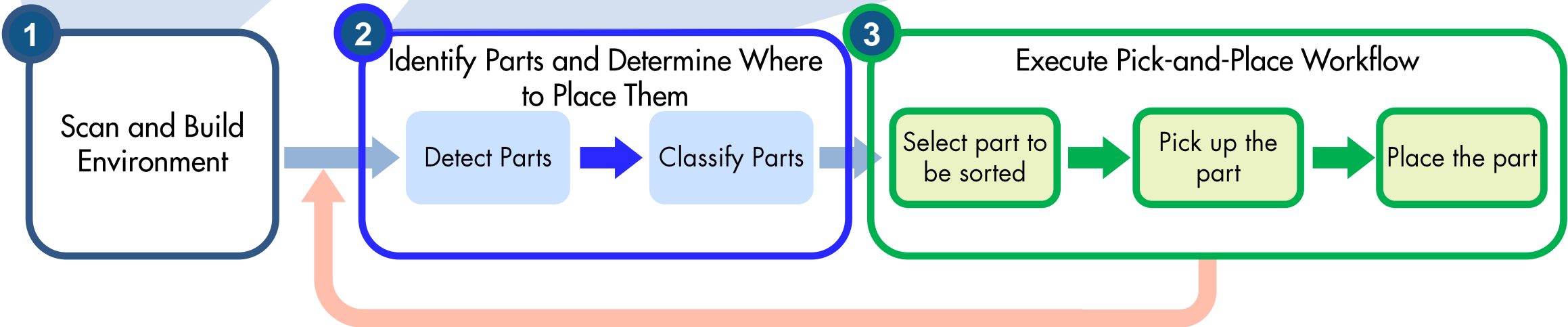
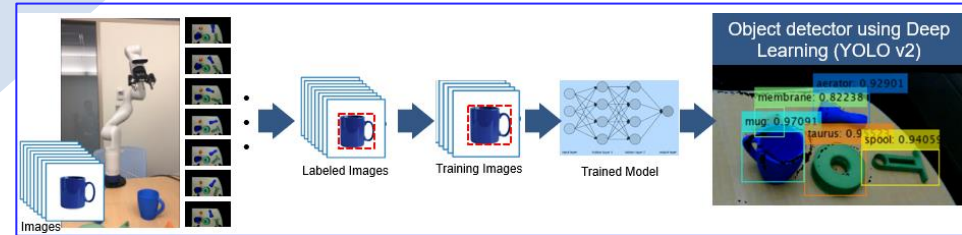
Setup an end-to-end pick and place workflow for a robotic manipulator like the KINOVA® Gen3.



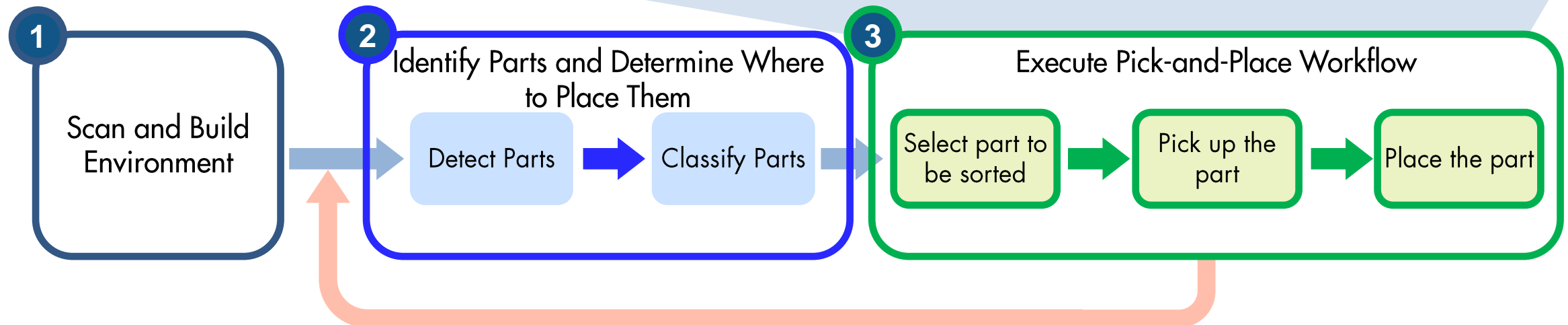
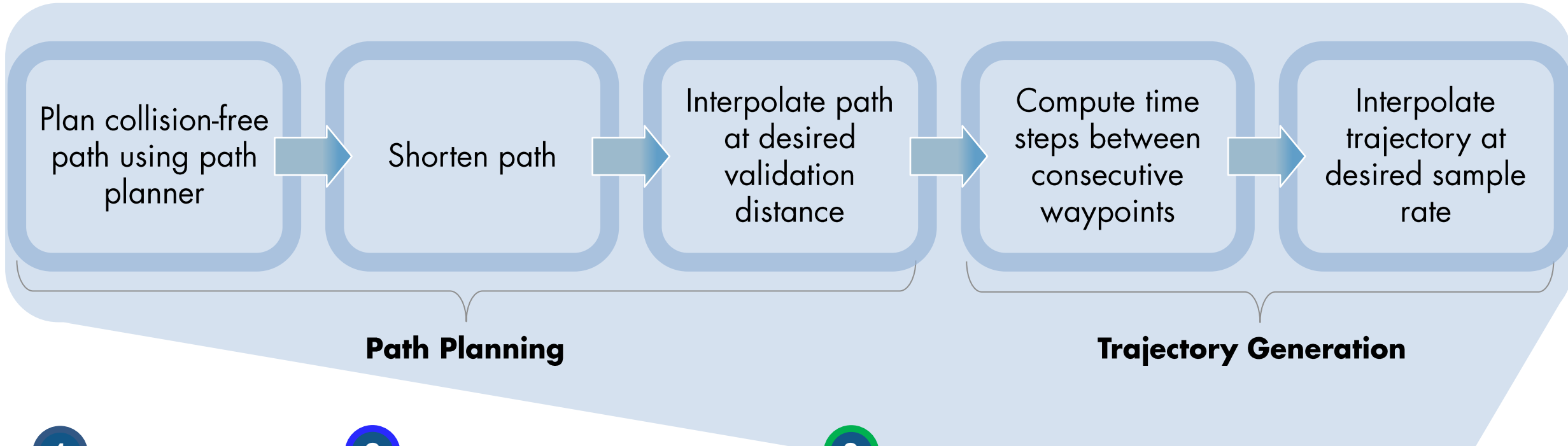
**Pick-and-Place Workflow in Gazebo using ROS**

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

## Scan-and-Build Environment

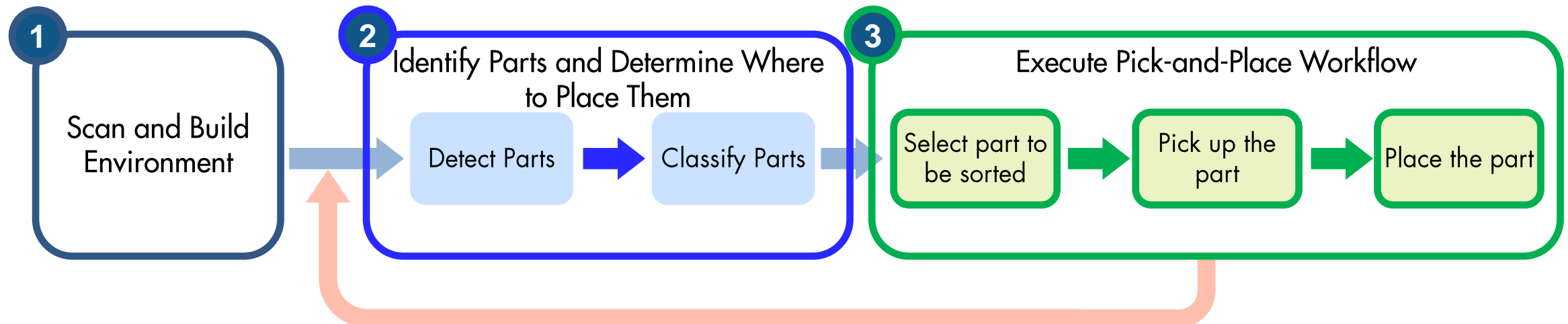
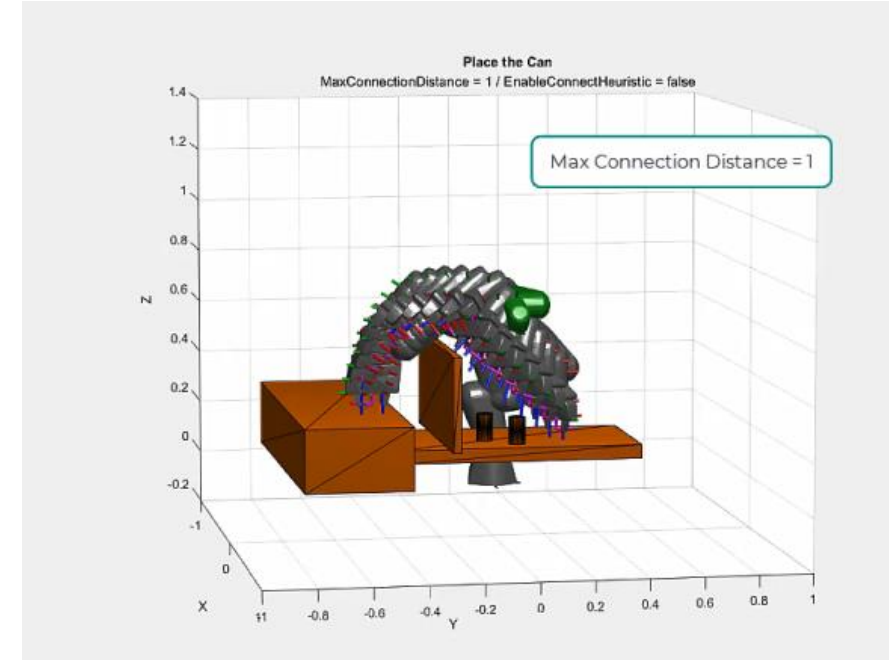
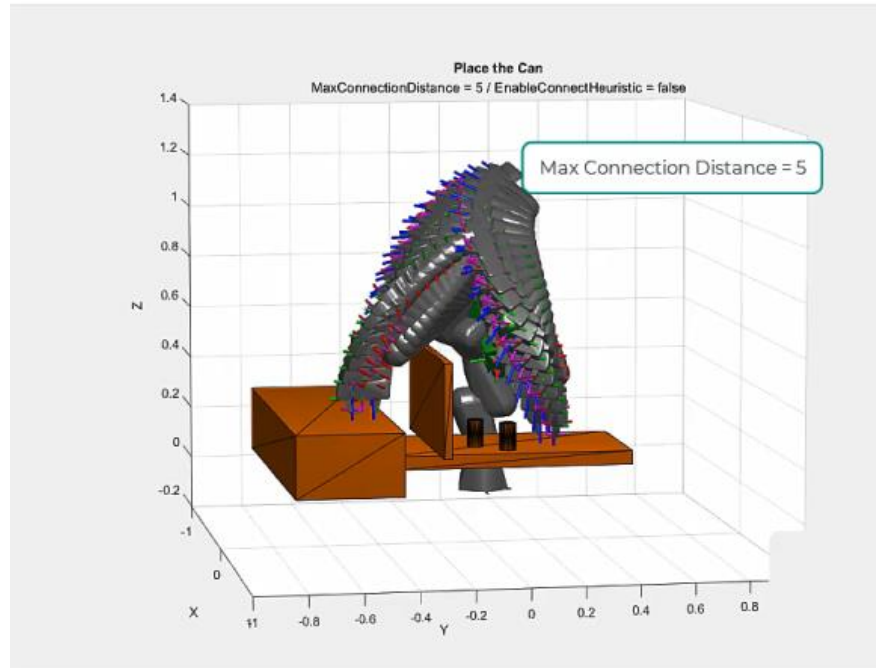


# Full Workflow of Pick-and-Place Robots

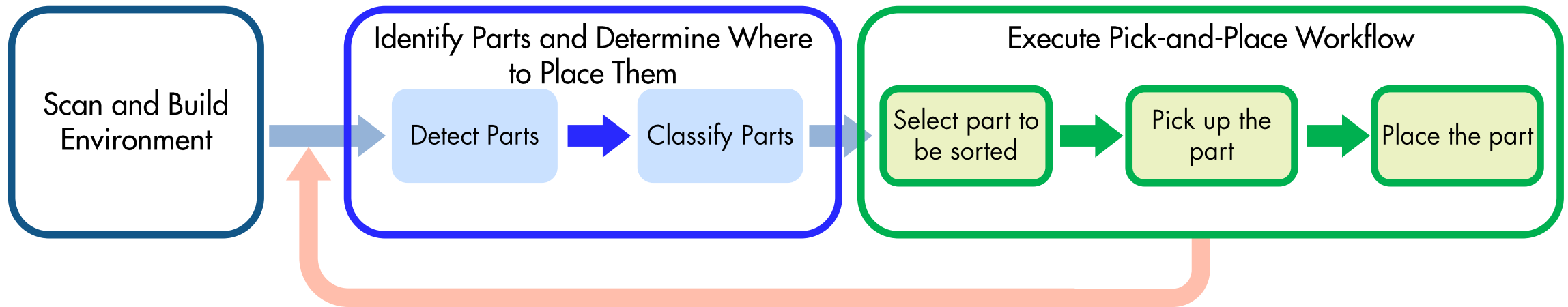


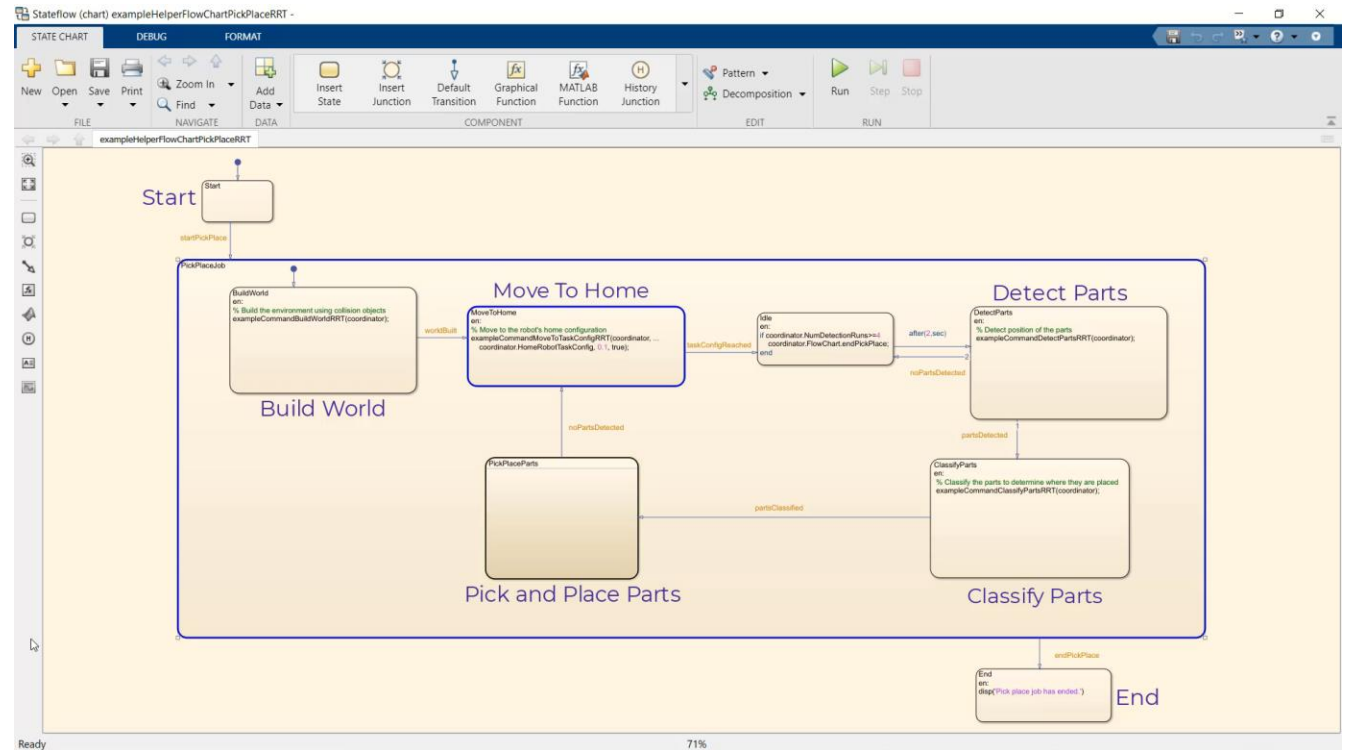
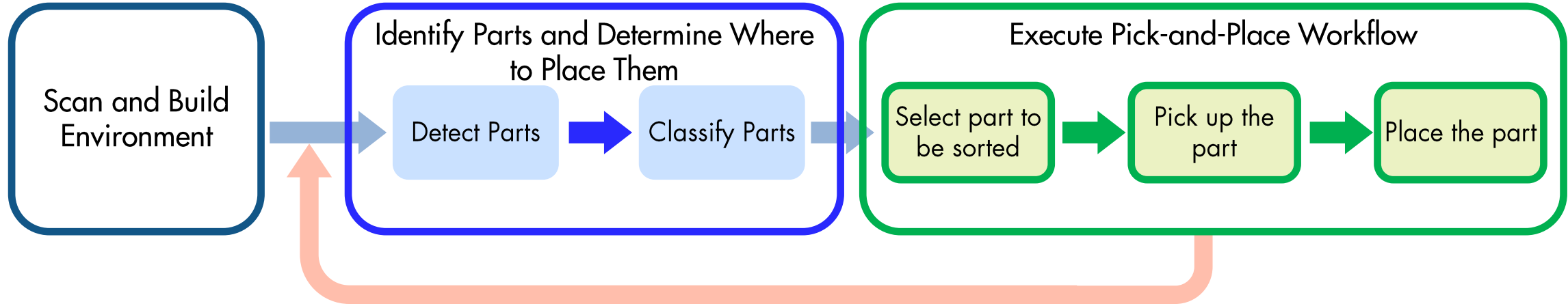


# Full Workflow of Pick-and-Place Robots



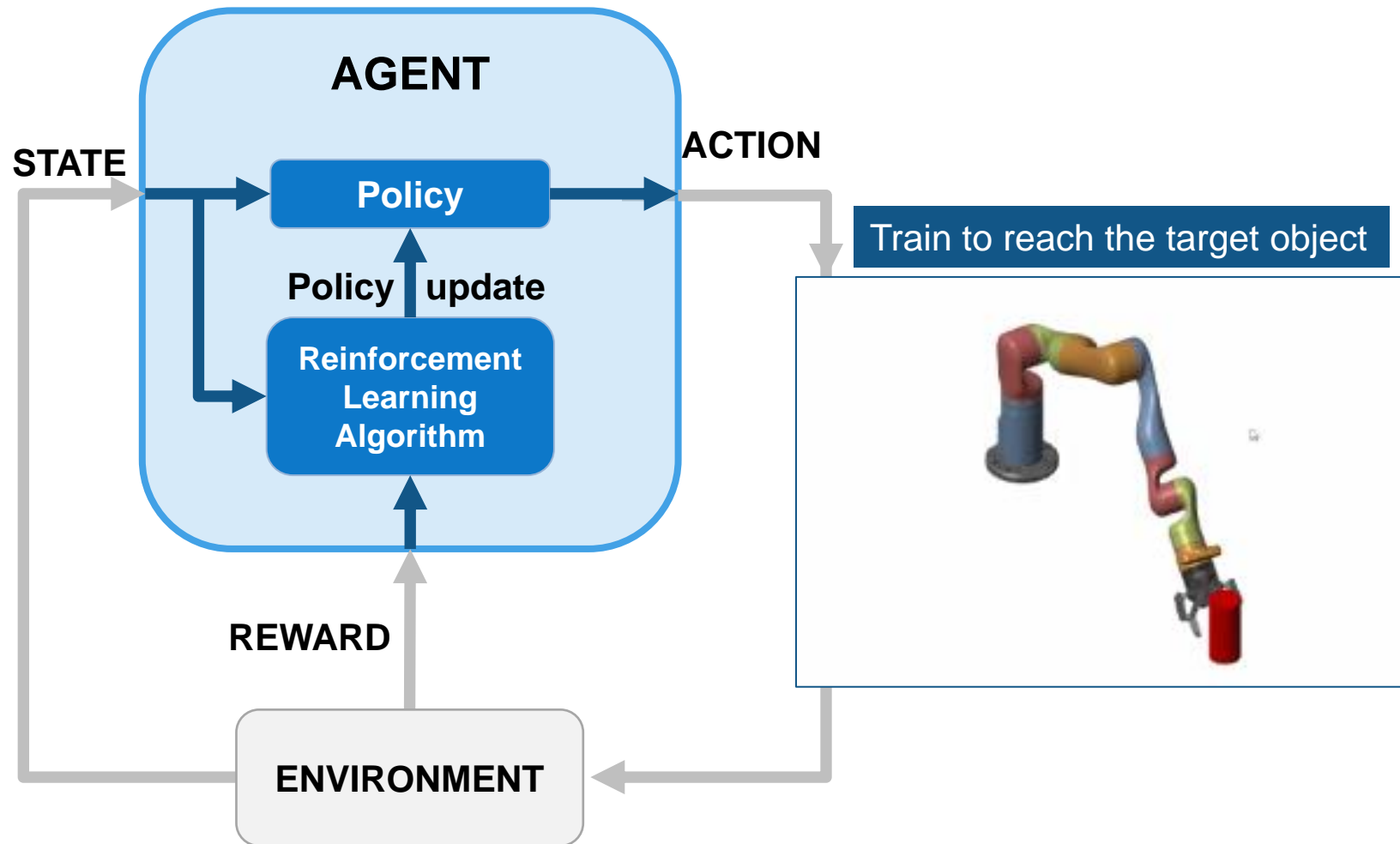
# Full Workflow of Pick-and-Place Robots





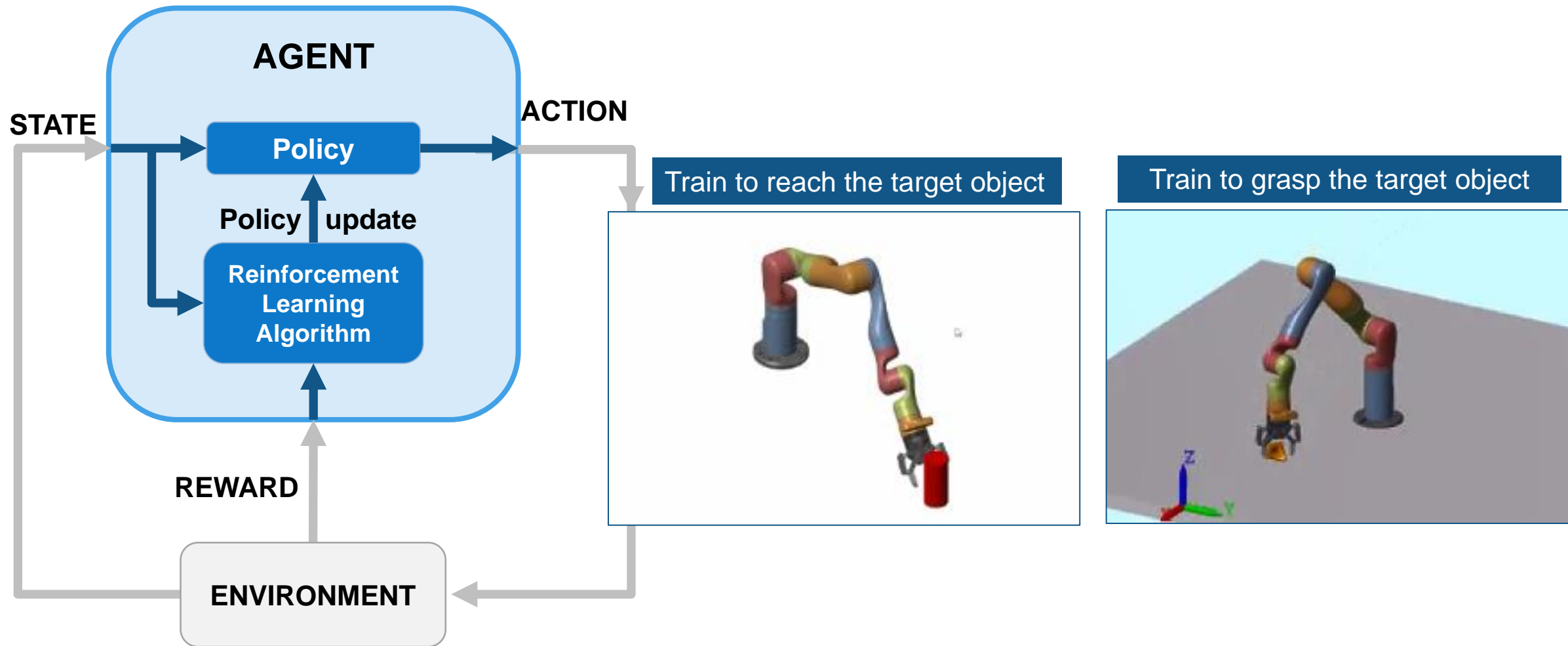
# Advanced Control: Reinforcement Learning

Train robot to reach the target object

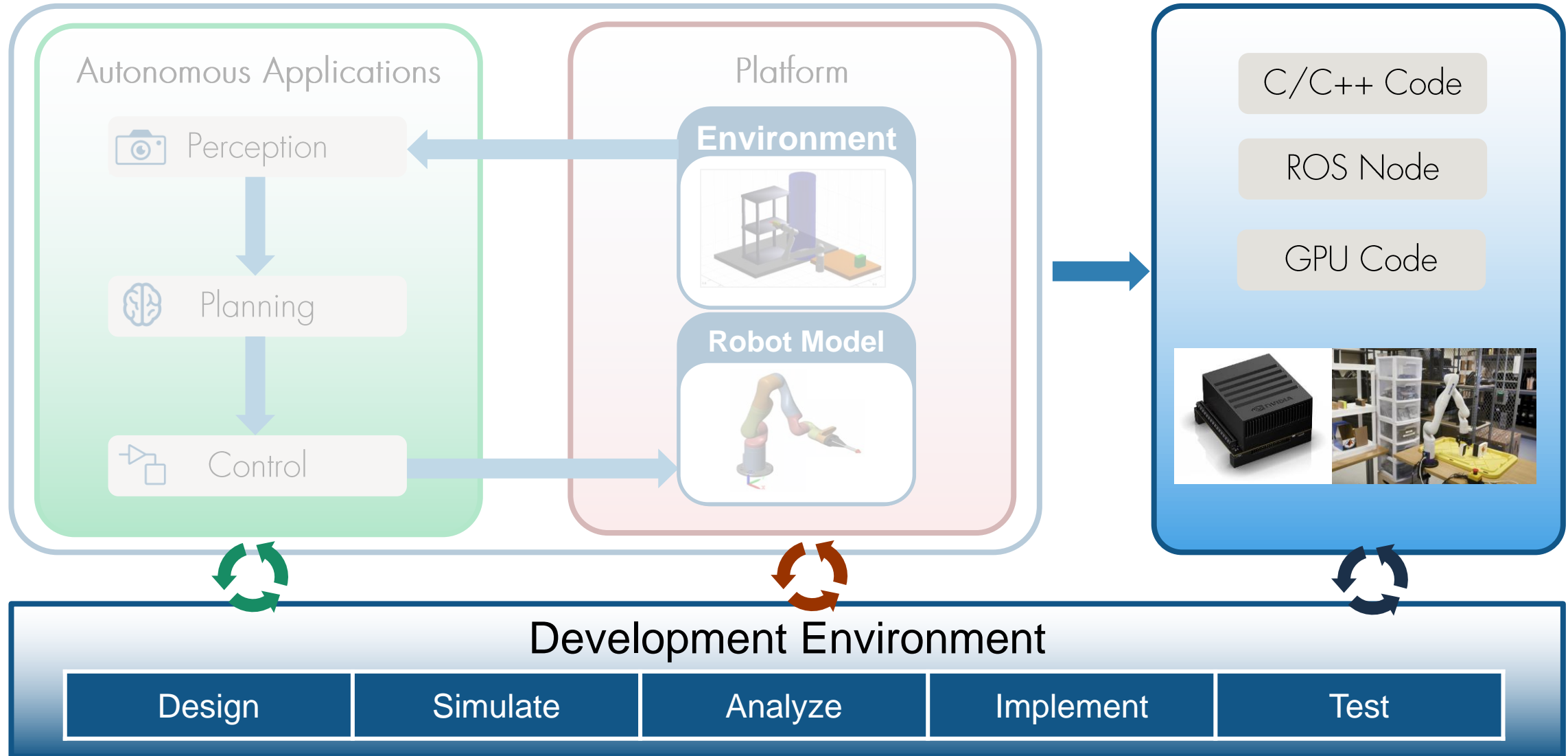


# Advanced Control: Reinforcement Learning

Train robot to reach the target object

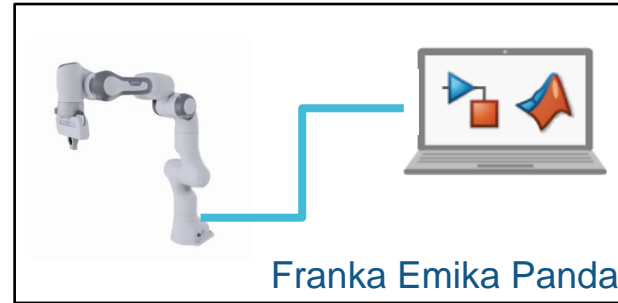
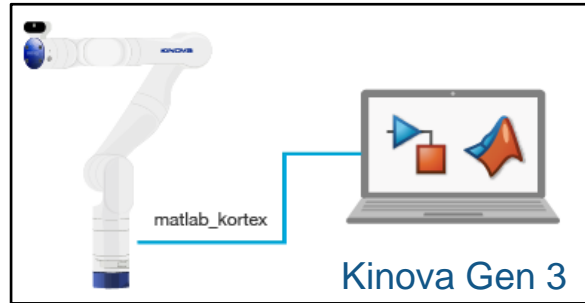


# Autonomous Robot Development with MATLAB & Simulink



# Hardware Connectivity & Deployment

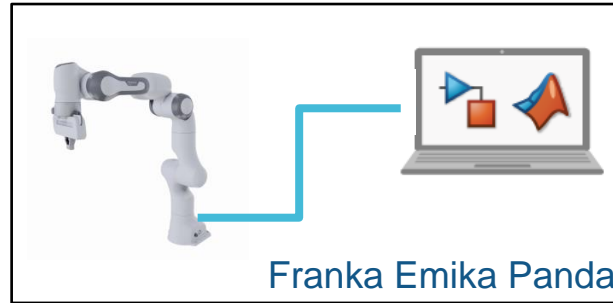
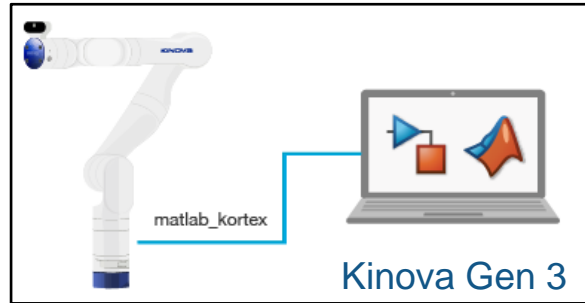
1



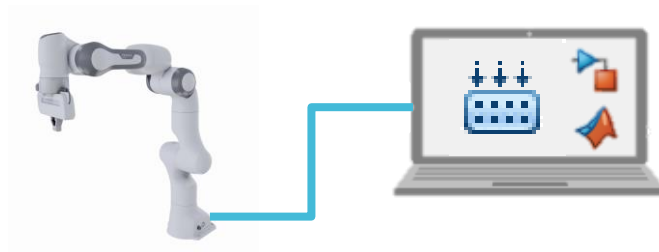
MATLAB APIs

# Hardware Connectivity & Deployment

1

**MATLAB APIs**

2



Design  
algorithms in  
MATLAB &  
Simulink

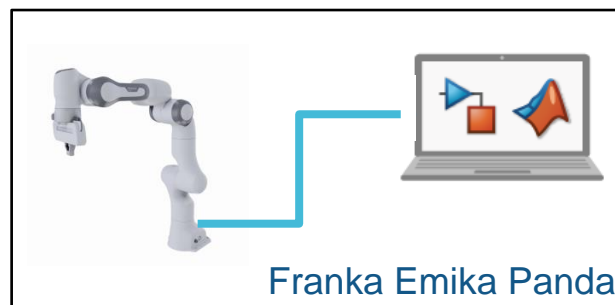
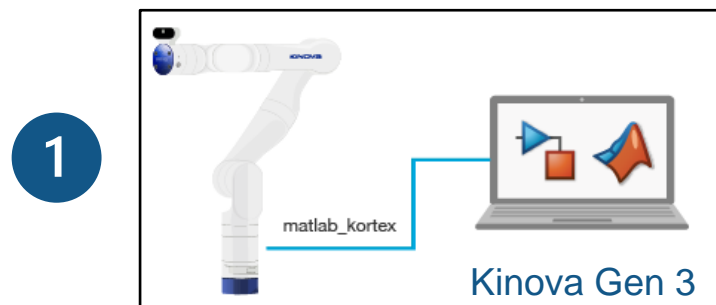
Generate  
code

Package  
using  
packNGo

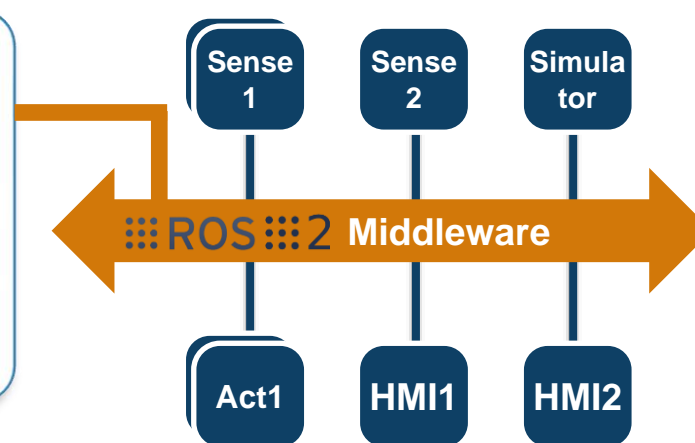
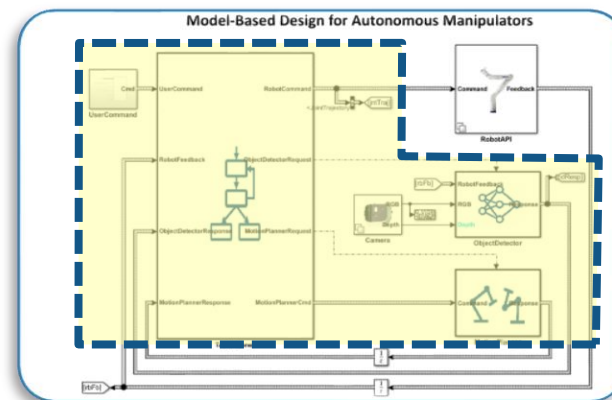
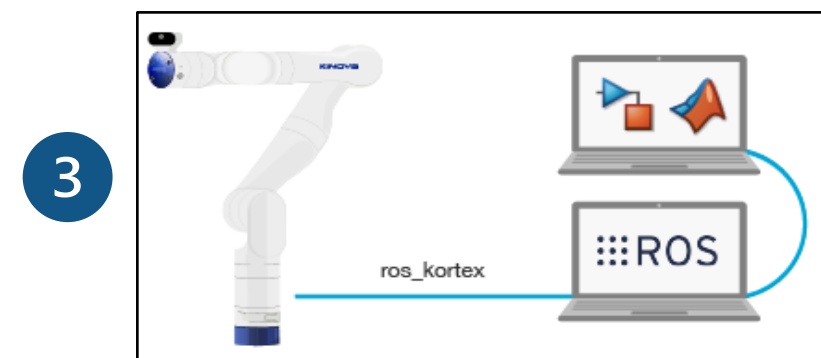
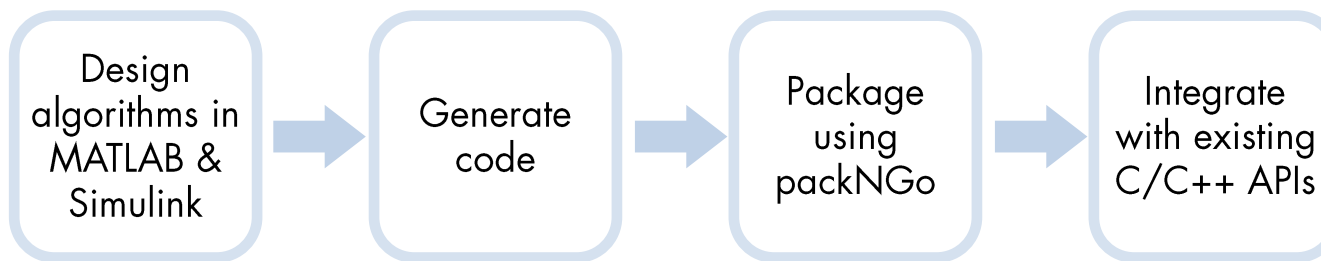
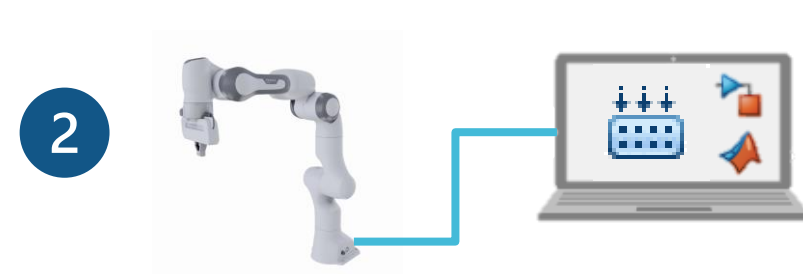
Integrate  
with existing  
C/C++ APIs



# Hardware Connectivity & Deployment

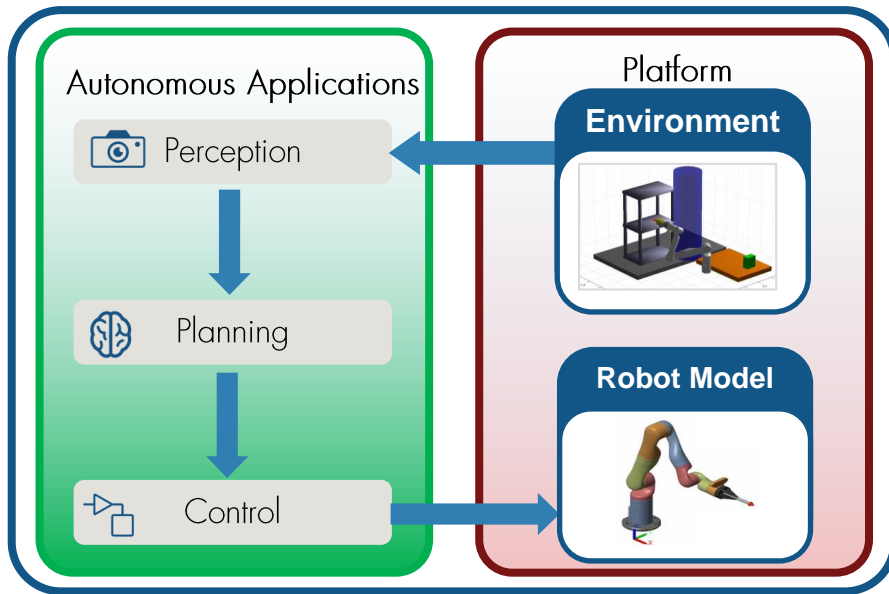


MATLAB APIs



Application

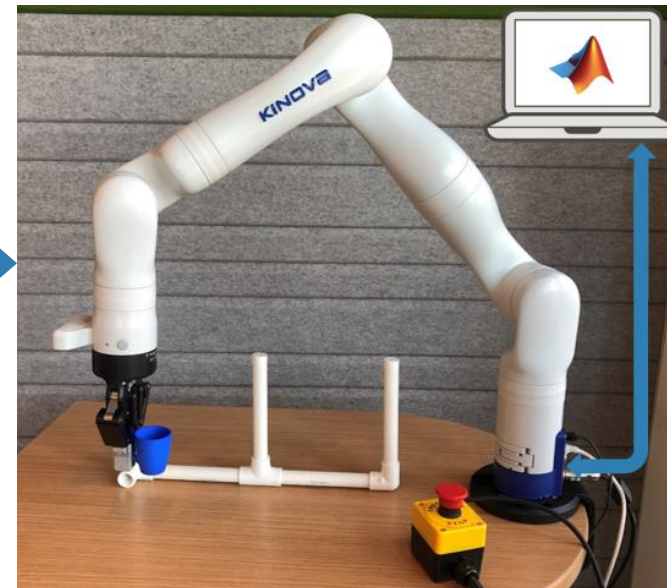
# Hardware Connectivity



## Robotics System Toolbox Supported Hardware

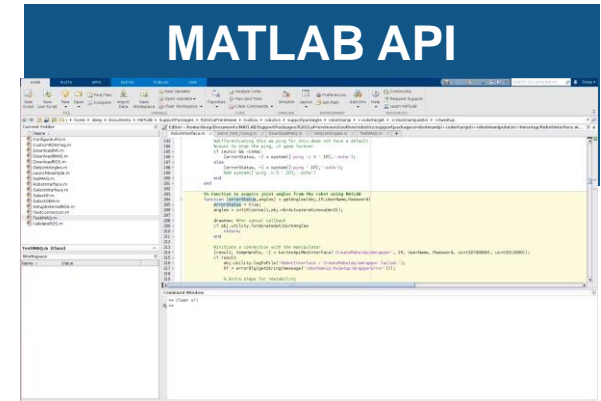
Support for third-party hardware

[Get Support Package Now](#)



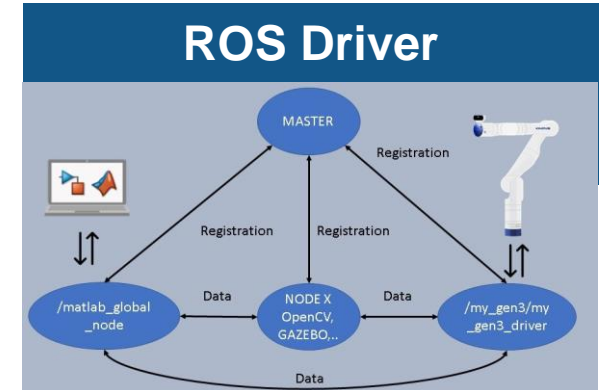
R2020b

## MATLAB API

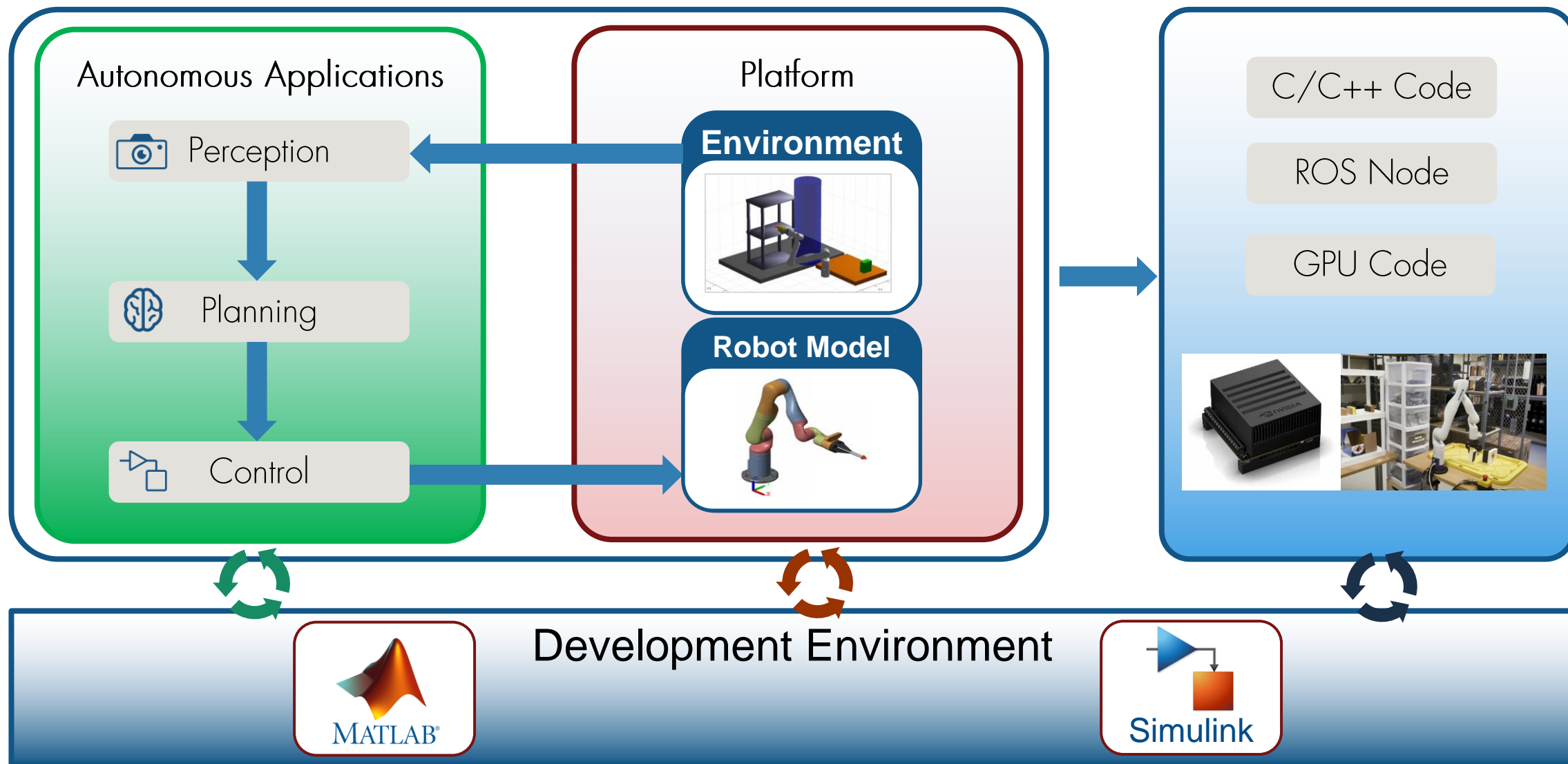


R2021a

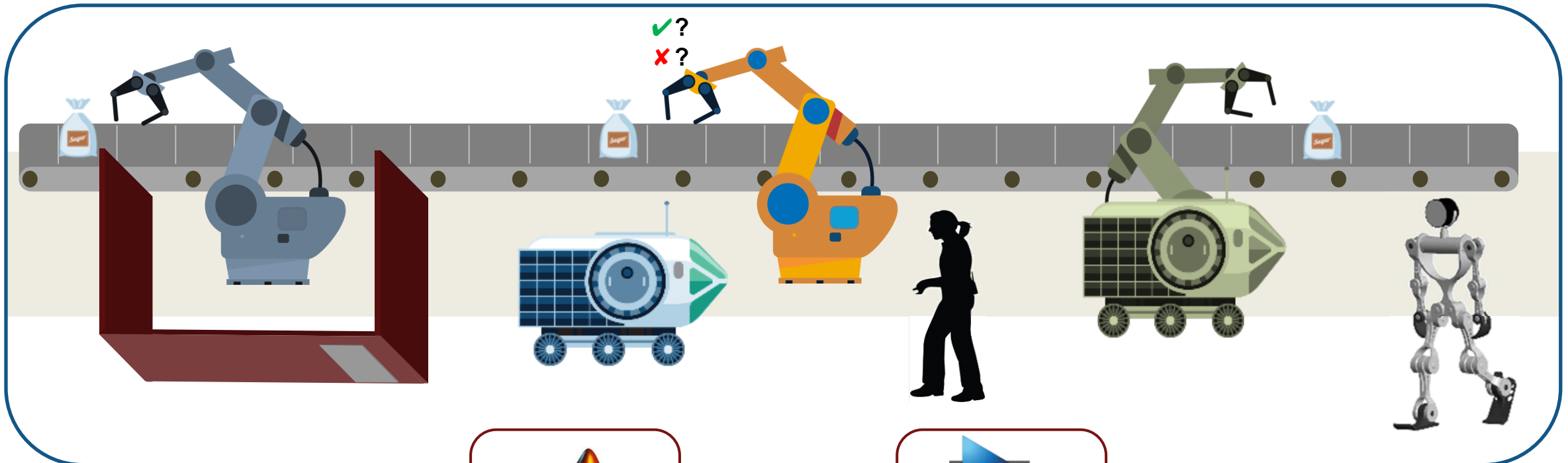
## ROS Driver



# Autonomous Robot Development with MATLAB & Simulink



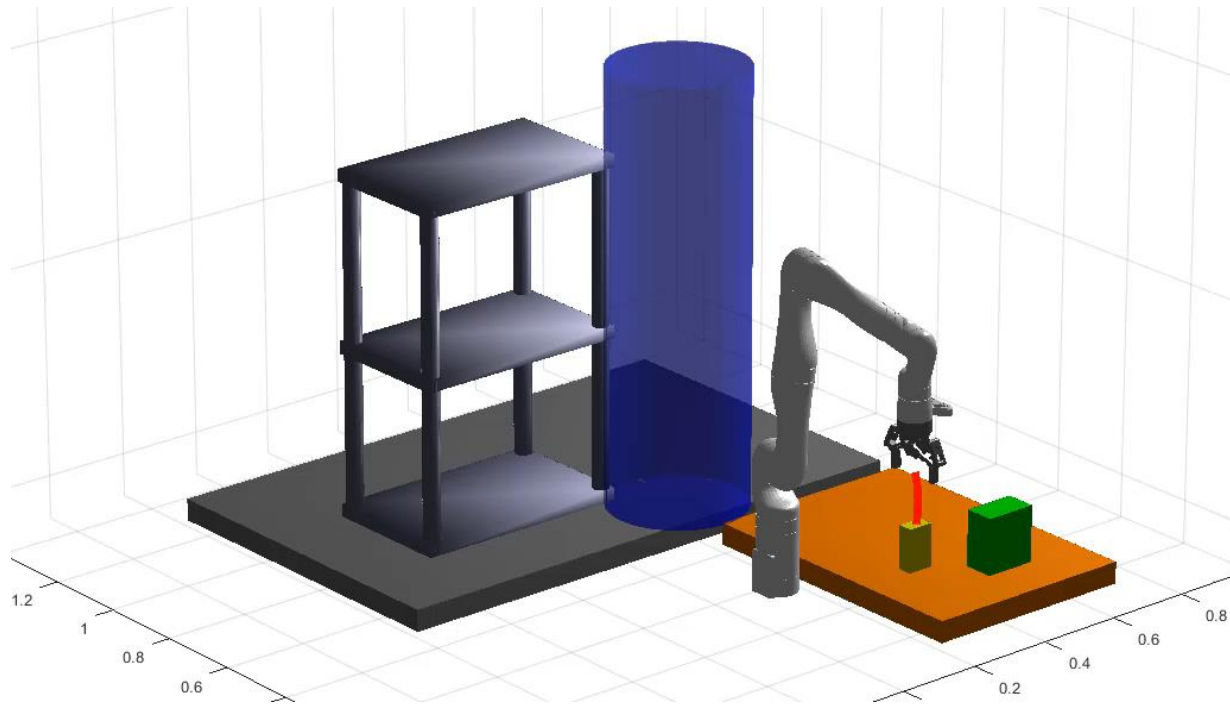
# Autonomous Robot Development for Smart Factories



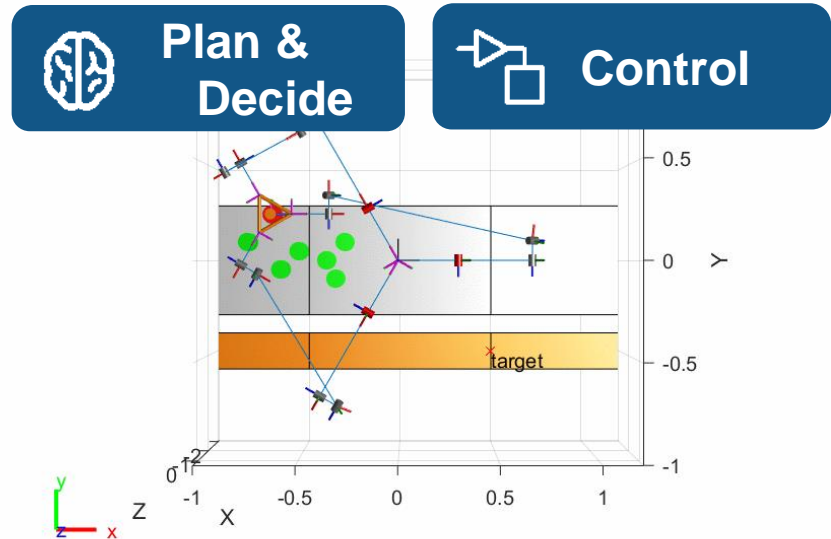
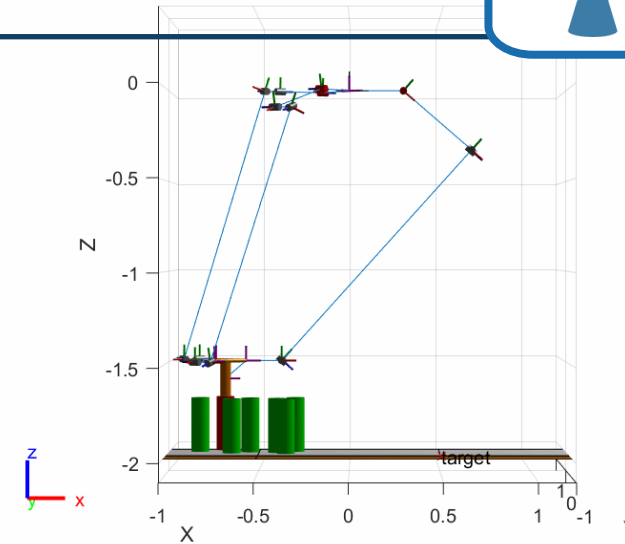
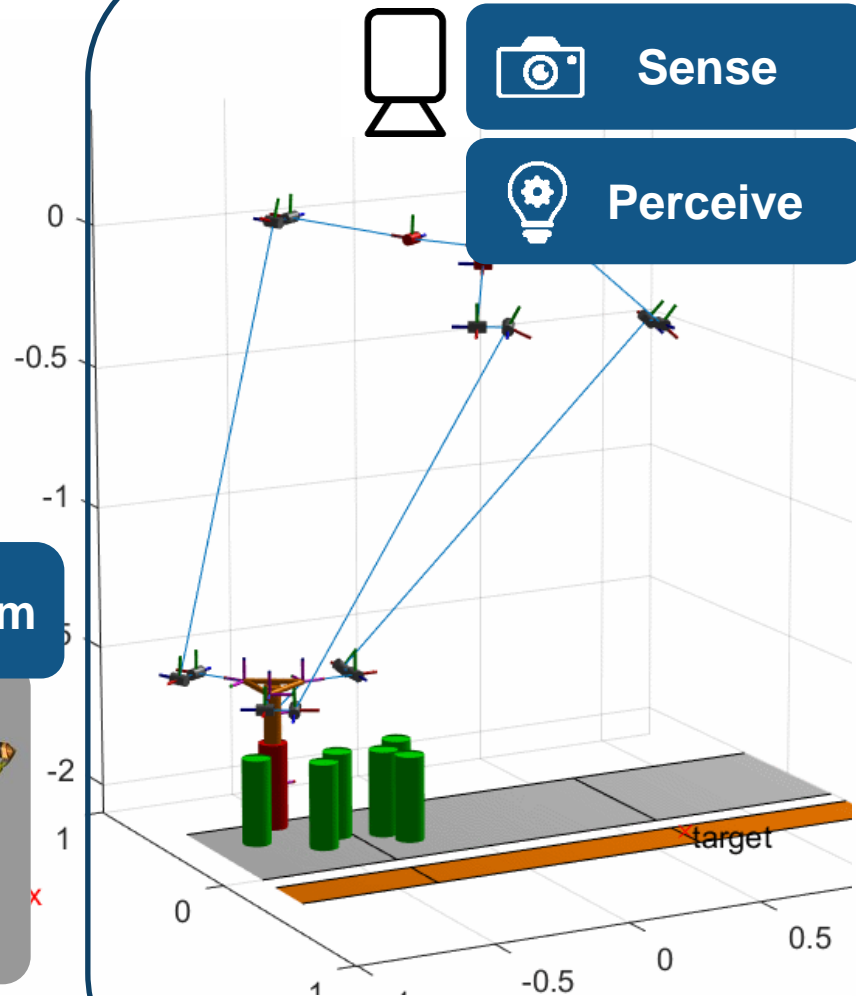
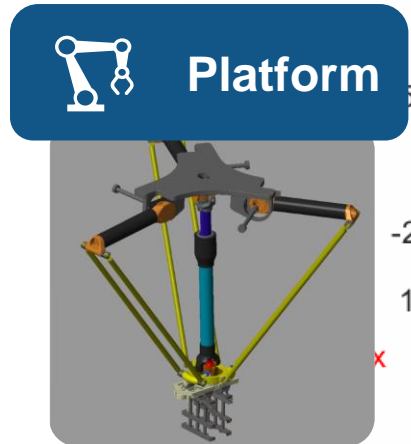
**Develop Autonomous Applications from Perception to Motion Planning  
and Optimize System-Level Behavior**

**Supported by MATLAB/Simulink Unified Environment**

# Use Cases: Warehouse Pick-and-Place Robot



# Use Cases: Delta Robot for Automated Parts Sorting



# Key Takeaways

- Advanced robotics systems in the factory of the future
  - > Collaborative Robots
  - > AI-enabled Robots
- Three pillars to develop autonomous robotics systems
  - > Platform design with environment models
  - > Autonomous application design
  - > Deployment
- MATLAB and Simulink is a unified development environment
  - > Develop autonomous applications from perception to motion planning
  - > Simulate and optimize system-level behavior

# Learn More

MathWorks® Products Solutions Academia Support Community Events

Robotics and Autonomous Systems

Overview Robot Manipulators Mobile Robots UAV Resources

## MATLAB and Simulink for Robotics and Autonomous Systems

Develop autonomous applications from perception to motion and optimize system-level behavior

Download a free trial

MathWorks® Products Solutions Academia Support Community Events

Robotics and Autonomous Systems

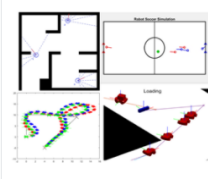
Overview Robot Manipulators Mobile Robots UAV Resources

## MATLAB and Simulink for Robot Manipulators

Download a free trial

### 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)



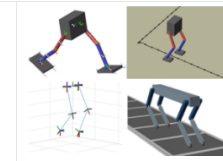
### Manipulation

- 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)



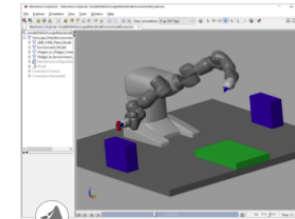
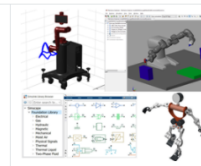
### Legged Locomotion

- 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



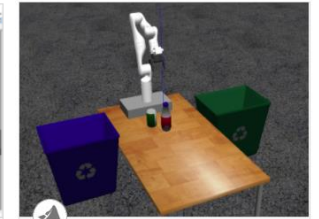
### Robot Modeling

- 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



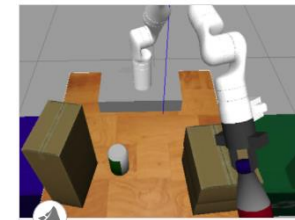
### Model And Control A Manipulator Arm With Robotics And Simscape

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



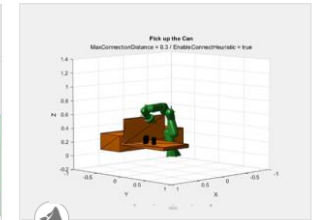
### Pick-and-Place Workflow in Gazebo using ROS

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

MathWorks Robotics Solution Page  
[mathworks.com/robotics](https://mathworks.com/robotics)

Awesome-MATLAB-Robotics  
[GitHub Repo](#)

[Robotics Examples](#)



# MATLAB EXPO

## 2021

## Thank you

Contact:



YJ Lim  
[yjlim@mathworks.com](mailto:yjlim@mathworks.com)



Ronal George  
[rgeorge@mathworks.com](mailto:rgeorge@mathworks.com)

