MATLAB EXPO 2019

Developing Autonomous Robots with MATLAB and Simulink

Veer Alakshendra
Challenges with Autonomous Robotics Systems

- Applying Multidomain Expertise
- Complexity of Algorithms
- End-to-End workflows
Today: Design Pick and Place Application

Platform

Sense

Perceive

Plan & Decide

Control
Today: Design Pick and Place Application

Platform

Sense

Perceive

Plan & Decide

Control

MATLAB EXPO 2019
Platform Design

How to create a model of my system that suits my needs?
Import models from common CAD Tools

**SolidWorks Model**

** Simscape Multibody Model**

Mechanics
% Import robot from URDF
smimport('j2n6s300_standalone_stl.urdf');
Actuators: Model other domains
Modeling Actuators
Design Pick and Place Application

Platform

Sense

Perceive

Plan & Decide

Control

Demo

MATLAB EXPO 2019
Design Pick and Place Application

- Support for Common Sensors
- Image analysis
- Apps
- Image enhancement
- Visualizing Point Clouds
Today: Design Pick and Place Application

Platform

Sense

Perceive

Plan & Decide

Control
Object Classifier and Pose Estimator

Images

Labels and Poses

Object 1

Object 2

Object 3

Object 4
MATLAB makes machine learning easy and accessible

Traditional Machine Learning approach

Traditional Feature Extraction → Classification

- Machine Learning
- Dog ✓
- Boy ×
- Bicycle ×

Deep Learning approach

Convolutional Neural Network (CNN)

- Learned features
- Feature learning + Classification
- End-to-end learning
- Dog ✓
- Boy ×
- Bicycle ×
% Detect regions
BW = createMask(videoFrame);

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

% Get bounding boxes
stats = regionprops('table',BW,'BoundingBox','Area');

% Filter based on area size
targetIndex = stats.Area > 500;

% Get bounding boxes from detected regions
testFeatures(k,:) = extractHOGFeatures(Icr);
Design Pick and Place Application

## Platform

### Sense

### Perceive

### Plan & Decide

### Control
Planning: Find a path

Map
Initial Pose
Final Pose

Path Planner

Path

\[ [x_a, y_a, \theta_a] \]

\[ [x_b, y_b, \theta_b] \]
Plan with Stateflow

Waypoint Control

Inverse Kinematics Control

Joint Control

Gripper Control

MATLAB EXPO 2019
Design Pick and Place Application

Platform

Sense

Perceive

Plan & Decide

Control
Control: Explore Built In Functions: Inverse Kinematics

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

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

% Parameters to pass to the solver
weights = [1, 1, 1, 1, 1, 1];
q_init = 0.1*ones(numel(q_home),1);
Key Takeaway of this Talk

Success in developing an autonomous robotics system requires:

- Multi-domain simulation
- Trusted tools which make complex workflows easy and integrate with other tools
- Model-based design
Clearpath Robotics Accelerates Algorithm Development for Industrial Robots

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

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

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

“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
Deep Learning with MATLAB

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

Topics include:

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