MATLAB EXPO 2016
KOREA
4월 28일 (목)
등록 하기 matlabexpo.co.kr
Robotics Applications Development Using Robotics System Toolbox

강효석
Training Engineer
MathWorks Korea
Complexities of Robotics Application Development

MATLAB® and Simulink® solves challenges with robotics application development
Agenda

- Introduction
- ROS
- Robotics Development Workflow
- What’s new in Robotics System Toolbox?
- Conclusion
Introduction: What is Robotics Development?

MathWorks tools are already being used in complex system development
Introduction: MATLAB and Simulink in Robotics

Algorithm prototyping

Wide variety of resources on using MATLAB/Simulink in Robotics
Introduction: MATLAB / Simulink in Robotics Development

Efficient system level design that yields higher quality robotics systems
Introduction: Robotics System Toolbox for Robotics Development

- Connecting MATLAB/Simulink to ROS
- ROS data exploration and analysis
- Algorithms and transformation functions

Features for flexible and convenient robotics development
Impact of Robotics System Toolbox

1. Top Automotive, Aero-Defense, and Software companies are using these tools to develop advanced robotics applications.

2. More than 500 universities worldwide are already using the toolbox.

Tutorials and exhibitions at ICRA and IROS
User Story
Automated Driving at BMW

AUTOMATED DRIVING WITH ROS AT BMW.

MICHAEL AEBERHARD, THOMAS KÜHBECK, BERNHARD SEIDL, MARTIN FRIEDL, JULIAN THOMAS, OLIVER SCHEICL.

BMW GROUP

USING MATLAB/SIMULINK WITH ROS.

- MathWorks released the Robotics System Toolbox this year for ROS integration with Matlab/Simulink.
- Easily read and analyze data from ROS Bags → useful for evaluating the system
- Some of our software is implemented as a Simulink model.
  - Use the Toolbox to easily integrated this software into the ROS eco-system:

http://roscon.ros.org/2015/presentations/ROSCon-Automated-Driving.pdf
ROS: What is ROS (Robot Operating System)?

- Architecture for distributed inter-process communication
- Multilanguage interface (C++, Python, Lua, Java, MATLAB)
- Tools for runtime and data analysis
- Packages for common algorithms and drivers
- Open source

With the intent to enable researchers to rapidly develop new robotic systems without having to “reinvent the wheel” through use of standard tools and interfaces.
ROS: Trend in Robotics Development

- **ROS**
  - #1 middleware for robotics applications development
  - Yearly increase in users
  - Simplify component compatibility through standalone interfaces
  - Integrate with simulation environments (e.g. Gazebo)

Popular in research and gaining great momentum in industry
ROS: Gazebo Simulator

- Gazebo is one of the most popular robotics simulators
- Many robot manufacturers provide plugins for Gazebo that help simulate their robots (TurtleBot, Baxter, Husky, …)
- Download a VM with Gazebo
  

Add visualization to simulations for effective algorithm evaluations
ROS nodes communicate through well-defined message interfaces
ROS: ROS Network Overview

Management of data transmissions through the ROS network
ROS: ROS Node Communication Methods

- **Topics**

  - ROS Node(s) \(\rightarrow\) /topic \(\rightarrow\) ROS Node(s)
    - Publish
    - Subscribe

- **Services**

  - ROS Node \(\rightarrow\) ROS Node
    - Request
    - Response
    - Service Server \(\rightarrow\) Service Client

ROS message selection based on data usage and needs
ROS: Challenges Using ROS

- Early Idea
  - MATLAB Code
  - Simulink Model
  - C/C++ Code

- Custom C

- Need to learn ROS and Linux
- Not integrated with MATLAB and Simulink
- Need to learn OOP and C++

- Convert to ROS Node by Hand

- ROS
  - ROS Node 1
  - ROS Node 2
  - ROS Node n
Connect MATLAB/Simulink to ROS for efficient algorithm development
Robotics Development Workflow

1. Explore Robot Interface
2. Develop Algorithm
3. Test and Refine in Simulation
4. Test and Refine on Real Robot

Verify algorithms at each step to refine design and prevent rework
Demo: Walking OP2 machine

Utilizing the power of MATLAB/Simulink and interfacing with ROS
Algorithms Developed in MATLAB/Simulink

Topics:
- Camera
- Joint State

State Controller
Motion Generator
Image Processing

Topics:
- Joint Commands

Low Level Control

Data processing and command calculations done in MATLAB/Simulink
Step 1: Explore Robot Interface

- Connect to simulated / real robot over ROS
- Explore available sensors and actuators
- Retrieve some sensor data
- Control the robot motion
Step 2: Prototype Algorithm

- Develop the algorithm in MATLAB/Simulink using image processing tools

- Run tests to ensure the algorithm behaves as expected
Step 3 : Test Algorithm in Simulator

1. Explore Robot Interface
2. Develop Algorithm
3. Test and Refine in Simulation
4. Test and Refine on Real Robot
Step 4: Test your algorithm with actual robot

- Explore Robot Interface
- Develop Algorithm
- Test and Refine in Simulation
- Test and Refine on Real Robot
Application Examples

EKF SLAM

Visual Odometry

Humanoid
Deploying your Algorithm

- Explore Robot Interface
- Develop Algorithm
- Test and Refine in Simulation
- Test and Refine on Real Robot
- Determine deployment methods based on application

- Generate ROS Node with Simulink
- Generate a shared library with MATLAB Coder™
- Create a Stand Alone Executable with MATLAB Compiler™
MathWorks Solution for Robotics Development
Robotics System Toolbox

- Connect MATLAB/Simulink to ROS
- Utilize useful toolboxes for algorithm development (Image Processing, Machine Learning, CVST, etc.)
- Use simulator to verify algorithms virtually
- Deploy algorithms through code generation

Integrate MATLAB with ROS using Robotics System Toolbox
1. Access ROS capabilities from MATLAB (I/O)
2. Access ROS capabilities from Simulink (I/O and C++ code generation)
3. Application Examples for working with robot hardware/simulator
   - TurtleBot and Gazebo (robot simulator)
4. Algorithms for autonomous wheeled robots
5. Simulink Support for ROS (New in R2016a)
   - Enable Raspberry Pi as target for ROS node generation
6. Support for Robotics Platforms (New in R2016a)
   - Support Package for TurtleBot
What’s New in Robotics System Toolbox?
TurtleBot Robot Support from Robotics System Toolbox
What’s New in Robotics System Toolbox?
Autonomous Ground Vehicle Algorithms

<table>
<thead>
<tr>
<th>Path Planning</th>
<th>• Probabilistic Roadmaps (PRM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kinematics Control</td>
<td>• Pure Pursuit path controller for differential-drive robots</td>
</tr>
<tr>
<td>Mapping</td>
<td>• Map representation using Occupancy Grid</td>
</tr>
<tr>
<td>Obstacle Avoidance</td>
<td>• Vector Field Histogram (VFH) algorithm</td>
</tr>
<tr>
<td>Localization</td>
<td>• Monte Carlo Localization  (New in R2016a)</td>
</tr>
</tbody>
</table>
| Utilities           | • Conversions between different rotation and translation representations  
                        • Particle Filter  (New in R2016a) |
Demo: Monte Carlo Localization

- Estimate pose of a robot using a known map
  - Estimate pose (location and orientation) of a differential drive robot in a known environment using a range sensor

```matlab
>> mcl = robotics.MonteCarloLocalization
>> [~, pose] = step(mcl, odom, ranges, angles)
```
Particle Filter

- Estimate the state of a non-linear system recursively
  - Estimate state for arbitrary non-linear systems and non-Gaussian noise distributions
  - Apply particle filter to diverse applications, such as robot pose estimation, object tracking, and sensor fusion

```matlab
>> pf = robotics.ParticleFilter
>> predict(pf)
>> correct(pf, [0 0 pi])
```
## Related Products for Robotics Applications Development

<table>
<thead>
<tr>
<th>Image Processing Toolbox™</th>
<th>Computer Vision System Toolbox™</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Contrast adjustment</td>
<td>- High-speed video I/O</td>
</tr>
<tr>
<td>- Geometric transformations</td>
<td>- Point Cloud processing</td>
</tr>
<tr>
<td>- Various filters</td>
<td>- Tracking</td>
</tr>
<tr>
<td>- Segmentation</td>
<td>- Stereovision</td>
</tr>
<tr>
<td>- Object analysis</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image Acquisition Toolbox™</th>
<th>Statistics and Machine Learning Toolbox™</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Image capture from standard H/W</td>
<td>- Multivariate statistics</td>
</tr>
<tr>
<td>- Analog, Camera Link, DCAM, GigE Vision, USB camera, etc</td>
<td>- Probability distribution</td>
</tr>
<tr>
<td>- Microsoft Kinect Support</td>
<td>- Machine learning</td>
</tr>
<tr>
<td></td>
<td>- Experimental design</td>
</tr>
<tr>
<td></td>
<td>- Statistical process control</td>
</tr>
</tbody>
</table>
### Related Products for Robotics Applications Development

<table>
<thead>
<tr>
<th>Control System Toolbox™</th>
<th>Simulink Design Optimization™</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Linear analysis</td>
<td>- Model parameter estimation from test data</td>
</tr>
<tr>
<td>- Classical control design</td>
<td>- Optimization of parameters</td>
</tr>
<tr>
<td>- Modern control design</td>
<td>- Response optimization</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Robust Control Toolbox™</th>
<th>Simulink Control Design™</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Robust control design</td>
<td>- Automatic tuning of PID Controller blocks</td>
</tr>
<tr>
<td>- Automatic tuning of gain-scheduled controllers</td>
<td>- Linearization of Simulink models</td>
</tr>
<tr>
<td></td>
<td>- Continuous-Discrete time conversions</td>
</tr>
</tbody>
</table>
Conclusion

- MATLAB enable you to develop algorithms efficiently
  - An advanced and abundant libraries
  - Interactive algorithm exploration by interpreter environment

- Your algorithms on MATLAB can directly connect to ROS network
  - Accelerate your verification process
  - Enable you to validate whole robotics system in early phase

- Robotics performance analysis by powerful MATLAB engine
  - rosbag

MATLAB/Simulink Tools to Increase Efficiency of Robotics Development