Developing Robotics Applications

with MATLAB, Simulink:

Introducing the

Robotics System Toolbox
What Are You Doing with Robotics?

Build Robots
Using MATLAB and Simulink for “Building Robots”

Recorded Webinar: How a Differential Equation Becomes a Robot

Festo Bionic Arm

DLR Humanoid Robot
Using MATLAB and Simulink for “Building Robots”

- High number of DoF
- Nonlinearity
- Multiple Layers of Control
- Multiple States or Modes
What Are You Doing with Robotics?

Teach/Learn Robotics With Low Cost hardware

Build Robots

Arduino
Raspberry Pi
Lego
Robotics with Low Cost Hardware:

Build Robots with Low-Cost Hardware?
- No need C/C++/Python
- Drivers Provided

Hardware Support Package

Visit: www.mathworks.com/hardware
Free LAB workbooks
Robotics with Low Cost Hardware:
What Are You Doing with Robotics?

Teach/Learn Robotics
With
More powerful hardware

Build Robots

Teach/Learn Robotics
With
Low Cost hardware

Arduino
Raspberry Pi
Lego
What Are You Doing with Robotics?

Teach/Learn Robotics
With
More powerful hardware

Use Powerful Robots Running ROS?
- ROS/Gazebo Interface
- ROS Node Generation
- Comprehensive Algorithms

Robotics System Toolbox

Build Robots

Teach/Learn Robotics
With
Low Cost hardware

Arduino
Raspberry Pi
Lego
What do we want to do today?
Agenda

1. MATLAB-ROS Interface

2. Robotics Algorithms

3. Simulink-ROS Interface
What Can You Do with Robotics System Toolbox?
Data Exchange Paradigms

- Topics

ROS Node(s) \(\xrightarrow{	ext{Publish}}\) /topic \(\xrightarrow{	ext{Subscribe}}\) ROS Node(s)

- Services

ROS Node Service \(\xrightarrow{	ext{Request}}\) ROS Node(s) \(\xrightarrow{	ext{Response}}\) Clients
Developing Robotic Applications with ROS

- **MATLAB**
- **Main CPU**
  - Simulink
- **Ethernet**
- **Robot (CPU 2)**
  - Global Planner
  - Map server
  - Image processing
  - Local Planner
  - Kinematics & Control
  - Localization & Mapping
- **Sensors**
- **Actuators**

**Nodes:**

- NODE
- NODE
- NODE
- NODE
- NODE
- NODE
The essentials: ROS communication

- `>> rosshutdown( )`  
  Shutdown or start global MATLAB node

- `>> rosinit( )`  
  Connect to existing ROS master

- `>> rosnodename( )`  
- `>> rostopic( )`  
- `>> rosmesage( )`  
- `>> rossubscriber( )`  
  - `>> receive( )`

- `>> rospublisher( )`  
  - `>> rosmesage( )`
  - `>> send( )`

- `>> robotics.ros.Node ( )`  
  Connect to multiple ROS masters
1. The MATLAB ROS interface – part 1
   – Practice on a MATLAB Robotic simulator

2. The MATLAB ROS interface – part 2
   – Practice on a GAZEBO Robotic simulator (Turtlebot)
   – A “teaser” on algorithm development

3. Processing ROS bag files
Agenda

1. MATLAB-ROS Interface

2. Robotics Algorithms ✔

3. Simulink-ROS Interface

Demo
1. Path planning and Pure pursuit controller – part 1
   – Practice on a MATLAB Robotic simulator

2. Path planning and Pure pursuit controller – part 2
   – Practice on a GAZEBO Robotic simulator
Overview: Robotics Algorithms

Test it with Gazebo or other Simulator through MATLAB-ROS Interface
Test it with a physical ROS-Enabled robot

\[ \text{Turtlebot\_IP} = \text{'XXX.XXX.XXX.XXX'} \]

rosinit(Turtlebot\_IP, 'NodeHost', my\_laptop\_IP, ...'
'nodeName', my\_name\_for\_global\_MATLAB\_node);
Agenda

1. MATLAB-ROS Interface

2. Robotics Algorithms

3. Simulink-ROS Interface
Generate a ROS Node from a Simulink Model

- Connect Simulink Model to a ROS network to simulate a Goal tracking algorithm
Simulink Algorithm development and DEPLOYMENT: Demo

Prototype algorithm in Simulink

Test algorithm with a ROS-enabled Simulator or Robot

Generate a standalone ROS node from Simulink
“How do I create custom messages?”
"How do I access the tf Transformation tree in ROS?"
Key Capabilities Demonstrated

- **MATLAB-ROS Interface**
  - Create a ROS nodes, publishers, subscribers, services
  - Design and test robotics algorithms on a robot simulator such as Gazebo
  - Test robotics algorithms on a physical robot
  - Import rosbag log files

- **Simulink-ROS Interface**
  - Publish and Subscribe from Simulink Models
  - ROS node generation from Simulink models

- **Algorithms**
  - Map representation with Binary Occupancy Grids
  - Path Planning with Probabilistic Roadmap
  - Path Tracking with Pure Pursuit Controller
  - Coordinate System Transformations
Thank You