From Simulink Models to ROS 2 Control  
- Streamlining Robotic Controller Development

Josh Chen  
Senior Software Engineer,  
ROS Toolbox  
MathWorks

Gijs van der Hoorn  
Researcher,  
Robot Dynamics Group  
TU Delft

YJ Lim  
Principal Product Manager of  
Robotics  
MathWorks
Our Products

MATLAB®, the language of engineers and scientists, is a programming environment for algorithm development, data analysis, visualization, and numeric computation.

Simulink® is a graphical programming environment for modeling, simulation, and analyzing dynamical systems. Control development tool.

More than 100 add-on toolboxes for specialized tasks.
MathWorks Supports Robotics and Autonomous Systems
Design, Simulate, Test, and Deploy

Manipulators / Cobots

Mobile Robots and Ground Vehicles

Marine Robots

ROS Toolbox
Design, simulate, and deploy ROS-based applications

Get a free trial
View pricing

Have questions? Contact Sales.
Agenda

- Introduction
  - Program ROS and ROS 2 using MATLAB and Simulink

- From Simulink to ros2_control generation
  - Automated ros2_control plugin

- Summary
ROS programming can be done in either Python or C++

and you can also do in MATLAB/Simulink!!
How to do ROS programming using MATLAB?

- Create ROS nodes, publishers, subscribers, services, and actions directly via MATLAB APIs.
- Enhance ROS programming with MATLAB’s Toolboxes, e.g., Navigation Toolbox and Computer Vision Toolbox.
- No CMAKE or C++ knowledge required.
- Leverage MATLAB/Embedded Coders for automatic C++ and CUDA ROS code generation.
How to do ROS programming using Simulink?

- Utilize Simulink blocks for publishers, subscribers, services, and actions.
- Modeling and simulation for Model-Based Design with Simulink
- No need for CMAKE or C++ expertise.
- Automatically generate C++ and CUDA ROS code with Simulink/Embedded Coders.
We support **major ROS functionalities in both MATLAB and Simulink for ROS and ROS2**

<table>
<thead>
<tr>
<th>ROS</th>
<th>ROS2</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Topic – Publish / Subscribe</td>
<td>• Topic – Publish / Subscribe</td>
</tr>
<tr>
<td>• Service server, Service client</td>
<td>• Service server, Service client</td>
</tr>
<tr>
<td>• Action – Client / Server</td>
<td>• Action – Client / Server</td>
</tr>
<tr>
<td>• Parameter Get / Set</td>
<td>• Parameter Get / Set</td>
</tr>
<tr>
<td>• ROS TF</td>
<td>• ROS2 TF</td>
</tr>
<tr>
<td>• Custom messages</td>
<td>• Custom messages</td>
</tr>
<tr>
<td>• rosbag reader, rosbag writer</td>
<td>• rosbag reader, rosbag writer</td>
</tr>
<tr>
<td>• Code generation</td>
<td>• Code generation</td>
</tr>
<tr>
<td>• CUDA ROS code generation</td>
<td>• CUDA ROS2 code generation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ROS Distro</th>
<th>ROS2 Humble</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROS Noetic</td>
<td>Switchable DDS</td>
</tr>
</tbody>
</table>
MATLAB and Simulink Simplify ROS and ROS 2 Programming

ROS Connection
Prototyping
ROS node generation
Standalone C++ node generation

ROS Data Analyzer
rosbag import/playback
Offline analysis
Visualize and analyze ros(2)bag and live ROS data

- Visualize both ROS 1 and ROS 2 bag files
- Visualize the live ROS data
- Use tags and bookmarks to ros(2)bag
- Read and visualize ros(2)bags stored in AWS S3

Please come over to the MathWorks table during the Lab Tours/Demo Session in the afternoon!!!
MATLAB connects with UR Cobots Via ROS and ROS 2

Perception → Pose Estimation → Publish Pose
Planning & decision → Motion Planning / Control → Subscribe Pose → Publish Trajectory
Control → Windows/Mac/Linux

Publish Raw image
Subscribe Trajectory

ROS 2

UR Cobots
Please come over to the MathWorks table during the Lab Tours/Demo Session in the afternoon!!!
Agenda

Introduction
  - Program ROS and ROS 2 using MATLAB and Simulink

From Simulink to ros2_control generation
  - Automated ros2_control plugin

Summary
Generate `ros2_control` plugin from Simulink

**Flowchart:**

1. Perception -> MATLAB
2. Planning & decision -> Simulink
3. Control

**Tools:**
- MATLAB
- ROS/ROS2 nodes
- Simulink
- MATLAB Simulator
- Real hardware

**Control Flow:**
- ros_control / ros2_control controller generation

**Software Environments:**
- Windows/Mac/Linux
- ROS/ROS 2
- Network

**Diagram Highlights:**
- Controller Manager
- Hardware Interface
- Real Robot
- Embedded Controller

**Text:**
- Generate `ros2_control` plugin from Simulink
Recap - *ros2_control* architecture

- Layered architecture
- Single process (multi-threaded)
- Determinism *within* node (execution)
- OEM provides up to the *interfaces* layer
- Hardware Interface transforms data:
  - From HW to ROS (ex: enc ticks → rad)
  - From ROS to HW (ex: rad → enc ticks)
- Combine Hardware Interfaces (OEM1, OEM2, ...)
- Controllers are user-facing
- Controllers inheriting from Lifecycle nodes
Case Study: Model-Based Design for Painting Robots

Perceive

Plan / Control

Plant

ROS Industrial Consortium Americas
Case Study: Model-Based Design for Painting Robots
Case Study: Model-Based Design for Painting Robots
Case Study: Model-Based Design for Painting Robots
Mapping between Simulink and Lifecycle Node
Custom implementation can be added to IRT blocks, which will generate code and get triggered at distinguished transition states.
Case Study: Model-Based Design for Painting Robots

- Simulink Model
  - Command Interfaces
  - State Interfaces
- Hardware interface (ie: driver)
  - Data Conversion
  - Safety (E-stops, ...)
  - Joint Limits
  - Sequencing
- Hardware / Simulation
  - System
  - Actuators
  - Sensors
Case Study: Model-Based Design for Painting Robots
Case Study: Model-Based Design for Painting Robots
Case Study: Model-Based Design for Painting Robots
Agenda

Introduction
  – Program ROS and ROS 2 using MATLAB and Simulink

From Simulink to ros2_control generation
  – Automated ros2_control plugin

Summary
Key Takeaways

MATLAB and Simulink simplify ROS and ROS 2 programming

- Leverage Simulink’s Model-Based Design with ros_control and ros2_control frameworks for robust controller development
- Go directly from algorithm prototyping to implementation
- Easily incorporate Simulink controllers into ros2_control framework

• Call-To-Action:
  - Try out the reference examples from ROS Toolbox
  - Reach out to us to work on real-world industrial applications
Thank you!

YJ Lim  
yjlim@mathworks.com

Josh Chen  
joshchen@mathworks.com