Automated ROS code and ros_control generation

YJ Lim
Technical Product Manager of robotics

Abhijeet Gadkari
Embedded Software Engineer

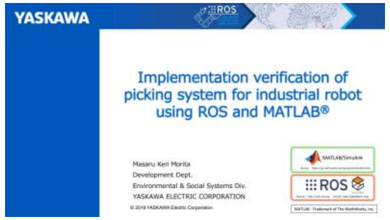
Murat Belge
Consulting engineer

What MathWorks customers are telling us about **ROS**…

voyage

“Simulink + ROS allowed us to deploy a Level 3 autonomous vehicle in less than three months.”

— Alan Mond, Voyage



“By combining various toolboxes of MATLAB, it is easy to develop advanced technology-based applications that are difficult to build with ROS alone”

— Masaru Ken Morita, Yaskawa Electric Corp.

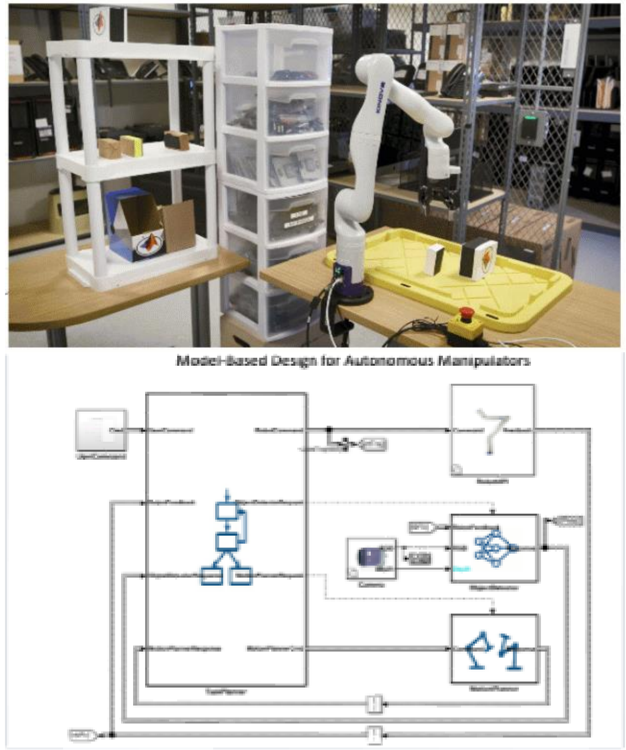
“.. ROS looks popular for industrial automation. How can I get started.”

— Lead Mechanical Engineering Manager, Industry Equipment

Agenda

- Introduction
- Automated ROS code generation
- Automated ros_control generation
- Summary

Introduction

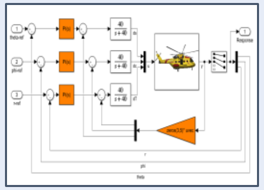


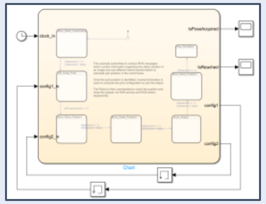
Code Generator

```
#include "AutomatedParkingValetAlgorithm.h"
#include "AutomatedParkingValetAlgorithm_private.h"

int32_T div_s32_floor(int32_T numerator, int32_T denominator)
{
  int32_T quotient;
  uint32_T absNumerator;
  uint32_T absDenominator;
  uint32_T tempAbsQuotient;
  boolean_T quotientNeedsNegation;
  if (denominator == 0) {
    quotient = numerator >= 0 ? MAX_int32_T : MIN_int32_T;

    // Divide by zero handler
  } else {
    absNumerator = numerator < 0 ? ~static_cast<uint32_T>(numerator) + 1U :
      static_cast<uint32_T>(numerator);
    absDenominator = denominator < 0 ? ~static_cast<uint32_T>(denominator) + 1U :
      static_cast<uint32_T>(denominator);
    quotientNeedsNegation = ((numerator < 0) != (denominator < 0));
    tempAbsQuotient = absNumerator / absDenominator;
    if (quotientNeedsNegation) {
      absNumerator %= absDenominator;
      if (absNumerator > 0U) {
        tempAbsQuotient++;
      }
    }

    quotient = quotientNeedsNegation ? -static_cast<int32_T>(tempAbsQuotient) :
      static_cast<int32_T>(tempAbsQuotient);
  }

  return quotient;
}

void AutomatedParkingValetModelClass::APV_emxInit_real_T(emxArray_real_T_T
  **pEmxArray, int32_T numDimensions)
```

Roles and Goals of Code Generation

User design in high level / domain specific languages





→ **Code Generator** →

General purpose programming languages

- **C code**
- **C++ code**
- **HDL code**
- **PLC code**
- **GPU code**
- ⋮

Roles

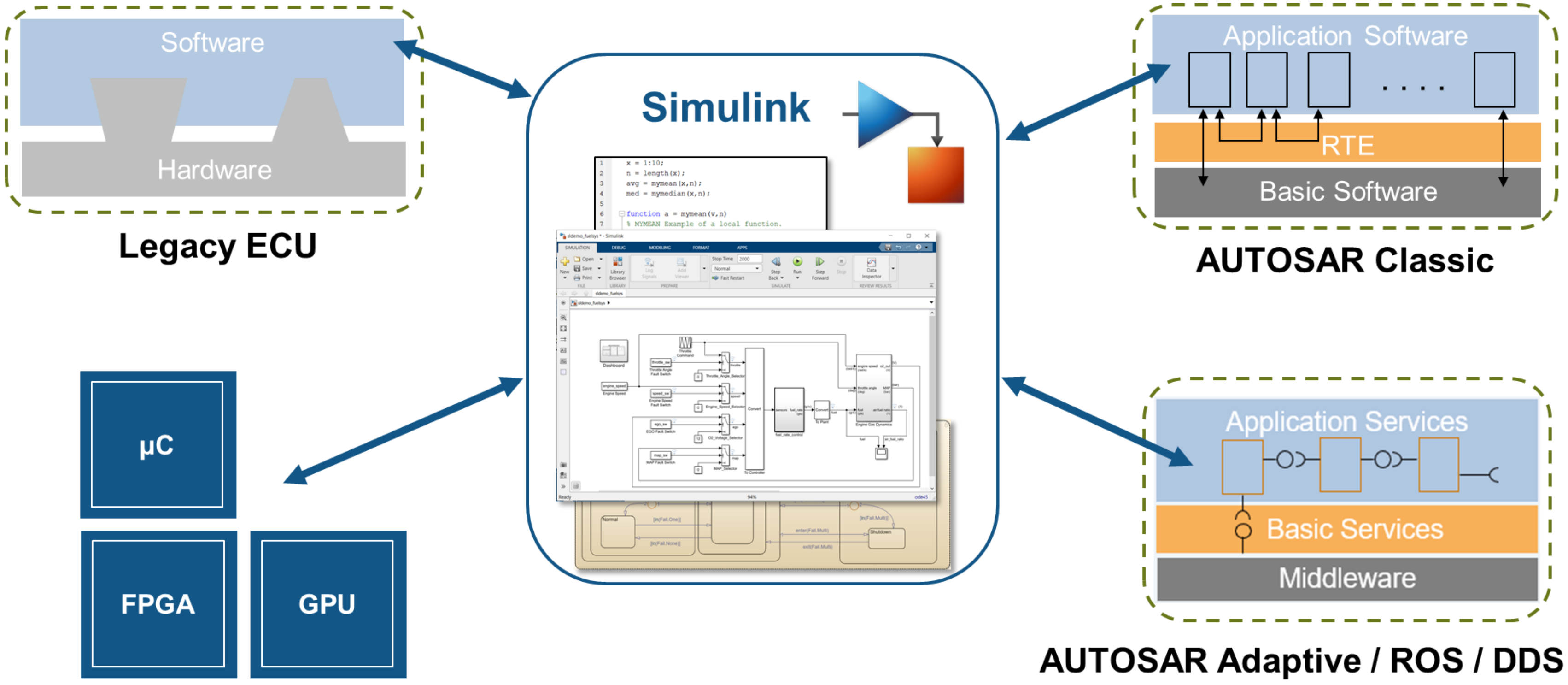
- Acceleration
- <u>Production</u>
- Verification

Goals

- Correct
- <u>Efficient</u>
- Customizable
- Certifiable
- Readable
- Scalable
- …

Design software once, deploy to many targets

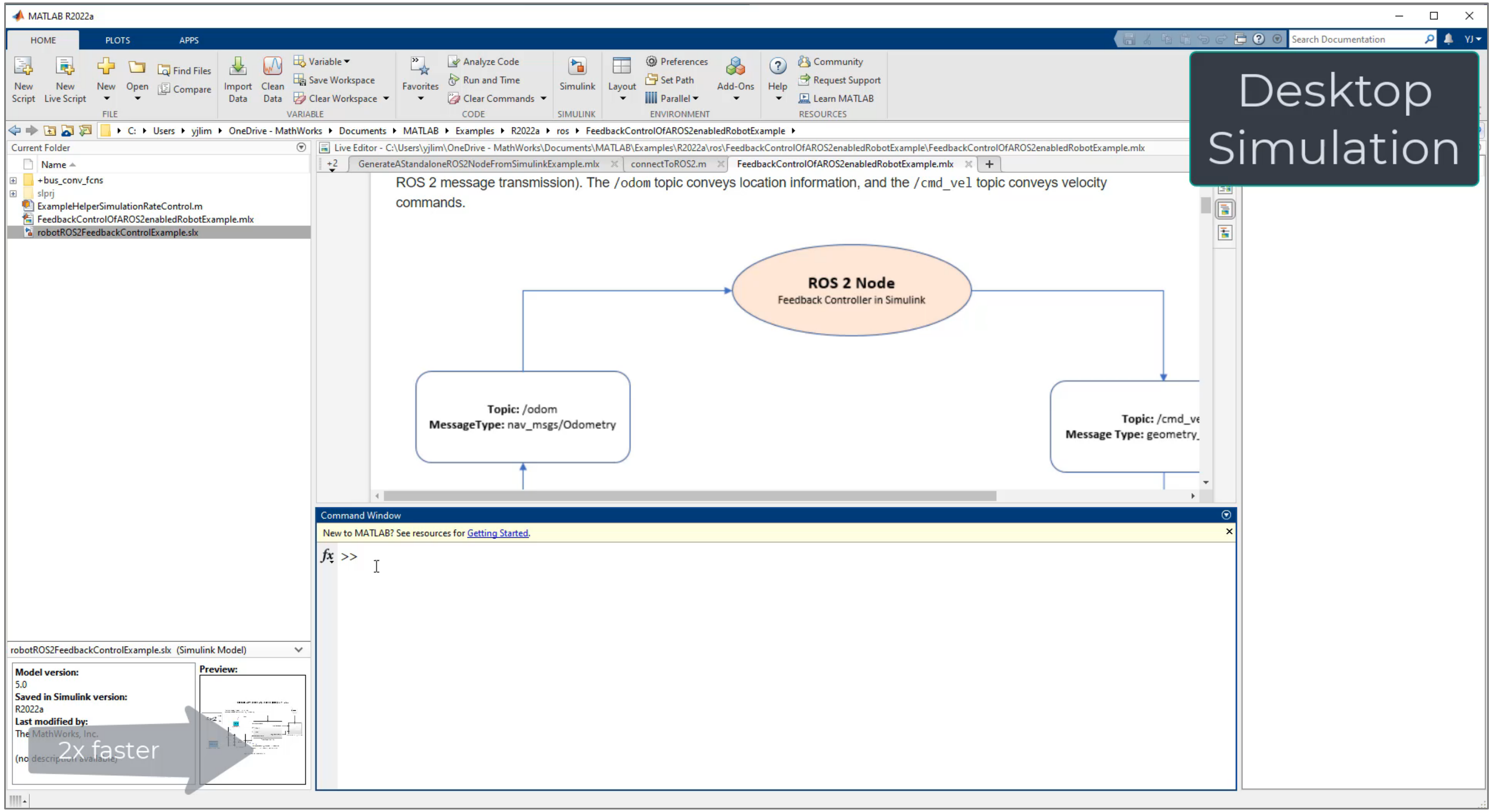
Migrate your existing tested components to new architectures while reusing existing workflows



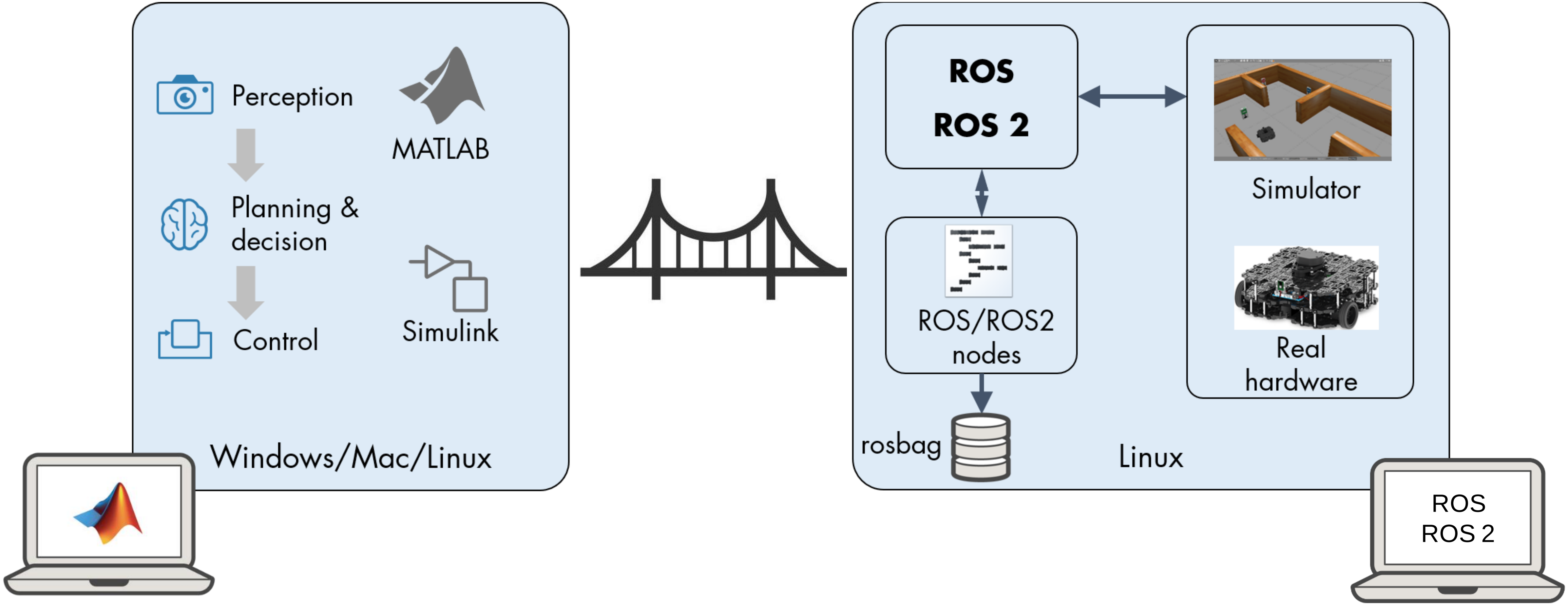
Agenda

- Introduction
- Automated ROS code generation
- Automated ros_control generation
- Summary

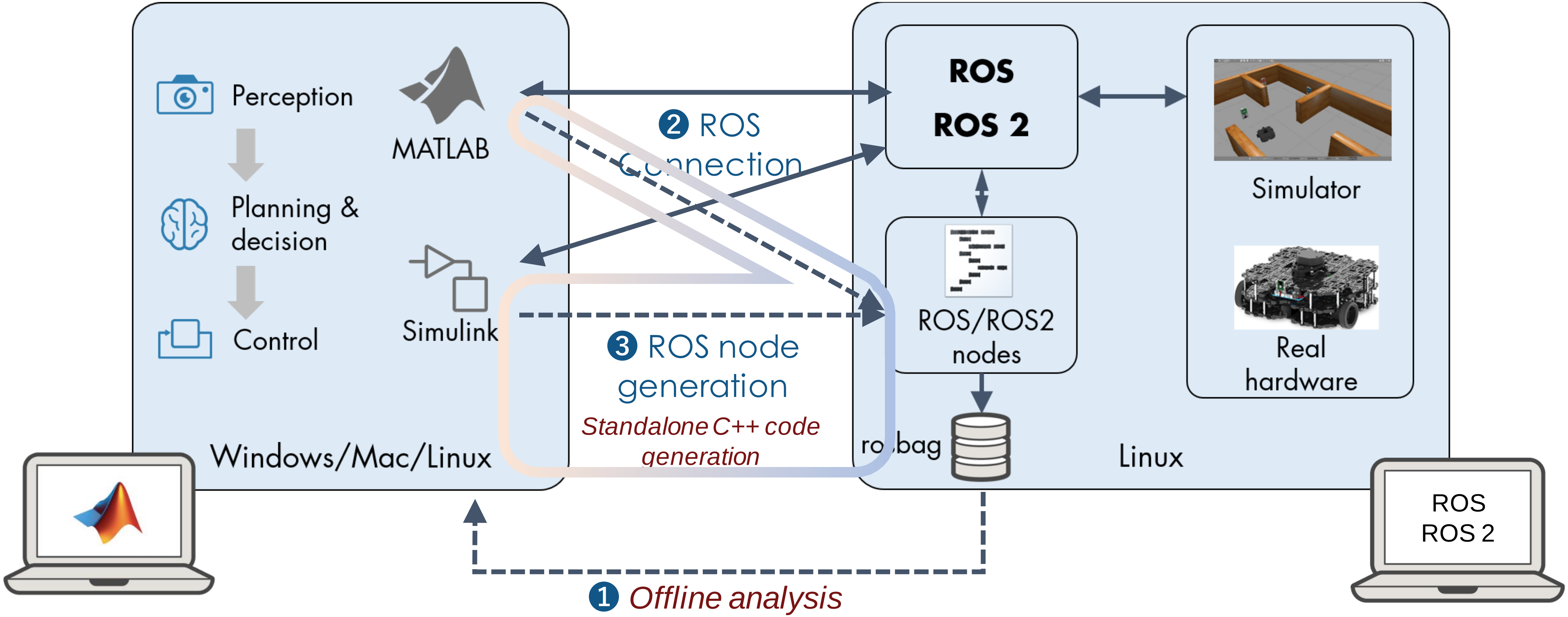
Automated ROS Code Generation



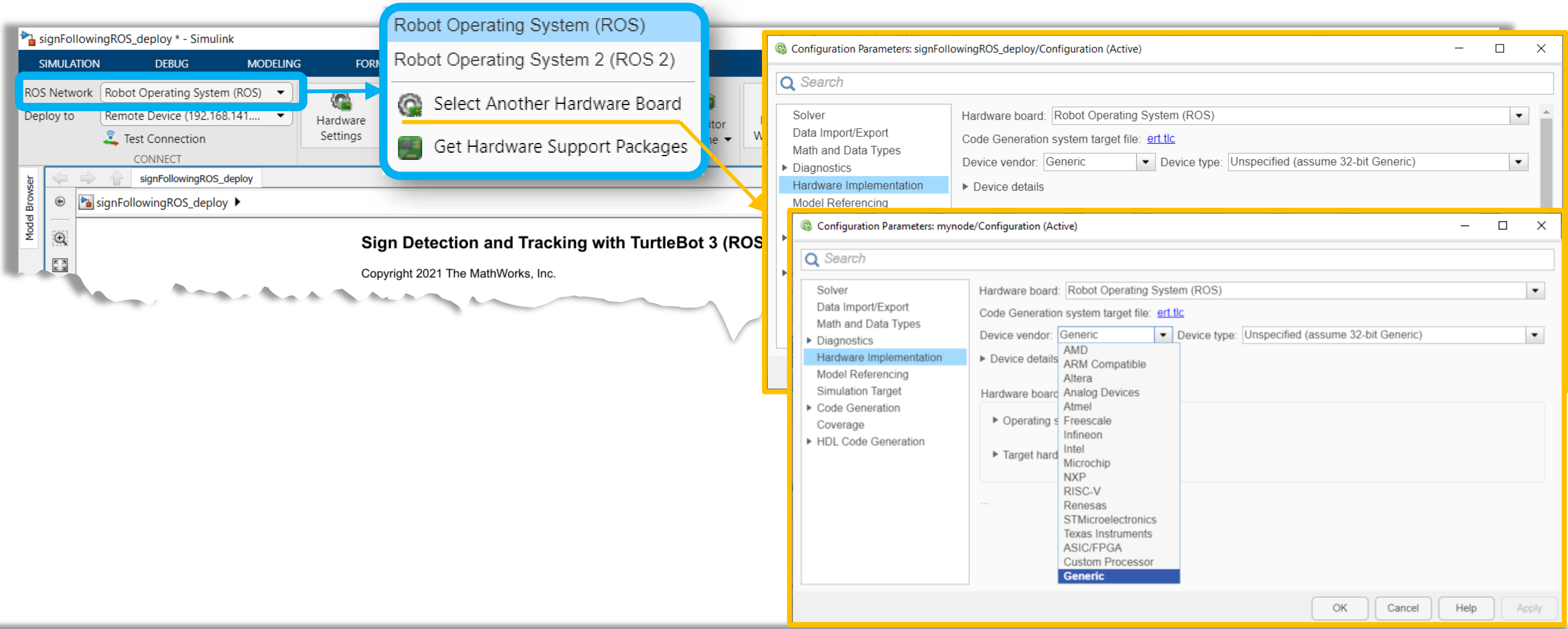
Bridging ROS / ROS 2 with MATLAB and Simulink



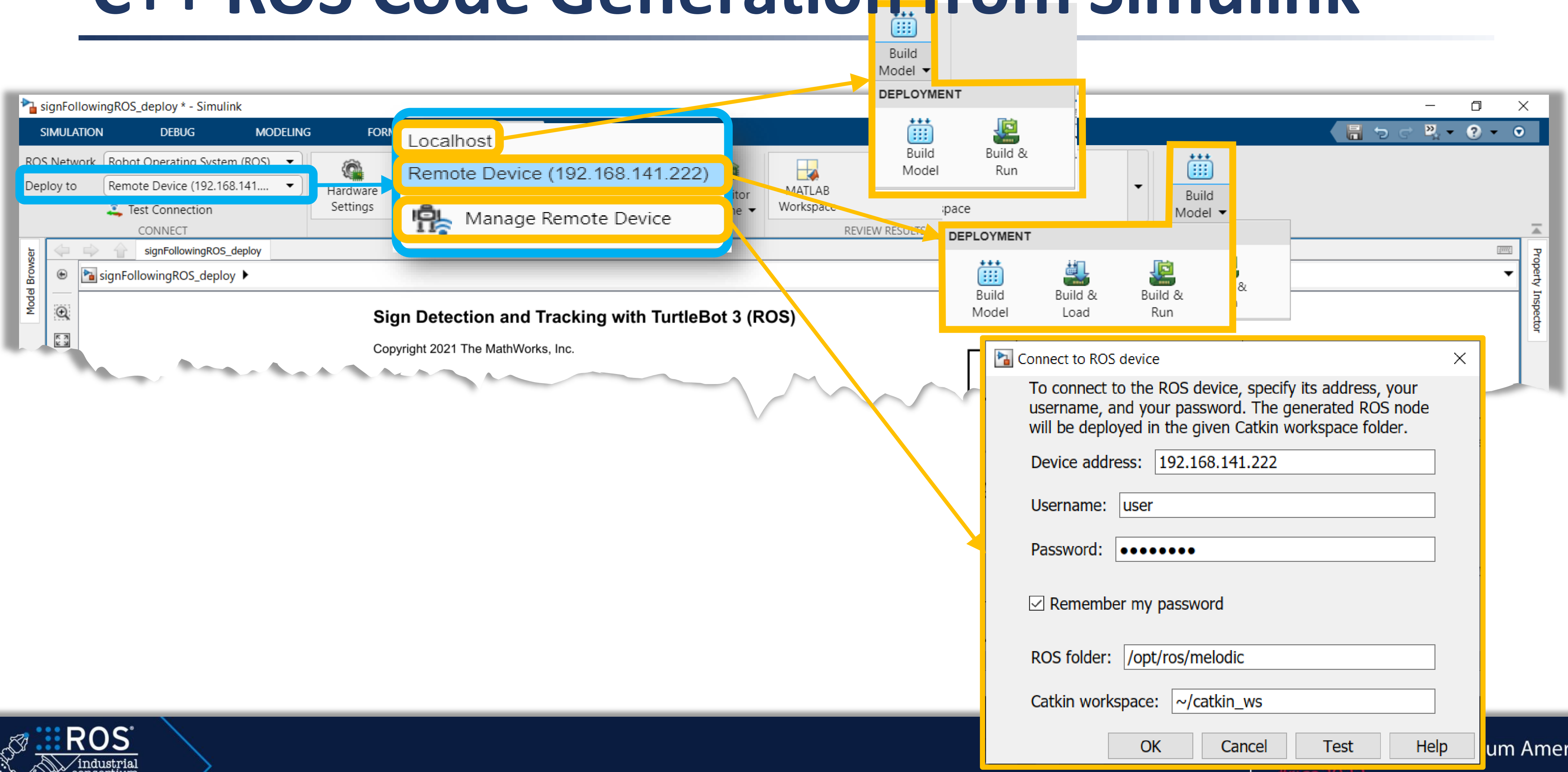
Bridging ROS / ROS 2 with MATLAB and Simulink



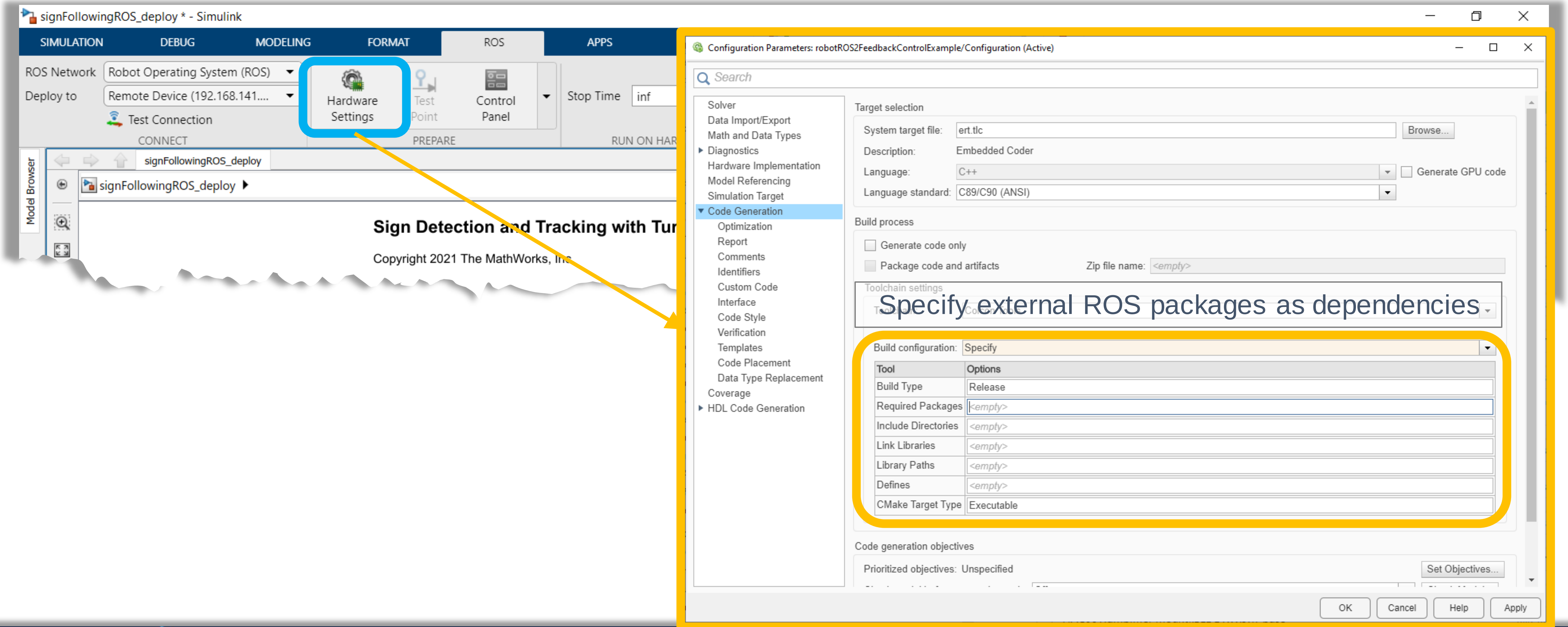
C++ ROS Code Generation from Simulink



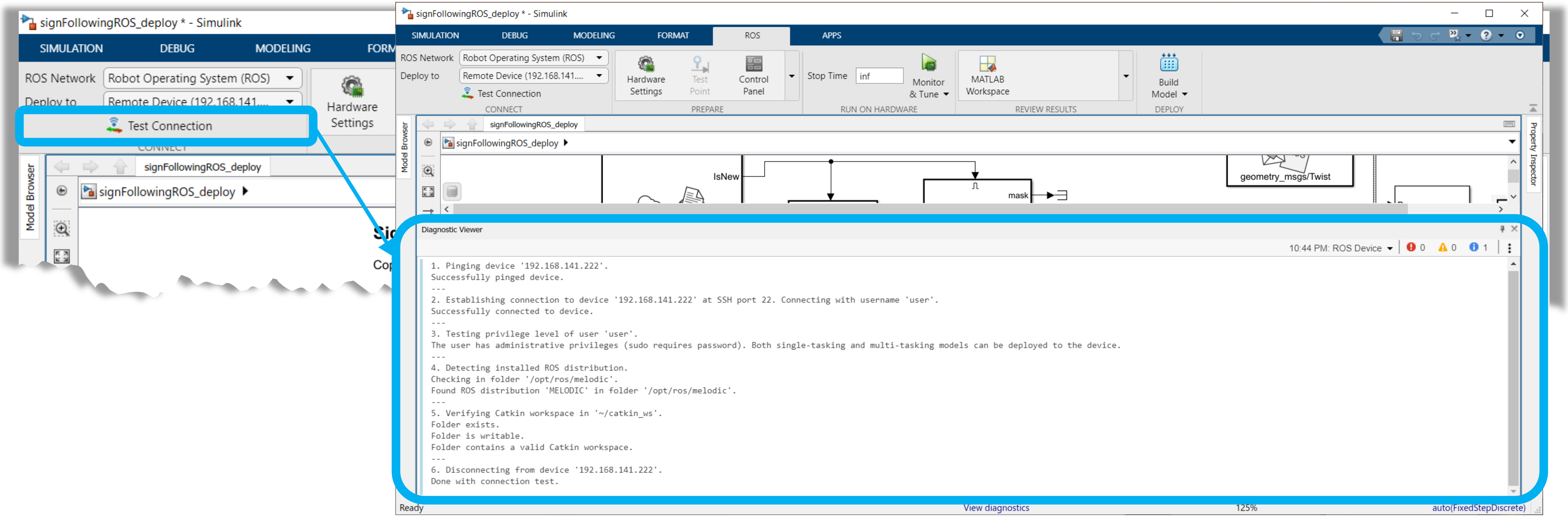
C++ ROS Code Generation from Simulink



C++ ROS Code Generation from Simulink



C++ ROS Code Generation from Simulink



C++ ROS Code Generation from MATLAB

Deploy MATLAB function as a C++ ROS code using MATLAB Coder

```
function myNode
% Copyright 2021 MathWorks Inc.

sub = rossubscriber('/point','geometry_msgs/Point',...
    @callback,...
    'DataFormat','struct');

fprintf('Created %s subscriber\n', sub.TopicName);
while (1)
    fprintf("Node is alive..\n");
    pause(3);
end

end % myNode

% Subsriber callback function
function callback(~,msg)
fprintf("(X,Y,Z): (%f,%f,%f)\n",msg.X, msg.Y,msg.Z);
end
```

```
cfg = coder.config('exe');
cfg.Hardware = coder.hardware(...
    'Robot Operating System (ROS)');
```

```
codegen('myNode.m','-config',cfg)
```

myNode

C++ ROS Node

C++ ROS Code Generation from MATLAB

MATLAB Code

```
function myNode
% Copyright 2021 MathWorks Inc.

sub = rossubscriber('/point','geometry_msgs/Point',...
    @callback,...
    'DataFormat','struct');

fprintf('Created %s subscriber\n', sub.TopicName);
while (1)
    fprintf("Node is alive..\n");
    pause(3);
end

end % myNode

% Subsriber callback function
function callback(~,msg)
fprintf("(X,Y,Z): (%f,%f,%f)\n",msg.X, msg.Y,msg.Z);
end
```

codegen →

C++ Code

```
void myNode()
{
  coder::ros::Subscriber sub;
  coderTimespec b_timespec;
  char varargin_1[7];
  if (!isInitialized_myNode) {
    myNode_initialize();
  }
  sub.matlabCodegenIsDeleted = true;
  sub.init();
  for (int i{0}; i < 6; i++) {
    varargin_1[i] = sub.TopicName[i];
  }
  varargin_1[6] = '\x00';
  printf("Created %s subscriber\n", &varargin_1[0]);
  fflush(stdout);
  while (1) {
    printf("Node is alive..\n");
    fflush(stdout);
    if (pauseState == 0) {
      b_timespec.tv_sec = 3.0;
      b_timespec.tv_nsec = 0.0;
      coderTimeSleep(&b_timespec);
    }
  }
}
```

③ Automated ROS Node Generation to

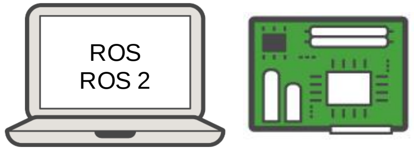
Local Host

MATLAB/Simulink

Build Model

Build & Run

Remote ROS Device



Build Model

Build & Load

Build & Run

Use Case: Deploy ROS Code of Deep Learning Algorithm onto Development Machine

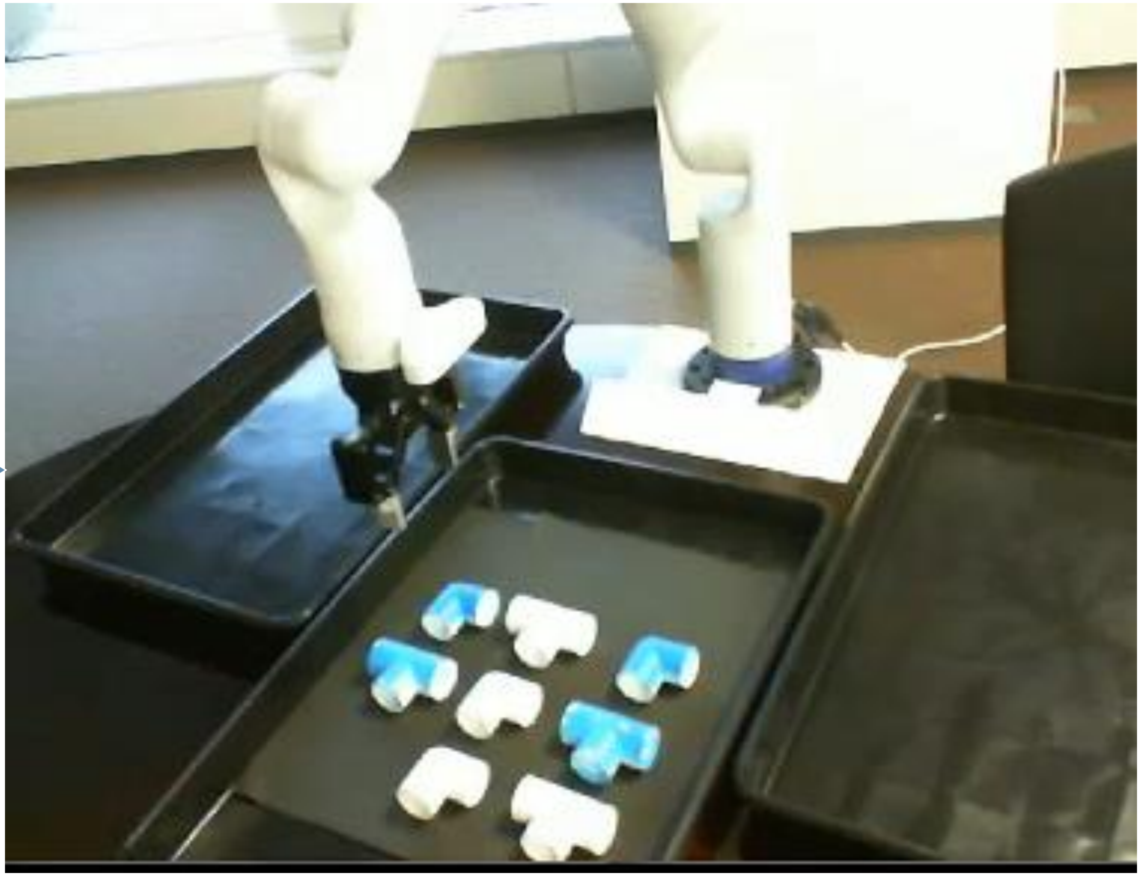
Build & Run

MATLAB/Simulink

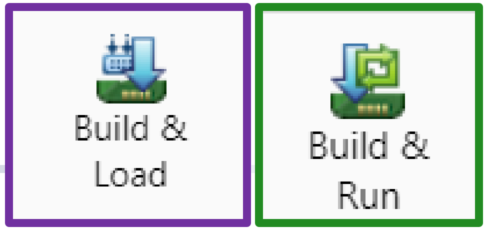
Standalone
ROS Node

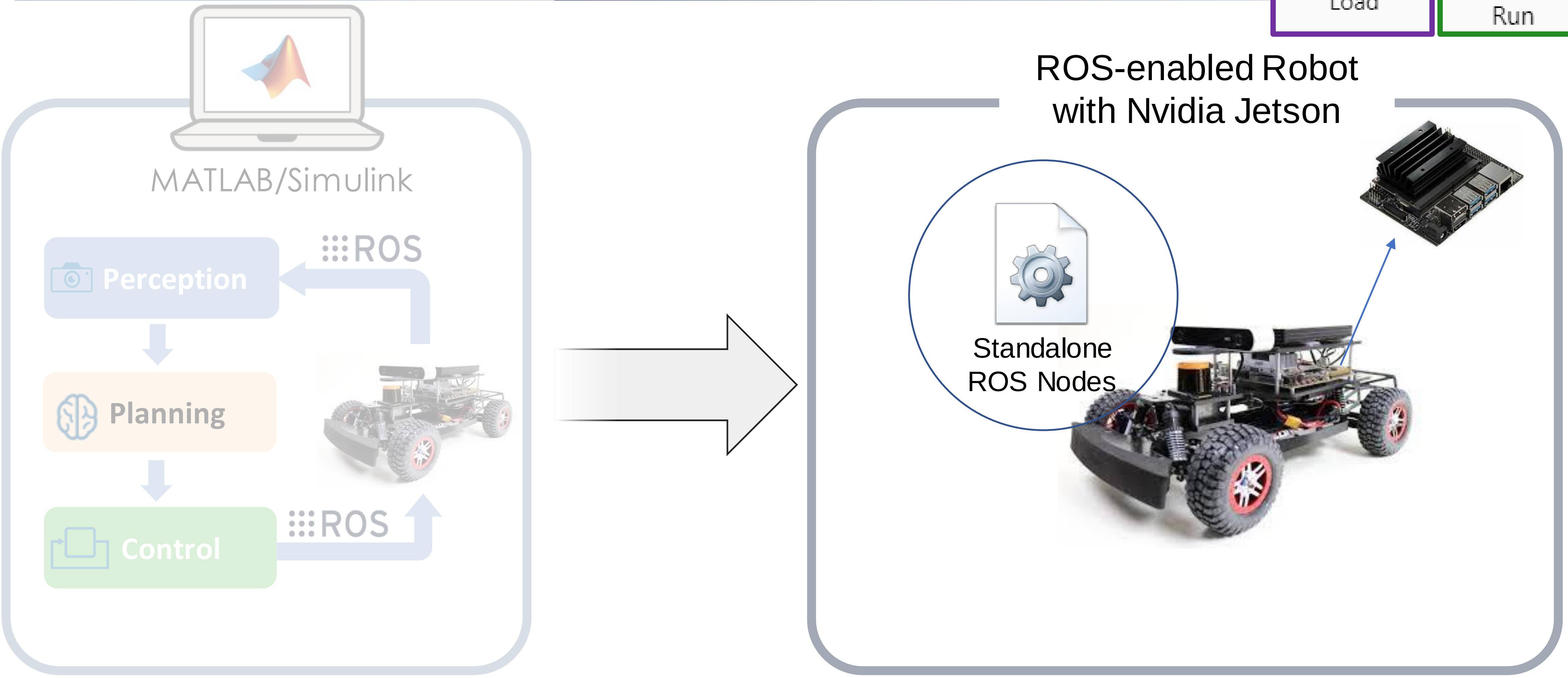
- **Host PC with high-computation power**
- **Deep learning engine**

ROS network

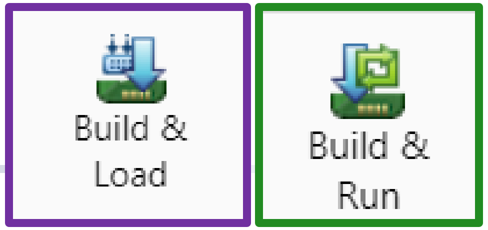


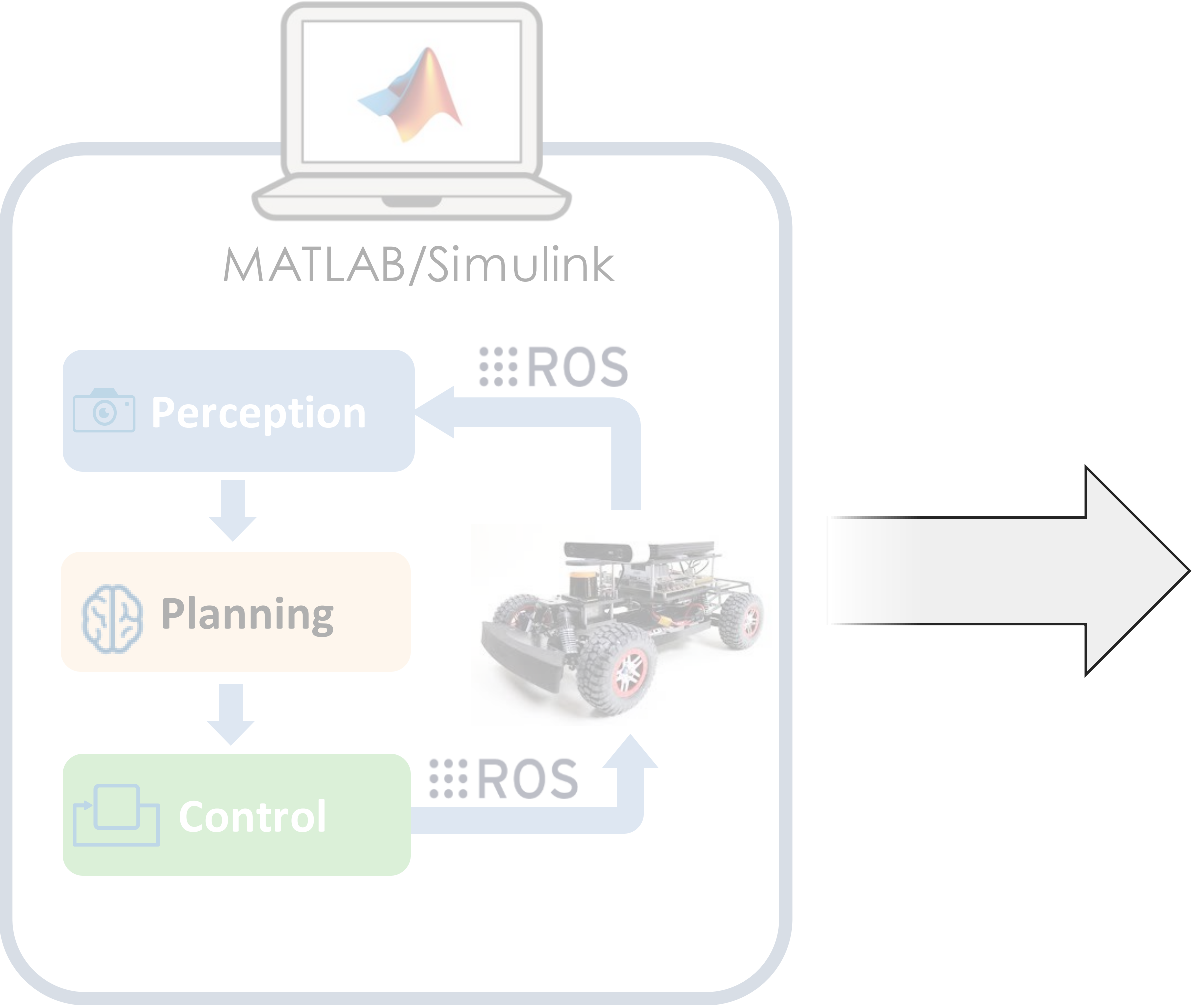
Use Case: Deploy a Standalone ROS Node to Remote Ground Robot



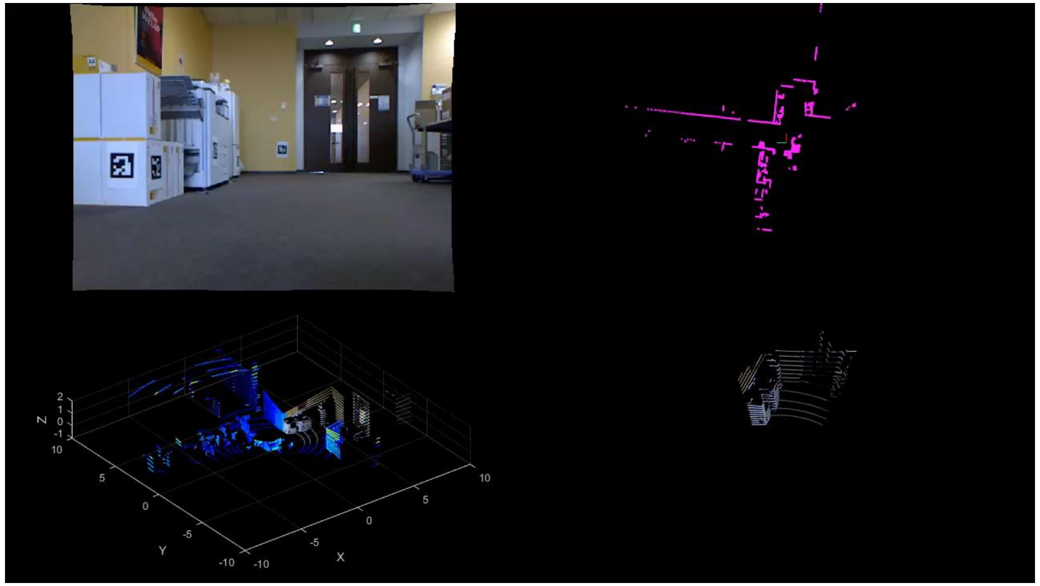


Use Case: Deploy a Standalone ROS Node to Remote Ground Robot





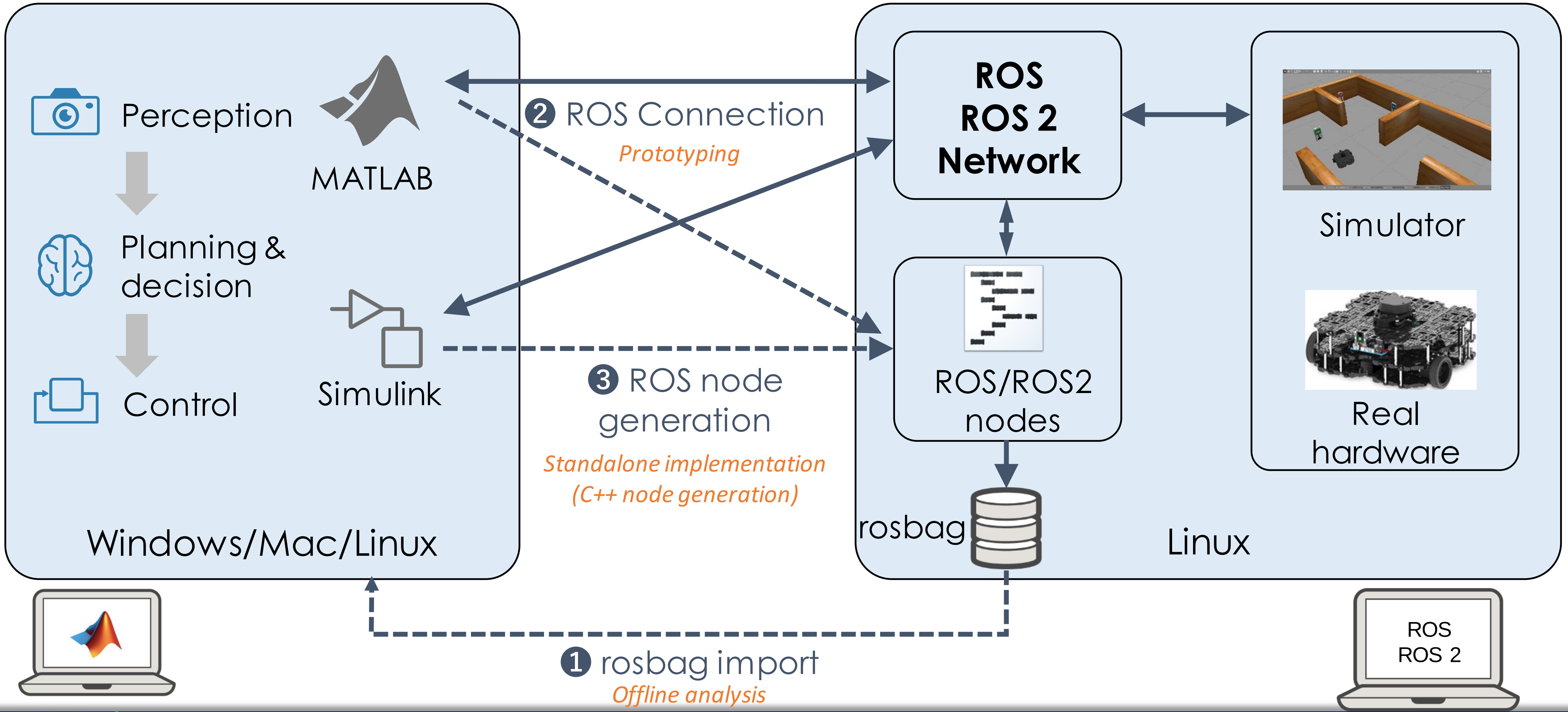
ROS-enabled Robot with Nvidia Jetson



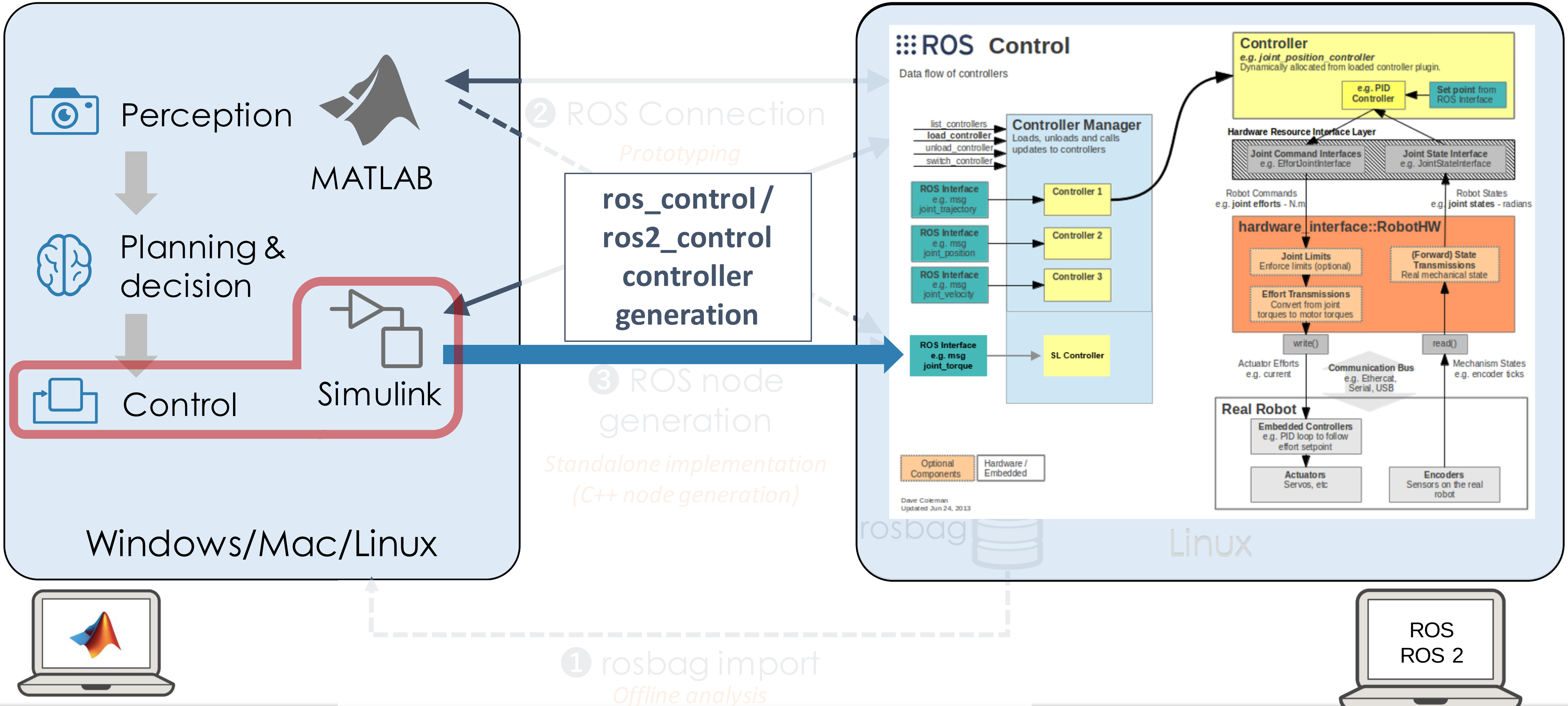
Agenda

- Introduction
- Automated ROS code generation
- Automated ros_control generation
- Summary

Connect MATLAB and Simulink with ROS

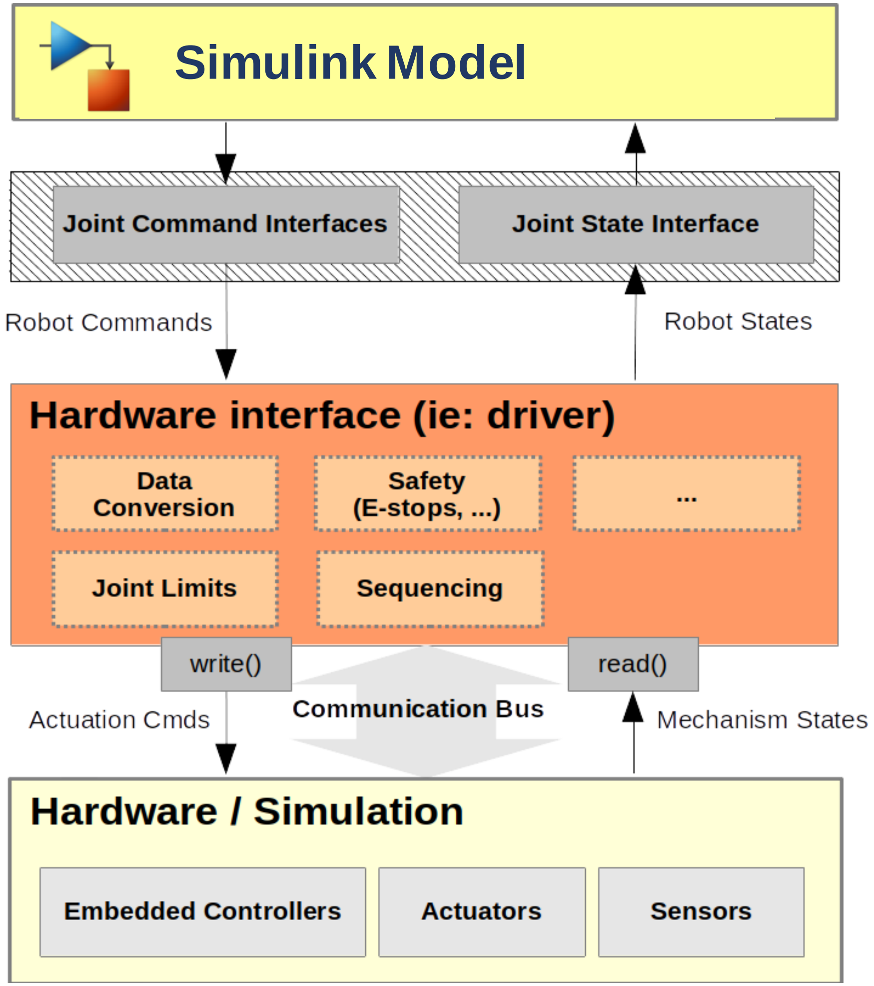


Connect MATLAB and Simulink with ROS

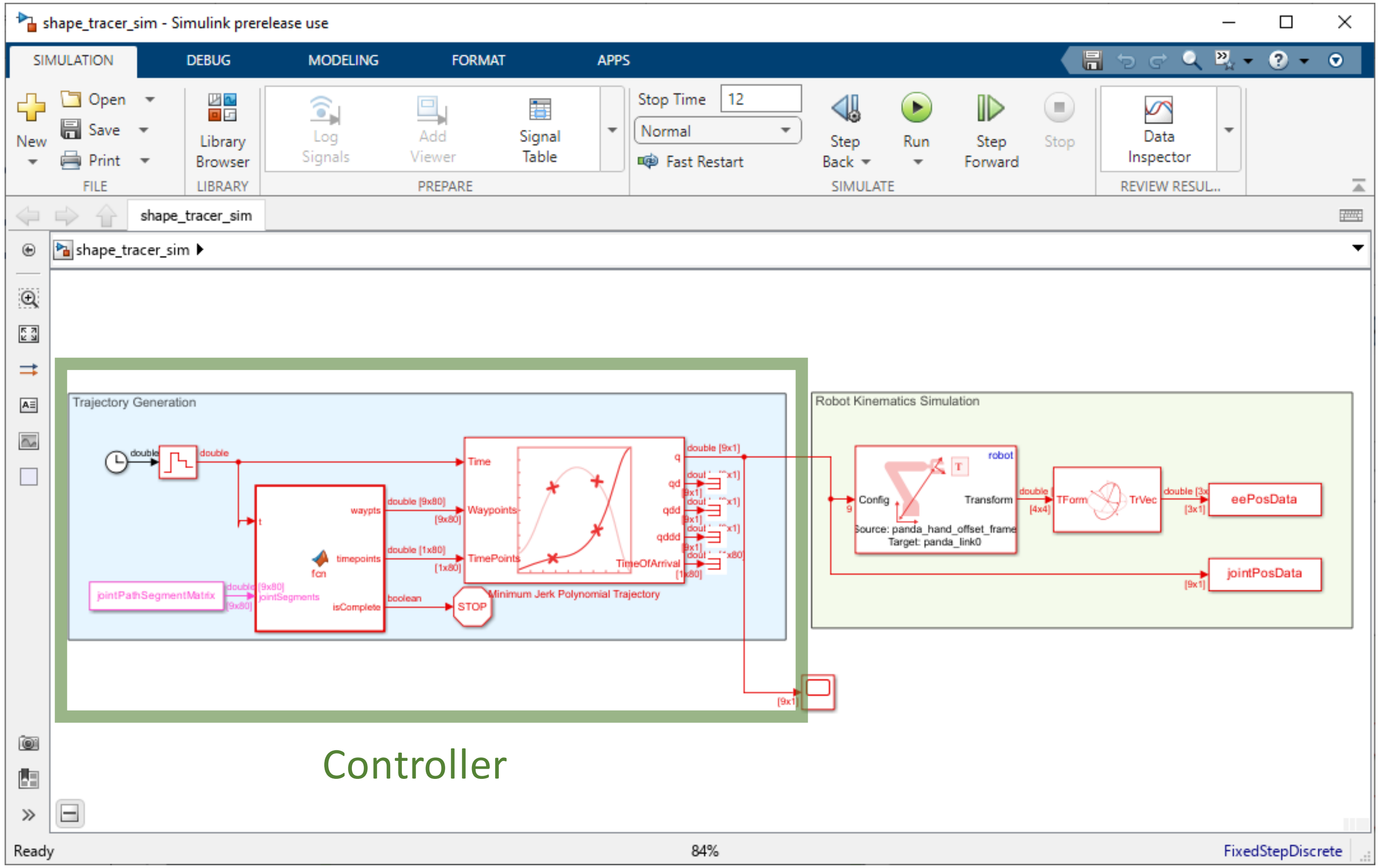


Recap (ros_control architecture)

- Layered architecture
- Single process (multi-threaded)
- Determinism *within* node (execution)
- OEM provides up to the *interfaces layer*
- RobotHW transforms data:
  - From HW to ROS (ex: enc ticks → rad)
  - From ROS to HW (ex: rad → enc ticks)
- Combine RobotHW (OEM1, OEM2, ...)
- Controllers are user-facing

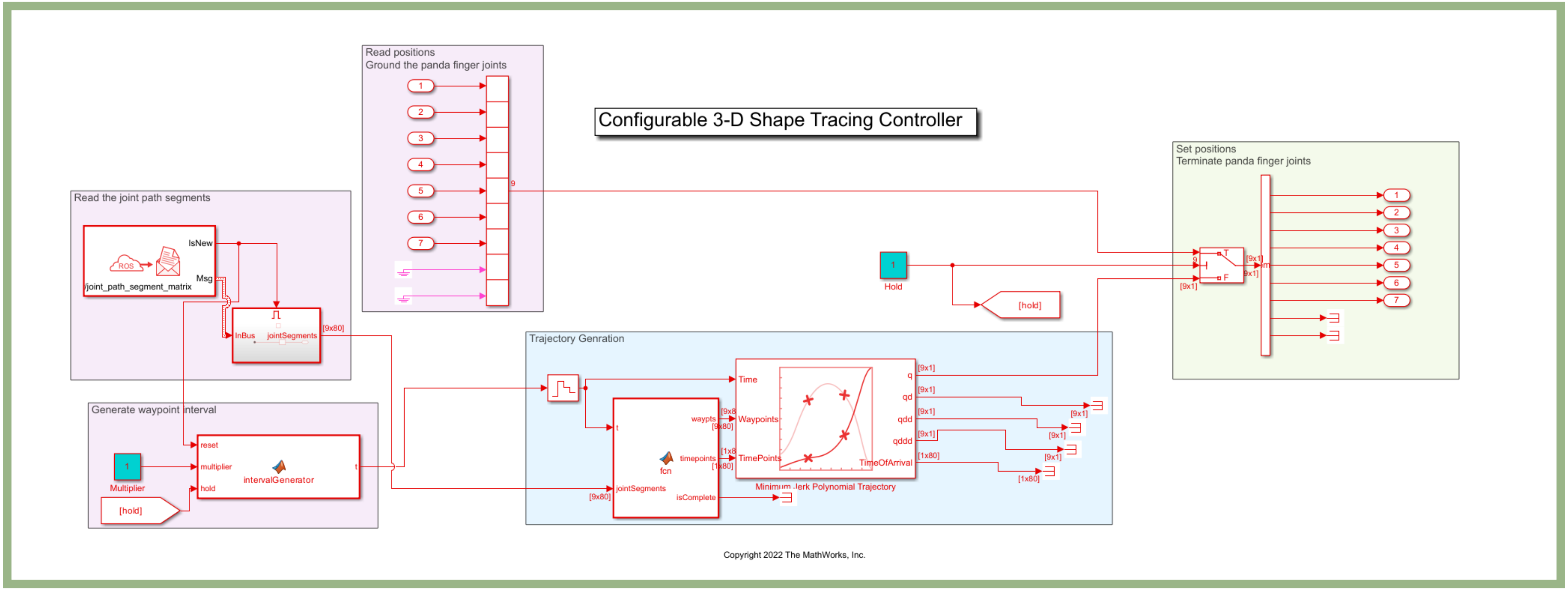


Example – 3D Shape Tracing

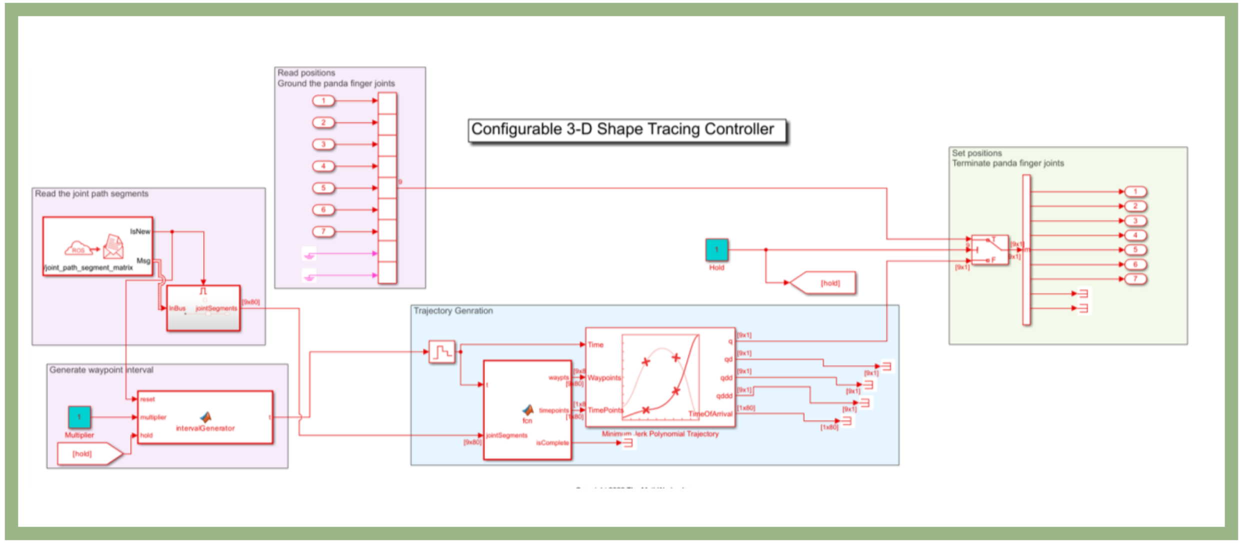


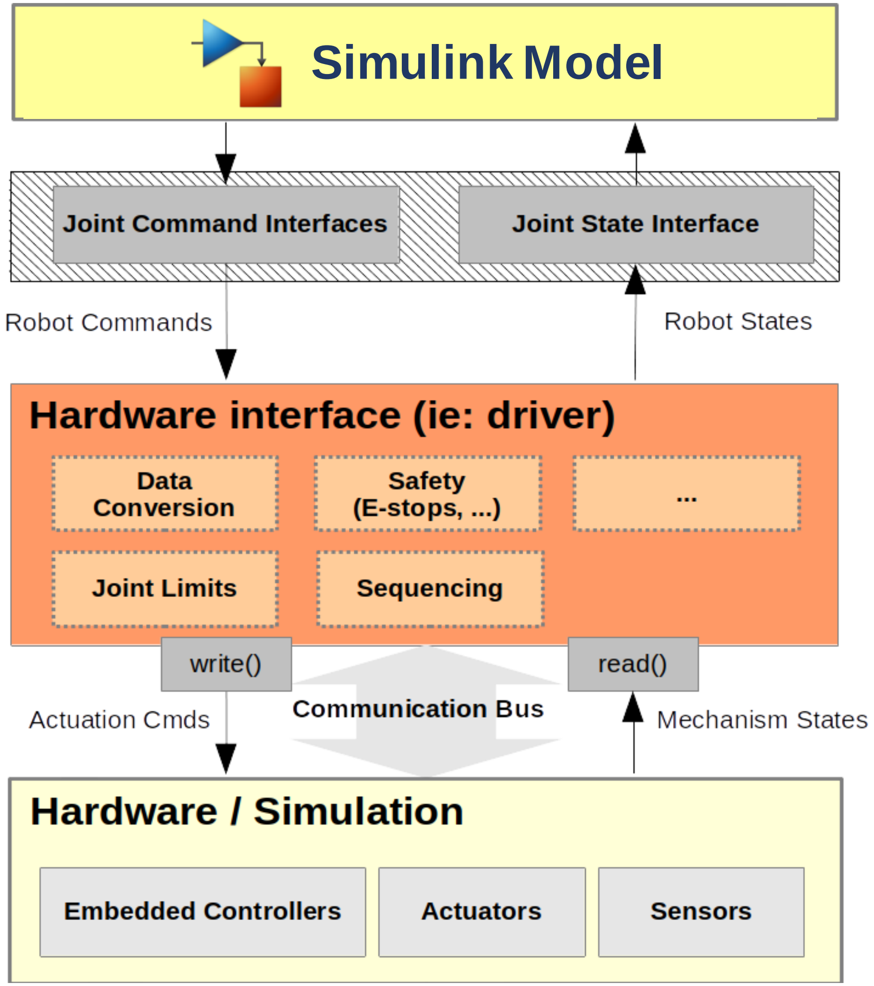
Controller

Example – Shape Tracing Controller Model

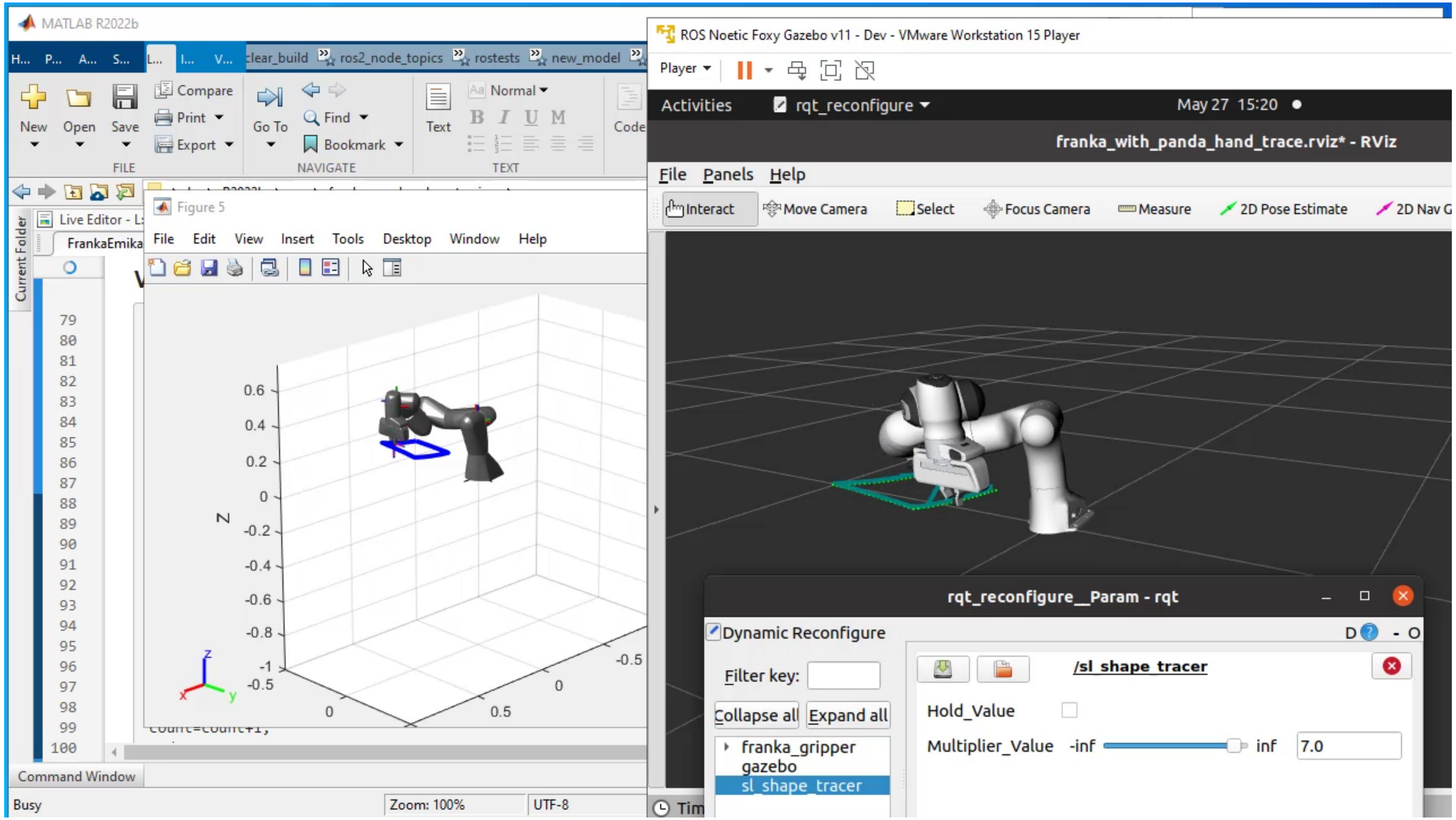


Example – 3D Shape Tracing





Video demo


MATLAB R2022b
ROS Noetic Foxy Gazebo v11 - Dev - VMware Workstation 15 Player
franka_with_panda_hand_trace.rviz* - RViz
rqt_reconfigure__Param - rqt
Dynamic Reconfigure
/sl_shape_tracer
Hold_Value
Multiplier_Value
7.0

Typical Applications using Shape Tracing

- Industrial Applications
  - Welding
  - Sanding
  - Painting
- Curved 3-D surfaces
  - Aircrafts
  - Ships

Agenda

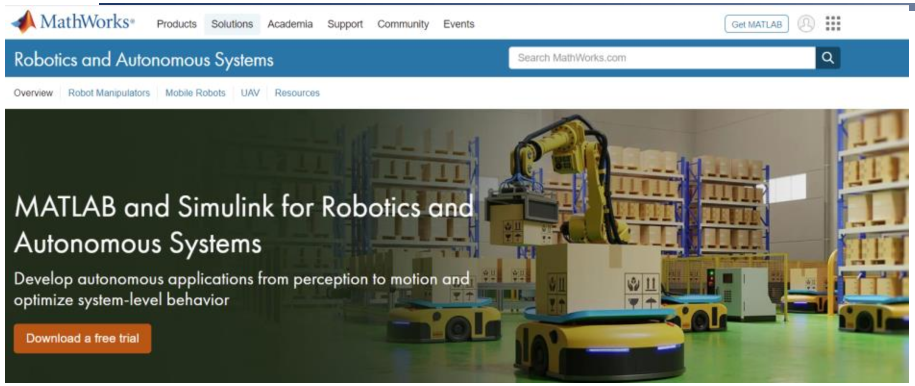
- Introduction
- Automated ROS code generation
- Automated ros_control generation
- Summary

Key Takeaways

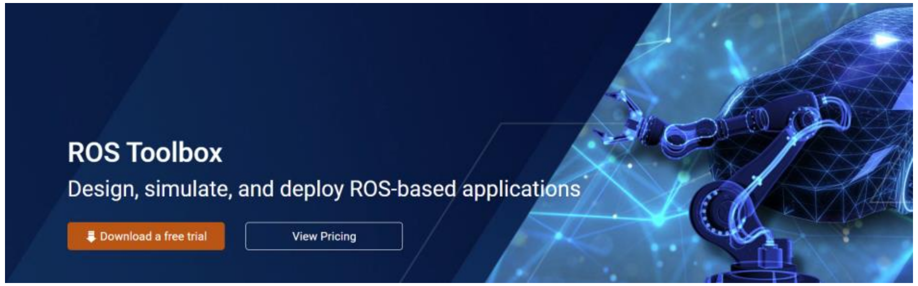
- Automated C++ ROS code and ros_control generation from MATLAB and Simulink
  - ▶ Incorporate ROS framework for implementation
  - ▶ Speed up the development process
  - ▶ Remove manual implementation errors
  - ▶ Go directly from algorithm prototyping to implementation
  - ▶ Easily incorporate Simulink controllers into ros_control framework

- **Call-To-Action:**
  - ▶ Try out the reference examples from ROS Toolbox
  - ▶ Reach out to us to work on real-world industrial applications

Learn More

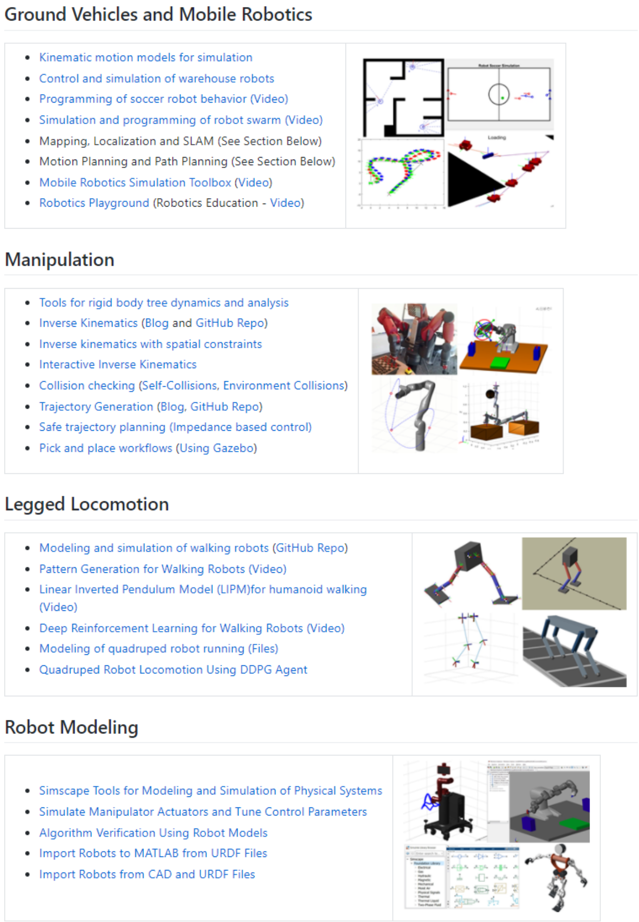


[Robotics Solutions Page]

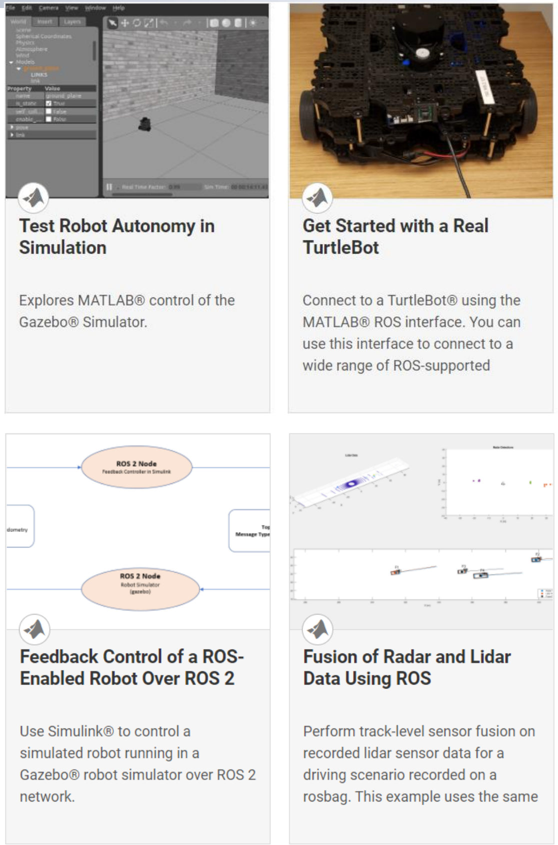


[ROS Toolbox]

MathWorks Robotics Solution Page



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



ROS Examples ([LINK])

Acknowledgement

We thank **Gijs van der Hoorn** for providing his guidance creating 'Automated ROS Control Plugin from Simulink'

Gijs van der Hoorn

TU Delft Delft University of Technology

YJ Lim
yjlim@mathworks.com

Abhijeet Gadkari
agadkari@mathworks.com

Murat Belge
mbelge@mathworks.com

MathWorks®

Accelerating the pace of engineering and science