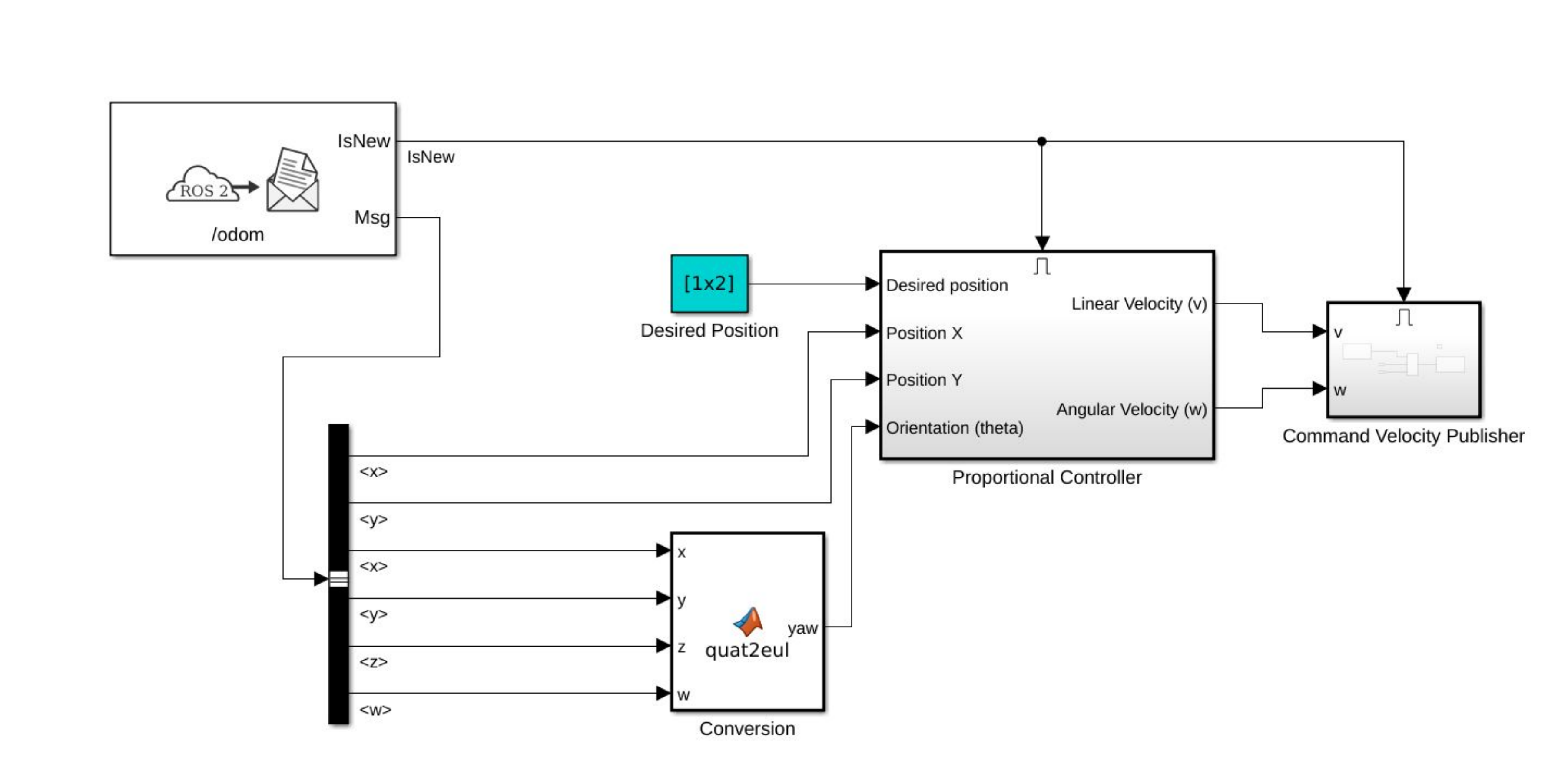
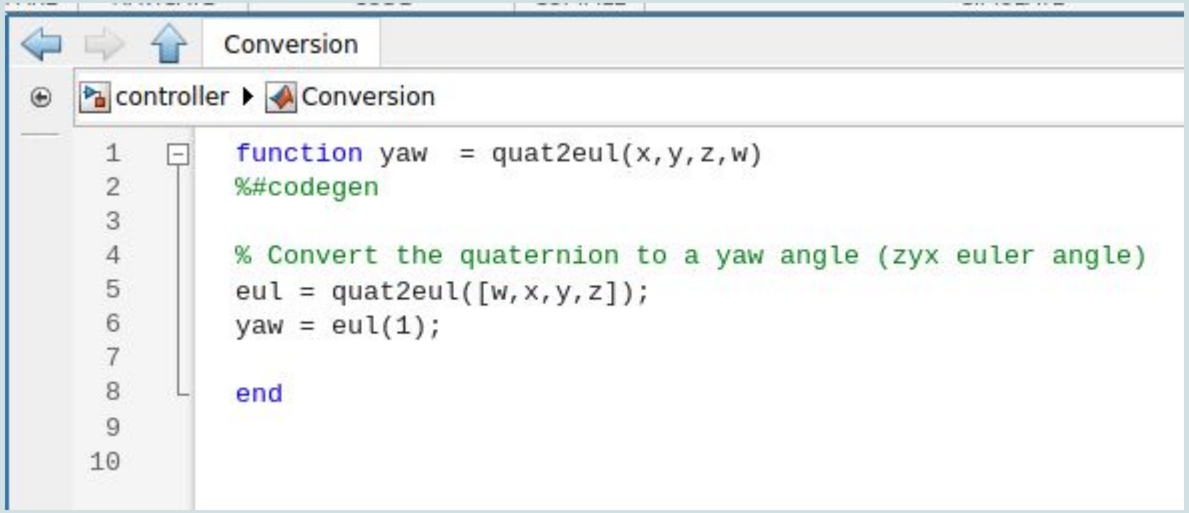
# Generating ROS node from Simulink Model :

Submitted By Anuj Yadav

# Step 1: Draw your Control Designs in Simulink





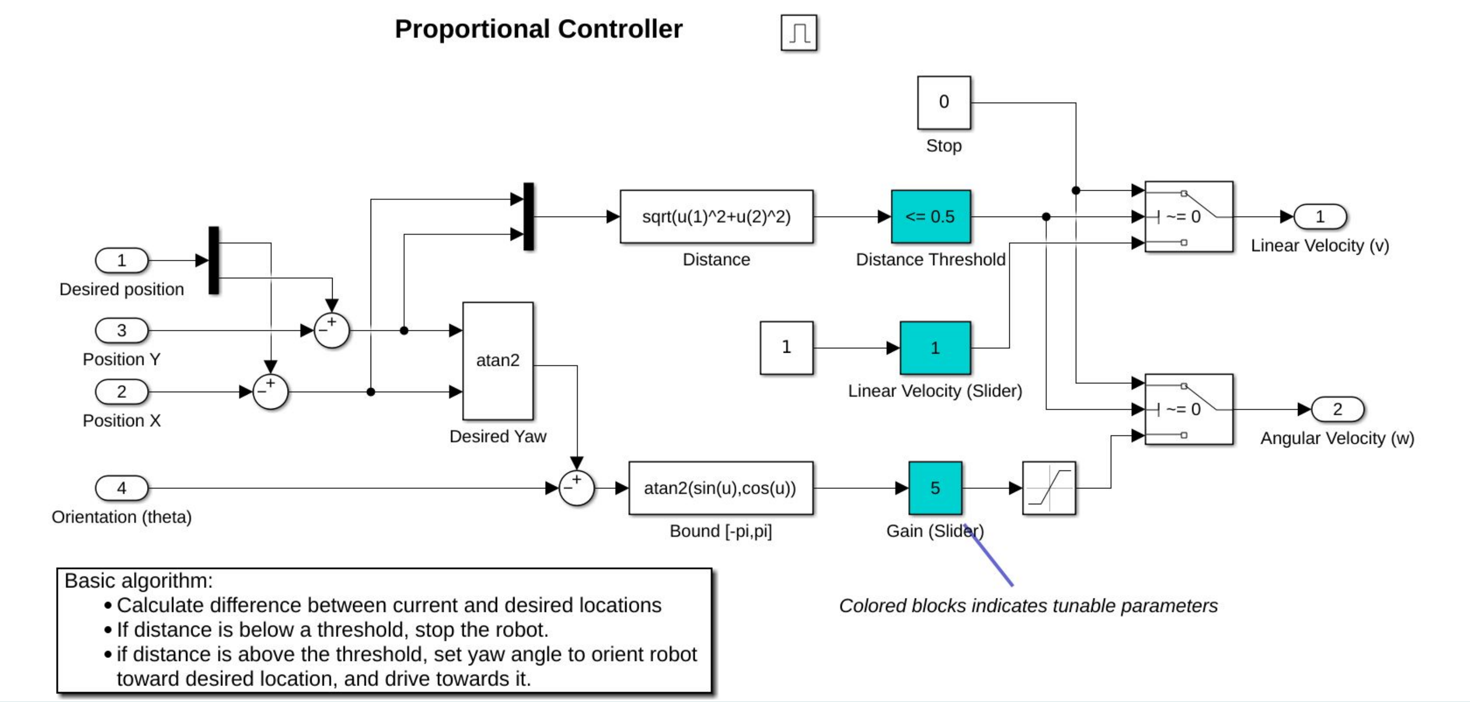
```
function yaw  = quat2eul(x,y,z,w)
%#codegen

% Convert the quaternion to a yaw angle (zyx euler angle)
eul = quat2eul([w,x,y,z]);
yaw = eul(1);

end
```

Quat2eul conversion

Proportional controller

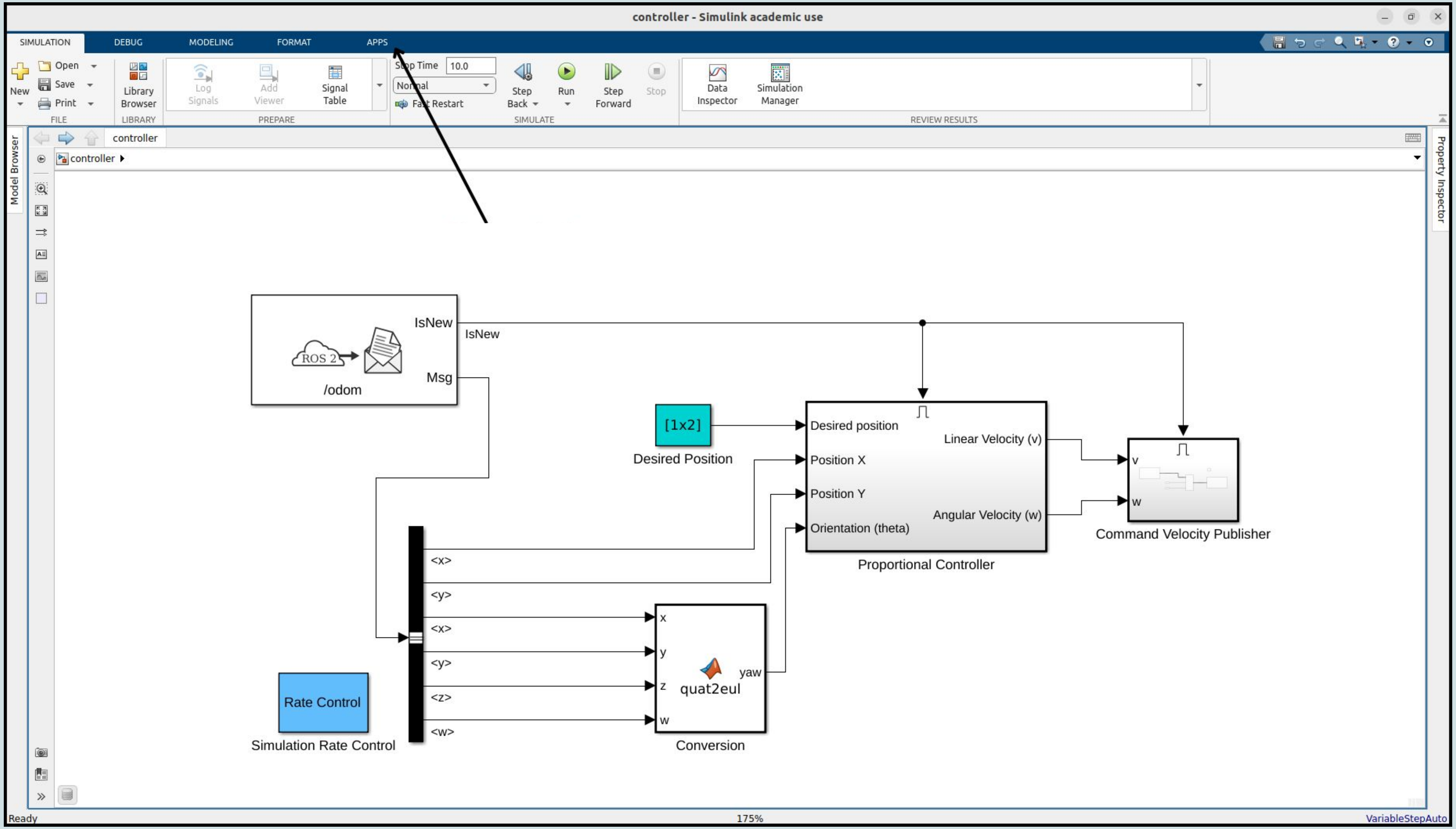


Basic algorithm:

- Calculate difference between current and desired locations
- If distance is below a threshold, stop the robot.
- if distance is above the threshold, set yaw angle to orient robot toward desired location, and drive towards it.

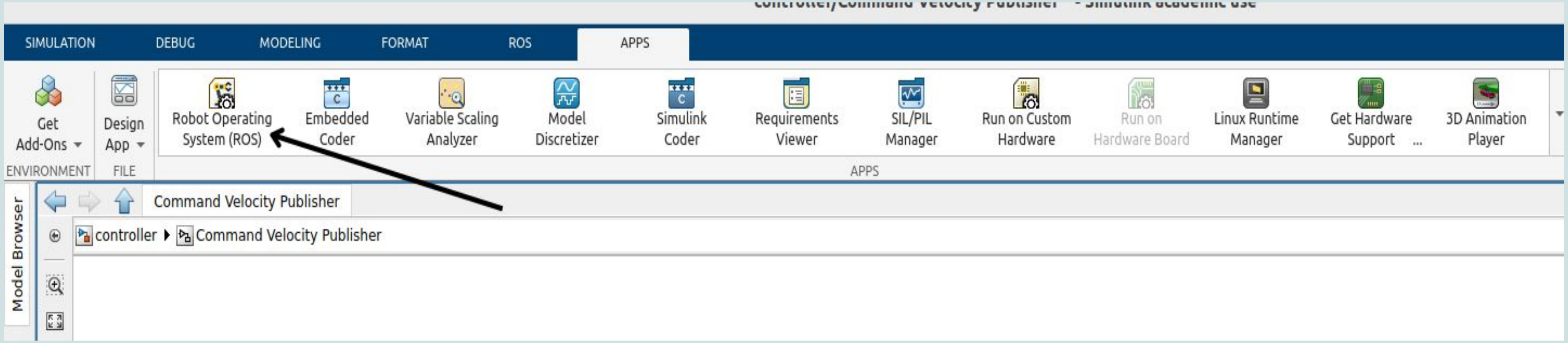
*Colored blocks indicates tunable parameters*

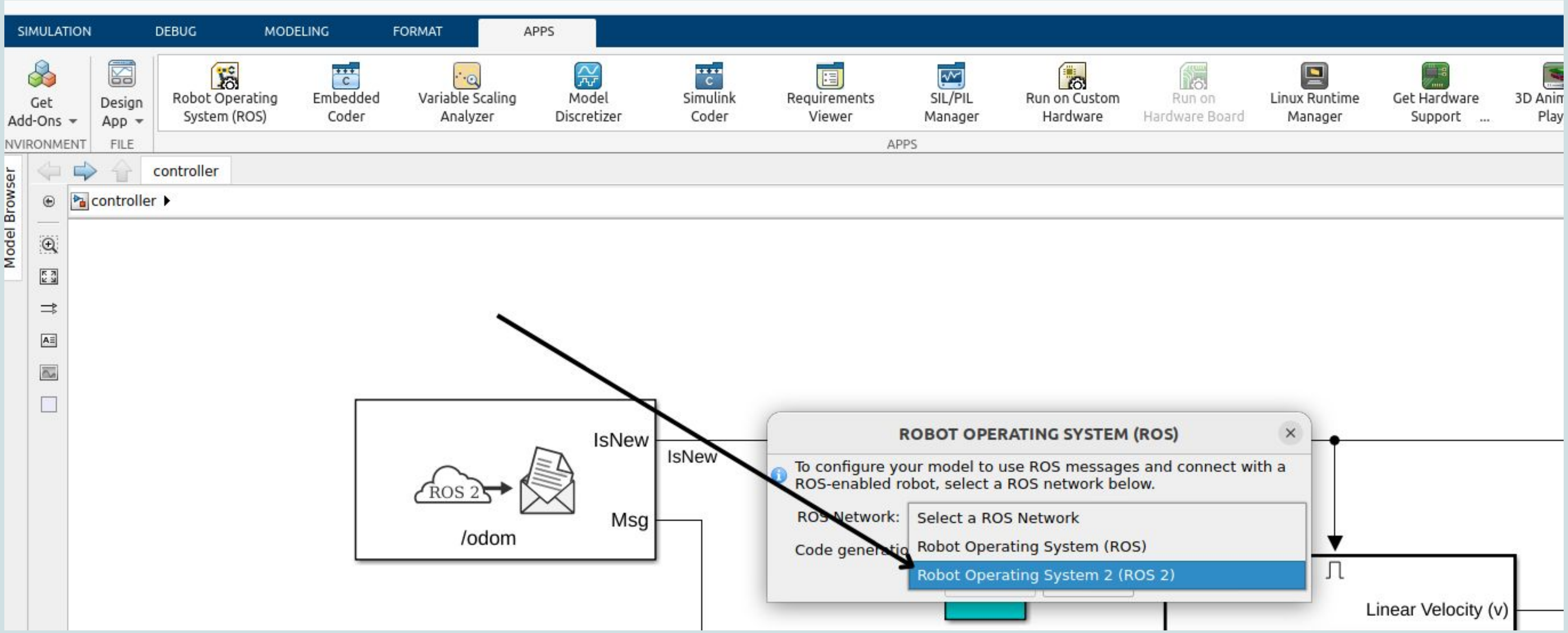
# Step 2 :



controller - Simulink academic use
SIMULATION
DEBUG
MODELING
FORMAT
APPS
New
Open
Save
Print
FILE
Library Browser
LIBRARY
Log Signals
Add Viewer
Signal Table
PREPARE
Stop Time 10.0
Normal
Fast Restart
Step Back
Run
Step Forward
Stop
SIMULATE
Data Inspector
Simulation Manager
REVIEW RESULTS
Model Browser
Property Inspector
controller
IsNew
Msg
ROS 2
/odom
IsNew
[1x2]
Desired Position
Desired position
Position X
Position Y
Orientation (theta)
Linear Velocity (v)
Angular Velocity (w)
Proportional Controller
v
w
Command Velocity Publisher
<x>
<y>
<x>
<y>
<z>
<w>
x
y
z
w
quat2eul
yaw
Conversion
Rate Control
Simulation Rate Control
Ready
175%
VariableStepAuto

# Step 3:





ROBOT OPERATING SYSTEM (ROS)

To configure your model to use ROS messages and connect with a ROS-enabled robot, select a ROS network below.

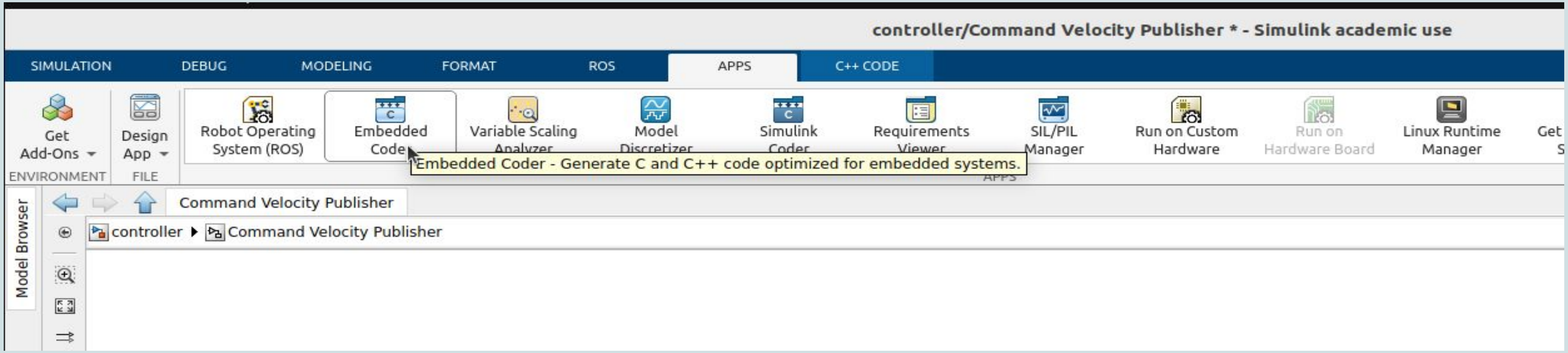
ROS Network: Select a ROS Network

Robot Operating System (ROS)

Robot Operating System 2 (ROS 2)

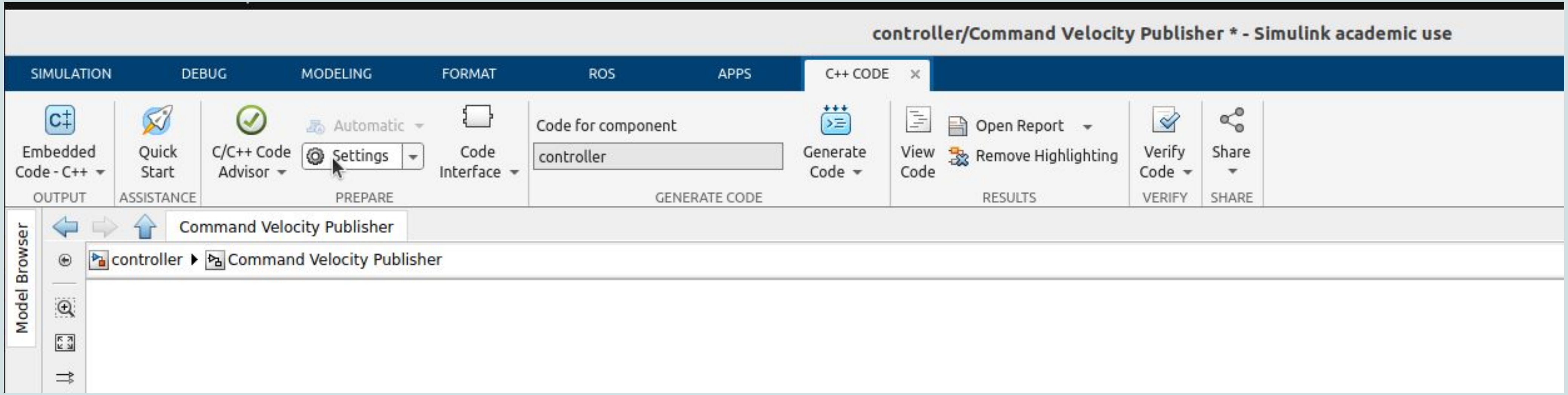
Code generatio

Step 4:



controller/Command Velocity Publisher * - Simulink academic use

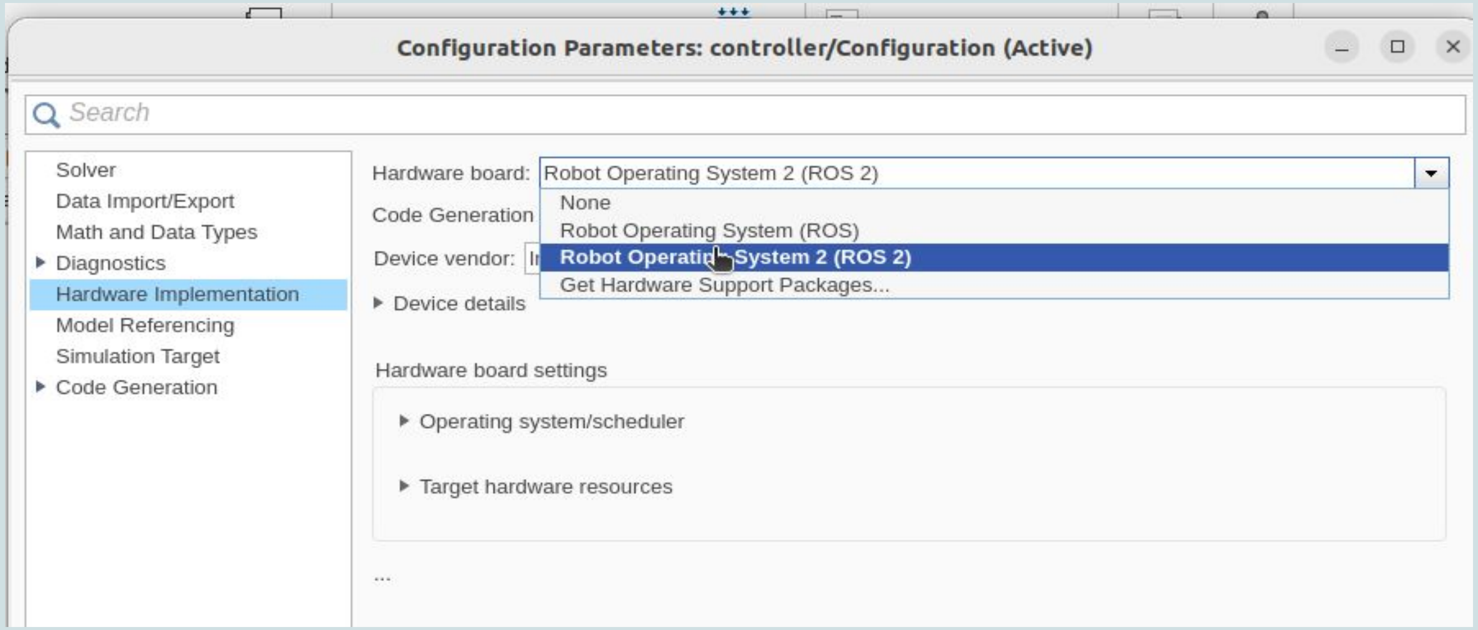
Embedded Coder - Generate C and C++ code optimized for embedded systems.



controller/Command Velocity Publisher * - Simulink academic use

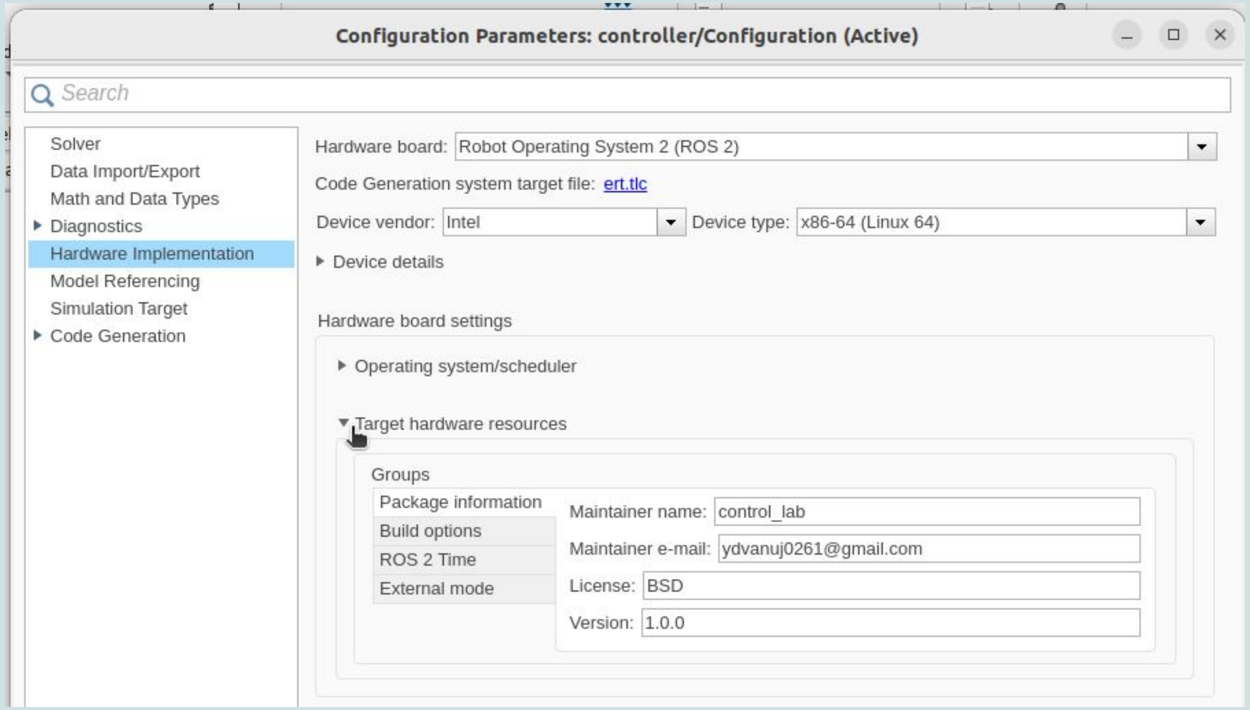
Code for component: controller

# Step 5:



Configuration Parameters: controller/Configuration (Active)

Hardware board: Robot Operating System 2 (ROS 2)



Configuration Parameters: controller/Configuration (Active)

Hardware board: Robot Operating System 2 (ROS 2)

Code Generation system target file: ert.tlc

Device vendor: Intel

Device type: x86-64 (Linux 64)

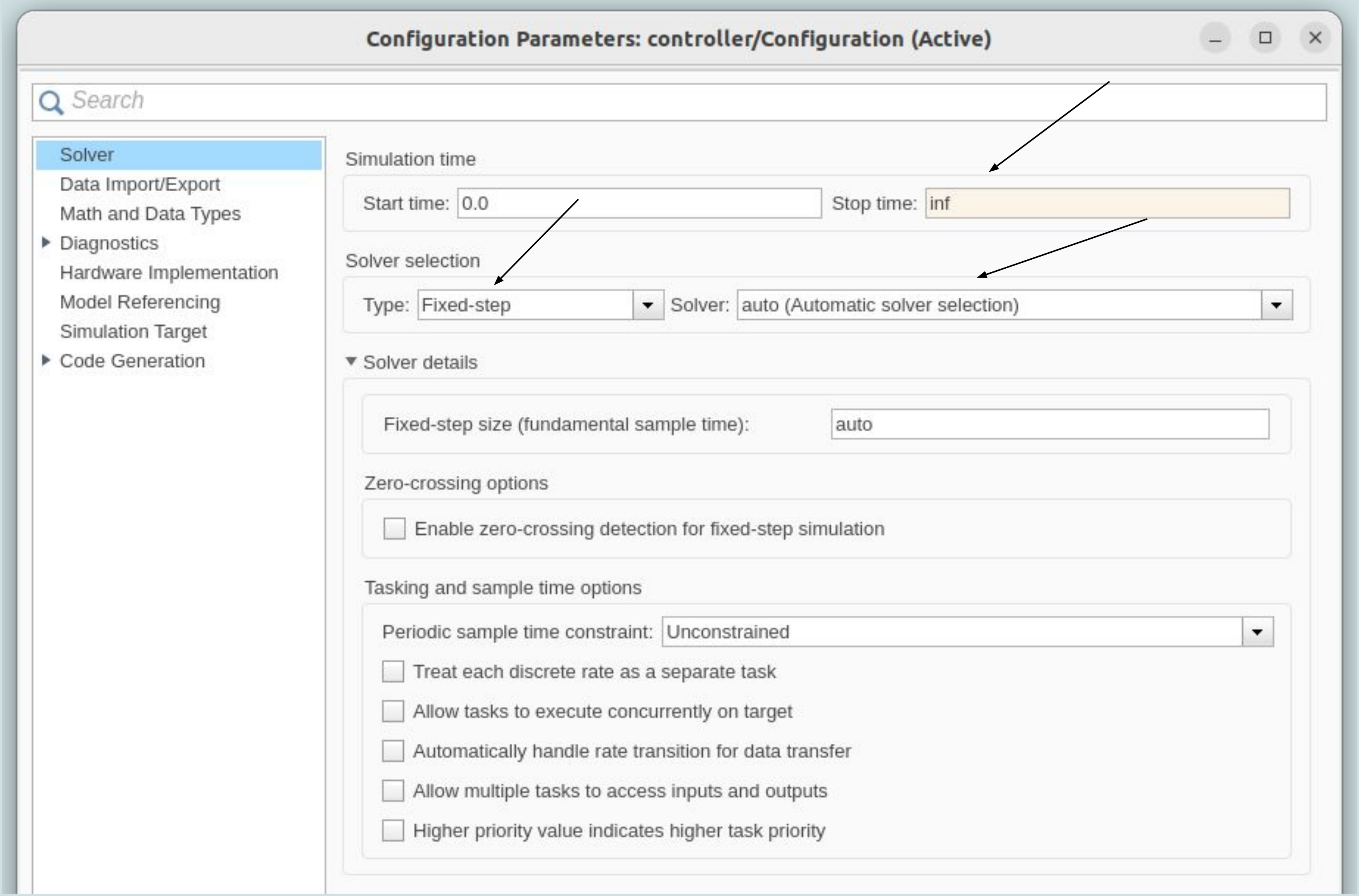
Maintainer name: control_lab

Maintainer e-mail: ydvanuj0261@gmail.com

License: BSD

Version: 1.0.0

# Step 6:



Configuration Parameters: controller/Configuration (Active)

Search

Simulation time

Start time: 0.0 Stop time: inf

Solver selection

Type: Fixed-step Solver: auto (Automatic solver selection)

Solver details

Fixed-step size (fundamental sample time): auto

Zero-crossing options

□ Enable zero-crossing detection for fixed-step simulation

Tasking and sample time options

Periodic sample time constraint: Unconstrained

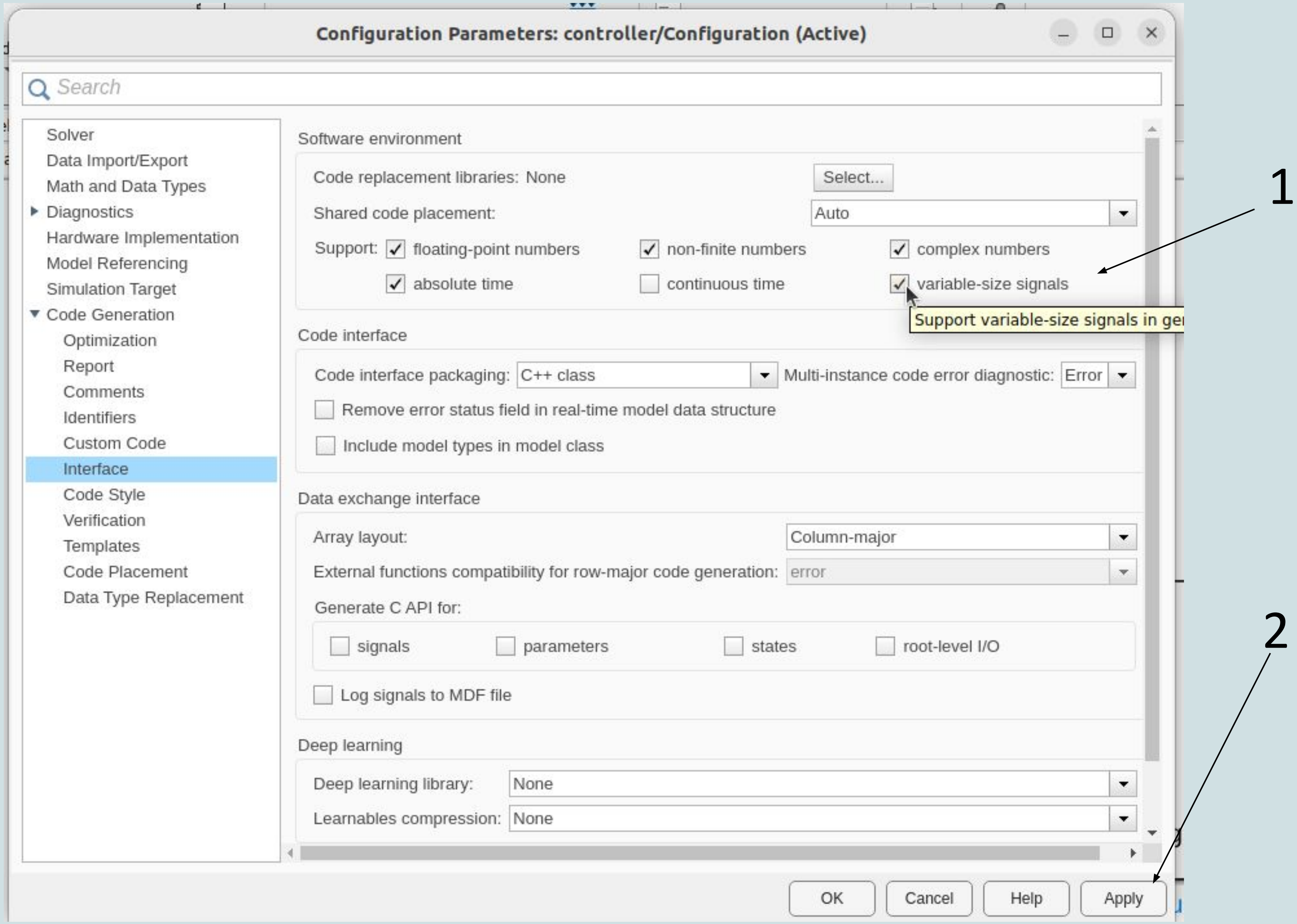
□ Treat each discrete rate as a separate task

□ Allow tasks to execute concurrently on target

□ Automatically handle rate transition for data transfer

□ Allow multiple tasks to access inputs and outputs

□ Higher priority value indicates higher task priority



**Configuration Parameters: controller/Configuration (Active)**

Software environment

Code replacement libraries: None

Shared code placement: Auto

Support: ☑ floating-point numbers ☑ non-finite numbers ☑ complex numbers

☑ absolute time □ continuous time ☑ variable-size signals

Support variable-size signals in ge

Code interface

Code interface packaging: C++ class Multi-instance code error diagnostic: Error

□ Remove error status field in real-time model data structure

□ Include model types in model class

Data exchange interface

Array layout: Column-major

External functions compatibility for row-major code generation: error

Generate C API for:

□ signals □ parameters □ states □ root-level I/O

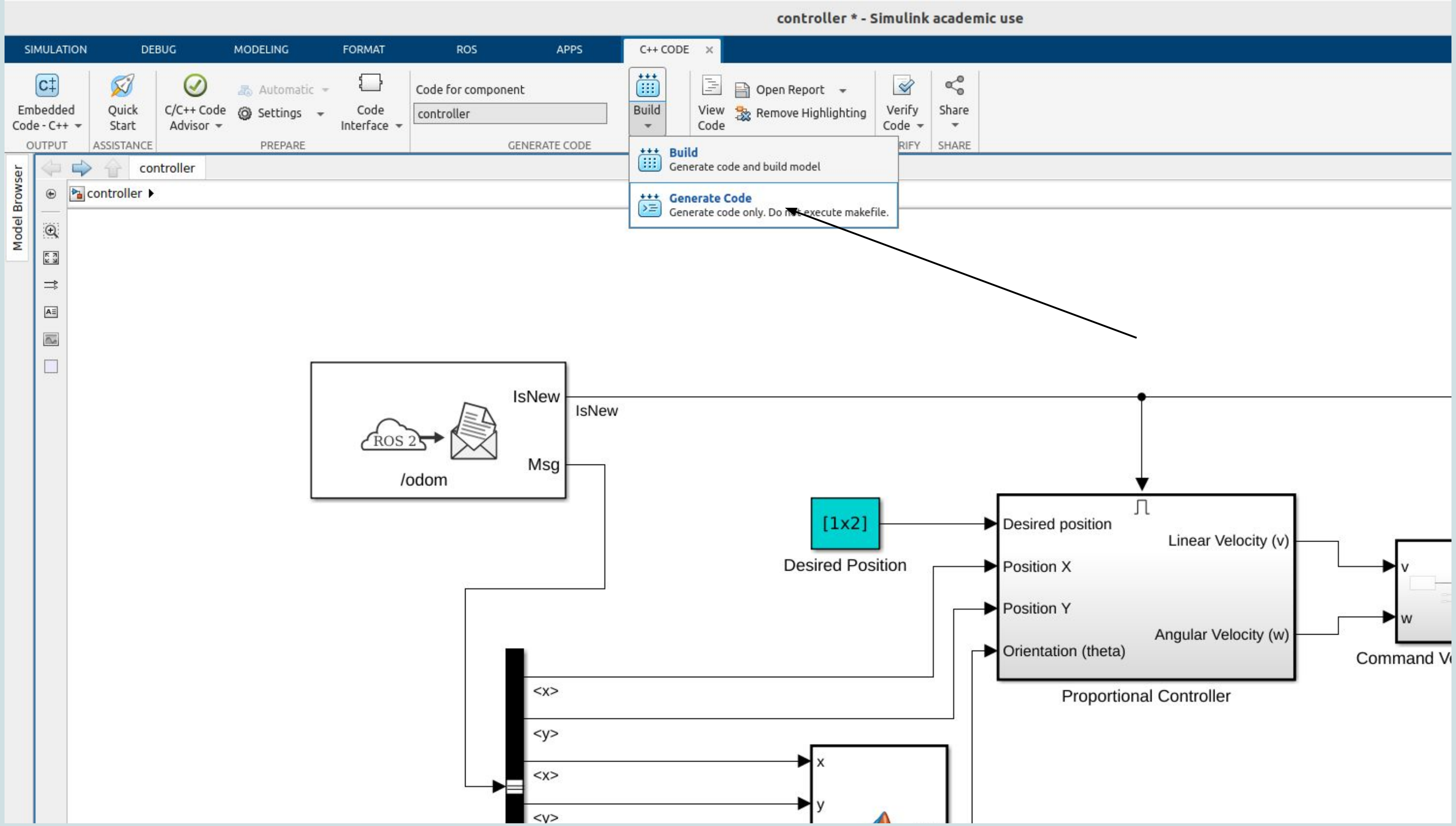
□ Log signals to MDF file

Deep learning

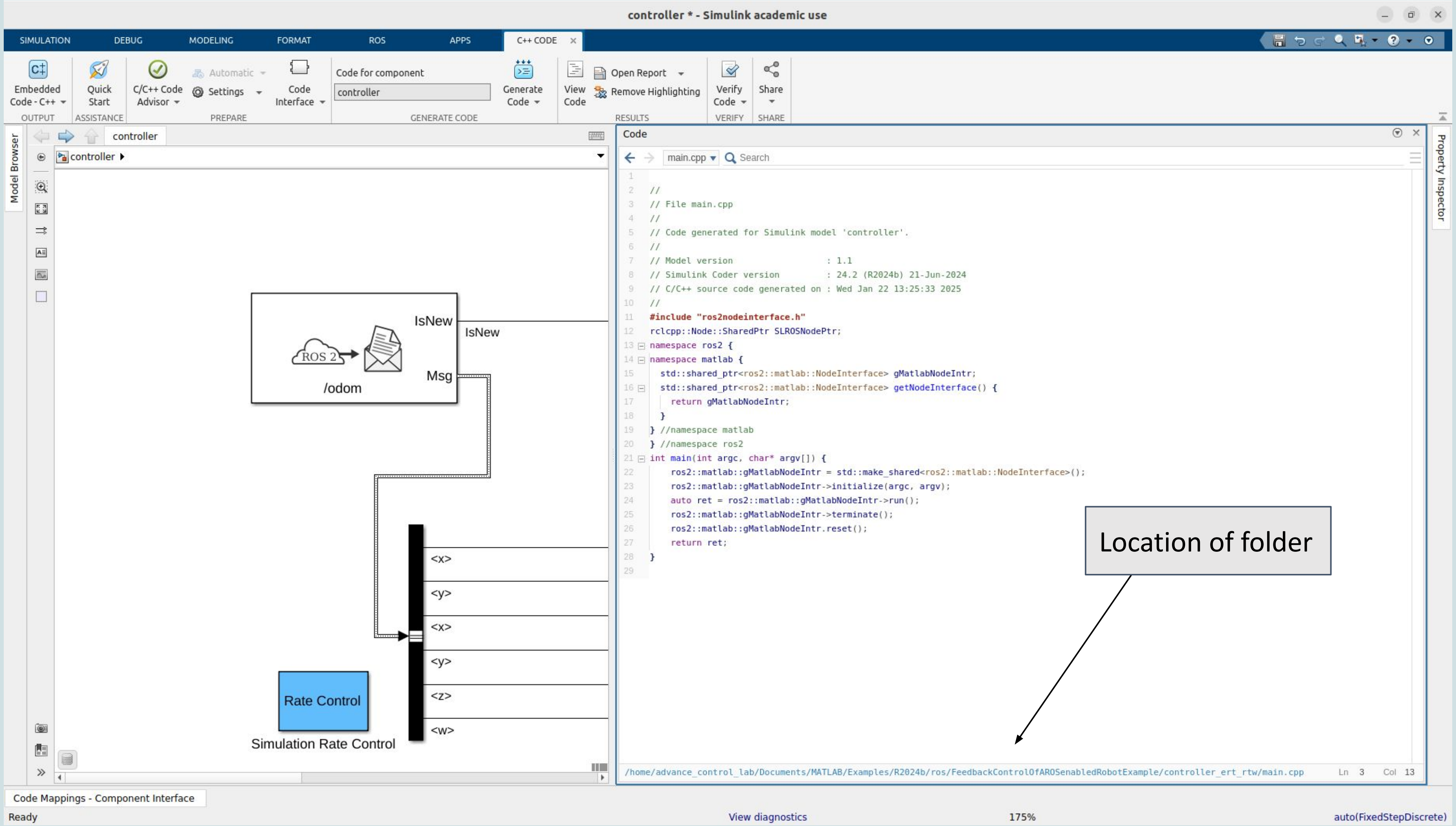
Deep learning library: None

Learnables compression: None

# Step 7:


controller * - Simulink academic use
SIMULATION
DEBUG
MODELING
FORMAT
ROS
APPS
C++ CODE
Embedded Code - C++
Quick Start
C/C++ Code Advisor
Automatic
Settings
Code Interface
Code for component
controller
Build
View Code
Open Report
Remove Highlighting
Verify Code
Share
OUTPUT
ASSISTANCE
PREPARE
GENERATE CODE
RIFY
SHARE
Build
Generate code and build model
Generate Code
Generate code only. Do not execute makefile.
Model Browser
controller
controller
IsNew
Msg
ROS 2
/odom
IsNew
[1x2]
Desired Position
Desired position
Position X
Position Y
Orientation (theta)
Linear Velocity (v)
Angular Velocity (w)
Proportional Controller
v
w
Command V
<x>
<y>
<x>
<y>
x
y

# Step 8:



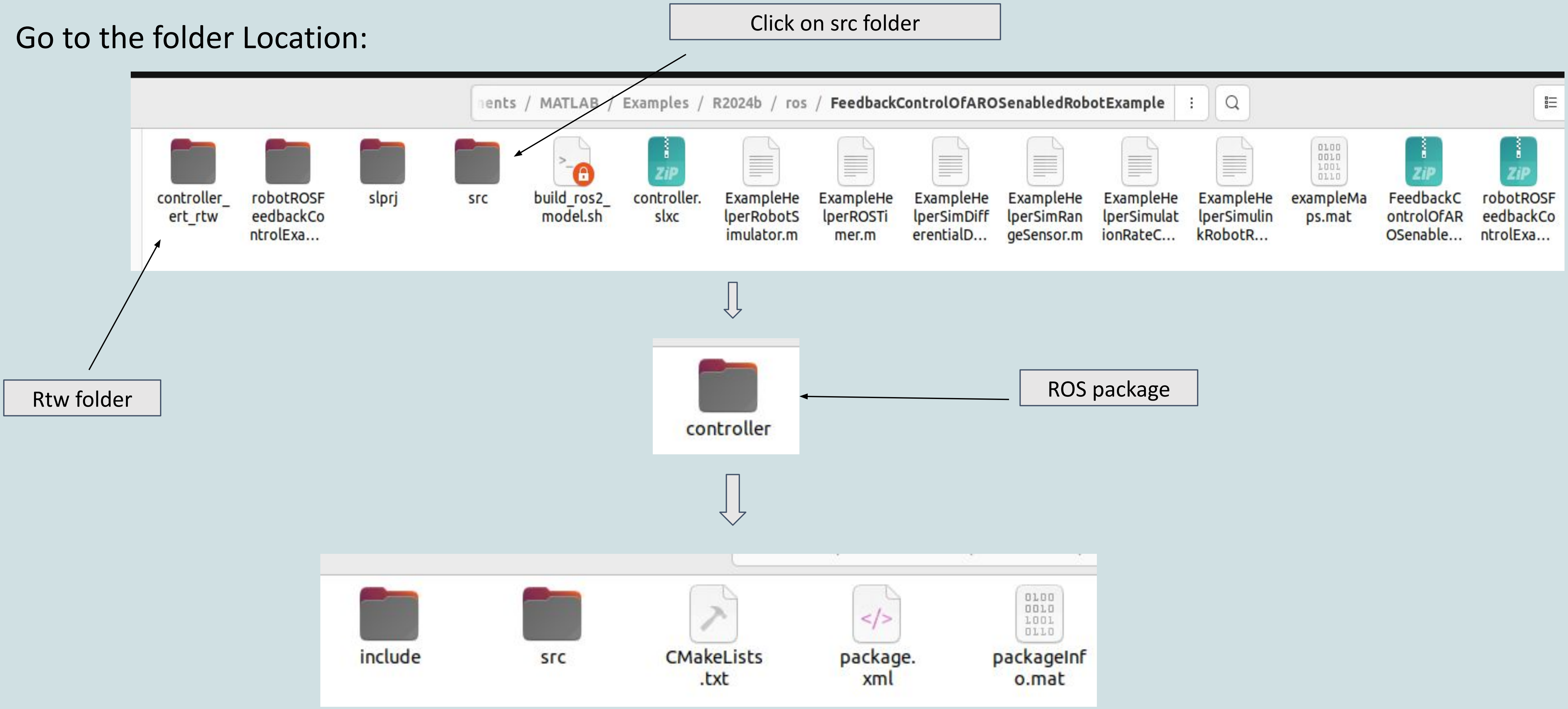
Code for component: controller

```
//
// File main.cpp
//
// Code generated for Simulink model 'controller'.
//
// Model version                  : 1.1
// Simulink Coder version         : 24.2 (R2024b) 21-Jun-2024
// C/C++ source code generated on : Wed Jan 22 13:25:33 2025
//
#include "ros2nodeinterface.h"
rclcpp::Node::SharedPtr SLROSNodePtr;
namespace ros2 {
namespace matlab {
  std::shared_ptr<ros2::matlab::NodeInterface> gMatlabNodeIntr;
  std::shared_ptr<ros2::matlab::NodeInterface> getNodeInterface() {
    return gMatlabNodeIntr;
  }
} //namespace matlab
} //namespace ros2
int main(int argc, char* argv[]) {
    ros2::matlab::gMatlabNodeIntr = std::make_shared<ros2::matlab::NodeInterface>();
    ros2::matlab::gMatlabNodeIntr->initialize(argc, argv);
    auto ret = ros2::matlab::gMatlabNodeIntr->run();
    ros2::matlab::gMatlabNodeIntr->terminate();
    ros2::matlab::gMatlabNodeIntr.reset();
    return ret;
}
```

```
/home/advance_control_lab/Documents/MATLAB/Examples/R2024b/ros/FeedbackControlOfAROSenabledRobotExample/controller_ert_rtw/main.cpp    Ln 3    Col 13
```

Location of folder

# Go to the folder Location:



Click on src folder

Rtw folder

ROS package

ents / MATLAB / Examples / R2024b / ros / FeedbackControlOfAROSenabledRobotExample
controller_ert_rtw
robotROSFeedbackControlExa...
slprj
src
build_ros2_model.sh
controller.slxc
ExampleHelperRobotSimulator.m
ExampleHelperROSTimer.m
ExampleHelperSimDifferentialD...
ExampleHelperSimRangeSensor.m
ExampleHelperSimulationRateC...
ExampleHelperSimulinkRobotR...
exampleMaps.mat
FeedbackControlOfAROSenable...
robotROSFeedbackControlExa...
controller
include
src
CMakeLists.txt
package.xml
packageInfo.mat

# Step 9:

- Deploy ROS package into you workspace directory
- Build your Package
- Now run your ROS node into your ROS environment