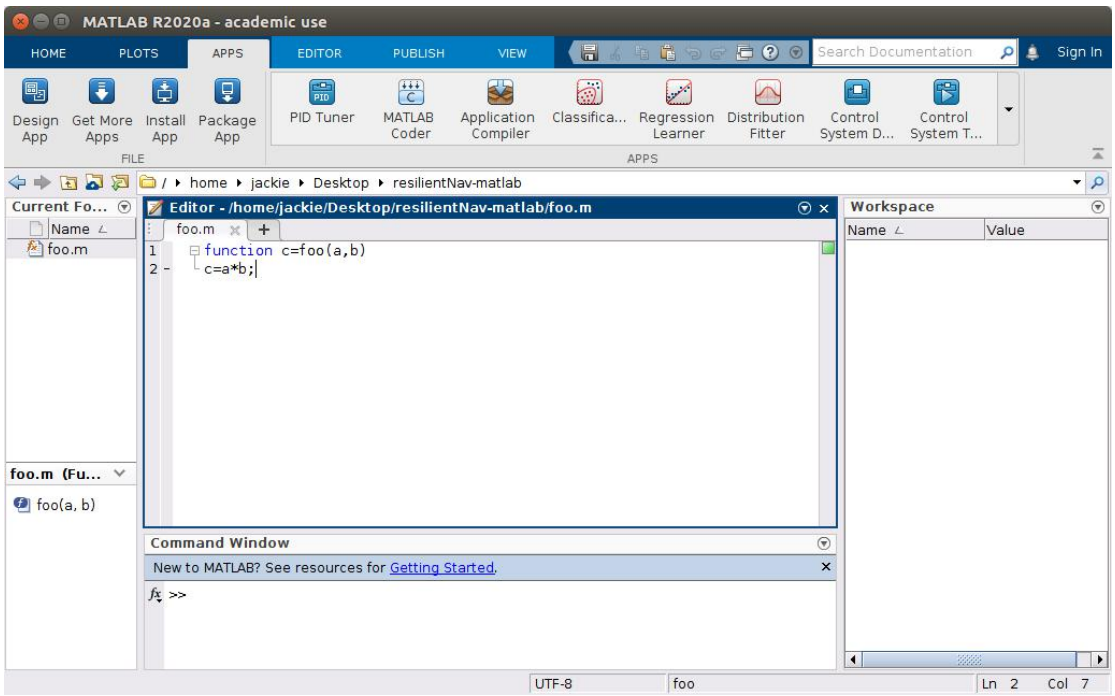
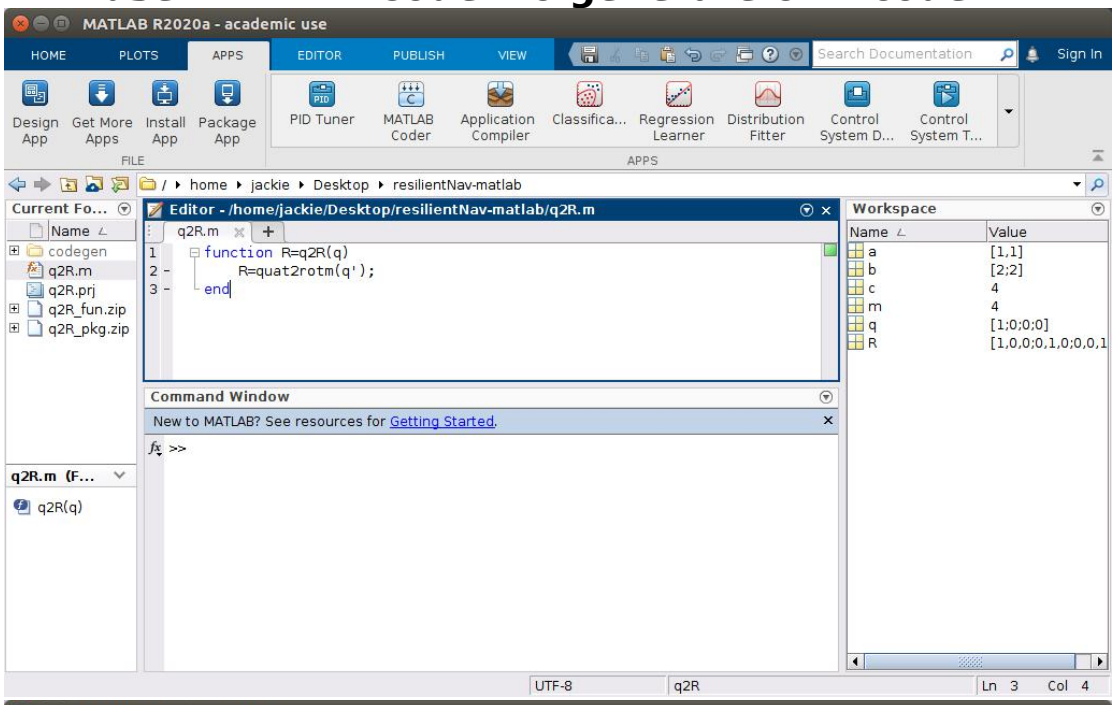
1. Environment

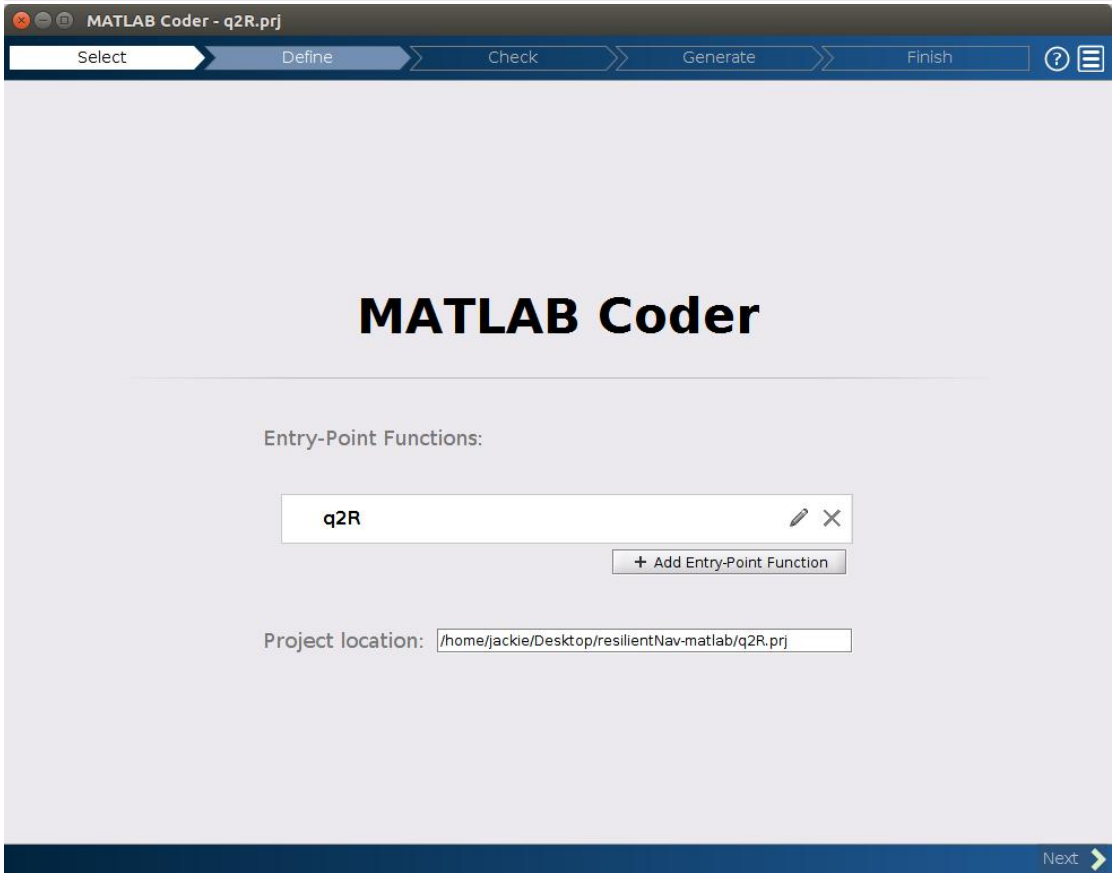
2. Generate c++ function from MATLAB function and using it in ROS workspace(such as catkin_ws)

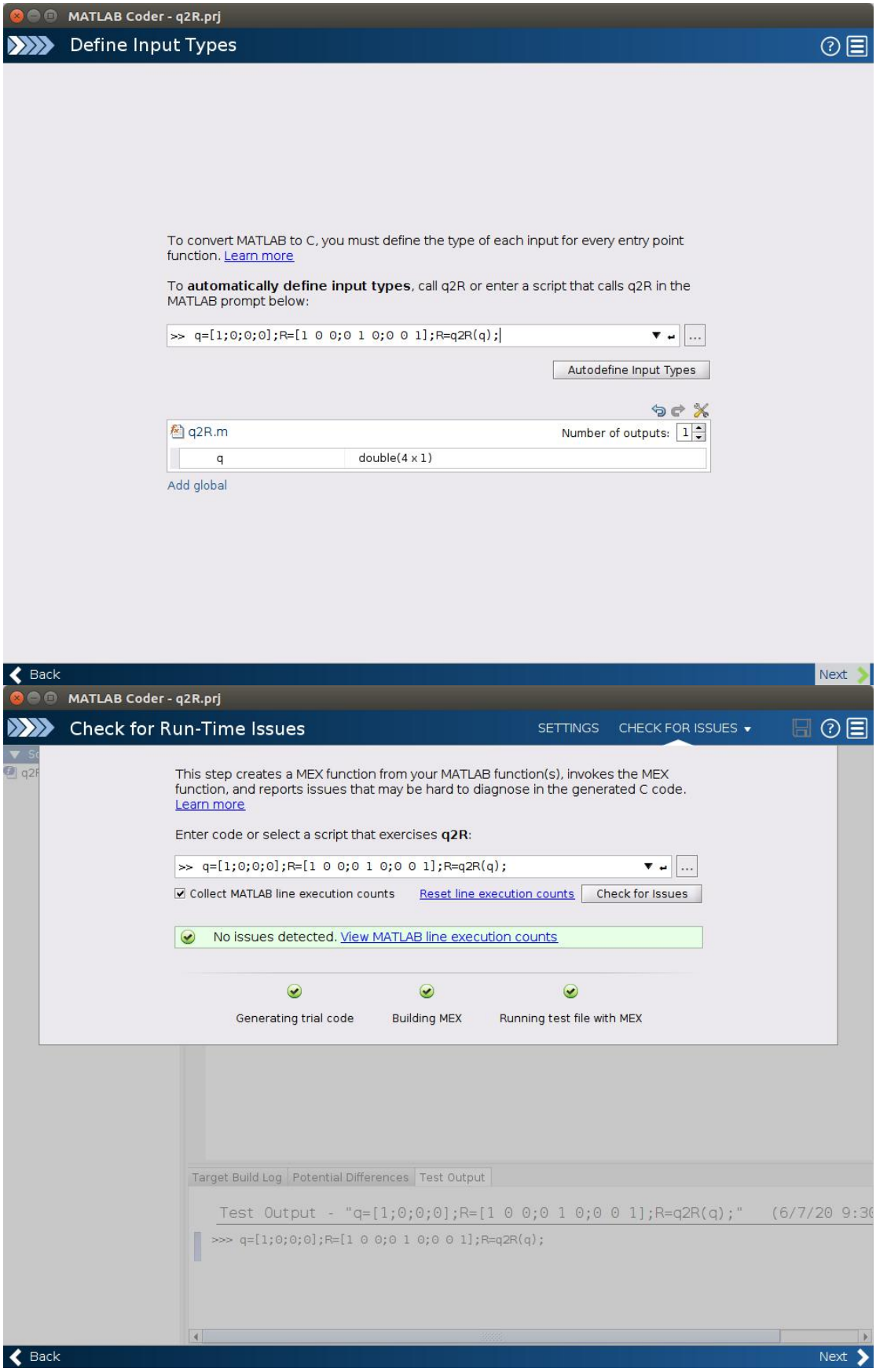
2.1. edit function in MATLAB

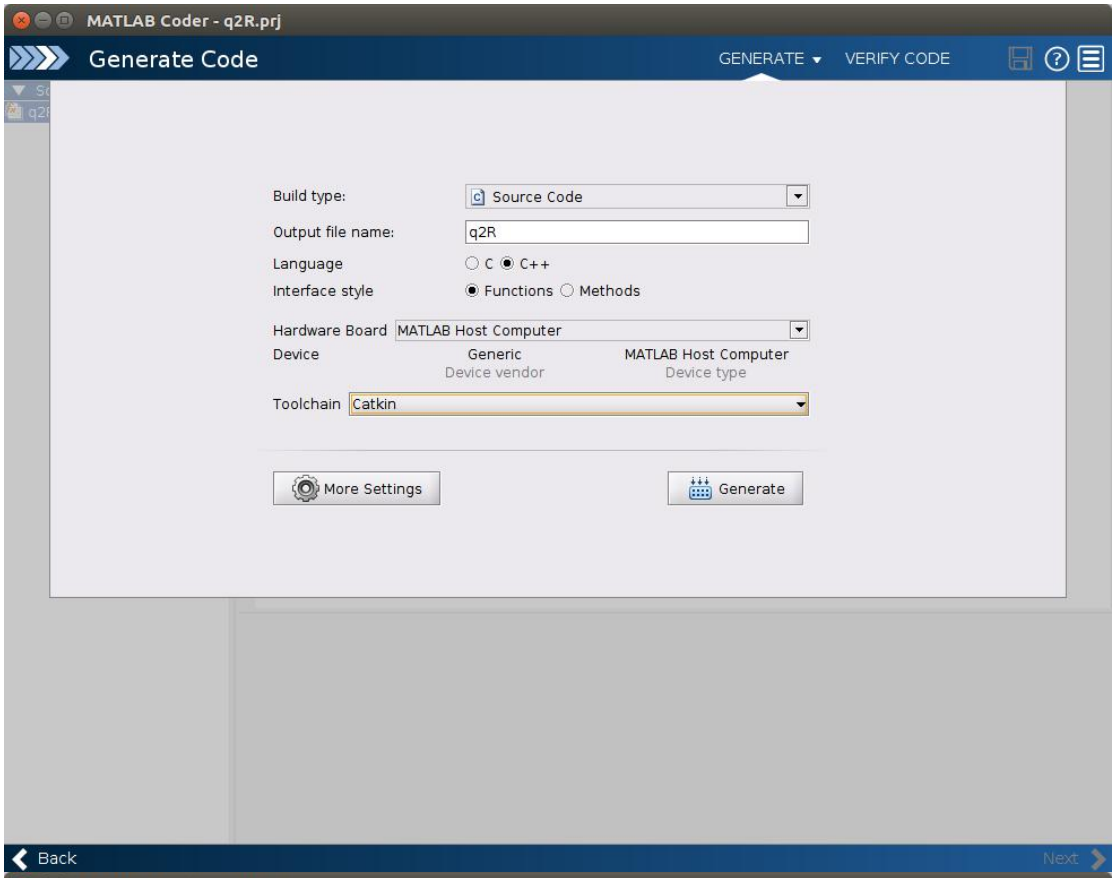


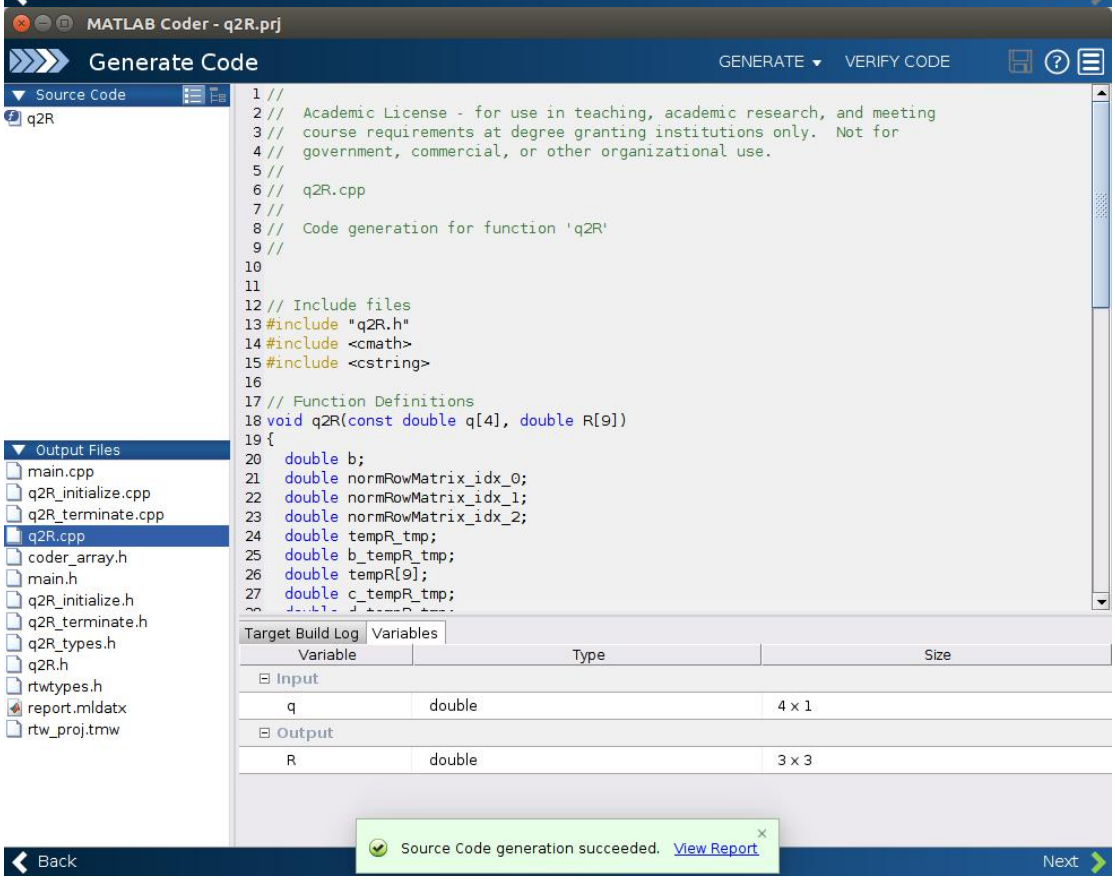
2.2. use MATLAB Coder to generate c++ code

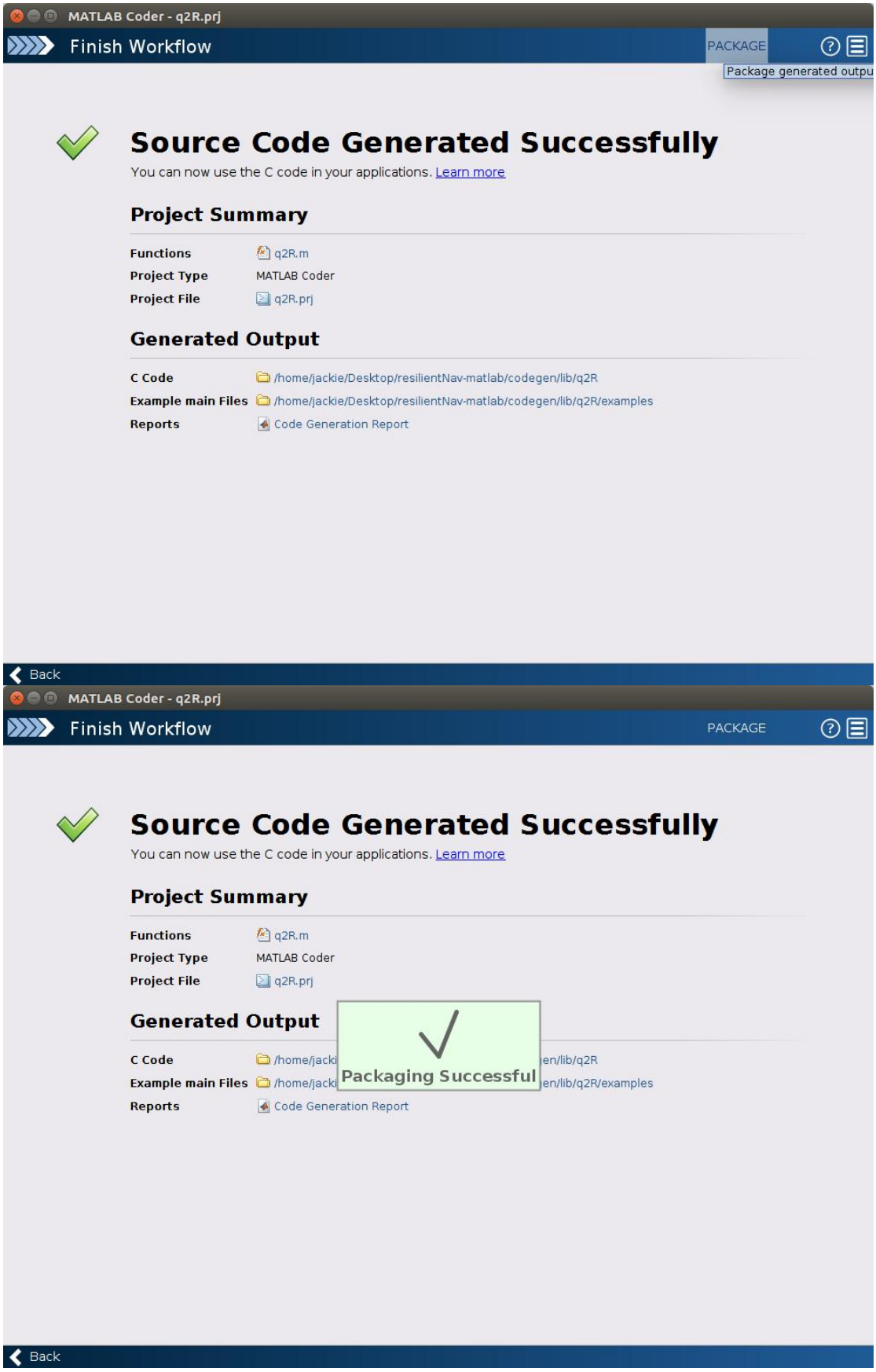




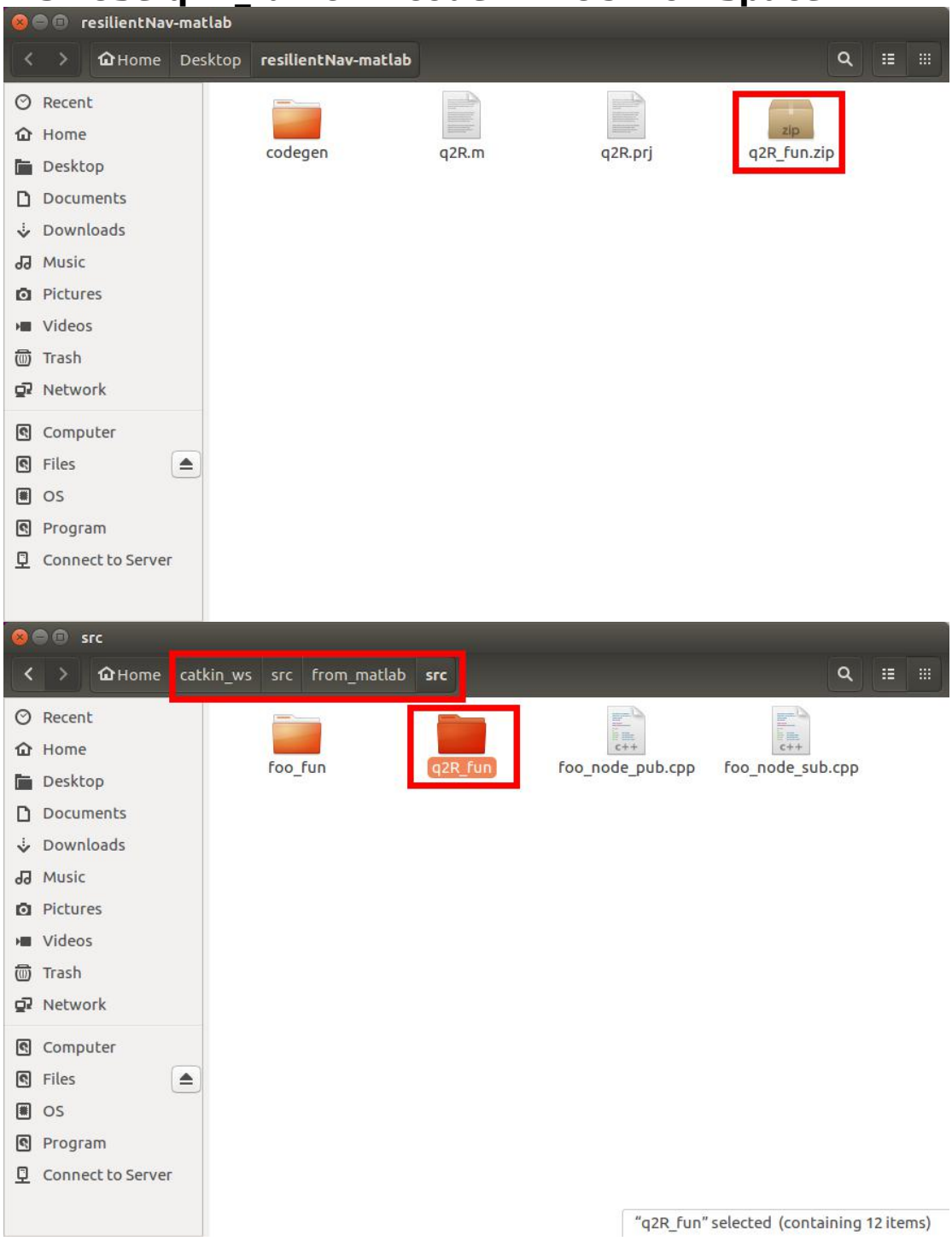

MATLAB Coder - q2R.prj
Define Input Types
To convert MATLAB to C, you must define the type of each input for every entry point function. Learn more
To automatically define input types, call q2R or enter a script that calls q2R in the MATLAB prompt below:
>> q=[1;0;0;0];R=[1 0 0;0 1 0;0 0 1];R=q2R(q);
Autodefine Input Types
q2R.m
Number of outputs: 1
q
double(4 x 1)
Add global
Back
Next
MATLAB Coder - q2R.prj
Check for Run-Time Issues
SETTINGS
CHECK FOR ISSUES
This step creates a MEX function from your MATLAB function(s), invokes the MEX function, and reports issues that may be hard to diagnose in the generated C code. Learn more
Enter code or select a script that exercises q2R:
>> q=[1;0;0;0];R=[1 0 0;0 1 0;0 0 1];R=q2R(q);
Collect MATLAB line execution counts
Reset line execution counts
Check for Issues
No issues detected. View MATLAB line execution counts
Generating trial code
Building MEX
Running test file with MEX
Target Build Log
Potential Differences
Test Output
Test Output - "q=[1;0;0;0];R=[1 0 0;0 1 0;0 0 1];R=q2R(q);" (6/7/20 9:30
>>> q=[1;0;0;0];R=[1 0 0;0 1 0;0 0 1];R=q2R(q);
Back
Next

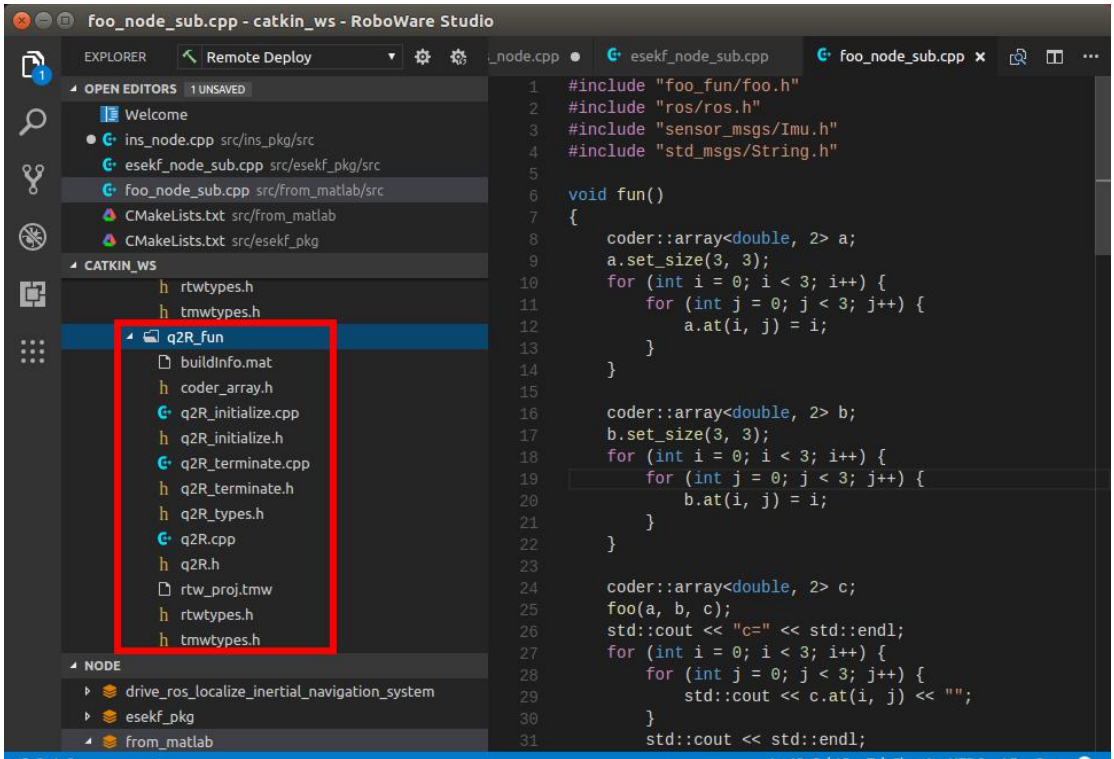

MATLAB Coder - q2R.prj
Generate Code
GENERATE
VERIFY CODE
Build type:
Source Code
Output file name:
q2R
Language
C
C++
Interface style
Functions
Methods
Hardware Board
MATLAB Host Computer
Device
Generic
Device vendor
MATLAB Host Computer
Device type
Toolchain
Catkin
More Settings
Generate
Back
Next

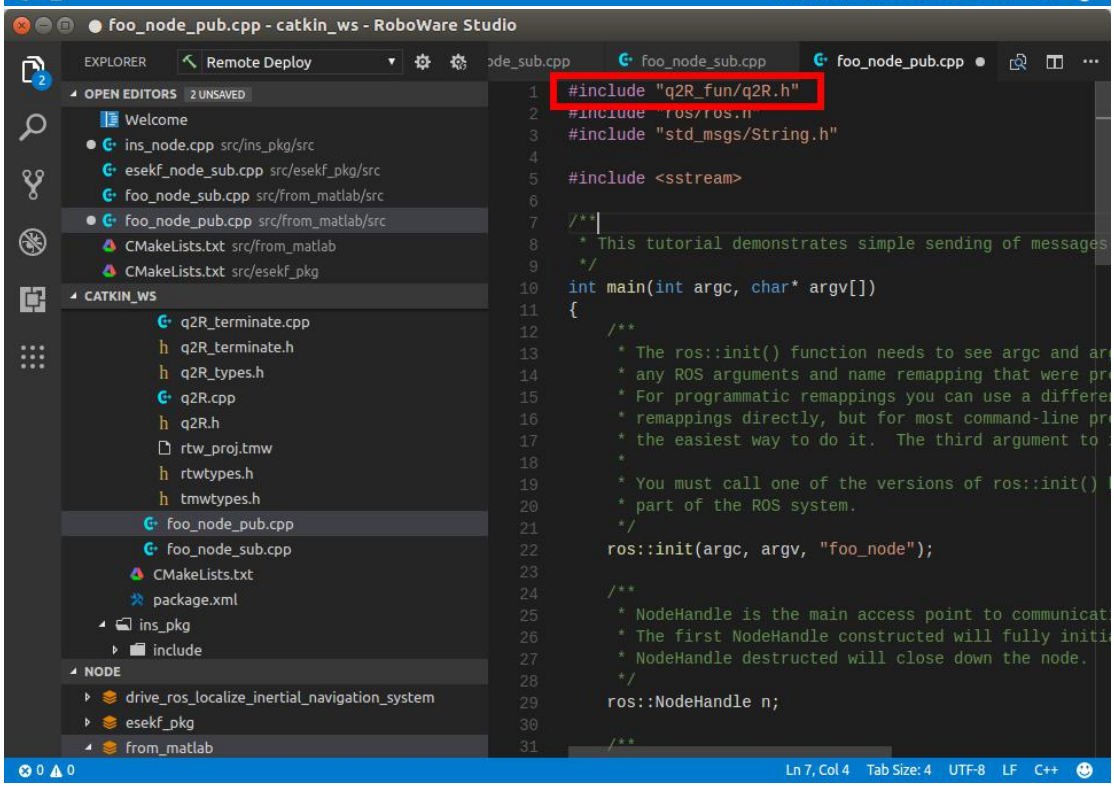

MATLAB Coder - q2R.prj
Generate Code
GENERATE
VERIFY CODE
Source Code
q2R
Output Files
main.cpp
q2R_initialize.cpp
q2R_terminate.cpp
q2R.cpp
coder_array.h
main.h
q2R_initialize.h
q2R_terminate.h
q2R_types.h
q2R.h
rtwtypes.h
report.mldatx
rtw_proj.tmw
1 //
2 // Academic License - for use in teaching, academic research, and meeting
3 // course requirements at degree granting institutions only. Not for
4 // government, commercial, or other organizational use.
5 //
6 // q2R.cpp
7 //
8 // Code generation for function 'q2R'
9 //
10
11
12 // Include files
13 #include "q2R.h"
14 #include <cmath>
15 #include <cstring>
16
17 // Function Definitions
18 void q2R(const double q[4], double R[9])
19 {
20   double b;
21   double normRowMatrix_idx_0;
22   double normRowMatrix_idx_1;
23   double normRowMatrix_idx_2;
24   double tempR_tmp;
25   double b_tempR_tmp;
26   double tempR[9];
27   double c_tempR_tmp;
Target Build Log
Variables
Variable
Type
Size
Input
q
double
4 x 1
Output
R
double
3 x 3
Source Code generation succeeded.
View Report
Back
Next

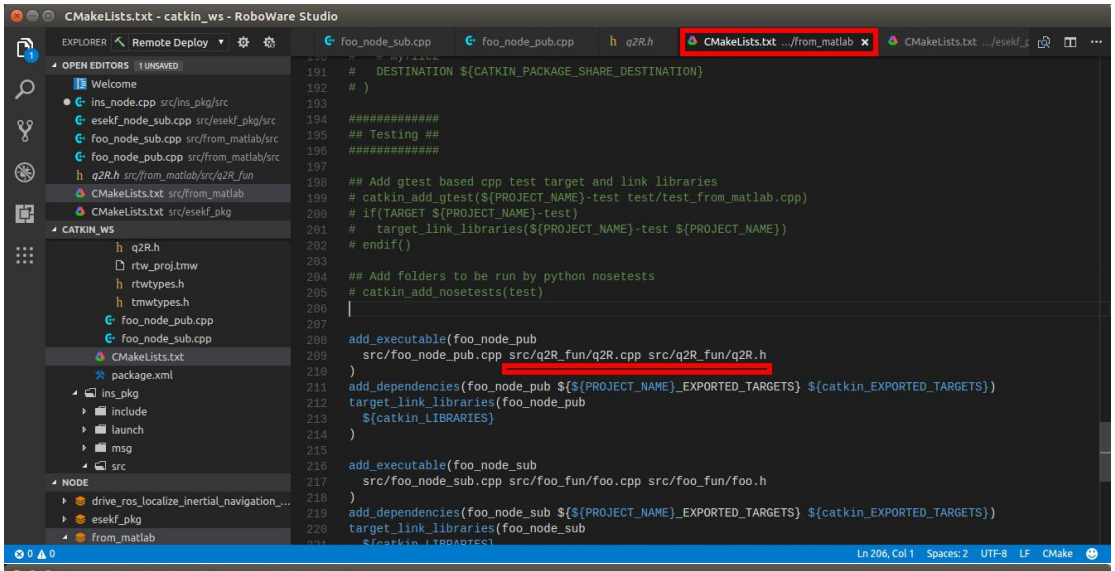

MATLAB Coder - q2R.prj
Finish Workflow
PACKAGE
Package generated outpu
Source Code Generated Successfully
You can now use the C code in your applications. Learn more
Project Summary
Functions
q2R.m
Project Type
MATLAB Coder
Project File
q2R.prj
Generated Output
C Code
/home/jackie/Desktop/resilientNav-matlab/codegen/lib/q2R
Example main Files
/home/jackie/Desktop/resilientNav-matlab/codegen/lib/q2R/examples
Reports
Code Generation Report
Back
MATLAB Coder - q2R.prj
Finish Workflow
PACKAGE
Source Code Generated Successfully
You can now use the C code in your applications. Learn more
Project Summary
Functions
q2R.m
Project Type
MATLAB Coder
Project File
q2R.prj
Generated Output
Packaging Successful
C Code
/home/jacki
en/lib/q2R
Example main Files
/home/jacki
en/lib/q2R/examples
Reports
Code Generation Report
Back

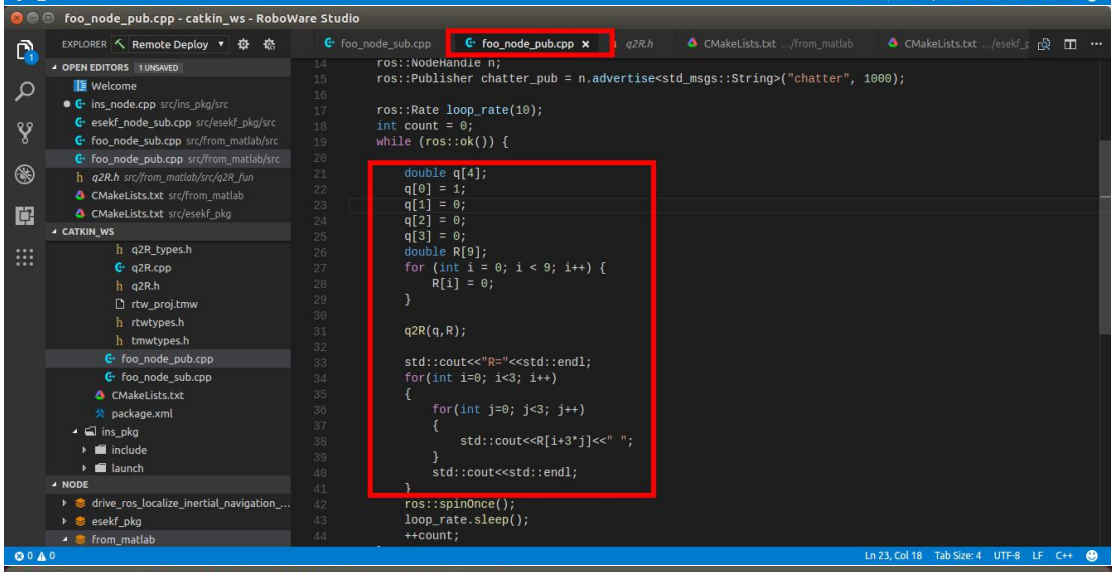
2.3. Use q2R_fun c++ code in ROS workspace

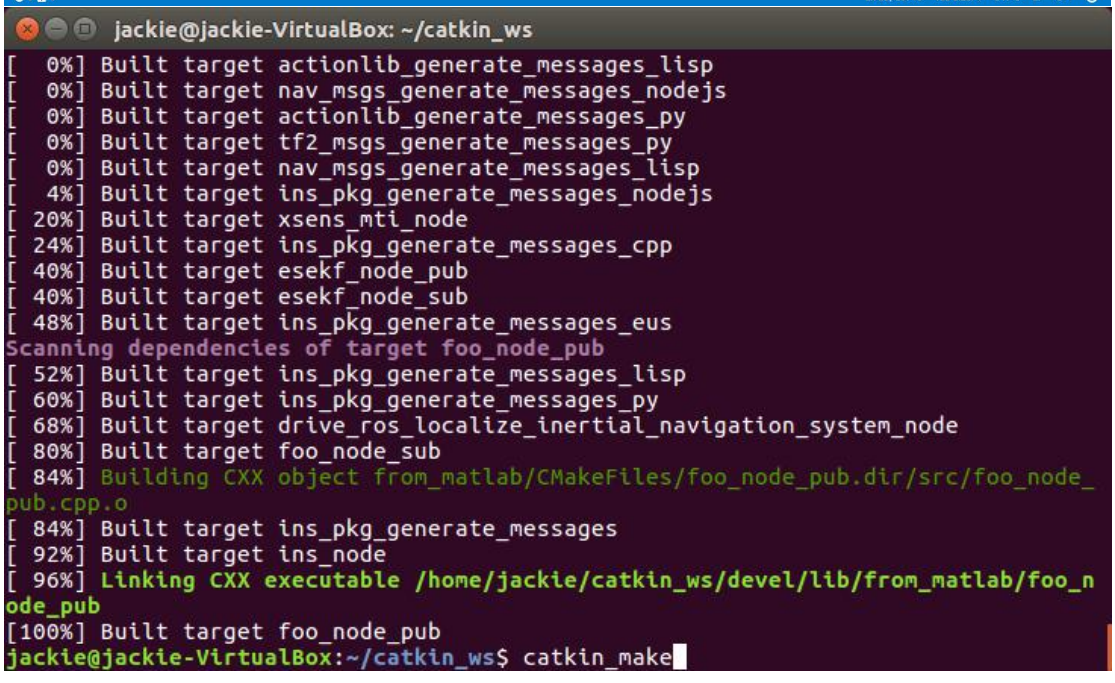


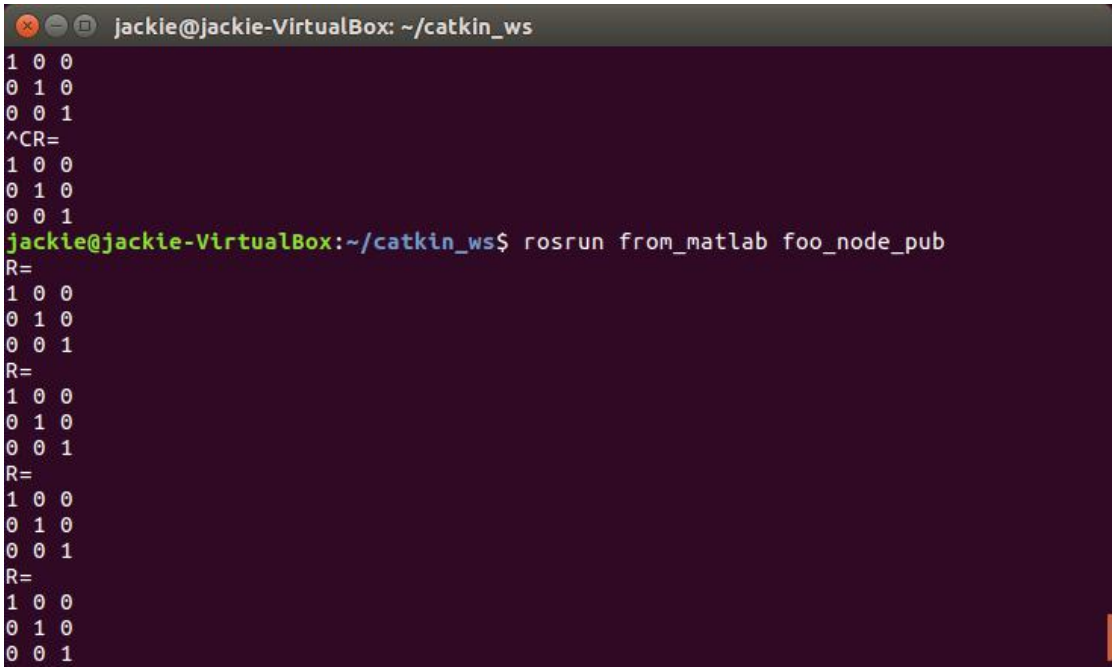

foo_node_sub.cpp - catkin_ws - RoboWare Studio
EXPLORER
Remote Deploy
OPEN EDITORS 1 UNSAVED
Welcome
ins_node.cpp src/ins_pkg/src
esekf_node_sub.cpp src/esekf_pkg/src
foo_node_sub.cpp src/from_matlab/src
CMakeLists.txt src/from_matlab
CMakeLists.txt src/esekf_pkg
CATKIN_WS
rtwtypes.h
tmwtypes.h
q2R_fun
buildInfo.mat
coder_array.h
q2R_initialize.cpp
q2R_initialize.h
q2R_terminate.cpp
q2R_terminate.h
q2R_types.h
q2R.cpp
q2R.h
rtw_proj.tmw
rtwtypes.h
tmwtypes.h
NODE
drive_ros_localize_inertial_navigation_system
esekf_pkg
from_matlab
esekf_node_sub.cpp
foo_node_sub.cpp
#include "foo_fun/foo.h"
#include "ros/ros.h"
#include "sensor_msgs/Imu.h"
#include "std_msgs/String.h"
void fun()
{
coder::array<double, 2> a;
a.set_size(3, 3);
for (int i = 0; i < 3; i++) {
for (int j = 0; j < 3; j++) {
a.at(i, j) = i;
}
}
coder::array<double, 2> b;
b.set_size(3, 3);
for (int i = 0; i < 3; i++) {
for (int j = 0; j < 3; j++) {
b.at(i, j) = i;
}
}
coder::array<double, 2> c;
foo(a, b, c);
std::cout << "c=" << std::endl;
for (int i = 0; i < 3; i++) {
for (int j = 0; j < 3; j++) {
std::cout << c.at(i, j) << "";
}
std::cout << std::endl;
Ln 19, Col 15
Tab Size: 4
UTF-8
LF
C++


foo_node_pub.cpp - catkin_ws - RoboWare Studio
EXPLORER
Remote Deploy
OPEN EDITORS 2 UNSAVED
Welcome
ins_node.cpp src/ins_pkg/src
esekf_node_sub.cpp src/esekf_pkg/src
foo_node_sub.cpp src/from_matlab/src
foo_node_pub.cpp src/from_matlab/src
CMakeLists.txt src/from_matlab
CMakeLists.txt src/esekf_pkg
CATKIN_WS
q2R_terminate.cpp
q2R_terminate.h
q2R_types.h
q2R.cpp
q2R.h
rtw_proj.tmw
rtwtypes.h
tmwtypes.h
foo_node_pub.cpp
foo_node_sub.cpp
CMakeLists.txt
package.xml
ins_pkg
include
NODE
drive_ros_localize_inertial_navigation_system
esekf_pkg
from_matlab
foo_node_sub.cpp
foo_node_pub.cpp
#include "q2R_fun/q2R.h"
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>
/**
* This tutorial demonstrates simple sending of messages
*/
int main(int argc, char* argv[])
{
/**
* The ros::init() function needs to see argc and ar
* any ROS arguments and name remapping that were pr
* For programmatic remappings you can use a differe
* remappings directly, but for most command-line pr
* the easiest way to do it. The third argument to
*
* You must call one of the versions of ros::init()
* part of the ROS system.
*/
ros::init(argc, argv, "foo_node");
/**
* NodeHandle is the main access point to communicat
* The first NodeHandle constructed will fully initi
* NodeHandle destructed will close down the node.
*/
ros::NodeHandle n;
Ln 7, Col 4
Tab Size: 4
UTF-8
LF
C++


CMakeLists.txt - catkin_ws - RoboWare Studio
EXPLORER Remote Deploy
OPEN EDITORS 1 UNSAVED
Welcome
ins_node.cpp src/ins_pkg/src
esekf_node_sub.cpp src/esekf_pkg/src
foo_node_sub.cpp src/from_matlab/src
foo_node_pub.cpp src/from_matlab/src
q2R.h src/from_matlab/src/q2R_fun
CMakeLists.txt src/from_matlab
CMakeLists.txt src/esekf_pkg
CATKIN_WS
q2R.h
rtw_proj.tmw
rtwtypes.h
tmwtypes.h
foo_node_pub.cpp
foo_node_sub.cpp
CMakeLists.txt
package.xml
ins_pkg
include
launch
msg
src
NODE
drive_ros_localize_inertial_navigation_...
esekf_pkg
from_matlab
foo_node_sub.cpp
foo_node_pub.cpp
q2R.h
CMakeLists.txt .../from_matlab
CMakeLists.txt .../esekf_p
DESTINATION ${CATKIN_PACKAGE_SHARE_DESTINATION}
)
############
Testing
############
Add gtest based cpp test target and link libraries
catkin_add_gtest(${PROJECT_NAME}-test test/test_from_matlab.cpp)
if(TARGET ${PROJECT_NAME}-test)
target_link_libraries(${PROJECT_NAME}-test ${PROJECT_NAME})
endif()
Add folders to be run by python nosetests
catkin_add_nosetests(test)
add_executable(foo_node_pub
src/foo_node_pub.cpp src/q2R_fun/q2R.cpp src/q2R_fun/q2R.h
)
add_dependencies(foo_node_pub ${${PROJECT_NAME}_EXPORTED_TARGETS} ${catkin_EXPORTED_TARGETS})
target_link_libraries(foo_node_pub
${catkin_LIBRARIES}
)
add_executable(foo_node_sub
src/foo_node_sub.cpp src/foo_fun/foo.cpp src/foo_fun/foo.h
)
add_dependencies(foo_node_sub ${${PROJECT_NAME}_EXPORTED_TARGETS} ${catkin_EXPORTED_TARGETS})
target_link_libraries(foo_node_sub
Ln 206, Col 1 Spaces: 2 UTF-8 LF CMake


foo_node_pub.cpp - catkin_ws - RoboWare Studio
EXPLORER Remote Deploy
OPEN EDITORS 1 UNSAVED
Welcome
ins_node.cpp src/ins_pkg/src
esekf_node_sub.cpp src/esekf_pkg/src
foo_node_sub.cpp src/from_matlab/src
foo_node_pub.cpp src/from_matlab/src
q2R.h src/from_matlab/src/q2R_fun
CMakeLists.txt src/from_matlab
CMakeLists.txt src/esekf_pkg
CATKIN_WS
q2R_types.h
q2R.cpp
q2R.h
rtw_proj.tmw
rtwtypes.h
tmwtypes.h
foo_node_pub.cpp
foo_node_sub.cpp
CMakeLists.txt
package.xml
ins_pkg
include
launch
NODE
drive_ros_localize_inertial_navigation_...
esekf_pkg
from_matlab
foo_node_sub.cpp
foo_node_pub.cpp
q2R.h
CMakeLists.txt .../from_matlab
CMakeLists.txt .../esekf_p
ros::NodeHandle n;
ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
ros::Rate loop_rate(10);
int count = 0;
while (ros::ok()) {
double q[4];
q[0] = 1;
q[1] = 0;
q[2] = 0;
q[3] = 0;
double R[9];
for (int i = 0; i < 9; i++) {
R[i] = 0;
}
q2R(q,R);
std::cout<<"R="<<std::endl;
for(int i=0; i<3; i++)
{
for(int j=0; j<3; j++)
{
std::cout<<R[i+3*j]<<" ";
}
std::cout<<std::endl;
}
ros::spinOnce();
loop_rate.sleep();
++count;
Ln 23, Col 18 Tab Size: 4 UTF-8 LF C++


jackie@jackie-VirtualBox: ~/catkin_ws
[0%] Built target actionlib_generate_messages_lisp
[0%] Built target nav_msgs_generate_messages_nodejs
[0%] Built target actionlib_generate_messages_py
[0%] Built target tf2_msgs_generate_messages_py
[0%] Built target nav_msgs_generate_messages_lisp
[4%] Built target ins_pkg_generate_messages_nodejs
[20%] Built target xsens_mti_node
[24%] Built target ins_pkg_generate_messages_cpp
[40%] Built target esekf_node_pub
[40%] Built target esekf_node_sub
[48%] Built target ins_pkg_generate_messages_eus
Scanning dependencies of target foo_node_pub
[52%] Built target ins_pkg_generate_messages_lisp
[60%] Built target ins_pkg_generate_messages_py
[68%] Built target drive_ros_localize_inertial_navigation_system_node
[80%] Built target foo_node_sub
[84%] Building CXX object from_matlab/CMakeFiles/foo_node_pub.dir/src/foo_node_pub.cpp.o
[84%] Built target ins_pkg_generate_messages
[92%] Built target ins_node
[96%] Linking CXX executable /home/jackie/catkin_ws/devel/lib/from_matlab/foo_node_pub
[100%] Built target foo_node_pub
jackie@jackie-VirtualBox:~/catkin_ws$ catkin_make



I succeed, and I may you succeed!