SC-635 Advanced Topics in Mobile Robotics

Experiment Module : Custom messages and trilateration

February 10, 2020

Systems and Control Engineering

Indian Institute of Technology Bombay

Overview

1. Custom messages

2. Assignment

Custom messages

In assignment 0: we implemented

- Name publisher with *String* message type
- Roll number publisher with *Numeric* message type

We created separate nodes to publish each type.

With custom messages we can encode specialized information such as collection of simple message types.
Some examples:

- Collection of robot poses in multi-agent setting.
- Sensor parameters

Custom message: Student_info

Lets create a custom message *student_info* with following fields

- Name
- Roll number

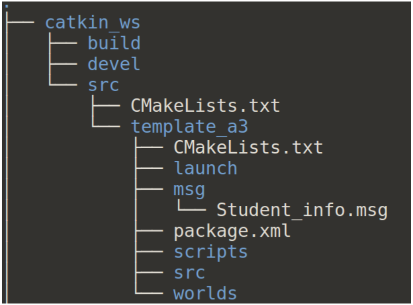
Custom messages are placed under the msg directory

Student_info.msg contents

```
string name
int64 roll_number
```

Directory structure

The directory structure should look as follows:



```
.
├── catkin_ws
│   ├── build
│   ├── devel
│   └── src
│       ├── CMakeLists.txt
│       └── template_a3
│           ├── CMakeLists.txt
│           ├── launch
│           ├── msg
│           │   └── Student_info.msg
│           ├── package.xml
│           ├── scripts
│           ├── src
│           └── worlds
```

Modifications 1

The custom message type we defined needs to be built with *catkin_make*

We notify *catkin_make* by modifying the contents of *package.xml* file

```
 1  <?xml version="1.0"?>
 2  <package format="2">
 3    <name>template_a3</name>
 4    <version>0.0.0</version>
 5    <description>The template_a3 package</description>
 6     ...
 7    <buildtool_depend>catkin</buildtool_depend>
 8    <build_depend>rospy</build_depend>
 9
10    <build_depend>message_generation</build_depend>
11    <exec_depend>message_runtime</exec_depend>
12
13      <build_export_depend>rospy</build_export_depend>
14    <exec_depend>rospy</exec_depend>
15
16    <export>
17      <!-- Other tools can request additional information be placed here -->
18    </export>
19  </package>
```

Line 10,11 are added to let the build system know that we want to use custom messages.

Modifications 2

We also need to modify CMakeLists.txt file

```
cmake_minimum_required(VERSION 2.8.3)
project(template_a3)
## Components
find_package(catkin REQUIRED COMPONENTS
  rospy
  roscpp
  std_msgs                          ## modified
  message_generation                ## modified
)
## Generate messages in the 'msg' folder
add_message_files(
    FILES
    Student_info.msg                ## modified
    Landmark.msg                    ## modified
    Trilateration.msg               ## modified
)
## Generate added messages and services with any dependencies listed here
generate_messages(
  DEPENDENCIES
  std_msgs                          ## modified
)
## catkin specific configuration
catkin_package(
 CATKIN_DEPENDS message_runtime     ##modified
)

## Specify additional locations of header files
include_directories(
  ${catkin_INCLUDE_DIRS}
)
```

Custom message: Publisher

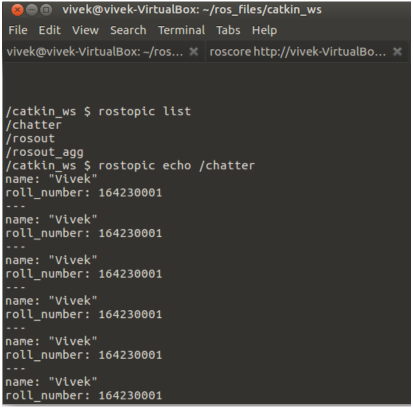
Lets use the custom message by writing a publisher:

```
1  #!/usr/bin/env python
2  import rospy
3  from template_a3.msg import Student_info
4
5  def talker():
6      pub = rospy.Publisher('chatter', Student_info, queue_size=10)
7      rospy.init_node('talker', anonymous=True)
8      rate = rospy.Rate(1)
9      while not rospy.is_shutdown():
10         t = Student_info("vivek", 164230001)
11         pub.publish(t)
12         rospy.loginfo("Sent␣a␣message!")
13         rate.sleep()
14
15 if __name__ == '__main__':
16     try:
17         talker()
18     except rospy.ROSInterruptException:
19         pass
```

In line 5, we are creating a custom message.

Custom message : Result

We expect to see the following results:



vivek@vivek-VirtualBox: ~/ros_files/catkin_ws
File Edit View Search Terminal Tabs Help
vivek@vivek-VirtualBox: ~/ros...
roscore http://vivek-VirtualBo...

```
/catkin_ws $ rostopic list
/chatter
/rosout
/rosout_agg
/catkin_ws $ rostopic echo /chatter
name: "Vivek"
roll_number: 164230001
---
name: "Vivek"
roll_number: 164230001
---
name: "Vivek"
roll_number: 164230001
---
name: "Vivek"
roll_number: 164230001
---
name: "Vivek"
roll_number: 164230001
---
name: "Vivek"
roll_number: 164230001
```

Assignment 3

The robot loads with pose (0, 0, 0). Three landmarks are placed at [7,7], [-7,-7], [7,-7], the distance and identity of the landmarks is being published under the topic name **/trilateration_data**.

- Calculate the position (x,y) of robot using the trilateration data. This part of code should be written inside the callback function (see line 30 of controller.py).
- Calculate the heading of the robot using two consecutive poses of the robot. This part will also go inside the callback function (see line 32 of controller.py)
- Track a circular trajectory with radius 5 meter centered at origin.
- Plot the waypoints and the tracked trajectory. Save figure with labels and title.
- Calculate the mean squared error for one complete traversal of the circular trajectory.
  - Sample 100 points along the tracked trajectory
  - For each robot pose $X_r = (x,y,\theta)$ find the closest point X_c on the circle $x^2 + y^2 = 5^2$
  - Calculate the distance to D_{error} and square it
  - Sum all D_{error}^2 terms and divide by 100 to obtain you MSE. Report this number in a file named RESULT.txt.

The template project is located at :

http://bit.ly/2tdpTemplateA4

Thank you