

Week 2 Assignment

You will be making a ros package in this assignment called week2 which will contain all the code for the various nodes you will write. You can use the command `catkin_create_pkg` for this (check out the tutorial [here](#)). All your code should go in the `scripts` folder inside the package.

1. Publisher - Subscriber Model

Write two nodes, one which publishes your first name on topic `/name` and another which publishes your surname on topic `/surname`

Now write a third node which will subscribe to the topics `/name` and `/surname` and publish your full name on the topic `/fullname`

E.g if your name is 'Vedant Shah', one node will publish 'Vedant' on topic `/name` and another will publish 'Shah' on topic `/surname` and the third node will publish 'Vedant Shah' on topic `/fullname`

2. Services

Write a server which accepts a message with either cartesian coordinates and converts them into polar or polar into cartesian. You will need to create a custom message for this. The message will have a third field which will determine what to convert to. For example the message can be of the form -

```
float32 first_coordinate  
float32 second_coordinate  
int32 to_polar
```

If the `to_polar` field is 1 then you need to convert to polar, otherwise to cartesian. This is an example of how you can construct the message. The server should return the converted coordinate in a similar message.

For a guide on services, look at Chapter 4 of Morgan Quigley. For more info on messages look at this [link](#) or Page 39 of Morgan Quigley. You may use the code you have written in last week's assignment.

3. Automation of Turtlebot

This question is divided into 3 interrelated parts. All three will work together to make the turtlebot move in a circular motion in Gazebo

3.1 Publisher

Write a publisher node which publishes the radius of movement for turtlebot on the topic `/radius`.

3.1 Service

Write a server which takes in radius and outputs the angular velocity required to move the turtlebot in a circle of that radius on a service named `compute_ang_vel`. Assume the linear velocity of the turtlebot will be set to 1.

3.1 Turtlebot

Write a node that subscribes to the `/radius` topic and is the client to the `compute_ang_vel` service. It will also be a publisher for `cmd_vel`. Whenever a radius message is received, the callback for the subscriber should execute the service proxy function to calculate the required angular velocity. Once the velocity is received from the server you need to publish it to `cmd_vel` to make turtlebot move. Make sure you set linear velocity to one.

3.2 Bonus

For bonus marks, you can try writing a publisher which makes the turtlebot move in the shape of the number *eight/infinity* in gazebo. You could create a [launchfile](#) for all the nodes.

4. Submission

Task 1 and 2 of the assignment are non evaluative. You are still encouraged to try them out for practice. To submit the results of the third task you will need to use rosbag. Rosbags are files which used to record a stream of messages on any topic. Follow [this](#) tutorial to learn how to record and play rosbag files. The bag file recorded should contain the messages published on the topics `/radius` and `/cmd_vel`. Record a bag file of around 20 seconds for the case $r=1$. Alongside this you should push your complete package (not workspace) to the repo created in the last

assignment. For this you can copy and paste the package directory into the directory with your git repo.