Challenges

Mini challenge 1

{Learn, Create, Innovate};

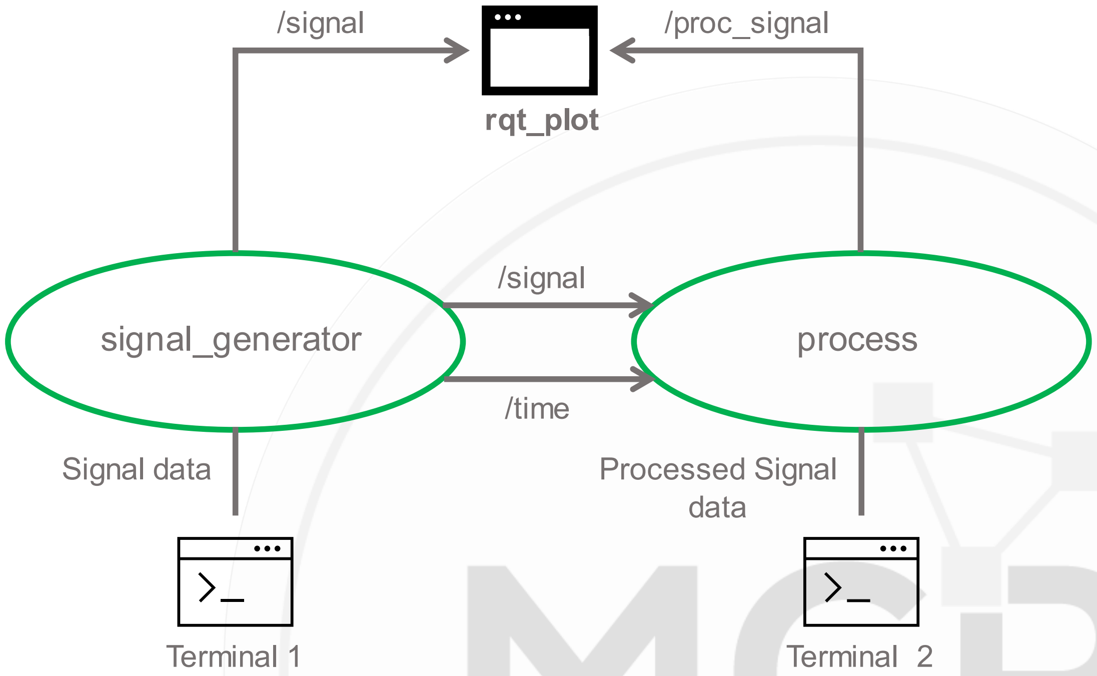
MCR^2

Manchester **Robotics**

Challenge 1

This activity is intended for the student to review the concepts introduced in this session.

- The activity consists of creating two nodes.
- The first node will act as a simple signal generator. That will generate a sinusoidal signal.
- The second node will act as a “process”, which will take the signal generated by the previous node and modify it, generating a “processed signal”.
- Both signals must be plotted using the rqt_plot.
- Different terminals must show the information of the generated signals (use it to debug).
- A launch file must be generated to launch both nodes, terminals and rqt_plot at the same time.



Challenge 1

Signal Generator Node

1. Create a new package called “courseworks” (std_msgs and rclpy)

2. Make a node called “signal_generator” to generate a sine wave with respect to time i.e., $y = f(t) = \sin(t)$.

    1. Publish the result using a `Float32` standard ROS message to a topic named “/signal”.

    2. Publish the time t into another topic called “/time” using the same type of message.

    3. Use a rate of 10 Hz for this node.

    4. Print the result on the terminal using `rospy.loginfo`

Process Node

1. Design a second node called “process” that subscribes to the “/signal” and “/time” topics.

2. Process the received signal as follows

    - Offset the received signal ($g(t) = f(t) + \alpha$) such that remains positive for all time $t \geq 0$,

    - Reduce the amplitude of the received signal in half

    - Add a phase shift to the received signal (as a user parameter or variable) to the original signal.

        ** For this exercise, this parameter can be hardcoded.

3. Use a rate of 10 Hz (you can ty different rates) for this node.

4. The result must be printed in the terminal.

5. The resultant signal must be published using `Float32` message into a topic called “/proc_signal”

Challenge 1

Launch File and Plotting

- Use the ROS tool “rqt_plot” to plot both signals.
  - ros2 run rqt_plot rqt_plot
- Make a Launch file to execute both nodes at the same time.
  - The Launch file must be able to open two different terminals (one for each node) to print the signals information.
  - The Launch file must open the rqt_plot and plot both signals in the same window.

Tips and tricks

The following tips and trick are not mandatory to be used, they are simply suggestions that could when designing the nodes.

- For this task, the student can use the NumPy python library, using the following command at the beginning of your code..

```
import numpy as np
```

- Students are encouraged to use trigonometric identities or other mathematical resources to modify the input signal.

$$\sin^2 x + \cos^2 x = 1\,,$$

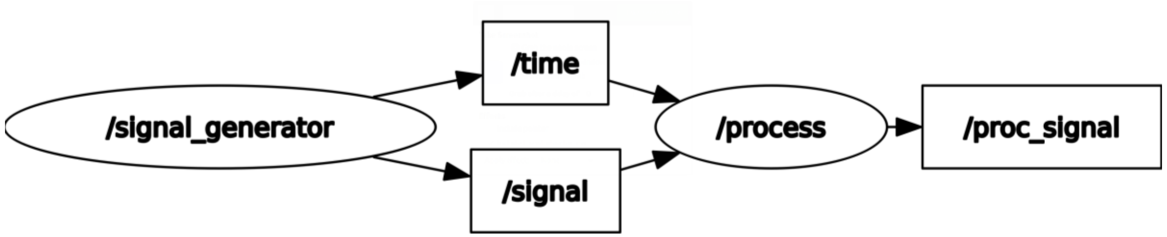
$$\sin(\alpha+\beta) = \sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta)\,, \text{... etc.}$$

- Wrap to pi function can also be used
- To use global variables inside callback functions you must declare them as “global” inside the callback function

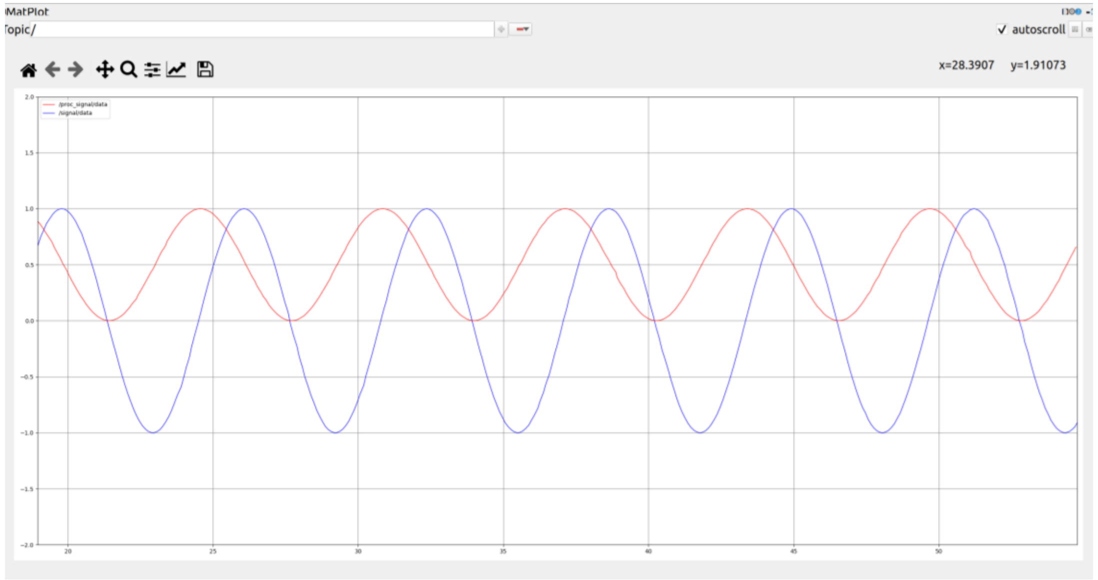
```
Foo = 0.0
def callback (parameter):
        global foo
```

Challenge 1 (expected results)

Nodes displayed using rqt_graph



Resultant signals displayed using rqt_plot



Rules

- This is challenge **not** a class. The students are encouraged to research, improve tune explain their algorithms by themselves.
- MCR2(Manchester Robotics) Reserves the right to answer a question if it is determined that the questions contains partially or totally an answer.
- The students are welcomed to ask only about the theoretical aspect of the classed.
- No remote control or any other form of human interaction with the simulator or ROS is allowed (except at the start when launching the files).
- It is **forbidden** to use any other internet libraires with the exception of standard libraires or NumPy.
- If in doubt about libraires please ask any teaching assistant.
- Improvements to the algorithms are encouraged and may be used as long as the students provide the reasons and a detailed explanation on the improvements.
- All the students must be respectful towards each other and abide by the previously defined rules.
- Manchester robotics reserves the right to provide any form of grading. Grading and grading methodology are done by the professor in charge of the unit.