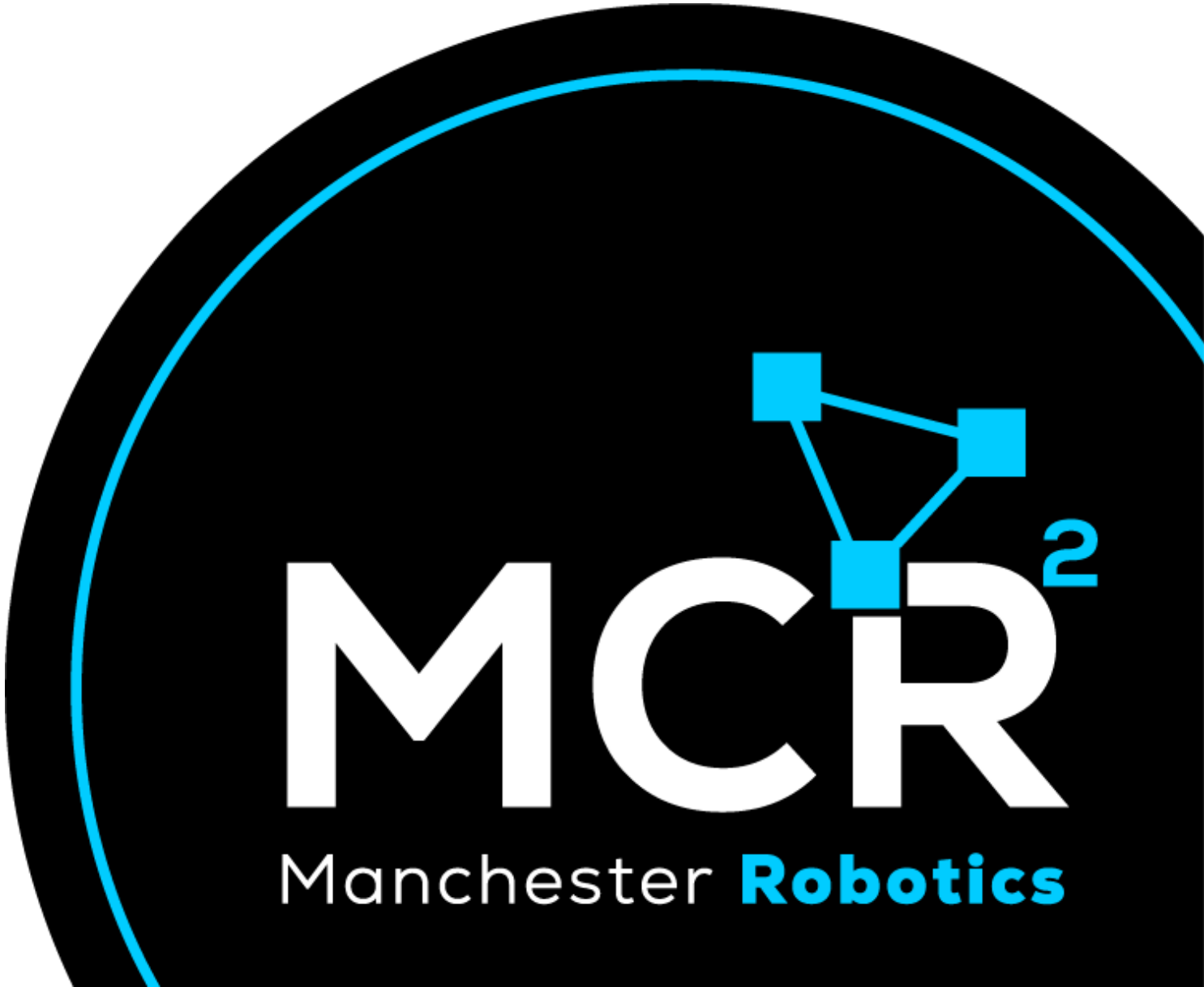
{Learn, Create, Innovate};

DC Motor Sim

Mini challenge 1

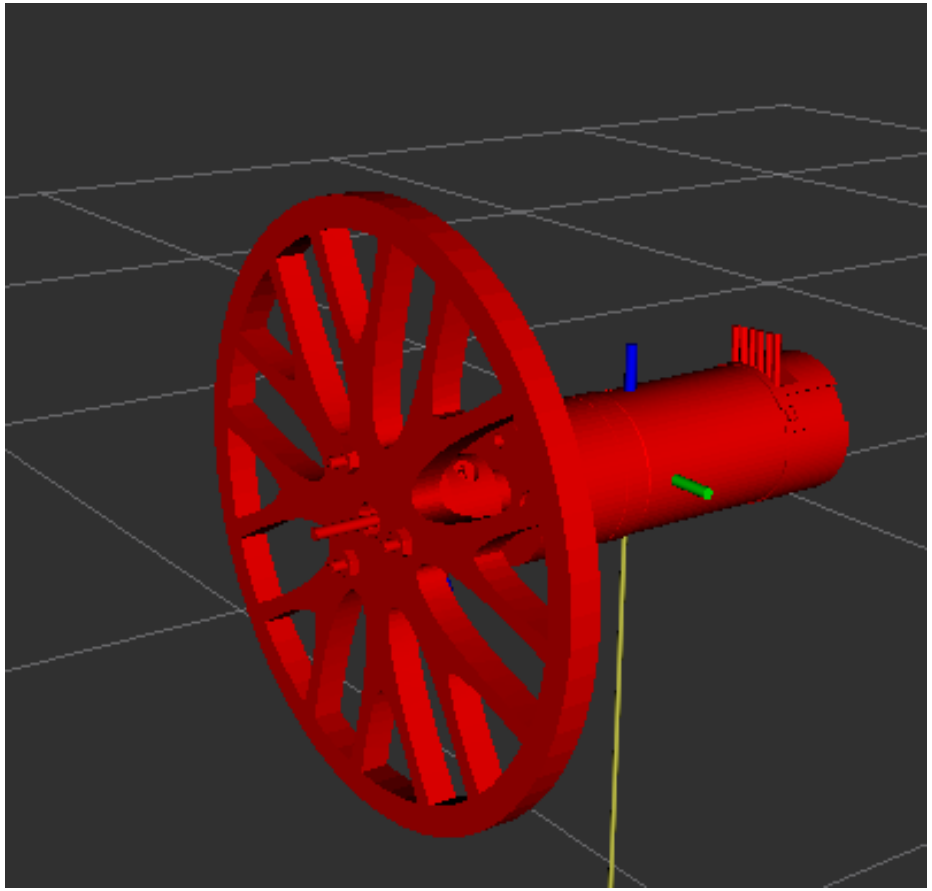


Introduction

Introduction

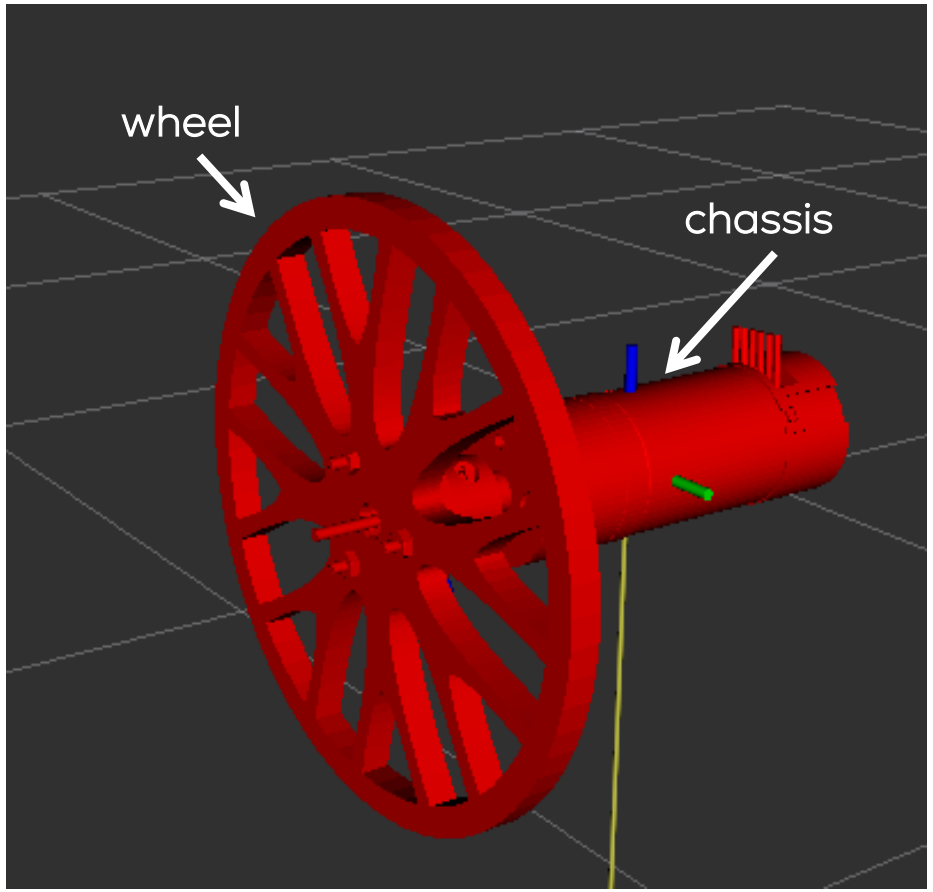
This mini-challenge is intended for the student to review the concepts introduced in the previous sessions.

- This activity consists of visualising the behaviour of a DC Motor using URDF files and joint state publishers in RVIZ.
- This activity employs a simple dynamical system simulation to dictate the motor's state behaviour.



Motor modelling

- The student must model their own DC motor using any CAD package (or use the files provided by MCR2).
    - The model must contain a robot chassis and a wheel to be attached to the motor shaft as shown in the figure.
- The student must use a URDF description file to describe the links and joints of the motor model.
- The motor chassis must be fixed to a “world” frame.



Motor modelling

- MCR2 provides the motor dynamical simulation in the package called “first_order_sys_sim”
- Make sure the following parameters are set for the dynamical simulation.

```
system_gain = 13.2
system_tau = 0.05
initial_conditions = 0.0
max_output = 13.0
min_input <= 0.05
sample_time >= 0.02
node_rate >= 100
```

- See the launch file “system.launch” for setting the parameters.

Caution!

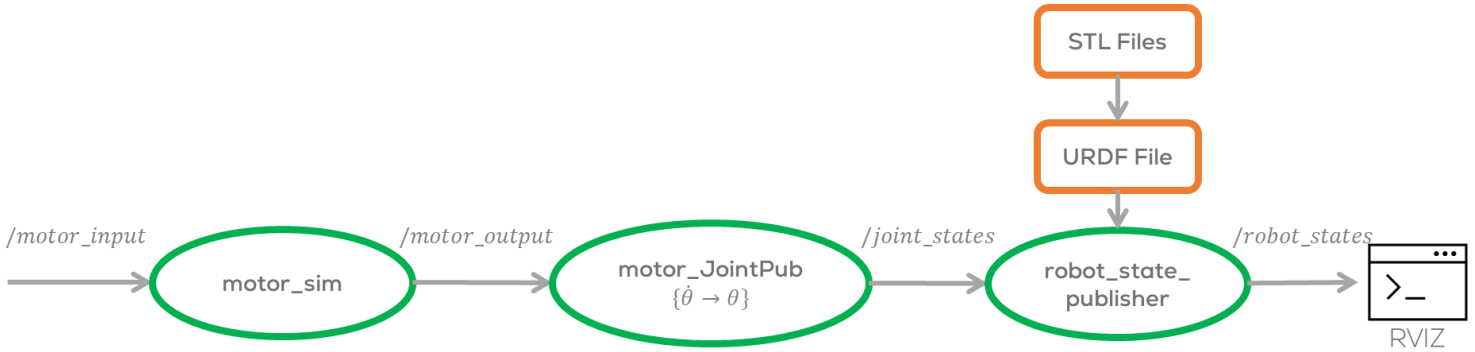
- The package “first_order_sys_sim” contains custom messages. Ensure you use them correctly when subscribing to or publishing messages to the topics.

```
System input message (/system_input):
float32 input #Input to the system
time stamp    #Time stamp to be sent to the system (ros Time message std_msgs)

System output message (/system_output):
float32 output #Output of the system
time stamp     #Time stamp of the system (ros Time message std_msgs)
string status  #Status of the motor “System Normal”, “System Stoppped”,
“System Max Output”
```

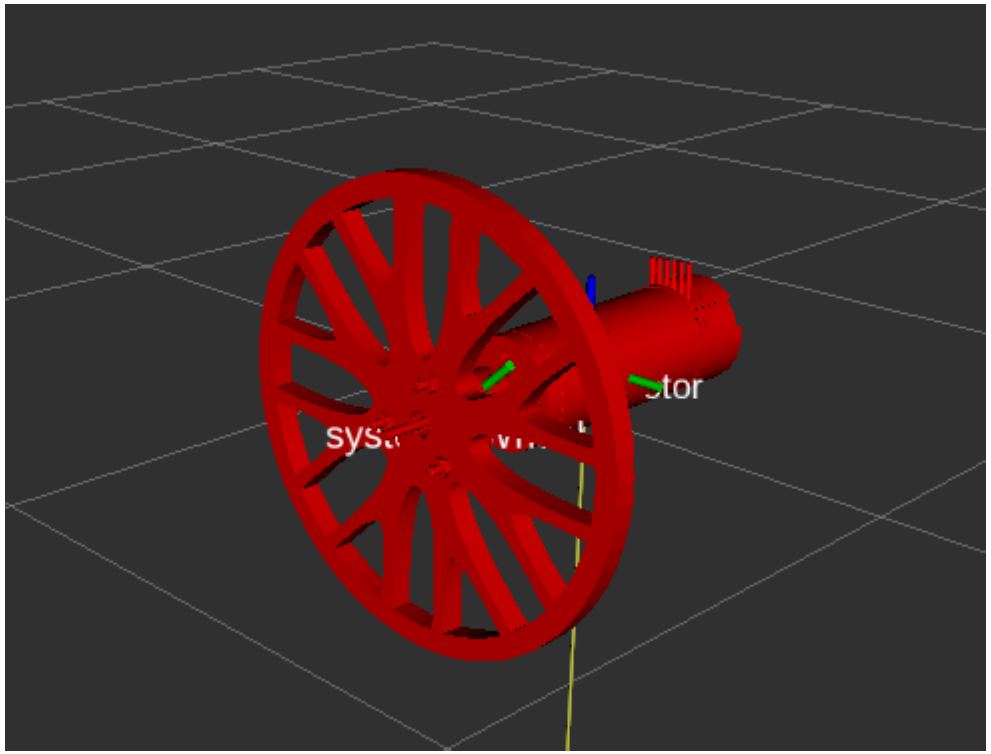
Motor modelling

- The student must develop their own joint state publisher for each one of the motor joints.
- The joint state publisher must read the speed of the motor and publish the required information to move the joint (angle in radians).
- The motor output speed is published in the topic "/system_output" (rad/s).
- The student must transform the motor speed into the positional angle of the joint before publishing the information to the joint.



Expected results

- The motor should spawn in the RVIZ world.
- The wheel must be able to rotate according to the system's dynamics.
- Different inputs to the system must be tested.



Challenge Extension

- Use the controller and the set point generator developed in the previous challenge (or develop new ones) to control the speed of the motor.
- The controller must take into consideration the custom messages used by the package “first_order_sys_sim”