{Learn, Create, Innovate};

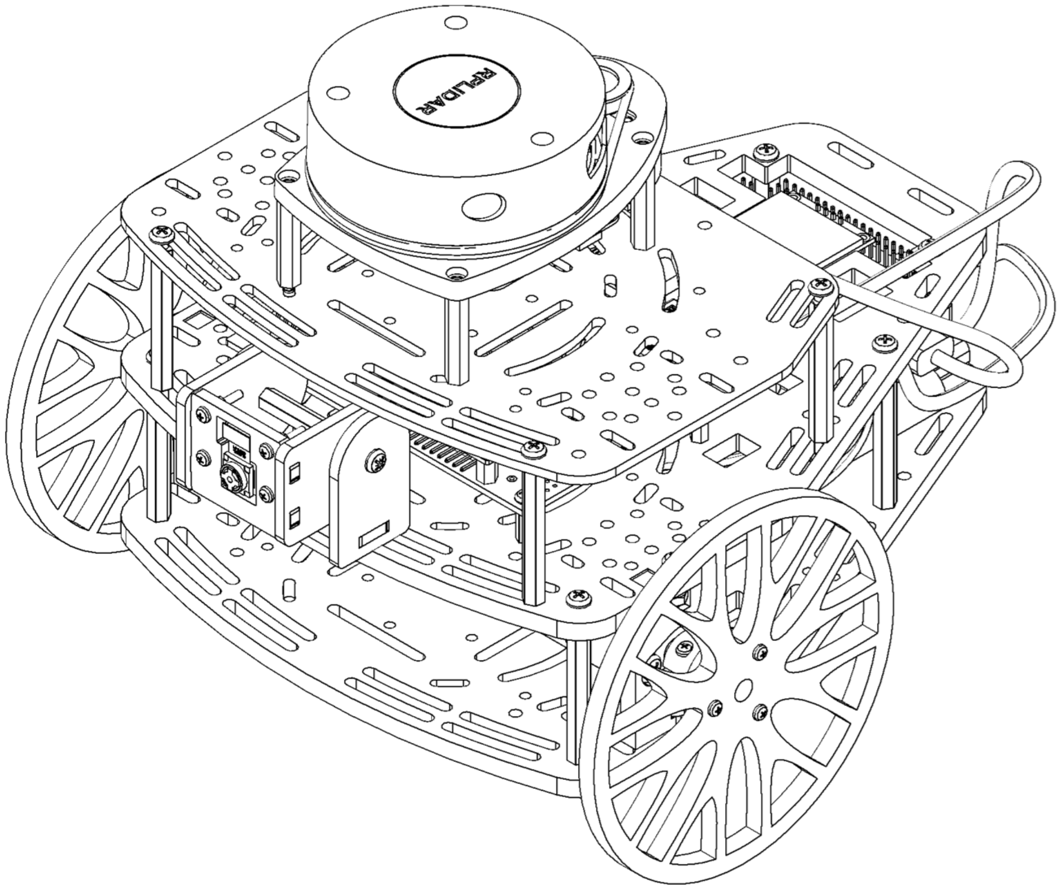
Challenges

Mini challenge 1

Mini challenge 1

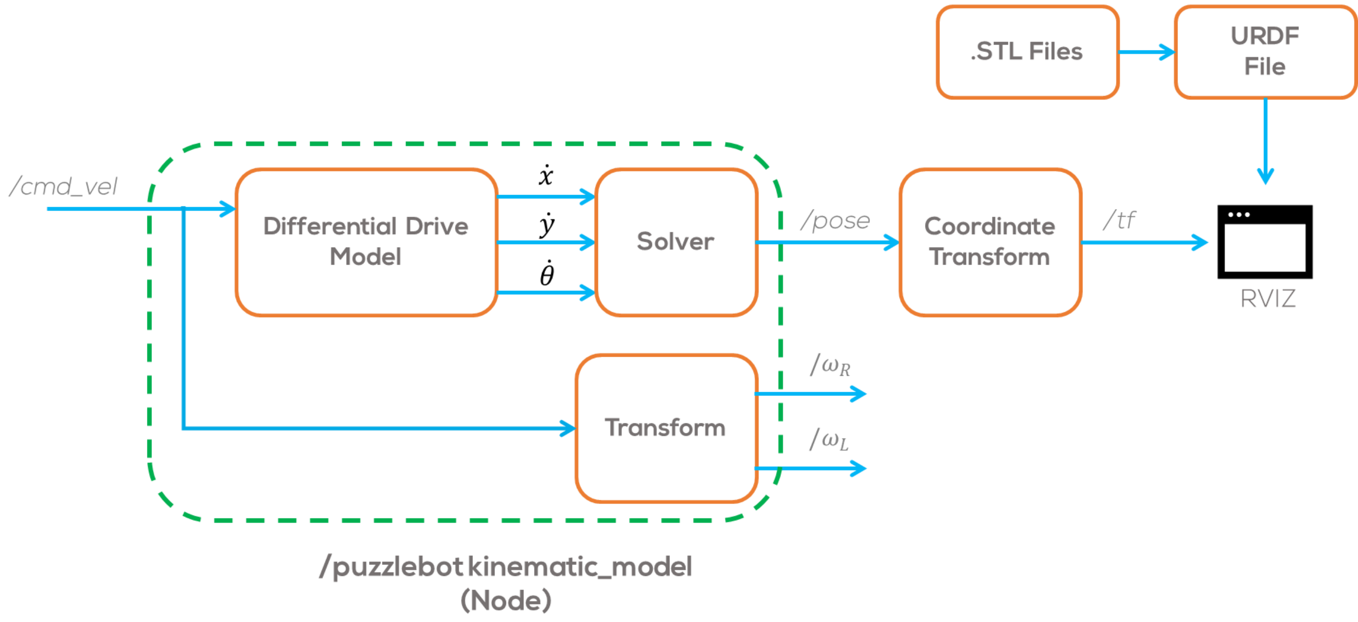
This challenge is intended for the student to review the concepts introduced in this week.

- The challenge is divided into two parts:
  - **Part 1:**
    - Development of a kinematic simulator for the Puzzlebot robotic platform, using the kinematic model of a nonholonomic robot.
  - **Part 2:**
    - Analysis of the uncertainties present in the real, gazebo simulation, and kinematic simulation of the Puzzlebot.



Mini challenge 1: Part 1

- This activity consists on creating a node that simulates a dynamical system.
- Simulate a nonholonomic robot (e.g., Puzzlebot) using ROS.
- The results of the simulation, must be visualised in RVIZ.
- The visualisation must be a 3D robot on RVIZ.
  - Use the "*.stl*" files provided by MCR2 to visualise the robot in RVIZ.

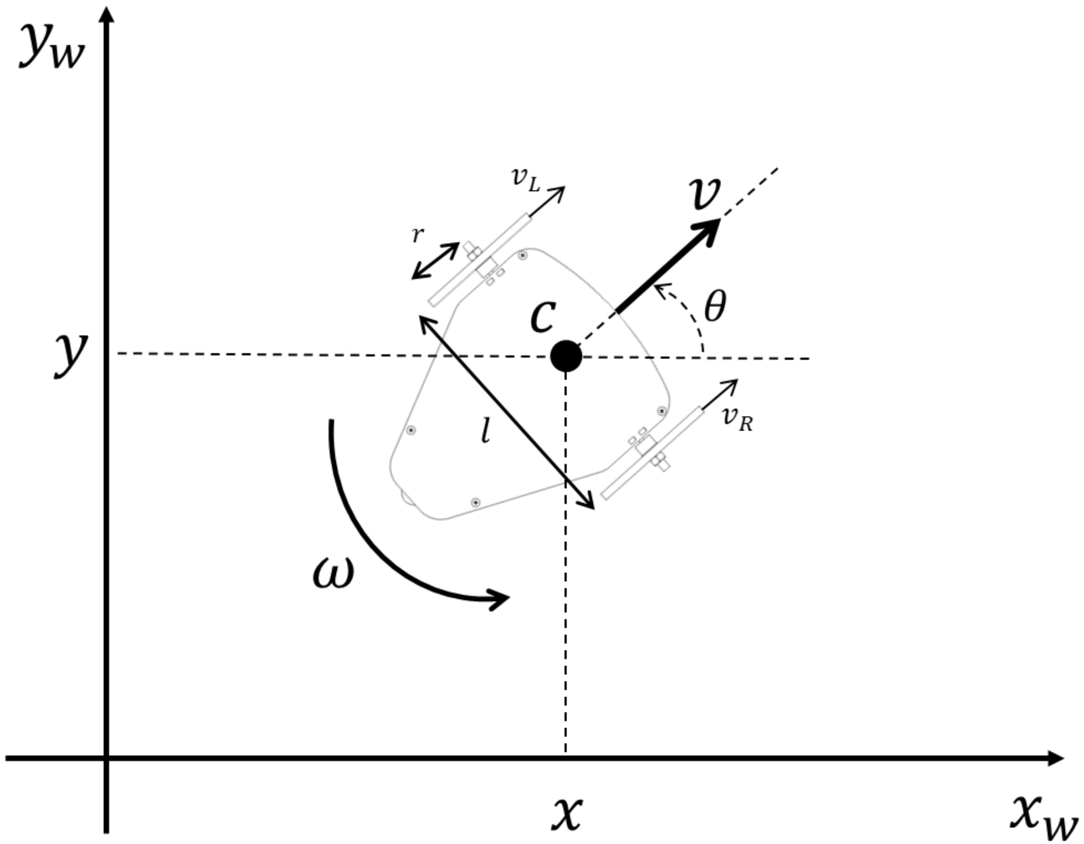


Mini challenge 1: Part 1

- The robot kinematical model is given by:

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = \omega \end{cases}$$

- The name of the package for the simulated node must be *"puzzlebot_sim".*
- The name of the package for the RVIZ configuration files must be *"puzzlebot_rviz"*
- For the pose of the robot use the "*PoseStamped*" Message.
  - The pose topic must be named *"/pose_sim"*
- For the input to the robot use ""*Twist*" message
  - The topic for commanding the robot must be named *"/cmd_vel"*



Mini challenge 1: Part 1

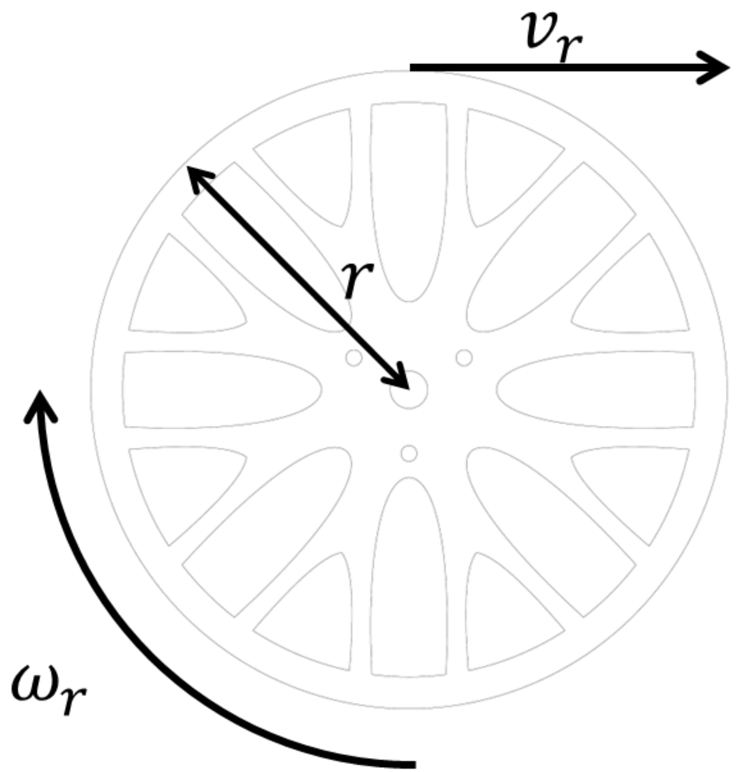
- The wheel speed must also be published using a *"Float 32" std_msg.*
- The topics for each wheel must be *"/wr"* and *"/wl"*, for the left and right wheels respectively.

Remember:

$$v = \frac{v_R + v_L}{2} = r\frac{\omega_R + \omega_L}{2}$$

$$\omega = \frac{v_R - v_L}{l} = r\frac{\omega_R - \omega_L}{l}$$

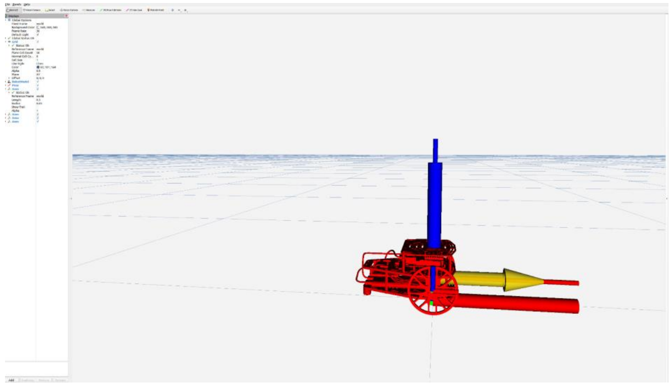
- As an **optional** task, the students are encouraged to make the wheels rotate with respect to the base of the robot.

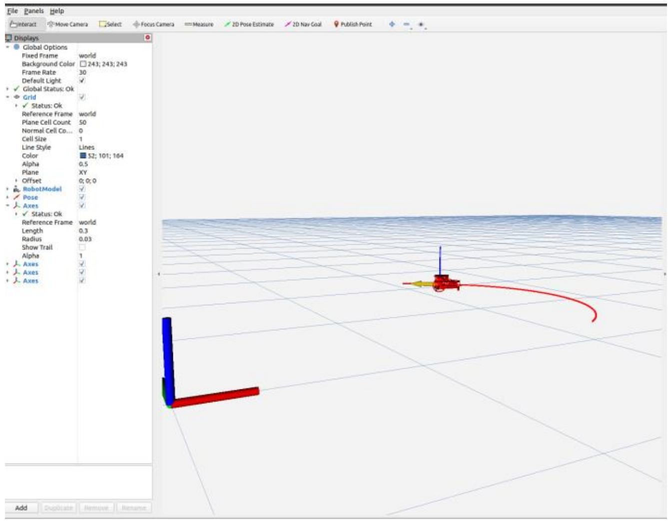


Mini challenge 1: Part 1

- The student is allowed to use "*tf*" coordinate transforms or "*URDF*" files for the simulation, or a combination of both.
- The student must define the coordinate frames and transformations to be used.
- The students must define the required launch files for this activity.
- The simulation must be tested under different conditions, i.e., different speeds.
- The students must define a correct sampling time for the simulation .
- The students must solve the differential equations using numerical methods.
- The usage of any library is strictly **forbidden**.

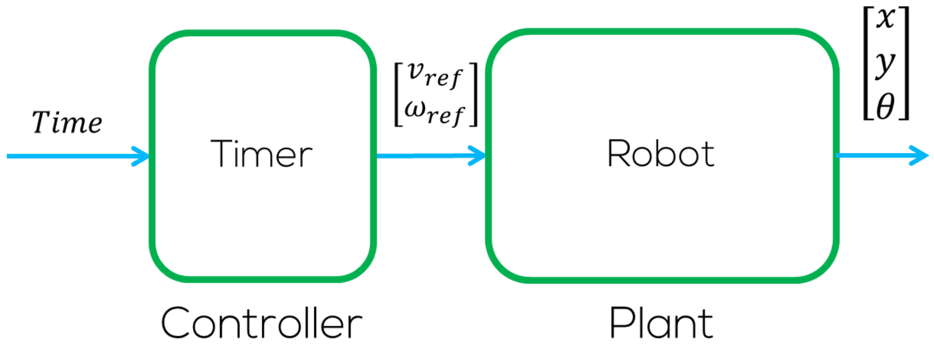
Expected results:





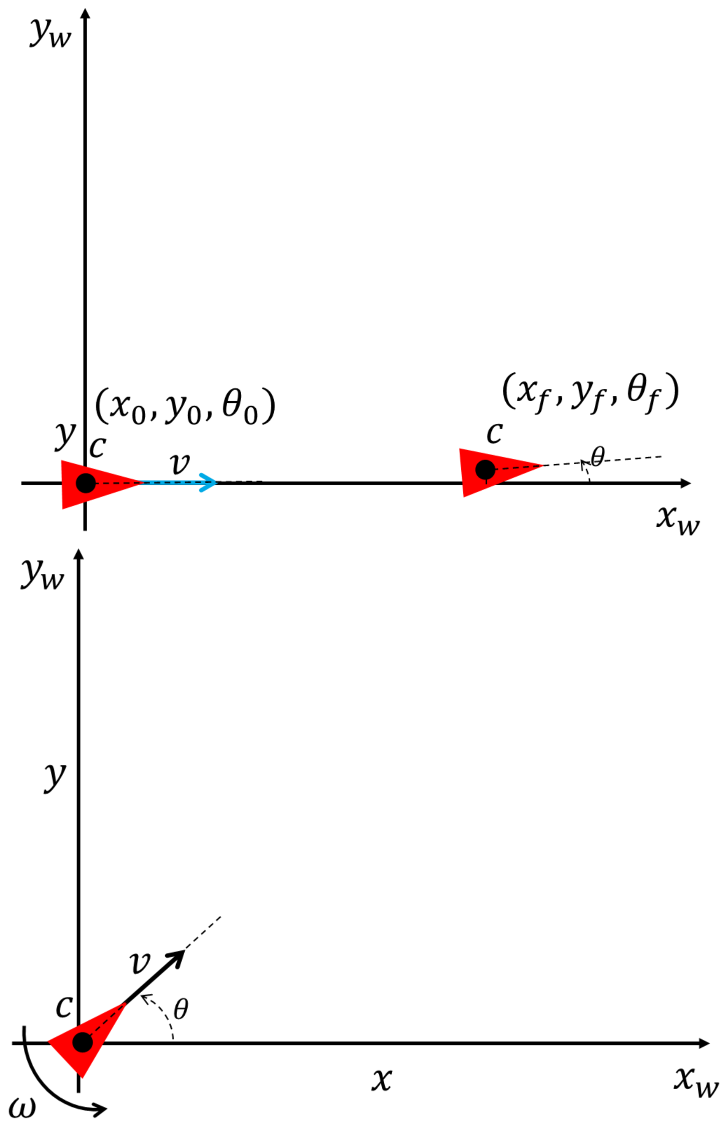
Mini challenge 1: Part 2

- This part of the challenge consists on moving the robot in different directions for a specified period of time, and analyse the position behaviour.
- Using an open loop control strategy, based on a specific time, move the robot in different directions.



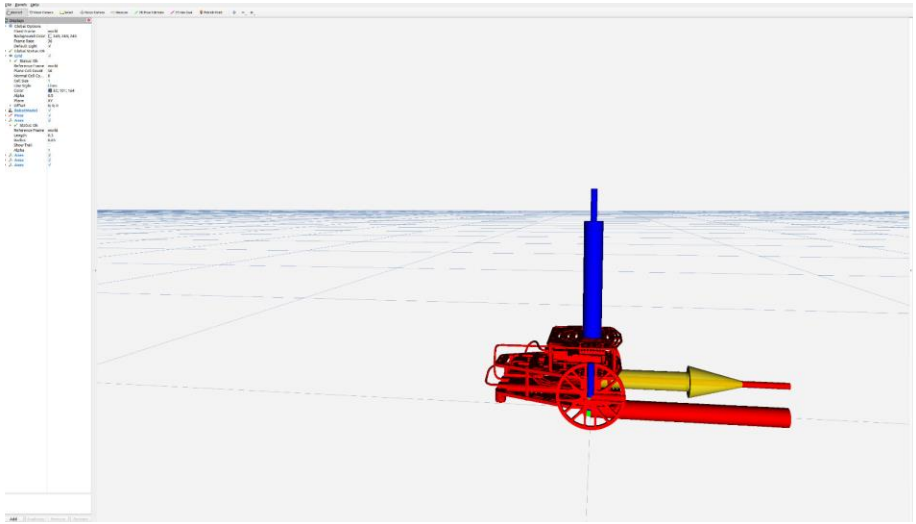
Mini challenge 1: Part 2

- Robot movement instructions:
  - Move the Puzzlebot in a **straight line,** with a fixed velocity (open loop) from an initial point A, for a specific time.
    - Record the position data (mark the spot of the initial pose).
    - Repeat the experiment 15 to 20 times and record the position data.
    - Return the robot to the initial position before repeating the experiment.
  - **Turn** the robot from an initial angle, with a fixed angular velocity for a specific time.
    - Record the position data (mark the spot of the initial pose).
    - Repeat the experiment 15 to 20 times.
    - Return the robot to the initial angle before repeating the experiment.



Mini challenge 1: Part 2

- Repeat the previous experiments for the real, gazebo model and kinematic model.
- Make a simple statistical analysis for each robot (e.g., histogram), and analyse the mean and spread of the results (variance).
- Make more tests if necessary.
- Analyse the correlation between the different movements. In other words, how the movement in one direction. affects the others.



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.