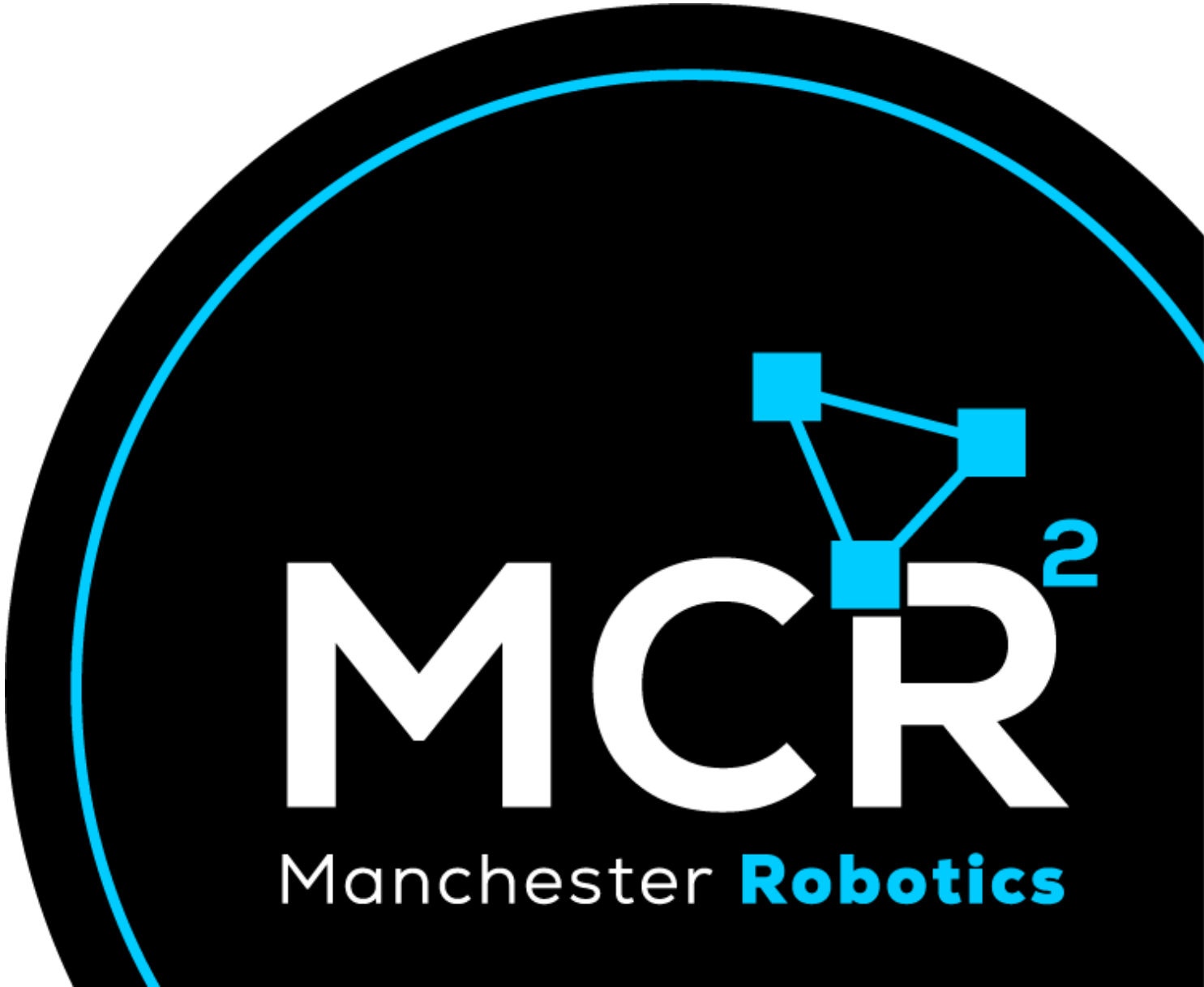
{Learn, Create, Innovate};

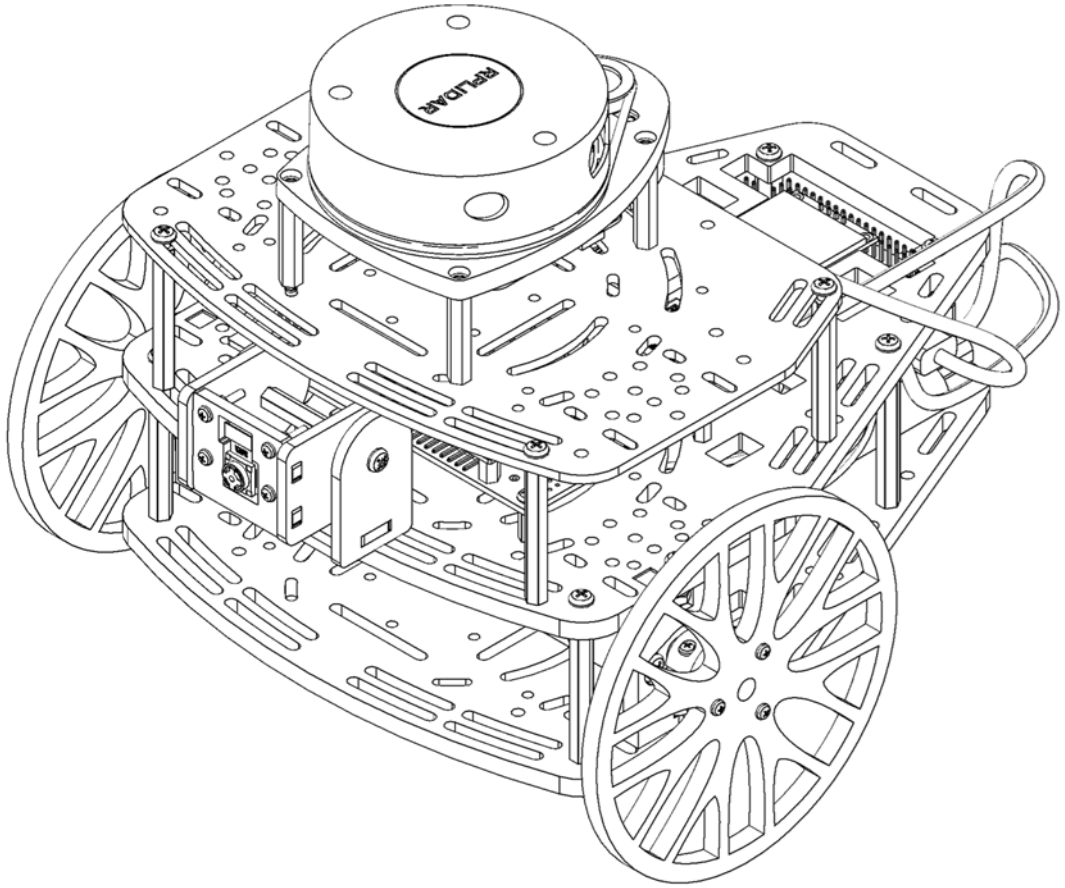
Challenges

Mini challenge 2



Mini challenge 2

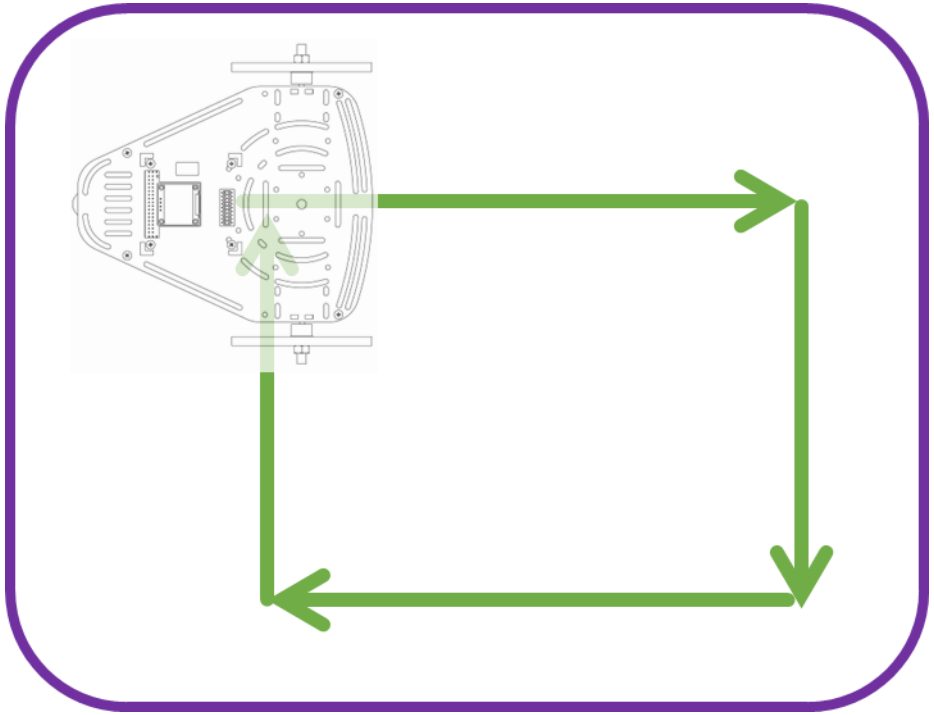
- This challenge is intended for the student to review the concepts introduced in this week.
- This challenge aims to show the behaviour of the closed loop control in robotics.
- This challenge will be divided in different sections.



Mini challenge 2

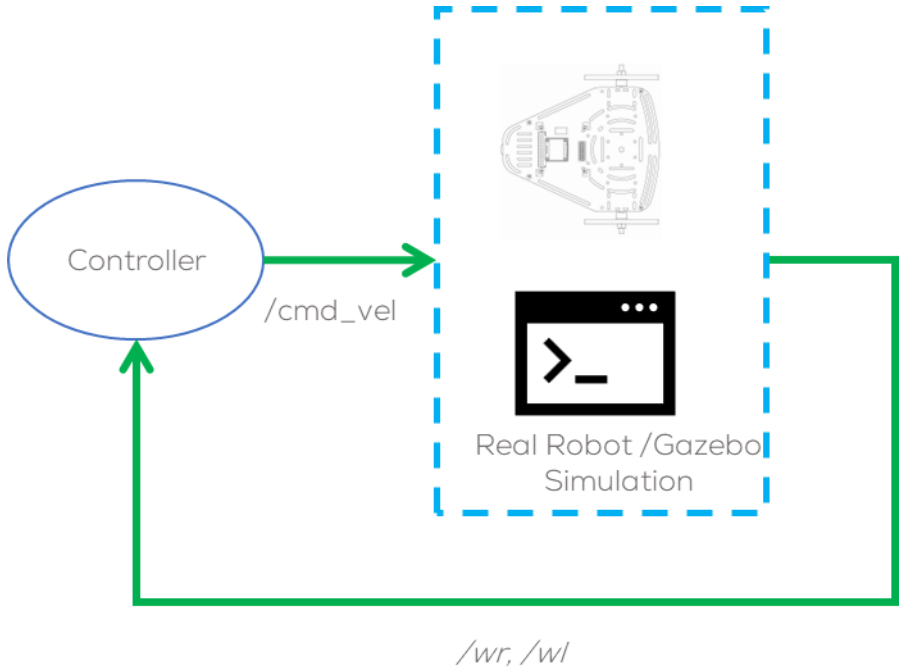
Mini challenge 2 - Part 1

- The task consists of using a PID (P, PI or PID) control to move the robot to different positions in the space.
- This task must be implemented using the Gazebo Puzzlebot Simulator and the real robot.
- No template is given for this activity (is a challenge) except for the Gazebo Simulator. The student must create each own nodes and packages in ROS (VM or dual booting is allowed).
- The robot must follow a square shape path of side length 2m.
- The initial pose of the robot must be $[0,0,0]^T$ for $[x_r, y_r, \theta_r]^T$ respectively.



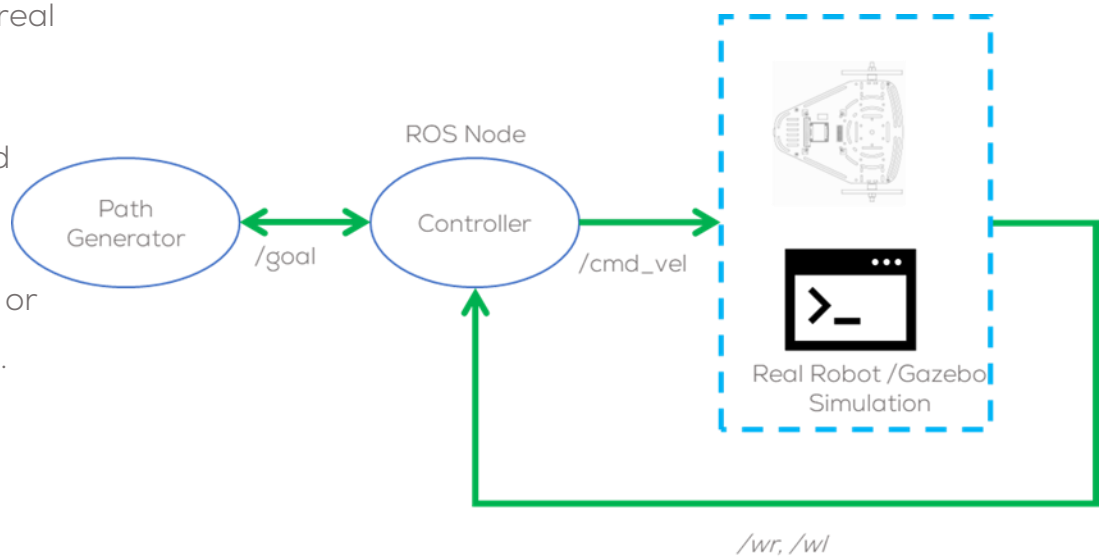
Mini challenge 2

- The closed loop controller must be **robust.**
  - The student must define what is robustness and implement strategies to achieve it with the controller.
- The controller must be tunned properly.
- The controller must take into consideration, perturbation, nonlinearities and noise.
- It is encouraged, but not required, for the student to use a config file or a parameter in the launch file to establish the goal targets such that they can be changed outside the code (not hardcoded).



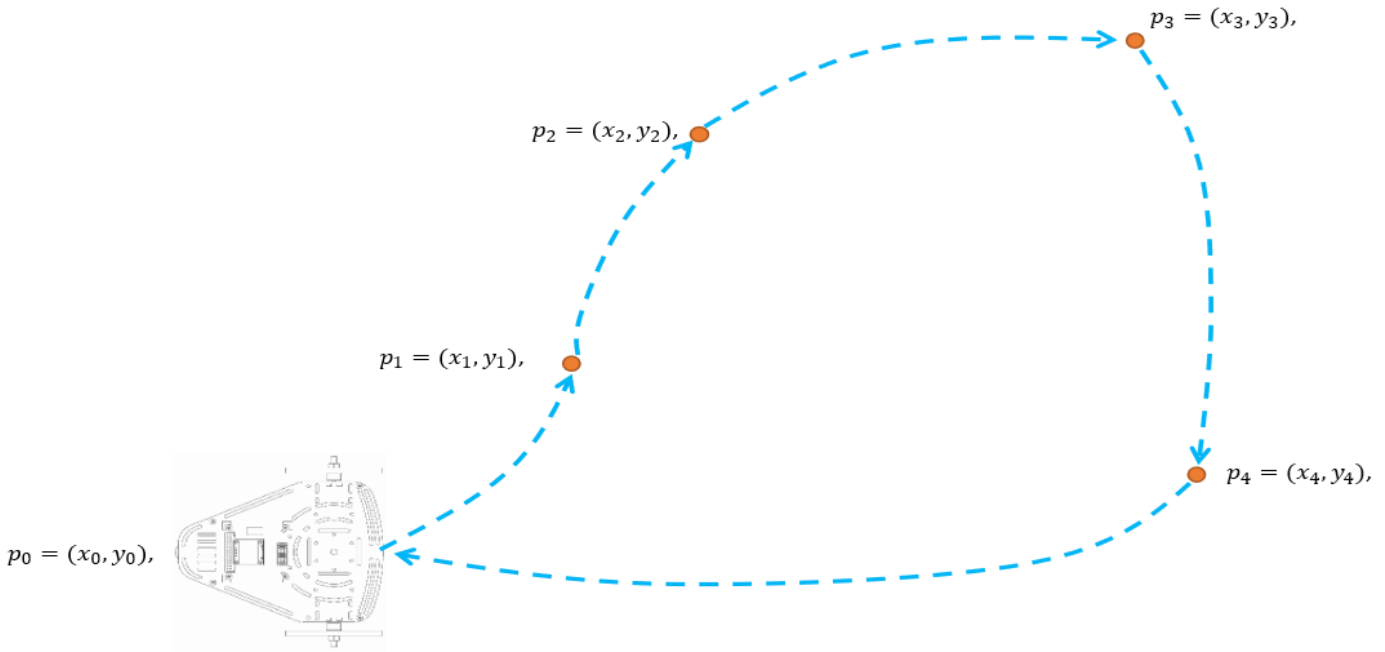
Mini challenge 2

- Mini challenge 2 - Part 2
- The task consists of using a PID (P, PI or PID) control to move the robot to different positions in the space.
- This task must be implemented using the Gazebo Puzzlebot Simulator and the real robot.
- The student must create a path generator node, that publishes the current and next goal once the robot completes the current goal.
- The path generator node can take other decisions (change the controller gains or control methodology) if deem necessary by the student and the circumstances.
- The node must let the user know if the point is reachable according to the dynamical behaviour of the mobile robot and controller (if applies).
- The number of targets $\boldsymbol{x}$ can be specified by each student (minimum required: 3 different positions, excluding the initial position).
- The initial pose of the robot must be $[0,0,0]^T$ for $[x_r, y_r, \theta_r]^T$ respectively.



Mini challenge 2

- The closed loop controller must be **robust.**
  - The student must define what is robustness and implement strategies to achieve it with the controller.
  - The controller must take into consideration, perturbation, nonlinearities and noise.
  - The controller must be tunned properly.
- It is encouraged, but not required, for the student to use a config file or a parameter in the launch file to establish the goal targets such that they can be changed outside the code (not hardcoded).
- The message for the topic /goals must be a custom message based on the *geometry_msgs "pose"*.



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.