{Learn, Create, Innovate};

Challenges

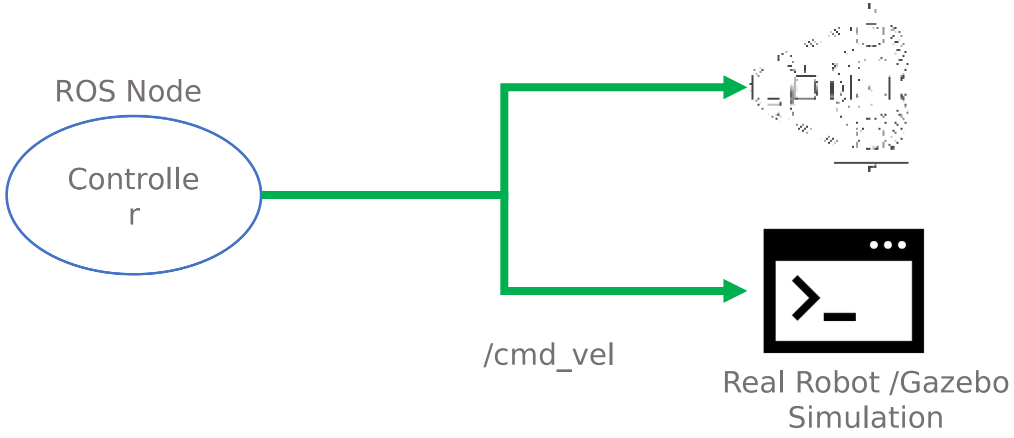
Mini challenge 1

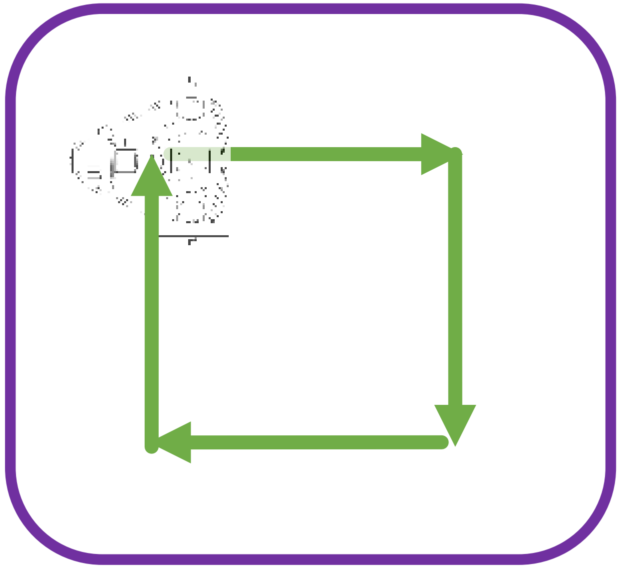
Manchester **Robotics**

Mini challenge 1

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

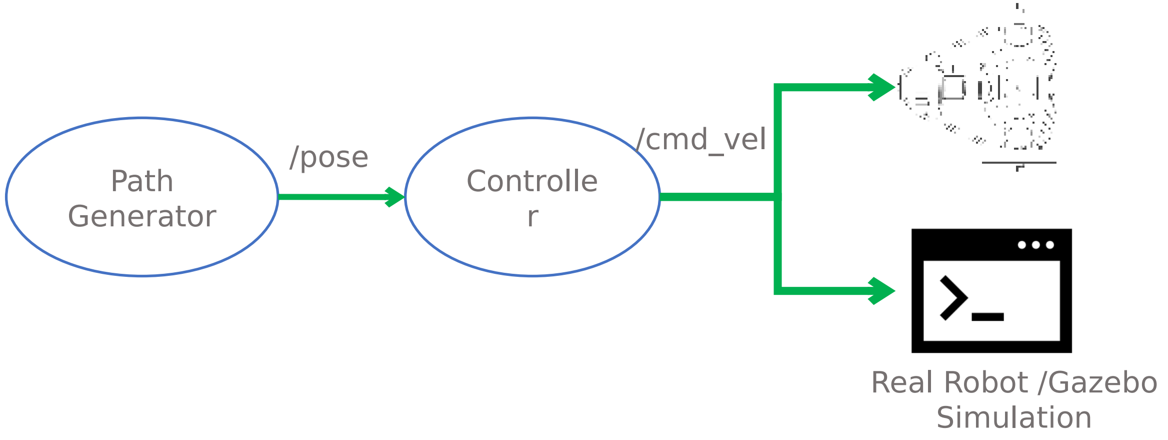
- Create a node to drive the gazebo simulated robot in a square path of a side length 2 m.
- Use the same node to move the real robot in a square of side length 2 m.
- The open loop controller must be **robust.**
    - The student must define what is robustness and implement strategies to achieve it with the controller.
- The controller must be, auto-tunned
    - The user selects the speed or the time to finish the path (square figure) and the controller must estimate either the velocities, acceleration or time required.
- The controller must take into consideration, perturbation, nonlinearities and noise.

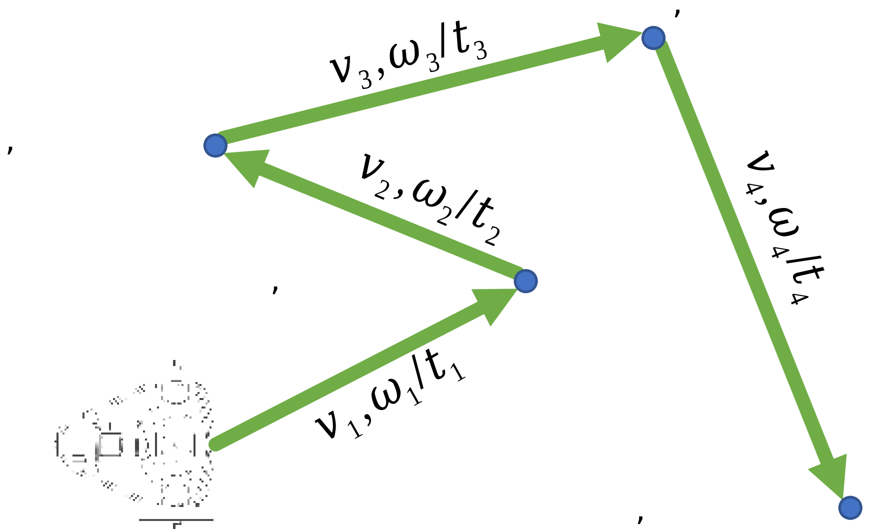




Mini challenge 1

- Create a node that generates different paths according to a user.
  - The path must be defined in the parameter files by the user.
  - The path must be defined by different points , velocities of the robot, or a time (depend on the user).
    - For each point, the node must estimate the linear and rotational speeds in case a time is given by the user or estimate the time in case the velocities are provided.
  - The node must let the user know if the point is reachable according to the dynamical behaviour of the mobile robot and the parameters that were input by the user .
  - The student must define what is robustness for this case, and the controller must take that into consideration.
  - The message for the topic /pose must be a custom message based on the *geometry_msgs “pose”* with an added field for *velocities* or *time*.





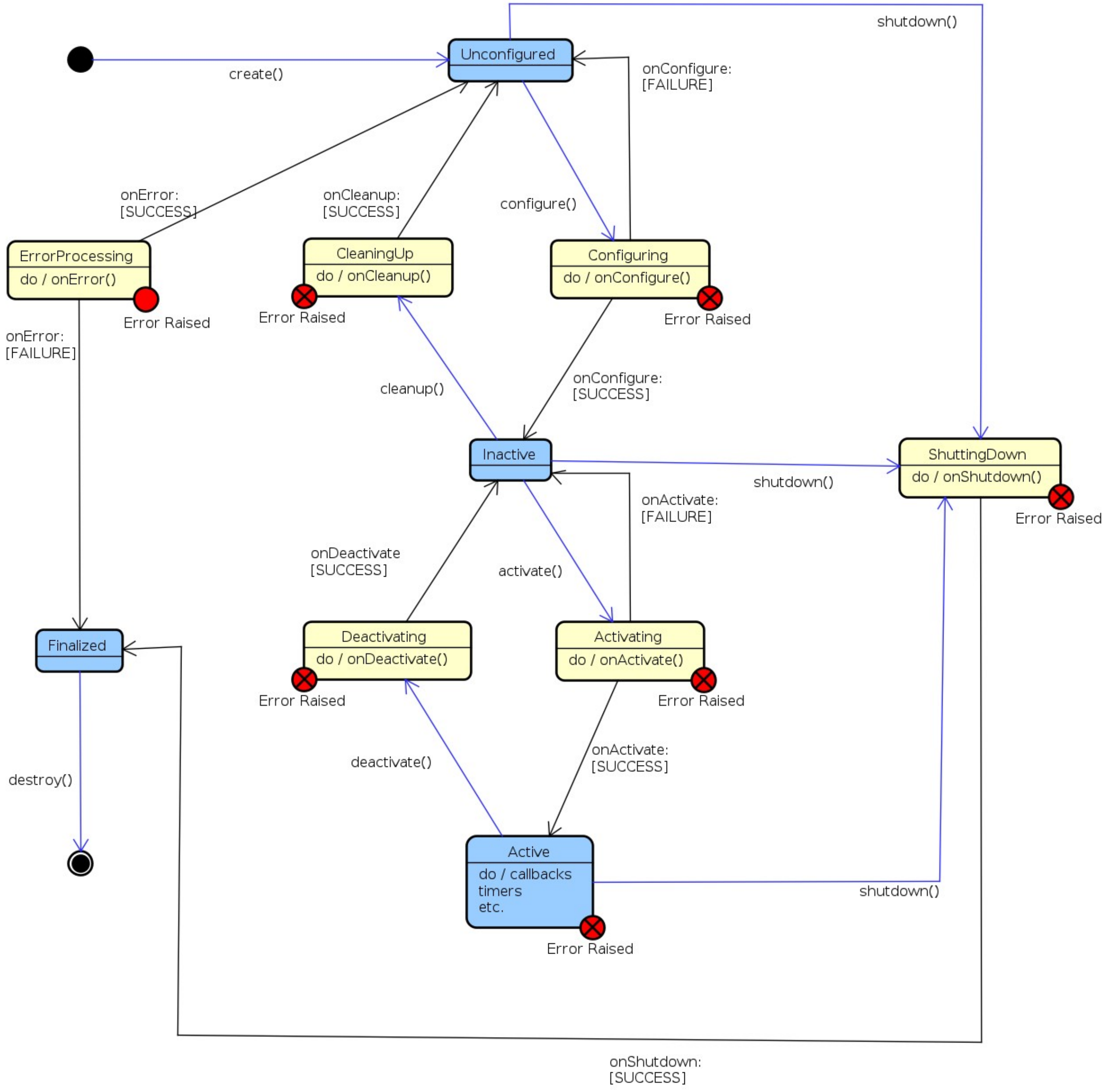
State Machines

Definition

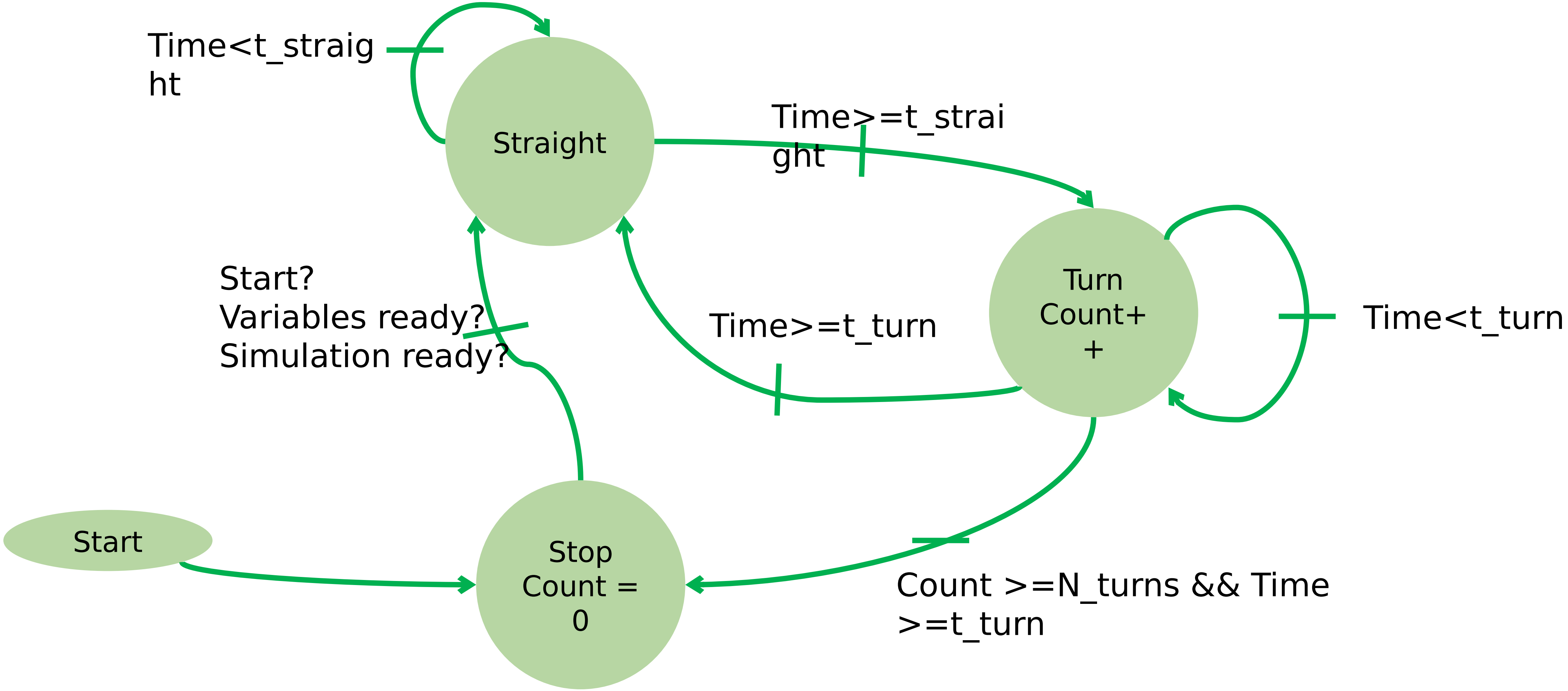
- A state machine is a computational model that transitions between predefined states based on inputs or conditions.
- **Finite State Machines (FSM)** - Fixed number of states.

Why use state machines?

- Structured way to control robotic behaviour.
- Ensures modular, scalable, and readable code.



Tips


Time<t_straig
ht
Straight
Time>=t_strai
ght
Turn
Count+
+
Time<t_turn
Start?
Variables ready?
Simulation ready?
Time>=t_turn
Start
Stop
Count =
0
Count >=N_turns && Time
>=t_turn

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.