Objective and Motivation

Welcome to Task 2 of Hologlyph Bots theme of eYRC 2023-24 🌌!!

In task 0...

- we setup Ubuntu and ROS

In Task 1A...

- we understood the working of ROS
- learnt about ROS Nodes, Topics, Services, Workspaces, Packages, etc...

And in task 1B...

- we explored Gazebo-ROS, and learnt about robot definition using URDF by creating and spawning a Hologlyph Bot in Gazebo.
- we manually controlled a “high-level model” of a generic holonomic robot (using Planar Move Plugin).
- implement some simple controller on the “high-level model” of a holonomic drive robot
  - Like implementing 3 P controllers for linear velocity in X, Y and angular velocity in Z axis.

In task 2A, we shall repeat Task 1B! But...

- with more realistic localisation:
  - Specifically: using **overhead camera** and **aruco marker** instead of simulator transforms
- with more realistic model of the holonomic drive.
  - Specifically: **three omni wheel robot** with **input (v1, v2, v3)** for three wheel velocities. Instead of any generic holonomic drive robot with inputs (Vx, Vy, W)

Recapping the Bigger Picture (Motivation)

The eventual goal of this theme will be to **draw shapes using three holonomic drive robots**.

Before that we will need to understand and implement the following building blocks to control a single holonomic drive robot:

1. Designing position control of **Holonomic Drive Robot** (ground vehicle). (this will be done on a “high-level” model of the robot as mentioned above in Task 1B objective).
2. **Understanding forward and inverse kinematics of three-omni-wheeled bot**. (the holonomic drive robot that we shall use in this theme) *We’ll only need to implement inverse kinematics for the purpose of taking the output of the above mentioned controller and controlling the three wheel velocities to achieve the desired (v_x, v_y, w)*
3. **Localisation of the robot**. In task 1B, we built the position control with localisation given (for free) by the simulator. But in the real world (outside simulator) we’ll need to worry about how

to localise the robot (i.e. answering the question: where is the robot?🤔).

4. At a level even higher than the controller (discussed above) lies the question; **where to even go?** We could call this **"Planning"**.

5. If we wish to draw smooth shapes, simple position control may not suffice. Therefore, we may need to explore more **fancy/advanced** ideas to control our bot.

6. All of the above can be implemented directly in hardware, but it is a good idea to simulate and find problems without actually breaking some hardware😛. Thus, here comes **Gazebo Simulator** and **ROS**!

As you know, in Task 1B we explored the first and last points listed above. In Task 2A, we shall explore the second and third point.

And finaly in Task 2B we shall multiple Task 2A by three!

> Task 2 is the last task of Stage 1. Therefore the deadline given in the announcement post (and the portal) is unlikely to change. And this is also the deadline for all tasks in Stage 1.

The performances of Task 0, 1 and 2 will determine if the team will be selected for Stage 2, so make sure to go all in!

All the Best!

With no further ado, let's look at the Task 2A: Problem Statement!