ROS 2 Powered Autonomous Navigation

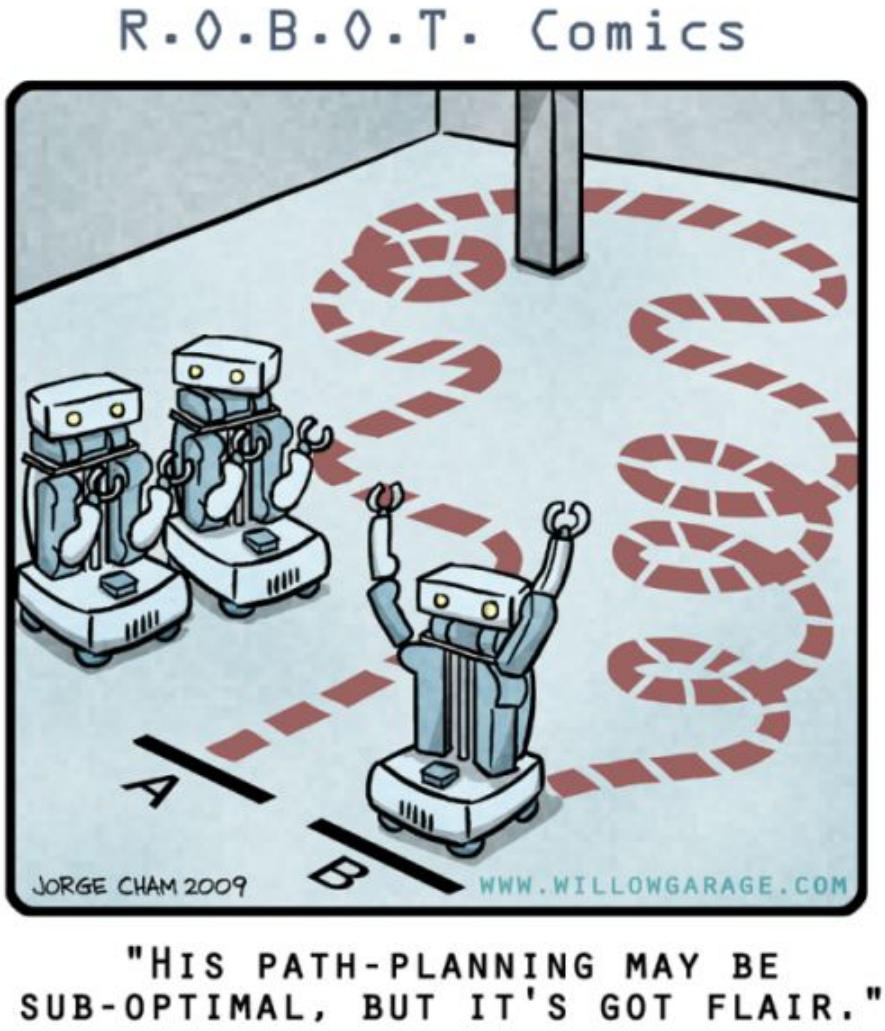
A Solution @ UCA based on LiDAR SLAM and Differential Drive

Lin Zhang (lzhang12@uca.edu)

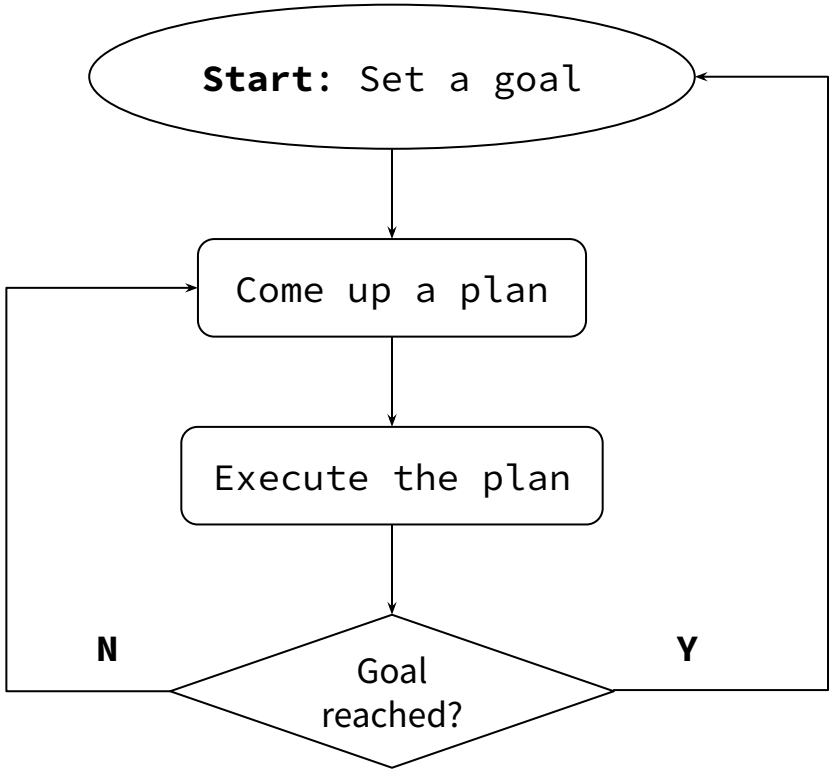
02/28/2026

Outline

- Autonomous Navigation Overview
  - Set a goal
  - Make plans
  - Execute
- Meet HomeR (Hands-On)
  - Motion control
  - ROS management



Autonomous Navigation Workflow



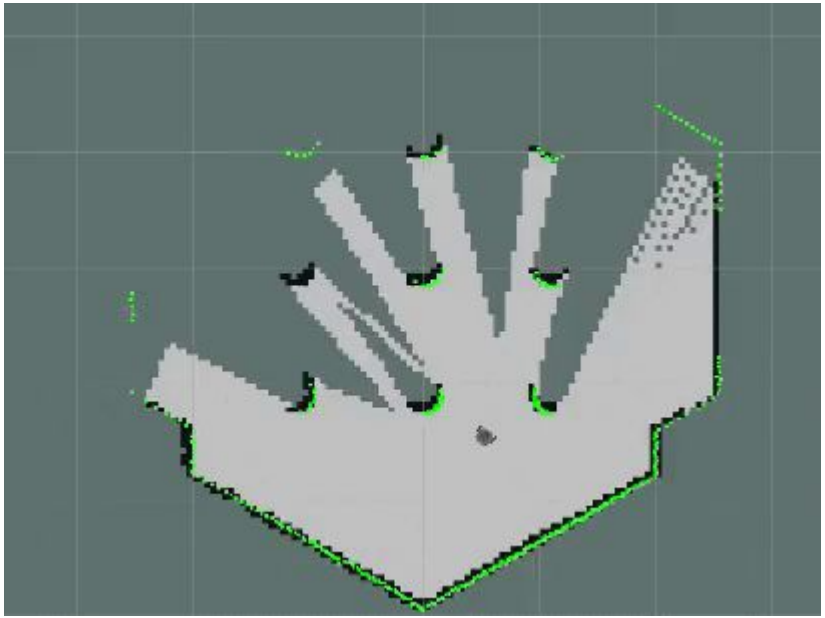
Step 1: Set A Goal

Nothing, but coordinates on a map.

Step 0.1 Make A Map

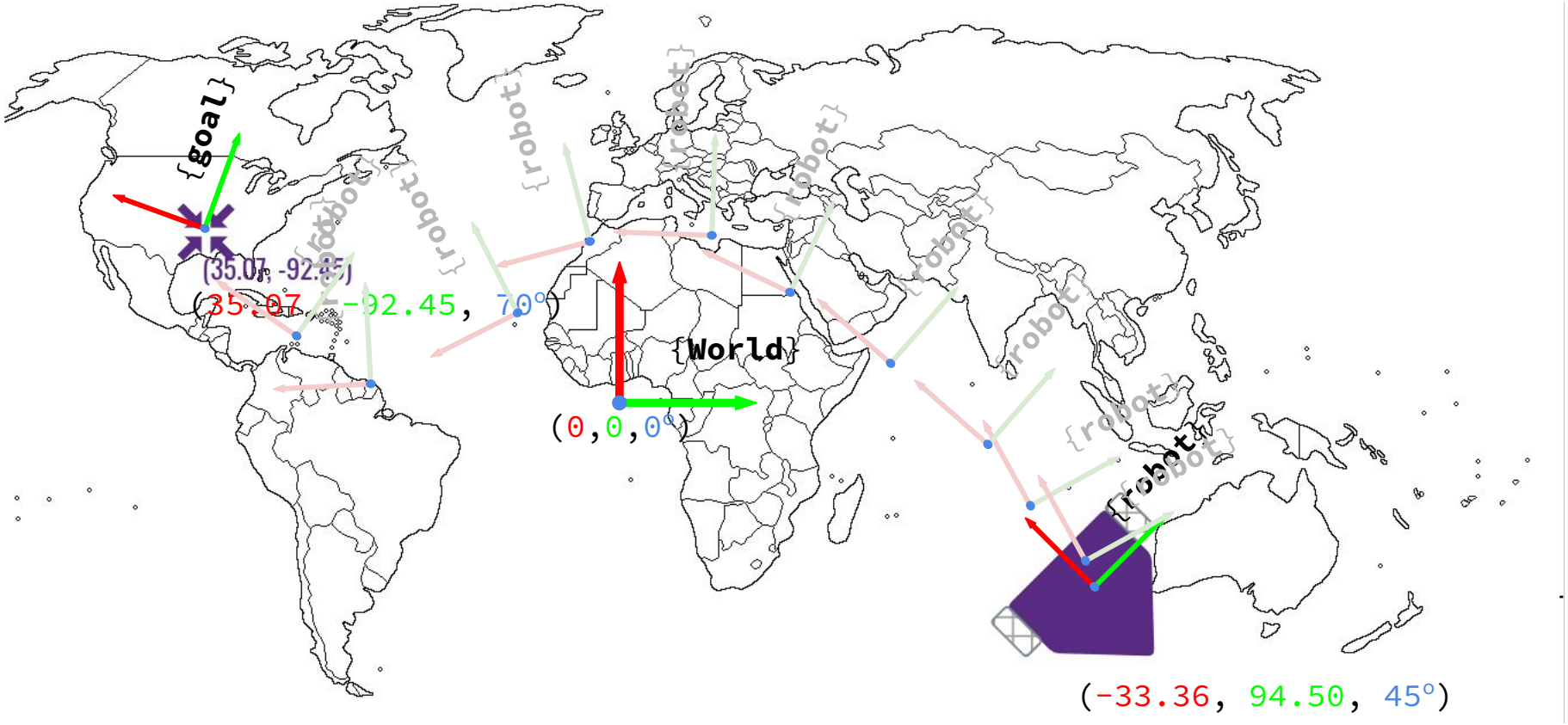
Synthesizing the big picture from fragmented information

Jigsaw Puzzle

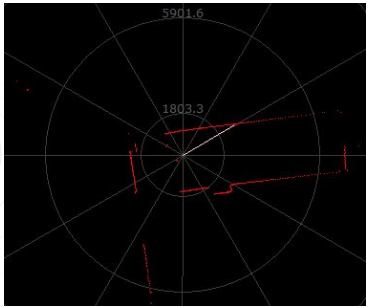


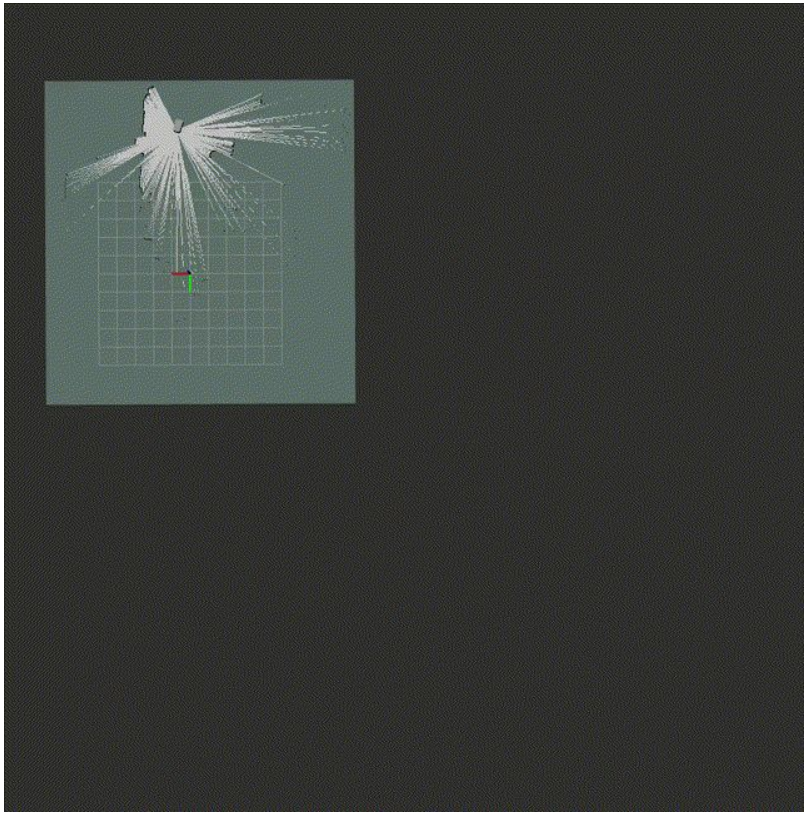
LiDAR SLAM

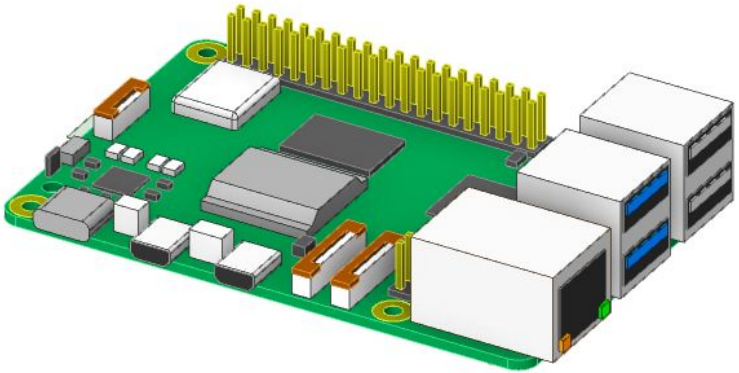
Step 0.2 Coordinate Frames



Step 0 Solution: RPLIDAR A1 + Raspberry Pi 5 + slam_toolbox







Step 2: Make a Plan

Global Plan

Local Plan

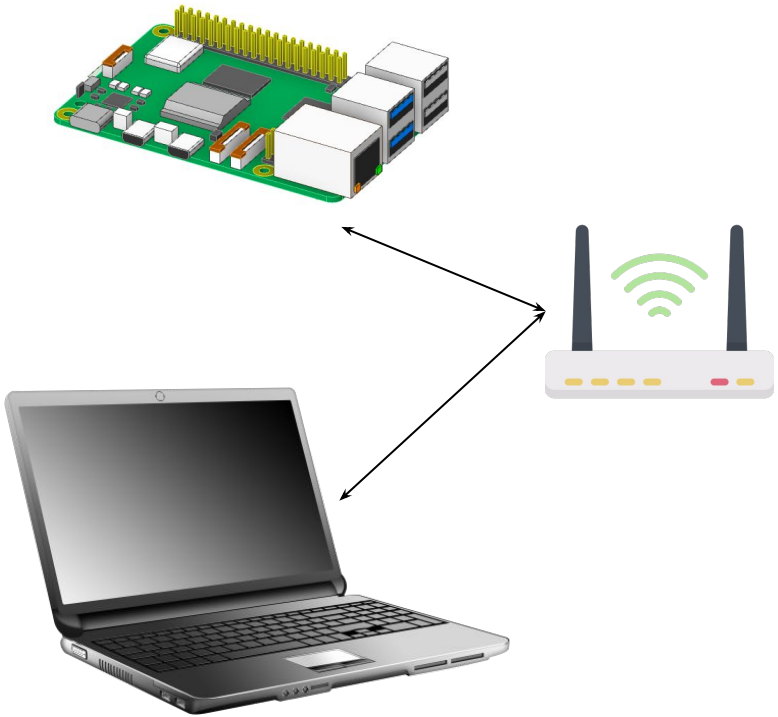
Step 2.1: Global Planner

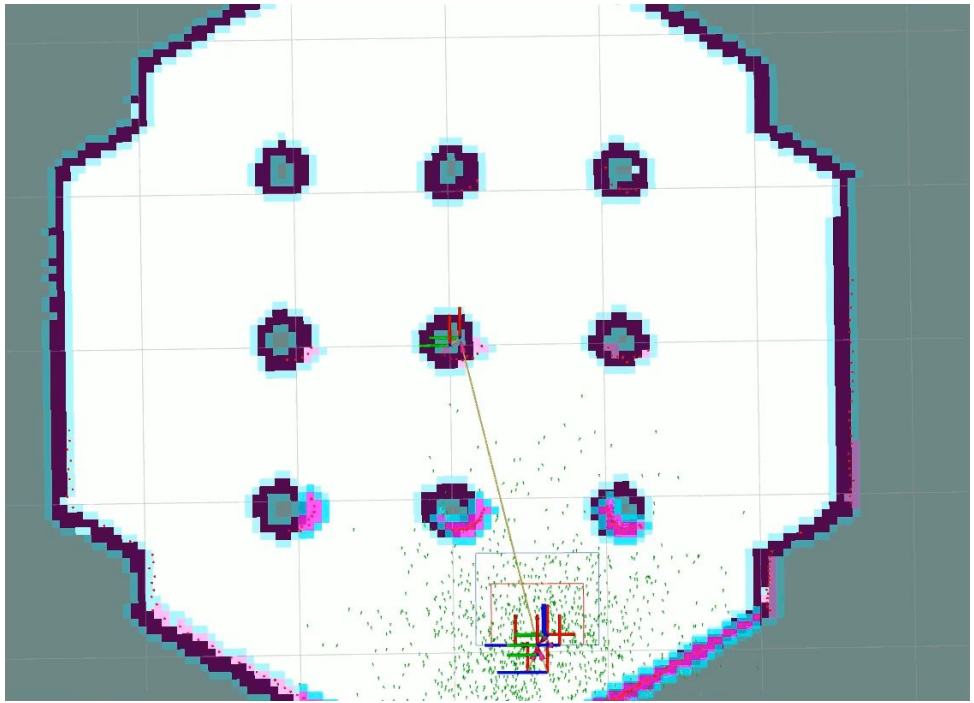
- Relies on current pose and goal pose.
- Computes the shortest, obstacle-free path from start to goal.
- ROS 2 default algorithm: Dijkstra
- Runs at lower frequency (~1 Hz) due to the heavy computing load.

Step 2.2: Local Plan

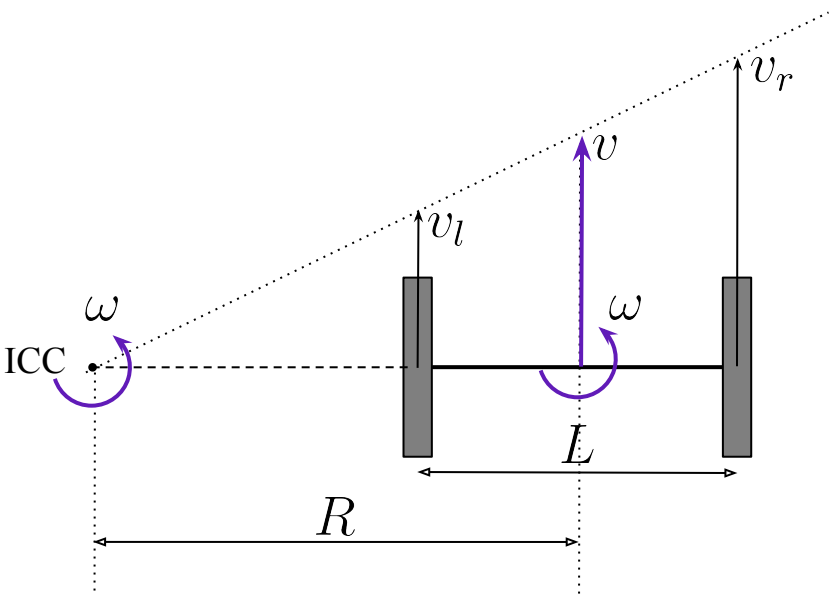
- Based on the global plan while aware of self motion status and the environment.
- Generates safe velocity commands to follow the global path while avoiding dynamic obstacles.
- ROS 2 default algorithm: Dynamic Window Approach (DWB).
- Does quick responding jobs at a higher frequency (20+ Hz).

Step 2 Solution: Server Computer + Raspberry Pi + Nav2





Step 3: Plan Execution



$$v = \frac{v_l + v_r}{2}$$

Robot's linear velocity

$$\omega = \frac{v_r - v_l}{L}$$

Robot's angular velocity

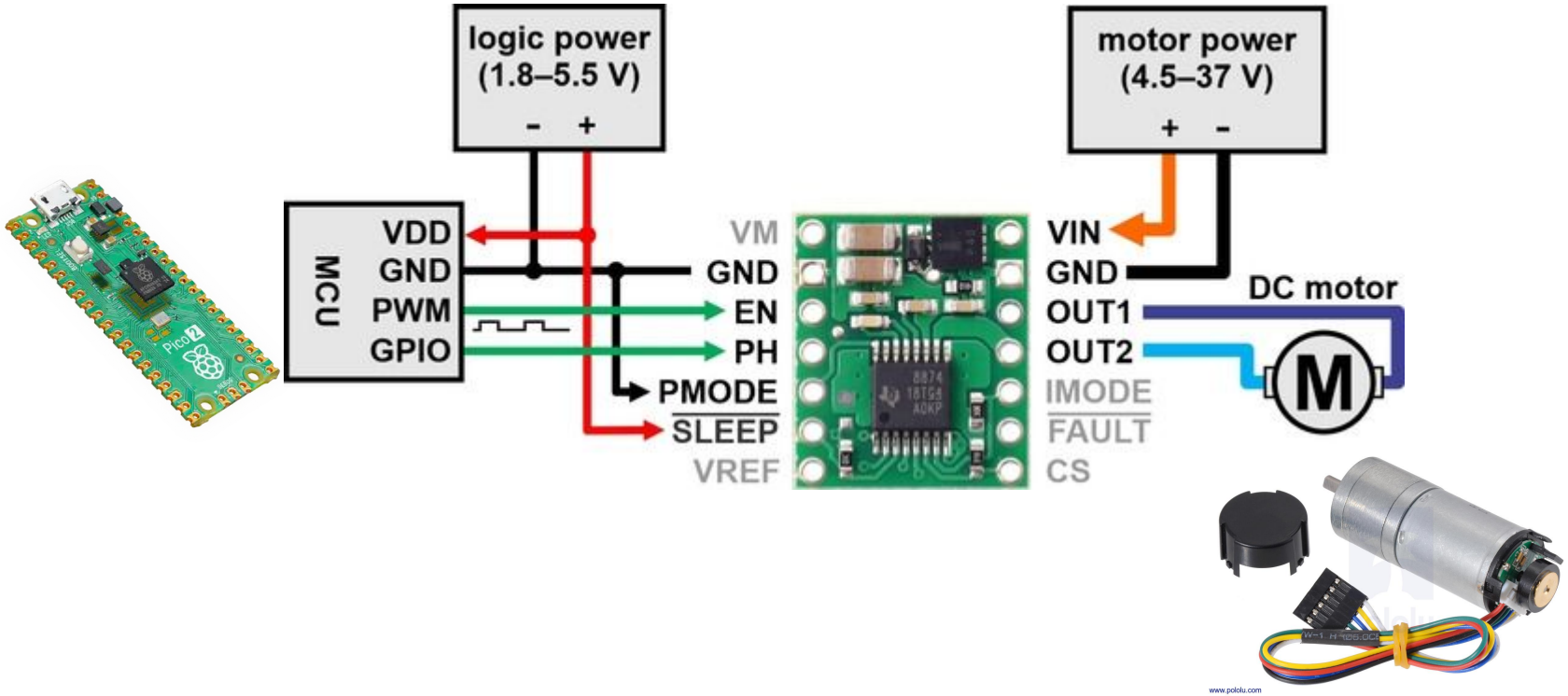
$$v_l = v - \frac{\omega L}{2}$$

Linear velocity of left wheel

$$v_r = v + \frac{\omega L}{2}$$

Linear velocity of right wheel

Step 3 Solution: Raspberry Pi Pico + Pololu 4805 + DRV8874



Software Stack: Ubuntu + ROS 2 + Slam_toobox + Nav2

- Ubuntu 24.04
- ROS 2 Jazzy Jalisco
- Nav2
- Slam_toolbox
- MicroPython

Take Away

HomeR @ UCA
Engineering Physics

https://linzhanguca.github.io/homer_docs/

Thank You