ROS 2 INTERFACE FOR A MOBILE ROBOT

Authors: Pat Truong, Saber Omidi

John Olson Advanced Manufacturing Center, University of New Hampshire

Abstract

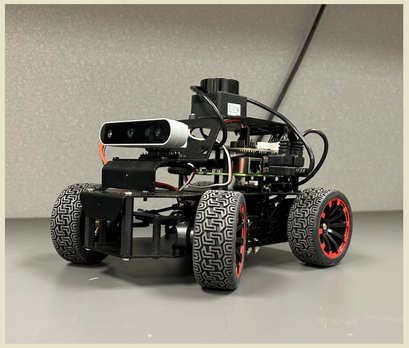
This project developed a simple Python interface for controlling a mobile robot with a basic PID controller. The interface allows users to start, stop, adjust speed, and manually move the robot. This interface serves as a stepping point and has the potential to work with advanced algorithms.

Introduction

- ROS 2 (Robot Operating System 2) is a versatile, open-source framework for creating robot software; it acts as a communicator between a computer and a robot.
- Gazebo Sim is a 3D robotics simulation environment that provides realistic physics simulation.
- This robot and interface are important because they simplify robot control, demonstrate feedback systems, and is vital to developing advanced processes.
- These systems can be applied to autonomous vehicles, manufacturing robots, delivery robots, and other machines that need precise and reliable movement.

HiWonder MentorPi

- Hiwonder MentorPi A1 is a Raspberry Pi–powered educational robot designed for ROS 2 projects.
  - It uses sensors including LiDAR and cameras for navigation, mapping, obstacle avoidance, computer vision, and Ackermann steering.
- The Dubins path is a technique for determining a quick route between an initial and final position.



HiWonder MentorPi A1 robot

System Workflow

Keyboard / GUI
↓
Python Interface
↓
ROS 2 Topics
↓
PID Controller
↓
MentorPi Robot
↓
Odometry Feedback

Scan to View the Ellipse Demonstration

MentorPi Control Panel

Status: Ready

Current Speed

0.000 m/s

Keyboard Controller

WASD or Arrow Keys

Drive: 0.00 m/s Turn: 0.00 rad/s

Enable Controller Mode

0.000 m/s

Drive Speed (m/s)

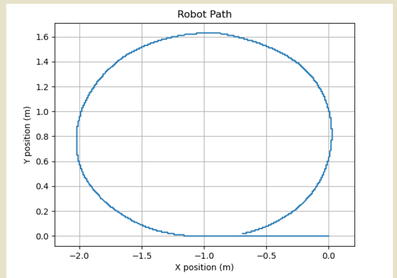
0.05

Start Ellipse

Stop Ellipse

EMERGENCY STOP

Python–based robot control panel

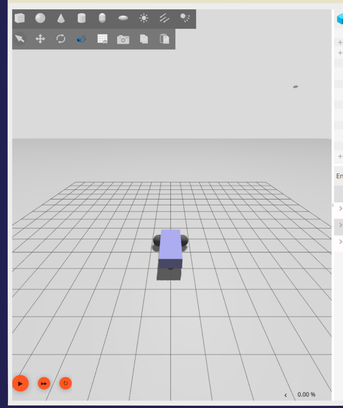


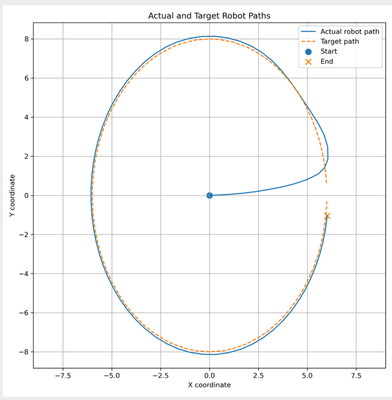
Recorded elliptical path using coordinates

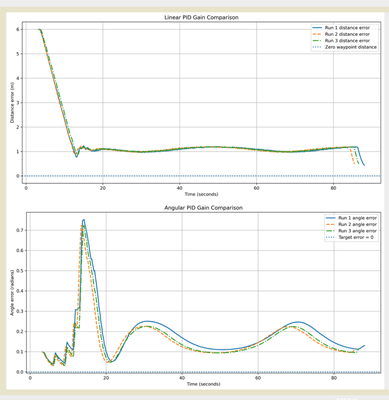
Future Work

- Establish a backup mechanism
- Add more autonomous movement options.
- Apply PID control to additional movements and autonomous tasks.

GAZEBO SIM AND PID







	Linear	Angular
Run 1	Kp = 0.7 Ki = 0.0 Kd = 0.08	Kp = 0.55 Ki = 0.0 Kd = 0.06
Run 2	Kp = 0.9 Ki = 0.01 Kd = 0.08	Kp = 0.65 Ki = 0.005 Kd = 0.08
Run 3	Kp = 0.08 Ki = 0.01 Kd = 0.08	Kp = 0.65 Ki = 0.005 Kd = 0.06

Result: The completed system successfully supports manual keyboard control and PID–controlled elliptical path following.

Aknowledgements

I would like to express my gratitude to my project advisor, Saber Omidi for helping introduce me to this initiative and special thanks are offered to the National Science Foundation and its EPSCoR Track 2 grant (NSF #2218063) for providing support for this research. Furthermore, I would like to thank the John Olson Advanced Manufacturing Center, the University of New Hampshire, and the faculty--including Dr. John Roth, Dr. Se Young Yoon, Dr. Stephen Hale, and Cordelia Norris--for making this project possible.