CSE 276A HW1: Waypoints

Varun Vupparige
Mechanical and Aerospace Engineering
University of California San Diego
La Jolla, USA

I. Introduction

With ever increasing use of robots in the different fields such as aviation, marine, automotive, healthcare, and military exploration, the human-robot interaction has increase exponentially. An autonomous system consists of various subsystems such as sensing, perception, path planning and control systems, and thus study of principles of robotics has been an important subject in recent times. Examples of successful robotic systems include mobile robots for planetary exploration, industrial robotics arms in assembly lines, autonomous driving cars and manipulators that assist surgeons. Robotics systems are situated in the physical world, obtain information from their local environments through on-board sensors, and make reactions.

In this course, we are provided with a metal structured four-omnidirectional-wheeled robot powered by Qualcomm RB5 development kit. In the next sections, we will formulating the problem statement and explain the implementation of open-loop control to traverse the given way-points.

II. Problem Formulation

In this section, we will describe the various different components provided for completing this coursework, and the underlying physics of four mecanum-wheeled mobile robot. We are provided with robot chassis consisting of upper shell, body shell, motor brackets and connector bracket (including their relevant fasteners and tools). Furthermore, we are provided with MegaPi, a programmable micro-controller which is capable of functions such as gesture recognition, line following, and obstacle avoidance with the usage of relevant sensors. Lastly, the star of the show is Qualcomm Robotics RB5 Development Kit, which is equipped with a QRB5165 processor capable of complex computational tasks.

Omni-differential systems work by applying rotating force of each individual wheel in one direction similar to regular wheels with a different in the fact that Omni-differential systems are able to slide freely in a different direction, in other word, they can slide frequently perpendicular to the torque vector. The forward and inverse kinematics of four mecanum-wheeled robot is as follows:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & -1 & -(l_x + l_y) \\ 1 & 1 & (l_x + l_y) \\ 1 & 1 & -(l_x + l_y) \\ 1 & -1 & (l_x + l_y) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} \tag{1}$$

$$\begin{cases} \omega_1 = \frac{1}{r}\left(v_x - v_y - (l_x + l_y)\,\omega\right), \\ \omega_2 = \frac{1}{r}\left(v_x + v_y + (l_x + l_y)\,\omega\right), \\ \omega_3 = \frac{1}{r}\left(v_x + v_y - (l_x + l_y)\,\omega\right), \\ \omega_4 = \frac{1}{r}\left(v_x - v_y + (l_x + l_y)\,\omega\right). \end{cases} \tag{2}$$

where,

l_x = half of the distance between front wheel
l_y = half of the distance between front wheel and the rear wheels
ω = wheel angular velocity
r = radius of the whee
v_x = linear velocity in x direction
v_y = linear velocity in y direction

This kind of wheeled robot is holonomic, and is capable of full omnidirectional motions. The kinematic equations above can be employed to formulate the motion model. Given a robot's state x_t, and control input u_t, the state x_{t+1} can be modelled as a probabilistic/deterministic function given by,

$$x_{t+1} = f(x_t, u_t) = p_f(.|x_t, u_t) \tag{3}$$

III. Implementation

In this section, we will discuss the overview of robot setup, methods employed to calibrate the robot control and its implementation to traverse the given way-points.

A. Assembly and setup

The assembly and setup of MegaPi robot powered by QRB5 consists of following steps:

- Mechanical Assembly of chassis/structures, motor, sensors
- Wiring of different sensor, motor drivers
- Verification of Mbot Motors
- MegaPi firmware update to remove random behaviours
- Flashing RB5 system with Ubuntu 18.04
- Setting up of Wifi and SSH access
- Installing ROS Melodic on RB5
- Testing keyboard control

B. Calibration and Implementation of the Controller

In this particular assignment, we are provided with robot which has no feedback from any sensors. Due to this reason, its important calibrate the motors to know the amount of PWM

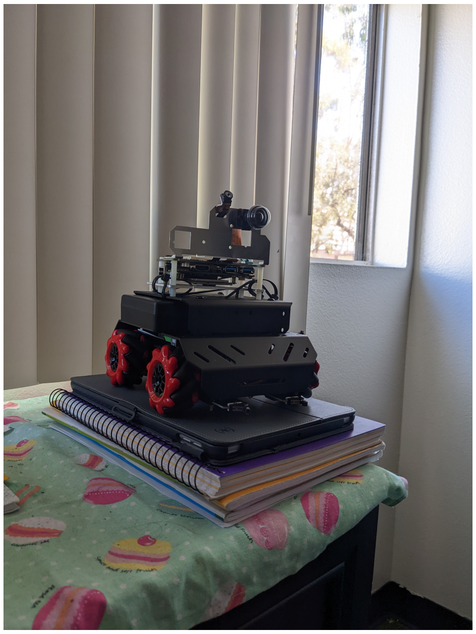

Fig. 1. MegaPI bot powered by QRB5

signal required to traverse a particular distance in known time. In order to do this, I performed experiments where in the robot was run in straight for a particular amount of time and distance travelled was noted, and thus linear velocity was calculated.

But before this we ran into issue of slight lateral drift when the robot was supposed to travel in a straight line. In order to address this issue, we decrease the PWM signals of the outside wheels. For example, if the robot is drifting towards left then it can be inferred that the outside wheels are rotating at higher RPM. In a similar way, the robot was calibrated for slide and rotate motions to find the lateral velocity and angular velocity of the robot.

After the calibration process, the markers were laid out which represented different way-points. Then the motion was broken down into smaller actions, and each sequence of motion was hard coded according to the values of linear and angular velocity. The ROS node basically published messages to the topic joy which was subscribed by the megapi controller. Due to irregular weight distribution, all kinds of motion does not represent the same performance so slight tuning was done to meet the waypoints.

C. Results

The controller performs fairly well due to same conditions such as type of surface, external weather conditions .etc. Due to irregular weight distribution, the robot produced inconsistent performance interms of time required to traverse straight versus traverse reverse. So, we had to do some slight tuning with times. Any change in the conditions would result in poor performance, thus this type of open loop controller based on kinematic model is not scalable or efficient.

Video Link: click here