DD2425 - Robotics and Autonomous Systems

Group 6

December 15, 2014

Project Report

Authors

Carlos Gálvez del Postigo
Mathias Lindblom
Tingyu Liu
Gundars Kalns

Abstract

Abstract Abstract

Contents

1 Introduction

This is the project report in course Robotics and Autonomous systems. The purpose of this report is to reflect ideas and solutions we provided for trying to reach the goal of this course – making autonomous robot that can explore maze, create map of it and detect and recognize different geometrical objects located in maze.

2 Mechanical design

Robot size is 23 cm x 23 cm width and length (including wheels) and 29 cm height. Same width and length were chosen due to ease of maneuverability. For the same reason wheels are located in the middle of its sides and corners of the robot are cut off in 45$\breve{8}$ angle. Two aluminium plates are connected with pillars in such way making two floors. On the first floor are located motors, acumulator battery and speaker whereas on the second floor is located NUC and Arduino sandwitch. For camera there is pillar which raises it high enough so it reaches 29 cm height. IMU is located under the second plate as well as long range IR sensors which points to front and back direction. Short range IR sensors are located on the pillars which connect first and second floor.

$$y = f(x) \tag{1}$$

Equation 1 shows that...
Thrun *et al.*[**?**] show that...

3 Electronics

3.1 NUC

3.2 Arduino

3.3 IMU

3.4 IR sensors

3.4.1 Calibration

Because of non-linear sensor output vs distance curve it is not practical to use raw values from IR sensors in higher level nodes. Therefore it was needed to create node which converts raw data from sensors to distance. As the raw output changes values more rapidly in lower distances, measurements in low range were made quite densely (by measuring sensor output every 0.5 cm) whereas in longer distances such accuracy were not needed because output values were approximately the same even within 5 cm range. Sensor converter node is the only one which subscribes to raw sensors node and other nodes in ROS subscribes directly to distance data.

3.4.2 Filtering

To filter out noise from IR sensors data Kalman filter is used.

4 Motion control

To be able to reliably use motors instead of sending just PWM data to motors there needs to be a node which translates linear and angular speeds to necessary PWM values. However as PWM vs speed

relation is highly non-linear and depends on many factors such as ground friction it is not possible to just find suitable function for translating speed to PWM. Use of motor controller is therefore crucial. Its task is to control PWM values at the same time getting feedback from encoders. This data then can be used to adjust PWM values so that motor reaches the necessary speed and keeps it. In this project PID controller is used for motor control. Equation 2 shows general PID controller algorithm. In case of motor controller error e(t) is difference between desired angular speed and current angular speed which can be calculated with equation 3. N describes ticks per revolution.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \tag{2}$$

$$\omega(t) = \frac{2\pi \Delta encoder}{\Delta t * N} \tag{3}$$

With PID controller it is very important to find right parameters otherwise motion can be with oscillations, too slowly rising or could have other problems.

5 Software

In this project ROS software is used. It creates good and feature rich environment for simple nodes which can easily communicate to each other.

5.1 Global picture

5.2 Odometry

Odometry node takes raw data from motor encoders and translates it into global coordinates and angle according to equations 5.2. This data can be used for mapping and localization. Problem with odometry is that it tends to drift especially when speed increases.

$$distance_{left} = \frac{2\pi * Wheelradius * encoder_{left}}{tickspervolution} \tag{4}$$

$$distance_{right} = \frac{2\pi * Wheelradius * encoder_{right}}{tickspervolution} \tag{5}$$

$$\Delta x = \frac{distance_{left} + distance_{right}}{2} * cos(\Phi) \tag{6}$$

$$\Delta y = \frac{distance_{left} + distance_{right}}{2} * sin(\Phi) \tag{7}$$

$$\Delta \Phi = \frac{distance_{left} - distance_{right}}{Wheelbase} \tag{8}$$

5.3 Mapping

For map we chose to use occupancy grid package from ROS. IR sensors provide data about distances to the walls. This data together with position information from odometry node are used to make a map. Initially all area is unknown and by driving around and getting corresponding data from IR sensors free area and obstacles are detected and drawn in map. Data from sensors are quite noisy therefore some processing is needed to mitigate noisy results.

5.4 Exploration

For exploration there was created navigation node.

5.4.1 Wall following

5.4.2 BFS Search

5.5 Path planning

5.5.1 Global

5.5.2 Local

5.6 Localization

6 Computer Vision

6.1 Camera extrinsic calibration

6.2 Object detection

6.3 Object recognition

6.4 Obstacle detection

7 Discussion

7.1 Results

7.2 System Evaluation

7.3 Project management

8 Conclusions

Conclusions