

Contents

1	Introduction	2
2	Mechanical design	2
3	Electronics	2
3.1	NUC	2
3.2	Arduino	2
3.3	IMU	2
3.4	IR sensors	2
3.4.1	Calibration	2
3.4.2	Filtering	2
4	Motion control	2
5	Software	3
5.1	Global picture	3
5.2	Odometry	3
5.3	Mapping	3
5.4	Exploration	4
5.4.1	Wall following	4
5.4.2	BFS Search	4
5.5	Path planning	4
5.5.1	Global	4
5.5.2	Local	4
5.6	Localization	4
6	Computer Vision	4
6.1	Camera extrinsic calibration	4
6.2	Object detection	4
6.3	Object recognition	4
6.4	Obstacle detection	4
7	Discussion	4
7.1	Results	4
7.2	System Evaluation	4
7.3	Project management	4
8	Conclusions	4

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° angle. Two aluminium plates are connected with pillars in such way making two floors. On the first floor are located motors, accumulator battery and speaker whereas on the second floor is located NUC and Arduino sandwich. 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

4 Motion control

In motors pwm input vs speed relation is highly non-linear and depends on many factors. Use of motor controller therefore is crucial. In this project PID controller is used for motion control. The purpose of motor controller node is to control PWM values for motors according to input linear and angular speeds. It adds abstraction layer for motors so in higher level nodes we can just care about sending desired linear and angular speeds to motors.

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.

$$distance_{left} = \frac{2\pi * Wheelradius * encoder_{left}}{ticksperrevolution} \quad distance_{right} = \frac{2\pi * Wheelradius * encoder_{right}}{ticksperrevolution}$$

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

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