ROB 301 Introduction to Robotics Project Report

Introduction

The final robot design project summarizes all the applications and knowledge exposed through the entire term and apply them into an actual robot that embedded with an objective and have the ability to sense and extract informations from the environment and act accordingly. The robots are equipped with a designed control system in order to perform the pizza delivery task. The mission of the robot is to deliver pizza to a destination that is randomly chosen from a set of predefined locations based on the user's instruction. During the delivery process, robot must detect and avoid randomly placed obstacles. Then the robot are required to return to its initial pose after the pizza is delivered successfully.

In this design report, a detail illustration of overall structure and platform design of robots are presented. Moreover, explanations about design strategies in each components and analysis and justifications about the algorithm along with detail figures and flowcharts are provided. Furthermore, this report discusses and summarize the actual performance of the robot in the design demonstration and offer constructive suggestions on how to improve the performance of robots for future benefits.

Problem assessment

To simplify the analysis of the project, the project is divided into 4 sub-tasks and all considerations are based on those 4 missions.

1. Pizza pick-up

There are two possible locations assigned to the 'Pizza' within the workspace. Each of these 'pizzas' are placed on a wooden pedestal with its diameter slightly smaller than that of the pizza. The pizzas are placed at a specific height upon the pedestal and this height should be taken into consideration while building the clamp. The clamp should be able to pick up the pizza without moving the stand by any means.

2. Obstacle avoidance

After picking the 'pizza', robot delivers it to a predefined address. It is necessary for the robot to avoid building and parked vehicles (i.e. obstacles) during its path. No contact with any obstacles is allowed during its delivery.

3. House Finding

Upon reaching the desired street, the robot should be able to travel along the street and then specify the house where the pizza should be delivered to. Then the robot should put down the pizza just in front of the target house.

4. Back navigation

The last mission of the robot is to navigate back to its initial pose (i.e. its initial location and initial heading angle).

Solution strategy overview

1. Pizza pick-up

Since the starting point and the pizza location are fixed, the robot is driven directly to the desired location follow a fixed path. Then the clamp is used to grab the pizza. This method not only simplifies the task but also avoid the noise which might appear by using ultrasonic sensor to locate pizza.

On the other hand, there is a built-in algorithm that keep updating the pose of the robot after each step based on the reading of the gyro sensor and the robot's step size.

2. Obstacle avoidance

The robot is loaded with a coordinate of the desired POSE for the destination. Moving the robot to the desired POSE is performed by an A* like algorithm. At the beginning of each iteration, the motor-connected ultrasonic sensor is rotated to test the existence of obstacles along three different directions. If there is an obstacle locating at the detected direction, then the algorithm will set the distance from that location to the destination as infinity. Otherwise, the algorithm will calculate the distance from the possible next locations to the desired destination and then choose the smallest one as its next location.

3. House Finding

Upon avoid all obstacles in its path, the robot now has reached a predefined location which it can reach the desired house. The ultrasonic sensor is then rotated to face the side where the target house locate. The house-finding algorithm takes a counter to find the target house. As the robot moving along the desired path, the counter decrements itself everytime a house is found. Once the counter becomes 0, which means that the robot reaches the target house, the robot will rotate to face the house and then put down the pizza just in front of that house.

4. Back navigation

Upon finishing the pizza delivery, 'self-homing' algorithm will drive the robot to return to its initial pose. The task is accomplished by using the built-in coordinate and simple unit functions such as going forward or self rotate.

Robot Platform Design

Mobile Base

The robot base platform designed in this project are constructed using the LEGO Mindstorms EV3 Kit. The mobile base designed is a modified version of the robot used in previous labs due to simplicity and the proven functionality, this robot uses the tripod differential steering, two motors are utilized to control left and right wheels independently and allow inplace rotations and a single unpowered universal wheel is used as support. All the sensors and components are assembly to the side of the robot in order to reduce the overall length of the robot for easy rotation and keep the the majority of the weight close the the center of gravity to improve stability.

Robot Arm

The robot arm used in this project are constructed using a medium servo motor and gear system to control the open and close of a clamp. Upon reaching the pizza location, the medium servo motor rotate to close the clamp and apply a firm grip on the pizza box.

Ultrasonic Sensor

The ultrasonic sensor are used to determine the distance, the sensor emits ultrasonic waves and use the echo to measure distances. In this project, it is the primary tool to detect obstacles and in between in advance thus allow robot to react. The sensor is mounted onto a large servo motor to measure distance in multiple angles by rotating the motor. By designed the sensor this way helps the algorithm to determine the best proceeding path to the destinations. The sensor-motor set is installed at the front of the robot to allow information are collected in advance and leave enough space and time for robot to respond.

Gyro Sensor

On top of the robot, a gyro sensor is installed in order to keep track of the current pose and the angle the robot deviate from the initial position, also the information collected by gyro sensor are used to calculate the current position in the map and update information. The gyro sensor are placed on a level platform facing the front of the robot for minimizing measurement error.

Technical details

Demonstration

In the actual demonstration, the robot successfully pick up the pizza identified by the user. However, it did not reach the desired location due to the number of obstacles and the measurement error. Project wise the robot did not achieve the overall objective.

Potential Improvements

Localization and Kalman Filter

For the current localization method applied in this project, the state of the robot is determined based on the state model which takes its previous state and the readings of the gyro sensor to predict its present state. However, it cannot prevent the error accumulation caused by the input noise and measurement noise of the gyro sensor in each step. For improvements, the

Kalman Filter can be applied to the control system to eliminate the noise and thus yield a more accurate estimation of the present locations of the robot.

PID
House locating, tune gain
The PID controller can be applied in this project in order to help the robot travel along the straight path between houses. The PID controller with fine tuned gains allow robot to travel without oscillation and overshoots except the initial stage.

Bayesian Tracking (more destinations)
Is house location is known, use Bayesian tracking to determine current location
For this project, six wooden blocks are used to represent houses for the pizza delivery. In this design, the robot used a counter and ultrasonic sensor to record number of houses it traveled pass thus delivery the pizza to the correct location. For a much more realistic project where the the environment of the houses are more complicated (tree, empty spot, etc) the method of Bayesian localization can be applied where the robot calculate the probability of current location when a certain object is detected, and updates its estimation when it moves to the next step. This method allow the robot to be applied in applications that is closer to real life.

Dead end
Project vs real
Object avoidance (vertical + horizontal not efficient) not the shortest path
SLAM(based on ultrasonic sensor reading, we can build a map around present location then use the map to find an optimized path

For the object avoidance method used in this project, the ultrasonic sensor measures distance between the robot and obstacles in three directions. The robot is directed to move only in vertical and horizontal direction. However, this method does not consider the condition when the robot is stuck in a dead end. The simultaneous localization and mapping can be applied in this condition, where the robot utilized the ultrasonic sensor reading to build a map around current location and uses the map to find the optimized path to the destination.

Return trip(cannot follow exact path) not efficient

If the noise of the state model and measure noise can be eliminated such that the present locations can be determined accurately. The the robot can return to its initial pose efficiently by following its exact path back.

Conclusion

This project provide an opportunity to demonstrate and assess knowledge learned in this course and previous labs. The technique about PID controller, kalman filtering, and path finding can be utilized and applied in many ways in this project. The project also reveals the environment which have lots of noise and randomness to students and reflects the difficulty to develop a robot that can achieve one simply task in real life environment. Most importantly, this project motivates and present the excitement to students in pursuing future robotics studies.

Overview problem statement: Our task is to
In our robot view, the world is a square with ()() grid based map view. Every step he moves, he will ultimately know his pose based on the information that gyrometer provides.

Fetch pizza

Problem: There are two given spots where we can grab pizza. The height of pizza holder is provided. The dimension of the pizza is given.
Solution: We hard coded the trajectory, and decomposed the process into 3 steps, namely turn left with path whose length are predefined, turn right with a predefined angle, and then approach to the pizza with the hand open and clamped it. The difficulty here is to test the holding force so that we ensure that the pizza will be held firmly throughout the whole run once got the pizza.
Our motivation is to grab the pizza with less unknowns. In this process, all the parameters have been celibately chosen based on real dimension from the test field and our trails of experiments.

Block Avoidance:

Problem: given a forest of blocks, the robot is expected to avoid the block and arrive to the destination.

In our algorithm, our desired final pose is the blue circle, facing north. We firstly initialized our field into a grid-map. We consider to use a modified A* star algorithm to complete the task. Similar to original A*, The blocks are viewed as obstacles where the distance of that node is infinite. From the starting point, the robot is moving in 3 degrees of freedom, namely, left, front and right depending on its own orientation.

Basic steps are as followed:

(1) Checking the neighbours' availability: rotate the ultrasonic and keep the reading

(2) Based the readings from the three directions, the distances of each neighbour to our destination will be calculated.

(3) Comparing the distances in the neighbourhood, the neighbour with the smallest distance is therefore chosen as the next moving point.

(4) Rotate our robot body and arrive to the point, facing north again.

Our algorithm will iterate the steps above until reaching the destination, i.e. the blue circle illustrated in Fig.., which can be expressed as the coordinates of (153, 500) with a tolerance of 10 cm.

It's easy to implement that we constrain our robot only moves in 3 directions. Moving in the grid can be considered relatively easy since there are no other direction allowed. The sources of error are the accumulation of uncertainty of each step-forward and rotation. Another important error comes from the coordinates updating based on our movement. Given there exists uncertainty in the actuators, our updated coordinates won't be identical to where we are in reality, i.e. there are error between robot's location in the global frame and in the updated map in robot's mind. We ignore the case where there are blocks in all three directions with the assumption that the distribution of obstacles are ideal and less complicated than in the real-world.

Pizza Delivery

When it comes to the stage of real **pizza delivery**, as such, our robot will be acknowledged the house in selected lane as his final destination. The road can be identified based on the colour circle which is in the beginning of this road.

Our approach is firstly aligning ourselves with the road, . Then acquiring the specific side of the road the house located and which house is along that row, i.e. first, second and third from the input. The real implementation is to follow the road, and then turn our ultrasonic sensor to the specific side with robot moving a certain step-size. Once the ultrasonic sensor reports that there's a house detected, we record it in our list of houses that have been detected. Once it reaches the desired house, we turn to the house and drop the pizza, i.e. loosing the clamp.

It's efficient to just reach the house that we are given and ignore the rest of the houses in that road. It's effective to just use initial detection by ultrasonic as the confirmation that the house has been detected since our step-size here is about the half of the house length.

Returning

Last part of the task, of course, is to go back to its starting point. In the end of the previous sub-task, our robot is facing the house. To set-up the trajectory of returing home, our robot firstly reaches the end of the lane and get to outer . Our robot turn to face west and follow the line to go back. The outer path is certainly pre-calculated. The difficulty is to address the length of the path from his starting point of the outer path to end of line 1.

Parameter Delivery

Parameter Delivery: The Design of User Interface

Our solution relies heavily on the knowledge that user passed into our robot, i.e. the location of the piazza, the road where the house located in specific side. These information is delivered through a initial prompt and options been chosen, i.e. which button has been pressed.

Justification: This way can be easily implemented and the information extraction is effective.