Capstone Project

Runaway Robot

1, Definition

Project Overview

This project came from an optional final projecy in Artificial Intelligence for Robotics. This project assumes a situation like: A robotics company named Trax has created a line of small self-driving robots designed to autonomously traverse desert environments in search of undiscovered water deposits. More specifically, the project programes an agent to track a target by using some methods in robotics

Problem Statement

This project aims to programe an agent to track a target, which moves in circle with some noise. All the information the agent can get from a target is sensing data that where the target is. However, this information is not necessarily equal to the exact location because of the noise. Both the agent and the target move in the same speed. This makes the problem a little bit complicated. Three steps are needed to solve this problem. First, the agent will track the target based on the sensing information. Second, the target intend to move in smaller circle than the target because the target never catch up with the target if it moves in the same circle with the target’s. Third, when the distance between the target and the agent become less than the distance one can afford in one step, the agent will try to catch the target by estimating the target’s next position. In this project, the expected result is that the agent can catch the target in less than 1,000 steps.

Metrixs

The model is measured by how often the agent can catch the target in 50 trials. In the process of running 50 trials, the target is given to random paramators in x, y, heading and turning: x and y are from -10.0 to -10.0, heading is from 0 to 2.0*pi and turning is from -2.0*pi/30.0 to 2.0*pi/30.0. All of them are including decimal points. If it would work well, the steps taken to goal could be analysd.

2, Analysis

Environment Specifications

Target Specifications

The target is the one object the agent try to catch. It basically moves in circle and every step the target moves, the agent can obtain sensing information like (x,y) from the target. Basic information like, initial location and how the target moves is given when the target is made. In other words, the target gets the information like location, heading, turning and distance. Except for distance, they are set by using random function, and distance is consistently set in 1.5.

Agent Specifications

The target is made in the same way with the target, using the same robot class. This means that the agent has the same information like location, heading turning anf distance. Unlike the target, all the factors except for distance can be coordinated by any functions, but as the same with the target, distance is the only consistent factor set to 1.5.

Algorithms and Techniques

Localization

In order to catch the target, the agent needs to estimate a location for the target. One of localization techniques is used for speculating the location: it is particle filters. In my model, 500 particles 8 landmarks are set in virtual environment. Particles measure the distance between them and landmarks in every step. The model comapares it with the distance between the target and the landmarks. If the two distances show some similarity, the particle more likely to get survived. By repeating this processes, particle filters would tell us where the target is now.

Control

The key point in this project to solve the problem is that the agent needs to move in smaller circle than the target's since they move at the same spped. However, without controlling the agent, the agent would not move smoothly along the small circle. PID control makes it possible for the agent to move as it is expected. PID control is short for Proportional, Integral and Derivative control. My model applys PDI control to Turning the agent takes. An equation shown below represents the PDI control: Kp, Ki and Kd are paramators and e(t) is a heading difference between the taeget and the agent in my model.

$$u(t) = K_{\mathrm{p}} e(t) + K_{\mathrm{i}} \int_0^t e(\tau)\, d\tau + K_{\mathrm{d}} \frac{de(t)}{dt}$$

3, Methodology

Implementation

First 20 trials, the agent just try to chase the past location of the target to minimize the distance between the agent and the target since particle filter needs some trials to start estimatig location more closely. Originally, particle filter is for knowing where the object is, but here it needs to estimate next location. In this project, the target's speed and turning is stable, but the agent has to receive those information with some noise. Therefore, to minimize the noise, the agent collect the past data and calculate average of the data. This is how, the agent estimates the target's turning and location, and this enables the agent to estimate next location by particle filter.

After 20 trials, the agent starts moving in smaller circle than the target's by using both particle filter and PDI control. By introducing PDI control, the agent changes its heading slightly. They enables the agent to move along the target. Once the target is close enough to catch the target in next step, it trys to catch the target without using PDI control since the control cahges the heading sloghtly.

4, Result

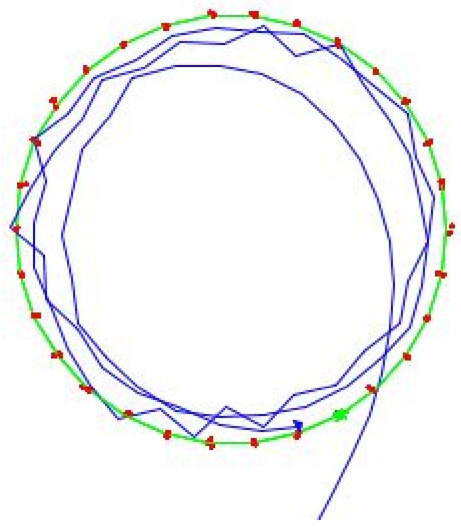
Model evaluation

The agent sucessfully catch the agent 40 times in 50 times. The sucess rate is 80%.

5, conclusion

Visualization

This visualization shows how the agent track the the target. After 20 steps, the agent starts moving in small circle and the agent is close enough to catch the target, it tries to cathch the target.



Reflection

In this project, I programmed two basic methods in robotics, localization and control. Since I am start enrolling in Self Driving Car Nanodegree from December, I am gonna learn a lot more about robotics and deep learning. This project is basically 2 dimention, but pretty sure that programming real self driving car requires 3 dimetion and should be much more difficult than this project. However, this project could be a milestone for my future learning.

Improvement

The paramators for PDI controll could be improved. I tried twiddle function to choose three paramators, but it did not work well. I guess that is because in the process of running particle filter, the model sets random partcles in the environment. It could cause an unstablity in choosing paramators. If I could find the better paramators for PDI control, the model would be imporved.

Resources

Artificial Intelligence for Robotics