Project 6 : Kidnapped Vehicle localization using a Particle Filter

Localization is the problem of an autonomous vehicle identifying its position with-in a map. Once localized, the robot may be able to plan its motion towards its destination. Commonly used Global Positioning System(GPS) measurements are not accurate enough for a self-driving car and are also unreliable.

Particle filter is a type of Bayesian filter used for localization of a robot within a known map; The map is assumed to be fixed in this project. The localization is achieved by the robot sensing landmarks in the map. Bayesian filters are iterative techniques i.e they have repeated operations: prediction and update.

Following table gives a comparison of filters studied in the course

	State space	**Belief**	**Efficiency (growth w.r.t dimensions)**	**Nature**
Histogram/Monte carlo	Discrete	Multimodal	Exponential	Approx
Kalman Filter	Continuous	Unimodal (Gaussian)	Quadratic	Approx (works for linear systems)
Particle FIlter (Markov process)	Continuous	Multimodal	Depends on the case	Approx

The iterative structure is achieved by incorporating Markov assumptions which simplifies prediction of future state by solely basing it on the current state without the need for information of any other previous states. The knowledge of the robot's position at the start of an iteration is called a priori and the estimations made after are called posterior.

The prediction step uses a motion model to calculate the probabilities of being at a future state from the current state. In this project for the autonomous vehicle, we are using a 'Bicycle model' of the car. The equations of motion in a bicycle model are given by

$$x_f = x_0 + \frac{v}{\theta dot}(sin(\theta_f) - sin(\theta_0))$$

$$y_f = y_0 + \frac{v}{\theta dot}(cos(\theta_0) - cos(\theta_f))$$

$$\theta_f = \theta_0 + \theta dot * dt$$

θdot is the yaw rate, dt is the time interval, v is the velocity of the car

When θdot =0, EOM are updated to

$x_f = x_0 + v(sin(\theta_0))$*dt

$y_f = y_0 + v(cos(\theta_0))$*dt

$\theta_f = \theta_0$

The update step involves taking in sensor measurements and adjusting the predictions.

A general form of equation for posterior estimation in Bayesian filters is

$bel(x_t) = \eta * P(z_t|x_t, m) * \widehat{bel}(x_t)$

Where $bel(x_t)$ is the posterior belief, $\widehat{bel}(x_t)$ is the predicted probability, $\eta$is the normalizing factor, $P(z_t|x_t, m)$is the likelihood of observations from the position x_t.

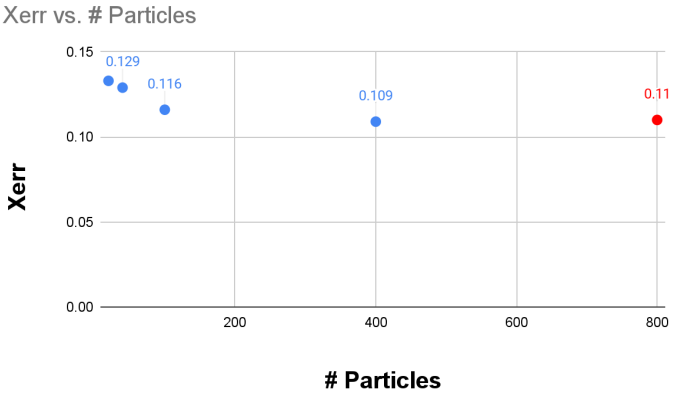
Particle filter utilizes particles representing a hypothetical car. Initially, the particles are spread out, by iterating through the update and prediction steps, the particles converge to the real car position. Steps for implementing a particle filter in this project

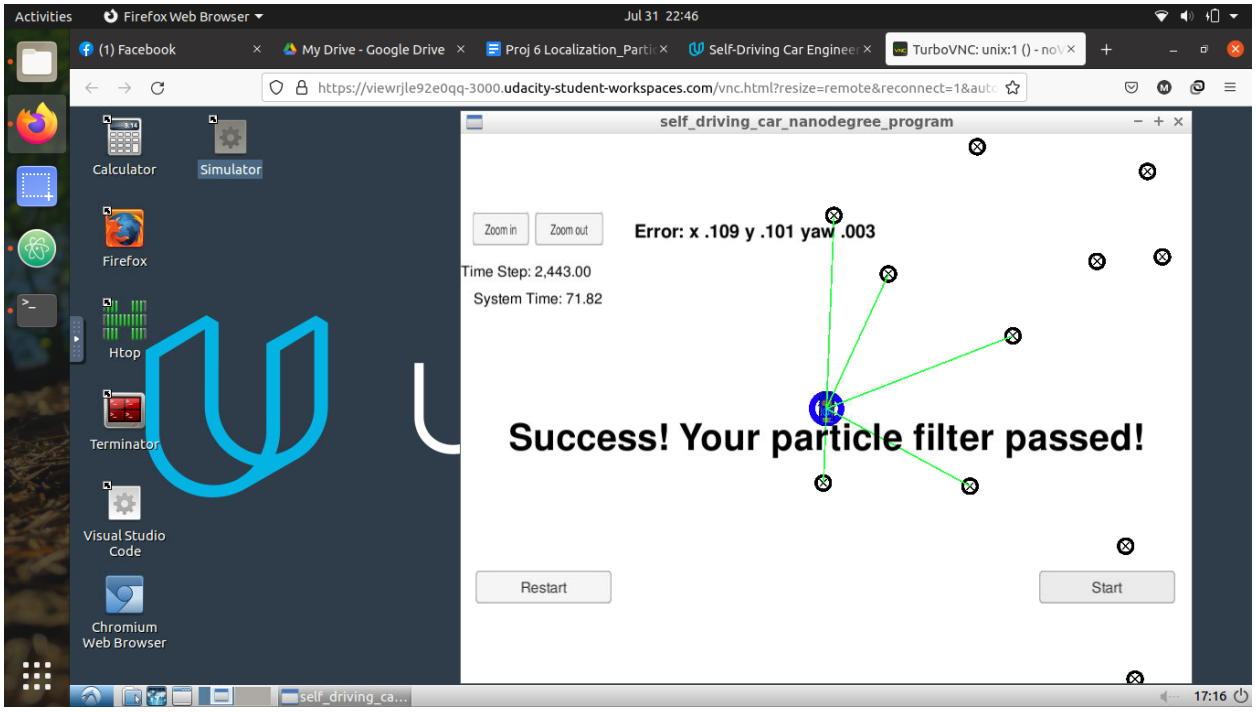
1. Initialize ‘n’ particles based on the GPS information and noises
2. Predict the next position of the particle using the motion model with noises added
3. Calculate a weight for each particle
    a. Transform the observations from vehicle coordinates to map coordinates
    b. Find the closest landmark for each observation
    c. Weight = Prob of landmark position in a multivariate normal distribution with mean as the observation.
4. Sample ‘n’ particles by replacement from the current pool based on their weights
5. Go to step 2

Implementation details

- Number of particles = 400
- Error : Xerr = 0.109, Yerr = 0.101, Yawerr = 0.004

I tested the program for a different number of particles. The following chart shows the decrease in the error with increasing number of particles. For 800 particles, however, the program failed to reach the goal within the time limit. There are no significant gains in increasing the number of particles beyond 400.




Activities
Firefox Web Browser
Jul 31 22:46
(1) Facebook
My Drive - Google Drive
Proj 6 Localization_Parti
Self-Driving Car Engineer
TurboVNC: unix:1 () - noV
https://viewrjle92e0qq-3000.udacity-student-workspaces.com/vnc.html?resize=remote&reconnect=1&aut
Calculator
Simulator
Firefox
Htop
Terminator
Visual Studio Code
Chromium Web Browser
self_driving_car_nanodegree_program
Zoom in
Zoom out
Error: x .109 y .101 yaw .003
Time Step: 2,443.00
System Time: 71.82
Success! Your particle filter passed!
Restart
Start
self_driving_ca...
17:16