Victoria Park SLAM

Project Overview

In this project you will be recreating the seminal Victoria Park SLAM demonstration. In this dataset, a truck is driven around an Australian park with an attached lidar functioning as a laser range finder. These range an bearing measurements combined with a model of the truck's motion and GPS measurements is fused together to simultaneously localize the truck in the park, and map the location of trees. Original article which contains more detail can be accessed here. The overall system is diagrammed in Figure 1 below.

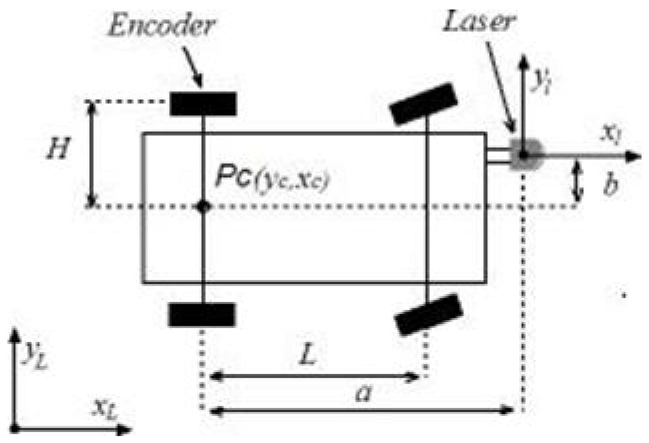

Figure 1: Vehicle geometry

The configuration of the system is that of a typical front axle steering and rear axle drive. The laser is mounted to the front left. The geometry constants are given below in Table 1. All lengths are in meters.

Table 1: Vehicle geometry constants

Constant	Value
L	2.83
H	0.76
a	0.5
b	3.78

System Description

Process Model

We will the Ackermann steering model to describe the motion in the body frame of the vehicle. However, since we also have GPS measurements available, we will need to convert the body frame velocities into local level frame (north east down or east north up, whichever you prefer). The Ackermann model uses a simple three-state vector: $x = \begin{bmatrix} p_x & p_y & \psi \end{bmatrix}$ which corresponds to x-

and y-axis position and the yaw angle relative to the long axis of the vehicle. The continuous time kinematic model relates these three states to the measured x-axis velocity as measured by an encoder on the rear axle (v) and the steering angle of the wheels (δ, or sometimes α).

$$\dot{x}(t) = \begin{bmatrix} \dot{p_x}(t) \\ \dot{p_y}(t) \\ \dot{\psi}(t) \end{bmatrix} = v(t) \begin{bmatrix} cos\,\psi(t) \\ sin\,\psi(t) \\ \frac{tan\,\delta(t)}{L} \end{bmatrix}$$

In discrete time:

$$\begin{bmatrix} p_{x_t} \\ p_{y_t} \\ \psi_t \end{bmatrix} = \begin{bmatrix} p_{x_{t-1}} + v_t t\ cos\ \psi_t \\ p_{y_{t-1}} + v_t t\ sin\ \ \psi_t \\ \psi_{t-1} + \frac{v_t}{L} t\ tan\ \delta_t \end{bmatrix}$$

Where the velocity v_t is related to the measured velocity at the encoder v_{e_t} as such:

$$v_t = v_{e_t}\left(1 - tan\,\delta_t \frac{H}{L}\right)$$

This now requires conversion to the local level frame. Since were are localizing a ground-based vehicle (and the dataset doesn't have any altitude measurements) we will neglect the altitude state and use a modified local level velocity navigation state vector

$$\boldsymbol{x_t} = \begin{bmatrix} p_L & p_\lambda & \psi & v_N & v_E \end{bmatrix}_t$$

We can then update the state in the local level frame according to equation 5.56 from Groves using an approximation that assumes that the velocity is constant across the time interval t :

$$p_{L_t} = p_{L_{t-1}} + \frac{t\left(v_{N_{t-1}} + v_{N_t}\right)}{2R_N\left(p_{L_{t-1}}\right)}$$

$$p_{\lambda_t} = p_{\lambda_{t-1}} + \frac{t}{2}\left(\frac{v_{E_{t-1}}}{R_E\left(p_{L_{t-1}}\right)} + \frac{v_{E_t}}{R_E\left(p_{L_t}\right)}\right)$$

Where R_N and R_E are the principal radii of the Earth in the northward and eastward directions respectively. These values are a function of the Earth's eccentricity ($e = 0.0818191908425$), equatorial radius ($R_0 = 6{,}356{,}752.31425$) and latitude ($L_b$).

$$R_N(L_b) = \frac{R_0\ \left(1\ -\ e^2\right)}{\left(1\ -\ e^2\ sin^2\ L_b\right)^{3/2}}$$

$$R_E(L_b) = \frac{R_0}{\sqrt{1 - e^2 sin^2 L_b}}$$

Measurement Model

We'll incorporate two measurement models for this system. The first will simple GPS measurements:

$$z_{GPS} = \begin{bmatrix} z_L \\ z_\lambda \end{bmatrix}$$

This is can be incorporated into a standard zero-mean normally distributed error model with a sensor noise value of 15 meters along both axis (typical of very old bad GPS, modern satellite signals can get down to approximately 5 meters). To translate degrees latitude and longitude you can use the haversine distance formula. You may also utilize the haversine library from PyPI or any other similar such library.

The second model will be for the laser range finders. There are 361 such beams evenly spaced across 180 degrees. These beams return a distance measured between zero and approximately 80m (consider any measurement greater than 80 meters to be a maximum value, i.e. no reflection and no target). This follows a standard range-bearing measurement model that relate back to the vehicle's position and yaw:

$$z_{RB} = \begin{bmatrix} r \\ \theta \end{bmatrix} = h(\boldsymbol{x}) = \begin{bmatrix} \sqrt{\left(p_{L_i} - p_L\right)^2 + \left(p_{\lambda_i} - p_\lambda\right)^2} \\ atan\dfrac{p_{L_i} - p_L}{p_{\lambda_i} - p_\lambda} - \psi + \pi/2 \end{bmatrix}$$

Where the coordinates of the landmarks are $(p_L, p_\lambda)_i$.

Dataset description

I have taken the original data files from the linked webpage and consolidated them into a single .csv file for you that is attached to this project. This file is indexed to the first column which is time in seconds. The data is sampled at 10Hz. The latitude and longitude columns should be used as either your ground truth or your GPS-based position measurements. Steering and speed are for the system propagation. The range measurements correspond to the laser_x columns. Again, you should filter these measurements to ignore any measurement greater than 80 meters as that represents a "no-reflection" measurement. You may also wish to consider ignoring some returns that are too close as well.

Project Requirements

Develop a SLAM implementation of your choice. I recommend EKF SLAM or FastSLAM. Show and discuss the performance of your system based on the following metrics: vehicle position error, overall path and the location and certainty of the number of landmarks. I will leave this project fairly open-ended. An acceptable project will implement one SLAM method that accurately tracks the vehicle's position and the position of the landmarks. Graphs, tables, and/or figures presenting relevant performance metrics will be professionally presented and discussed. A marginal project will make minimal effort to present this information in a report or have an implementation that is faulty.

Some suggestions for how to excel

In the linked original paper, there is a discussion (section 3.1) on how to consolidate and extract more relevant features by considering the diameter of the tree trunk. Using this consideration you can collect individual range and bearing measurements into the detection of a single feature and not several.

If you're up for it, you could compare and contrast two different SLAM methods analyzing their strengths and weaknesses in this dataset.

Do not use GPS! Initialize off of the the first position fix an use the speed and steering angles to perform dead reckoning and only bounding error with landmark detection. Use the measured GPS position as truth and compare the SLAM implementation's performance to those measurements.