Milan Gunasekera

Homework 3

Introduction:

The goal is to create a mapping of an environment using only LIDAR sensors, wheel odometry, and IMU readings. In this report, I show the results of displaying the odometry readings alone, along the map space.

Methods:

Dead Reckoning using wheel odometry data:

New positions are based on the following:

Instructions

* In the directory, run slam.py
* Enter path of where data can be found
  + Either data/ or test/
  + Specifically works for encoding readings

Results:

To best display trajectory along the odometry map, I decided to use a vector field, using the pose as well as the gradient vectors created at each update step. We can see the results for the two testing data below. The training plots can be seen in the Results folder:

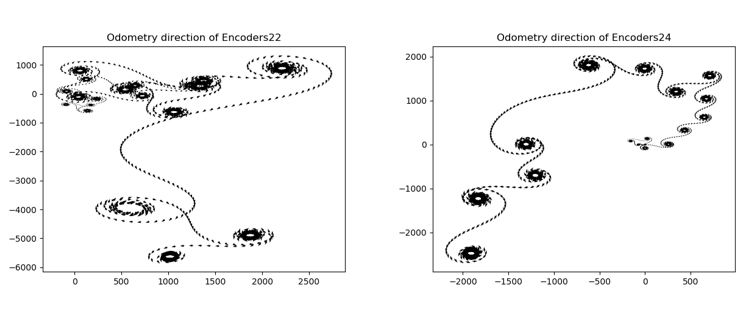
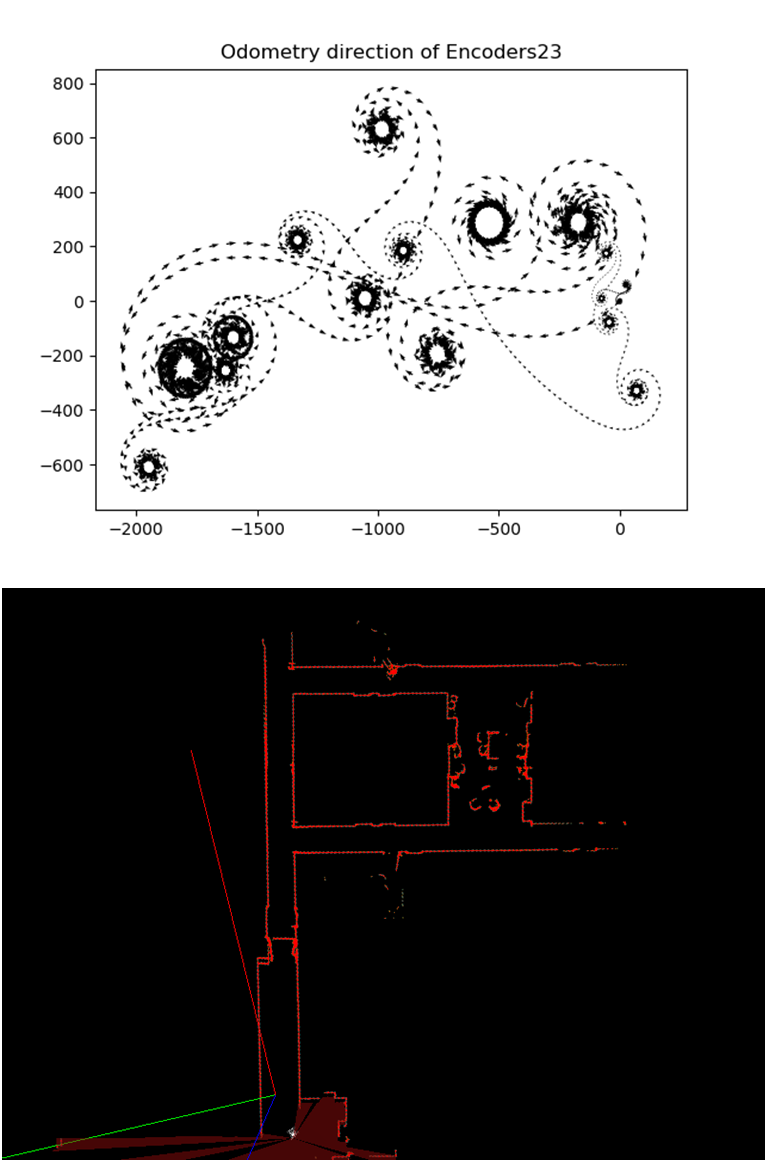


Fig. 1: Encoder values for test-set: **Top-left:** Encoder 22; **Top-right:** Encoder 24



Limitation of odometry:

* Integrative nature of Odometry readings leads to increased in localization over time (Fig. 3)
* When the robot is on standstill, it appears that the pose estimation continues to change
* This odometry reading does not stay consistent with localization if the values are subsampled with larger time gaps (Fig. 4)
  + This is likely because of the integrative nature of the dead Reckoning algorithm. Particle filters with a perception model (using LIDAR data) would have been needed.

Fig. 3: Odometry reading from encoder 23, compared to reference map. It is clear that the shape of the odometry does not appear to traverse the shape of the corridor.

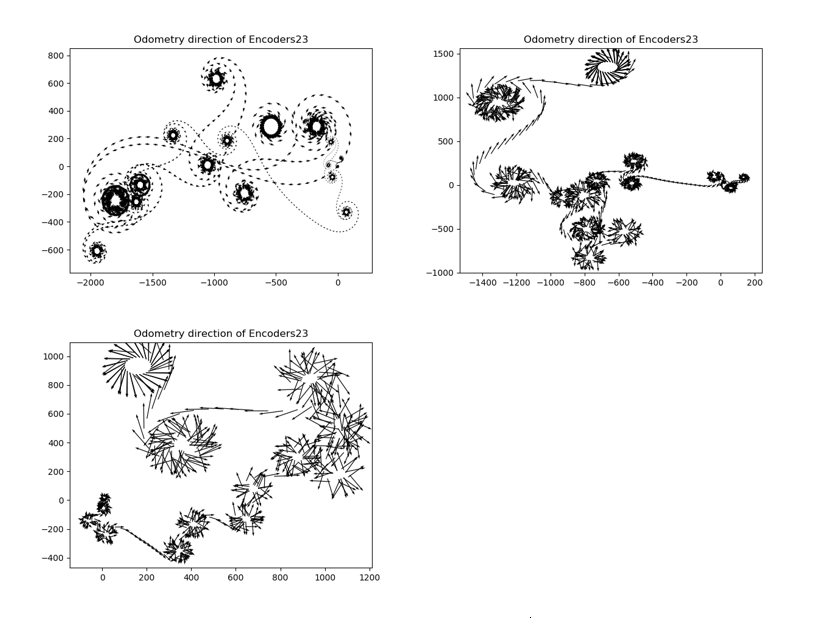


Fig.4 Encoder values with different subsampling schemes. **Top-left**: every sample collected ts=1; **Top-right:** ts=2; **Bottom-left:** ts=5. One can notice that the origin (0,0) shifts as well with each scheme as well.