

Лабораторная работа 9 — SLAM

FastSLAM — это метод для одновременной локализации и построения карты на основе фильтра частиц Рао–Блэквелла. Для оценки положения робота в пространстве используется фильтр частиц. При этом каждая частица “несет” карту окружающего пространства, которую использует для локализации. Для построения карты используются внешние ориентиры и фильтрация Калмана, с помощью которой оценивается среднее положение и ковариация ориентиров.

Реализуйте алгоритм FastSLAM на базе внешних ориентиров.

Используйте готовый проект **FastSLAM_framework**, в котором уже описана необходимая для работы алгоритма FastSLAM последовательность действий и реализована визуализация.

В архиве **FastSLAM_framework** содержатся следующие папки:

data содержит файлы с данными о мире и показаниями датчиков.

code содержит “заготовку” для написания алгоритма FastSLAM.

Вы можете попробовать запустить алгоритм FastSLAM в терминале: `python fastslam.py`. Однако он будет работать корректно только после того, как вы дополните код.

- Допишите функцию `sample_motion_model`, реализовав модель процесса на основе одометрии и осуществив сэмплинг. Функция генерирует новые положения, используя старые значения положений, измерения одометрии δ_{rot1} , δ_{trans} и δ_{rot2} и шумовые параметры модели процесса: $[\alpha_1, \alpha_2, \alpha_3, \alpha_4] = [0.1, 0.1, 0.05, 0.05]$.
- Допишите функцию `eval_sensor_model`. Эта функция реализует модель измерений в фильтре частиц Рао–Блэквелла. При этом используется датчик, измеряющий расстояние и относительный угол. На вход функции принимаются частицы и наблюдения ориентиров. Для каждой частицы функция обновляет карту и вычисляет вес w . Также необходимо учесть зашумленность датчиков:

$$Q_t = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

- Допишите функцию `resample_particles`, осуществив отсев. Функция принимает в качестве входных данных набор частиц и соответствующие веса и возвращает набор частиц, прошедших отсев.

Подсказки:

Для считывания данных, полученных с датчиков, и данных об ориентирах используются словари. Словари обеспечивают более простой способ доступа к структурам данных на основе одного или нескольких ключей. Функции `read_sensor_data` и `read_world_data` в файле `read_data.py` считывают данные из файлов и создают соответствующий словарь с отметками времени в качестве первичных ключей.

Чтобы получить доступ к данным датчика из словаря `sensor_readings`, используйте:

```
sensor_readings[timestamp, 'sensor']['id']
sensor_readings[timestamp, 'sensor']['range']
sensor_readings[timestamp, 'sensor']['bearing']
```

Чтобы получить доступ к данным одометрии, используйте:

```
sensor_readings[timestamp, 'odometry']['r1']
sensor_readings[timestamp, 'odometry']['t']
sensor_readings[timestamp, 'odometry']['r2']
```