

Graph SLAM Seminar

Mikhail Korobkin, Dmitri Kovalenko

Yandex SDC

2021.04.02

Зависимости

python3, nosetests, matplotlib, numpy

Необходимые правки

Участки кода, которые требуют реализации, можно найти поиском по `TO_IMPLEMENT`, для каждого такого участка указано, к какой задаче он относится, например: `Seminar.Task#1`, `Homework.Task#2`.

Задача 1

Обеспечить загрузку вершин графового представления задачи. Метод `_init_pose_vertices` класса `Optimization` будет вызван при конструировании класса и должен заполнить поле `_pose_vertices` позами автомобиля во все моменты времени. Индекс позы в списке `_pose_vertices` равен моменту времени. Представить позу в виде объекта: `SE2Vertex([x_position, y_position, orientation])`. Позы автомобиля получить по данным инициализации и управления (Подробнее в методичке, секции: данные, модель движения автомобиля).

Для правильно реализованной функции юнит-тест должен проходить:

```
nosetests -s unit.test_graph:test_graph_init
```

Визуализация конфигурации загруженного графа должна выглядеть качественно верно (показана пройденная автомобилем траектория), тест выведет путь к визуализационному изображению в терминал.

Задача 2

Обеспечить загрузку априорных ребер. Априорное ребро `PriorEdge` обеспечивает ограничение на положение 0-ой вершины, используя данные события инициализации. Подробнее про фабрики рёбер в методичке.

Реализовать метод `PriorEdgeConstraintBuilder.add_event` и класс `PriorEdge`, поддерживающий инициализацию полей `inf`, `error` и метода `compute_error`.

Для правильно реализованного кода юнит-тест должен проходить:

```
nosetests -s unit.test_edges:test_prior_edge
```

Задача 3

Реализовать функцию потерь одометрического ребра. Подробнее в методичке, раздел обратная модель движения.

Для правильно реализованной функции все 6 юнит-тестов должны проходить:

```
nosetests -s unit.test_edges:TestOdometry
```

Задача 4

Обеспечить загрузку одометрических ребер. Полученная реализация позволяет показать сходимость оптимизационной задачи даже при внесении шума в начальную конфигурацию вершин.

```
nosetests -s unit.test_graph:test_graph_optimization_without_landmarks
```