

Graph SLAM Theory

Mikhail Korobkin, Dmitri Kovalenko

Yandex SDC

2021.04.02

Семинар и домашнее задание посвящены оптимизации траектории движения автомобиля в искусственной среде. Время считается дискретным и определено в условных единицах. Скорости определены в пройденных метрах и радианах за условную единицу времени. Целью является нахождение оптимального (в терминах наименьшей квадратичной ошибки) решения для координат автомобиля во все моменты времени на траектории.

Данные

Каждому варианту соответствует файл, описывающий все наблюдения и сигналы управления, полученные автомобилем по мере прохождения траектории и имеет следующий вид:

example-timeline.json

```
[
  [...], # Список событий, произошедших в момент 0
  [...], # Список событий, произошедших в момент 1
  ... # и т.д
]
```

Где событием выступает ассоциативное хранилище данных о наблюдении или сигнале управления. В рамках задачи встретятся события трёх типов, при этом каждое из них будет содержать как минимум поля `time`, `type` и другие, специфичные для этого типа:

- **Инициализация** (`type=init`). Используется для сообщения начального положения автомобиля, хранит поле `pose = [x, y, orientation]`. В каждом задании только 1 событие такого типа.
- **Управление** (`type=control`). Сигнал управления в поле `command = [v, w]` хранит линейную и угловые скорости в системе координат, связанной с автомобилем; Кроме того, хранит калибровочные параметры в поле `alpha = [a1, a2, a3, a4, a5, a6]`.

- **Наблюдение маяка** (type=point). Считаем маяки неориентированными объектами (для них можно знать только картезианские координаты, но не их поворот), для которых задача распознавания решена: гарантируется, что сделанные в разные времена наблюдения маяков с одинаковыми значениями поля index - это наблюдения одного и того же объекта. Поле measurement - измеренное положение маяка относительно автомобиля в момент времени time. Ковариация измерения поставляется в поле $Q = [var_1, cross_{1,2}, cross_{1,2}, var_2]$

Системы координат

Локальной системой координат назовем систему, связанную с корпусом автомобиля, ось X направлена вперёд, ось Y - влево. Нулевая ориентация означает направление, сонаправленное оси X. $\frac{\pi}{2}$ - оси Y. Положительное изменение ориентации интерпретируется как поворот против часовой стрелки. Все системы координат в данной задаче евклидовы, и глобальная система координат определяется инициализационной позой автомобиля.

Модель движения автомобиля

При управлении автомобилем посредством угловой и линейной скоростей, можно показать, что на временном интервале, где скорости неизменны, автомобиль движется по окружности с радиусом $r = \left| \frac{v}{w} \right|$ и центром в x_c, y_c .

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} x - \frac{v}{w} \sin \theta \\ y + \frac{v}{w} \cos \theta \end{bmatrix} \quad (1)$$

Тогда, поза к моменту времени t примет следующее значение, при условии, что известны скорости, временной интервал, и поза в момент времени $t - 1$.

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} \frac{v}{w} \cdot [-\sin \theta_{t-1} + \sin(\theta_{t-1} + \omega \Delta t)] \\ \frac{v}{w} \cdot [\cos \theta_{t-1} - \cos(\theta_{t-1} + \omega \Delta t)] \\ \omega \Delta t \end{bmatrix} \quad (2)$$

Фабрики рёбер

Загруженный файл **example-timeline.json** с данными, описанными выше, обрабатывается методами `_init_pose_vertices` и `_init_constraints` из класса `Optimization`. Последний из них, создает всевозможные фабрики, которые по мере получения данных, создают рёбра (объекты-наследники класса `Edge`), которые затем встраиваются в граф с помощью классов `Constraint`. Существующие фабрики:

- `PriorEdgeConstraintBuilder`
- `OdometryConstraintBuilder`
- `LandmarkConstraintBuilder`

Обратная модель движения автомобиля

Для двух поз автомобиля, $[x_{t-1} \ y_{t-1} \ \theta_{t-1}]^T$ и $[x_t \ y_t \ \theta_t]^T$, соединенных ребром, получим оценки параметров движения.

Центр окружности, по которой автомобиль двигался на этом интервале, может быть выражен как

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} \frac{x_{t-1}+x_t}{2} + \mu(y_{t-1} - y_t) \\ \frac{y_{t-1}+y_t}{2} + \mu(x_t - x_{t-1}) \end{bmatrix}$$

Решение будет не вырожденным, если угловая скорость не близка к 0. Тогда:

$$\mu = \frac{1}{2} \frac{(x_{t-1} - x_t) \cos \theta + (y_{t-1} - y_t) \sin \theta}{(y_{t-1} - y_t) \cos \theta - (x_{t-1} - x_t) \sin \theta}$$

При этом, радиус окружности будет равен:

$$r = \sqrt{(y_t - y_c)^2 + (x_t - x_c)^2}$$

А изменение ориентации:

$$\Delta\theta = \text{atan2}(y_t - y_c, x_t - x_c) - \text{atan2}(y_{t-1} - y_c, x_{t-1} - x_c)$$

Оцененные параметры управления:

$$\hat{u} = \begin{bmatrix} \hat{v} \\ \hat{w} \end{bmatrix} = \Delta t^{-1} \begin{bmatrix} r \cdot \Delta\theta \\ \Delta\theta \end{bmatrix}$$

Подробнее в [1], глава Robot Motion.

Список литературы

- [1] Sebastian Thrun. “Probabilistic robotics”. в: *Communications of the ACM* 45.3 (2002), с. 117–137.