

Slide 1. Титульный

Здравствуйте, уважаемые члены аттестационной государственной комиссии и преподаватели.

Представляю вашему вниманию дипломный проект “Исследование алгоритмов локального планирования траекторий колесных роботов”

Slide 2. Цель и задачи

Навигация автономных мобильных роботов – быстроразвивающаяся область.

Автономные роботы должны безопасно и эффективно перемещаться из точки А в точку Б с учетом времени, расстояния, энергии и других факторов.

За построение маршрута от одной точки к другой отвечает глобальный планировщик, вооруженный статической картой.

Заранее составленная траектория движения робота не позволяет полностью автоматизировать перемещение

Окружение и препятствия постоянно изменяются, а некоторые маршруты не могут быть полностью достижимы для тех или иных роботов.

Мобильный робот может иметь ограниченный угол поворота и скорость движения.

Целью работы является проведение оценки различных алгоритмов локального для робота с дифференциальным приводом и ограниченной видимостью.

Для достижения заданной цели необходимо выполнить 4 последовательные задачи:

1. Изучить существующие алгоритмы локального планирования для мобильных роботов.
2. Создать симуляцию для исследования алгоритмов.
3. Определить метрики для сравнения алгоритмов.
4. Сравнить алгоритмы локального планирования по заданным метрикам.

Slide 3. Классификация

Исследование в области планирования пути для роботов начали бурное развитие в 1979 году. Тогда была представлена концепция конфигурационного пространства (набор конфигураций, которые направляются роботу)

Задача навигации роботов в основном развивалась для нахождения оптимальных решений, касающихся:

1. Планирования пути – то есть расчета маршрута, по которому необходимо следовать чтобы достичь заданной цели
2. Управления движением – то есть расчета сигналов, посылаемых роботу, для отслеживания заданного маршрута

Глобально алгоритмы планирования движения делятся на две большие категории, в соответствие с их принципами и эпохой, в которую были изобретены:

1. Традиционные
2. На основе машинного обучения

На сегодняшний день алгоритмы локального планирования на основе машинного обучения направлены на “беспилотные летательные аппараты”

Поэтому в данной работе исследованы традиционные алгоритмы локального планирования, больше направленные на колесных роботов

Slide 4. Традиционные методы

Несмотря на то, что исследование современных алгоритмов на основе машинного обучения не направлено для использования с колесными роботами

Среди традиционных алгоритмов тоже не всякий способен обеспечивать расчет траектории в неизвестной среде

Алгоритмы поиска по графу требуют знания о всей территории и занимают длительное время расчета для больших пространств

Сэмплинг-методы основаны на принципе случайного распространения возможной траектории, существуют методы, позволяющие брать в учет динамику робота, однако для таких алгоритмов требуется обширное знание об окружающей местности, то есть дальний обзор во всех направлениях

Интерполяционные кривые - представляют собой способ сглаживания глобального маршрута и расчета кривой, по которой может двигаться колесный робот

Реактивные алгоритмы - направлены на то, чтобы предоставлять реакцию на окружение путем постоянного перезапуска. Такие алгоритмы как можно быстрее реагируют на данные с сенсоров и выбирают наиболее подходящие команды скорости.

Slide 5. Алгоритмы к рассмотрению

Выбранные к рассмотрению алгоритмы также можно разделить на две категории:

- Конфигурационное пространство - выбор наилучшей команды скорости из возможных
- Нелинейная оптимизация - расчет траектории в пределах опорных точек с наложенными ограничениями

К первой категории относятся Dynamic Window Approach и Trajectory Rollout.

Ко второй - Timed-Elastic-Band и Model Predictive Control

DWA выбирает конфигурацию, согласно ограничениям ускорений, лишь для одного шага симуляции на всем временном горизонте

TR выбирает скорость, согласно ограничениям ускорений, на всем временном горизонте

TEB использует многокритериальную оптимизацию, применяется только к автомобилеподобным роботам.

Целевые функции фиксированы, а ограничения аппроксимируются полиномиальными функциями

MPC - задача сводится к квадратичному программированию, накладываемые ограничения не аппроксимируются

и могут принимать любой вид. Применим к любым динамическим системам (например, двухколесный велосипед)

Slide 6. Симуляция робота и пространства

Алгоритмы применялись для определенного типа колесных роботов. Робот с дифференциальным приводом.

Сама модель робота:

- Представляет из себя параллелепипед размером 50 на 30 на 7 см.
- По бокам которого находятся колеса диаметром 20 см и шириной 4 см.
- Спереди у робота находится лазерный сканер, сзади - опорный ролик.

Робот описан в URDF (расшифр. как unified robot description format). Это XML формат описания роботов, встроенный в ROS.

В качестве окружения выступает три различных полигона, построенных в симуляторе Gazebo. Каждый полигон может состоять из высоких стен, ограничивающих пространство и различных препятствий в виде простых геометрических объектов.

Роботу известная информация о стенах.

Slide 7. Метрики для сравнения

Для сравнения результатов работы алгоритмы определен набор метрик:

- Длительность движения - время, за которое робот проезжает от точки старта до точки назначения
- Средняя скорость - средняя линейная скорость движения робота
- Максимальная скорость - максимальная линейная скорость движения робота
- Колебание скоростей - насколько сильно менялась команда скорости от времени движения
- Колебание движения - насколько резким было движение робота
- Отклонение от маршрута - показывает, насколько траектория движения совпадает с глобальным маршрутом

Метрики, которые необходимо максимизировать нормировались наибольшим значением

Метрики, которые необходимо минимизировать брались обратными к нормализованным минимумом

Slide 8. Сравнение алгоритмов

После того как каждый алгоритм был запущен в каждой среде с одинаковой начальной и конечной позицией.

С помощью numpy, pandas и scipy были подсчитаны значения метрик для каждой полученной траектории.

После нормировки, выполненной по значениям результата каждого алгоритма для отдельной карты, значения критериев суммировались для каждого алгоритма.

Таким образом получена оценка траектории для разных типов среды, а именно:

1. Полностью известная среда - препятствия в среде и на карте робота совпадают
2. Частично известная среда - только часть препятствий среды есть на карте
3. Неизвестная среда - робот начинает с пустой картой

Наиболее подходящим вариантом для робота с дифференциальным приводом, согласно определенным метрикам, является Timed-Elastic-Band.

Slide 9. Заключение

Не смотря на существования множества подходов следования глобальному маршруту, исследование

алгоритмов локального планирования показало, что алгоритм должен удовлетворять требованиям динамики и кинематики.

Этим требованиями удовлетворяют:

- Dynamic Window Approach
- Trajectory Rollout
- Timed-Elastic-Band
- Model Predictive Control

Для применения алгоритмов разработано ПО для управления роботом на базе ROS, в частности Navigation Stack.

Данному ПО можно сообщить начальную карту местности, планировщик, и точку назначения.

Также создана симуляция пространства трех различных полигонов и робота и дифференциальным приводом.

Для сравнения алгоритмов определен набор метрик, позволяющий оценить траекторию и динамику движения робота,

построенную с помощью локального планировщика.

Стоит принять во внимание, что данные метрики оцениваются в соответствие со средой и условиями движения.

В экспериментальных условиях каждая метрика принята равнозначной.

В ходе выполнения экспериментов выявлено сильное преимущество алгоритма Timed-Elastic-Band, Который направлен на то, чтобы робот максимально быстро доехал до точки назначения.

Движение получается довольно резким, что не является проблемой для мобильного маневренного робота.

Slide 10. Апробация

Для того чтобы развернуть данную систему на роботе, необходимо:

- Ubuntu:Focal (20.04)
- ROS Noetic

Следовательно системные требования должны удовлетворять условиям Ubuntu и ROS:

- Процессор: ^1 ГГц (архитектура amd64, arm32/64)
- Оперативная память: ^1 ГБ
- Физическая память: ^2.5 ГБ

Все указанные эксперименте проводились на устройстве с:

- Процессором: AMD RYZEN 5 3500U
- Графическим ядром: AMD Radeon Vega 8 Graphics
- Оперативной памятью: 6 ГБ DDR4

Стоит принять во внимание, что большая часть ресурсов уходила на симуляцию.

Репозиторий проекта с инструкциями по запуску находится по указанной ссылке.