электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

УДК 62-529

Построение и анализ векторных топологических карт местности для движения робота

А.А. Киндяков

Студент, кафедра «Специальная робототехника и мехатроника»
МГТУ им. Н.Э. Баумана, г. Москва, Россия

Научный руководитель: Рубцов В.И., к.т.н., доцент кафедры
«Специальная робототехника и мехатроника»
МГТУ им. Н.Э. Баумана, г. Москва, Россия

МГТУ им. Н.Э. Баумана
akindyakov@gmail.com

Введение

Алгоритмы навигации широко используются сегодня в технике, начиная от бытовых уборщиков и заканчивая сложными комплексными решениями для движения по автодорогам. Алгоритмы делятся на несколько видов, наиболее используемые из которых это матричные алгоритмы для простых местностей и задачи на теорию графов, когда пути перемещения известны. Намного меньше используется векторные контурные алгоритмы.

В работе рассмотрены теоретические аспекты для реализации простейших векторных карт, их анализа, построения на их основе путей следования и использование в качестве локальных решений топологических задач методами теории графов. Такое решение является достойным конкурентом для матричных алгоритмов. Последние имеют достаточно высокую вычислительную стоимость и требуют достаточно больших объемов памяти для хранения даже с применением сжатия для сильно разряженных матриц.

Для конкретности определим, что рассматриваемое робототехническое транспортное средство построено по схеме с двумя независимо управляемыми соосными колесами, которые

управляются электродвигателями, с дискретными относительным датчиком угла поворота. Положим так же, что робот установлен на ровной, абсолютно шероховатой поверхности.

Способы представления плоских карт

Растровый способ.

Самый простой способ хранения информации о пространстве, в котором движется транспортное средство, аппроксимированного до плоскости. Способ отличается относительной простотой алгоритмов анализа карт на проходимость, построения путей следования, а так же в формировании карт на основе данных, например, с дальномера. Облако точек пропускается через фильтры, например, Кальмана или вероятностные фильтры, далее облако проецируется на плоскость. Далее плоское облако переносится в неподвижную систему отчета, разбивается на группы с заданным шагом. Группы оцениваются еще рядом фильтров, строится матрица растровой карты.

Векторный способ.

Назовем точку на плоскости движения робота достижимой, в случае, когда робот может самостоятельно попасть в такое положение, когда точка его внешней стороны совпадет с рассматриваемой. И назовем не достижимой - точку, когда робот не может самостоятельно попасть в нее. Отсюда, точки пространства по этому принципу можно сгруппировать. Группы разделяют пространство на зоны проходимости. Выделяем только границы этих зон и сохраняем в виде контуров [1]. Есть несколько способов хранения контуров: хранить точки контура в абсолютных координатах, хранить отрезки и т.п. Но самый простой — хранить контур в виде однонаправленного списка: сохраняем одну стартовую точку в абсолютных координатах, остальные участки контура храним в виде относительных приведенных векторов. Такой способ представления удобен для:

- переноса контура — смещаем только стартовую точку.

- поворота относительно стартовой точки — поворачиваем каждый вектор, в отдельности, умножая его на матрицу поворота — что приводит к повороту контура на заданный угол.

- поворота относительно произвольной точки — поворачиваем контур относительно базовой точки, затем поворачиваем положение базовой относительно заданной поворотной.

- операции замены участка в случае уточнения этого куска карты.

- операции масштабирования — просто масштабируем каждый вектор в отдельности.

Чтобы обозначить области внутри препятствий и снаружи ввели объект контур с

вектором направленным от тела препятствия. Отсюда карта в самом простом виде это просто набор препятствий, ими же обозначаются и границы карты.

Построение простого пути из точки А в точку Б в векторной карте

Задача сводится к выбору пути обхода встретившихся препятствий или другими словами построение так называемой тени контура на заданном расстоянии от него. Построение тени не что иное, как вычислению для всех векторов контура, векторов параллельных заданному начинающихся в определенной точке и заканчивающихся на прямой. Тень, построенная таким образом, есть простейший путь обхода. Вычислив точки пересечения этого пути с коридорами прямого движения и прочими препятствиями, получим точки входа и выхода из обходного маневра. Результат такого построения можно видеть на Рис. 1.

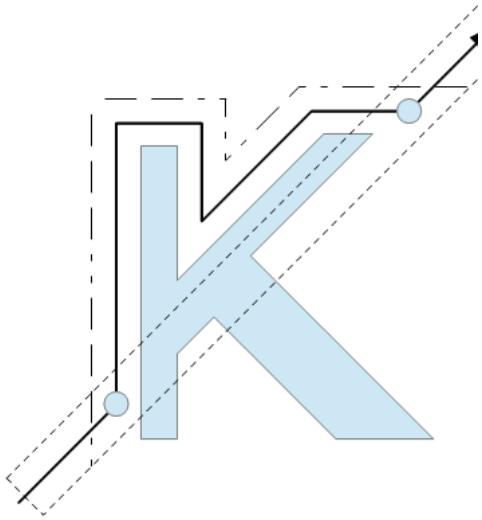

Рис. 1. Простой путь обхода

Отсюда вытекают следующие **задачи:**

1. Спрямление. Весьма очевидно, что таким образом построенный путь сложен для выполнения роботом, а также приведет к частым остановкам и сменам направления.
2. Выбор направления при получении 2х направлений обхода.
3. Решения задачи в условиях сложного контура. Например (рис 2)

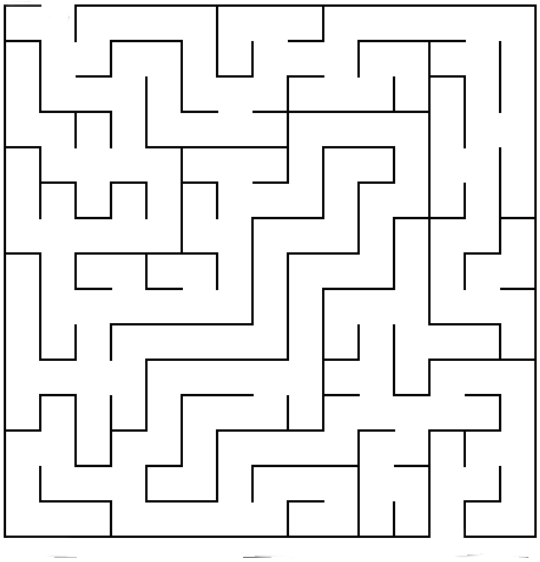

Рис. 2. Сложные помещения

Спрямление. Прохождения полученной траектории может быть осложнено, во-первых, наличием резких поворотов, а во-вторых, их количеством и, в-третьих, это увеличивает длину пути. Пронумеруем векторы в односвязном списке контура от 0 до n.

Алгоритм заключается в рекурсивном проходе по пути. Пронумеруем векторы в списке с 0 по n. Тогда алгоритм будет выглядеть как:

a) Объединяем 0 и 1 векторы заменяя их вектором их суммы, это вектор 0'.

b) Смотрим не пересекается ли участок коридора на основе 0' с рядом лежащими контурами препятствий.

c) Если он не пересекается с препятствиями, повторяем a и b пункты с векторами 0' и 2.

d) Если пересекается — выполняем пункты a и b для векторов 2 и 3.

Пример выполнения алгоритма (рис 3).

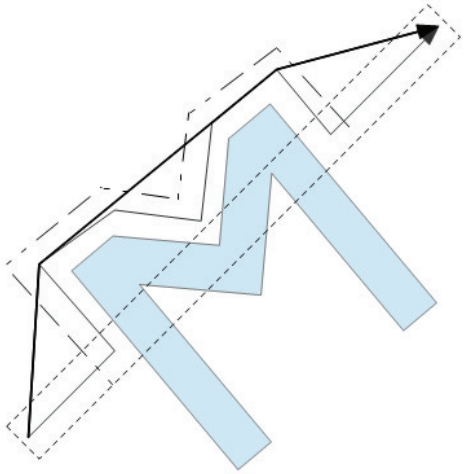

Рис. 3. Спрямление траектории

Направление. При обходе препятствия — получаем два пути — выбираем тот, длина которого меньше.

Траектория следования внутри коридора

Коридор, обозначенный ломаной линией и доверительной зоной заданной вектором, будем использовать для построения управления движения.

В предыдущей работе рассматривалось построение кривых траекторий по таким коридорам. Результатом была реализация поученного алгоритма на C++. Суть заключалась в построении траектории движения внутри коридора из отрезков и участков окружностей, в условиях дискретного задания скоростей мобильной платформы с достаточно большим шагом дискретизации, таким, что им уже нельзя пренебречь без ущерба для точности движения. У такого подхода есть проблема — динамический удар, возникающий при переходе с прямого движения на окружность, из-за скачка ускорения. Это плохо сказывается на точности, плавности движения(робототехническая платформа на которой тестировались алгоритмы, а также в клиентский софт которой планируется внедрение данного алгоритма, взаимодействует с человеком и визуальная плавность движения играет важную роль). Так же это станет причиной преждевременного износа механической части шасси робота.

Этот вопрос подробно исследуется в [2] и [3]. В ней рассматривается склейка окружностей переходными спиралями. Спирали легко получаются из уравнений связей.

$$\begin{cases} \dot{x} = (a \cdot \dot{\theta} + r \cdot \dot{\varphi_1}) \cdot \cos(\theta) \\ \dot{y} = (a \cdot \dot{\theta} + r \cdot \dot{\varphi_1}) \cdot \sin(\theta) \\ \dot{\varphi_2} = \dot{\varphi_1} + \frac{2 \cdot a}{r} \cdot \dot{\theta} \end{cases} \quad (1)$$

В работе рассматривается спирали с непрерывным изменением угловой скорости, что, очевидно, не применимо для общего случая мобильной платформы и, в частности, для нашей. Имеем дискретное задание скорости с конечным временем переходного процесса при смене скорости с одного значения на другое. Для реализации спирального движения необходимо непрерывное изменение скорости с постоянным ускорением.

Необходимо определить моменты переключения скорости.

В общем случае угловые скорости должны измениться на разные величины, а потому моменты переключения между шагами не будут совпадать. Стоит отметить что, так как система дискретная то скорости всегда кратны некоторому значению Δω.

Наибольшее количество переходов:

$$n_{max} = \frac{\max(\Delta\omega_1, \Delta\omega_2)}{\Delta\omega} \quad (2)$$

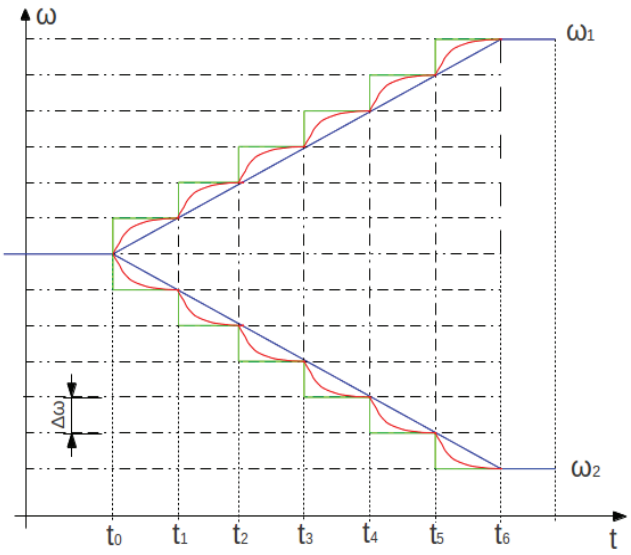

Рис. 3. Переходные процессы на спиральной траектории

Время всего перехода будет определяться максимальным количеством переходов:

$$t_\Sigma = t_{\text{п.п.}} \cdot n_{max} \quad (4)$$

В общем случае передаточную функцию САУ по скорости робототехнической подвижной платформы можно представить в виде апериодического звена в рабочей зоне частот

и входных воздействий. Тогда $t_{\text{п.п.}} = 3 \cdot T$

Но если $\Delta\omega_1 \neq \Delta\omega_2$

Тогда скорость второго колеса необходимо изменять медленнее. Чтобы определить момент включения следующей «ступеньки», поставим в соответствие нашему переходному процессу линейное нарастание скорости

$S = w_{i-1} \cdot \Delta t + \frac{\Delta\omega \cdot \Delta t}{2}$ (5)

$S' = \int_0^{\Delta t} \omega_{\text{п.п.}}(t)dt = \int_0^{3T} \omega_i - \omega_{i-1} \cdot \exp\left(-\frac{t}{T}\right) dt + v_1 \cdot \Delta t \cdot \omega_{i-1} + v_2 \cdot \Delta t \cdot \omega_i$ (6)

Где $v_1 < 1, v_2 < 1$ - означают промежутки времени движения до включения ступеньки и после окончания переходного процесса. Такие что

$v_1 \cdot \Delta t + v_2 \cdot \Delta t + 3 \cdot T = \Delta t$ (7)

$$\int_0^{3T} \omega_i - \omega_i \cdot \exp\left(-\frac{t}{T}\right) dt + v_1 \cdot \Delta t \cdot \omega_{i-1} + v_2 \cdot \Delta t \cdot \omega_i = \mu \cdot \omega_i \cdot T + v_1 \cdot \Delta t \cdot \omega_{i-1} + v_2 \cdot \Delta t \cdot \omega_i$$

Приравняв это к (5)

Обозначив $\Delta t = 3 \cdot T \cdot f, f \geq 1$ получим:

$v_1 \cdot (i-1) + v_2 \cdot i = \left(i - \frac{1}{2}\right) - \frac{\mu \cdot i}{3 \cdot f}$ (8)

Из (7) и (8) получаем

$v_1 = \frac{1}{2} - \frac{1}{f} + \frac{\mu \cdot i}{3 \cdot f}$ (9)

$v_2 = \frac{1}{2} - \frac{\mu \cdot i}{3 \cdot f}$ (10)

Иерархический поиск пути

Для областей пространства достаточно сложных, чтобы их анализ можно было проводить в реальном времени, необходимо иное решение. Достаточно изящное решение применил автор в работе [4]. автор разбивает пространство на отдельные части с простым поиском пути внутри, далее объединяет эти части в связный циклический двунаправленный граф используя в качестве вершин порталы перехода между локациями. Анализирует его алгоритмами Дейксры и A*. Применили идею частично, взяв только идею разбиения большой карты и построения графа на основе частей. В качестве алгоритма анализа графа было решено использовать — классические алгоритмы на Евклидовых сетях [5]. Неоднозначно стоит вопрос о разбиении пространства на локации.

Локации должны отвечать следующим требованиям:

1. Иметь конечное число порталов.

2. Удовлетворять критерию сложности вычисления (два критерия сложность контура и объем локации).

3. Все порталы одной локации должны быть попарно связаны друг с другом внутри.

4. Порталом преимущественно называем узкие проходы между контурами препятствий.

Полученные результаты

В качестве выхода была получена библиотека инструментов для построения векторных карт, разбиения их на локации, получения путей следования внутри локаций, преобразования контуров. А также построение связного двунаправленного графа и анализа его алгоритмами Евклидовых сетей. Хеширование полученных путей, и сохранение графа с локациями и посчитанными путями в xml.

Библиотека написана на C++, в качестве сборочного программного обеспечения использовался «Cmake», что позволит использовать библиотеку на различных платформах.

Исходный код открыт и доступен на «GitHub»: **https://github.com/AKindyakov/neron.git**

В дальнейшем планируется реализовать и другие алгоритмы анализа графов. Дополнить виды локаций растровыми. Так же планируется с помощью контуров реализовать разбиение пространства на весовые области приоритетов и использование их при получении траектории.

Список литературы

1. Введение в контурный анализ; приложение к обработке изображений и сигналов. Фурман Я.А., Кревитский А.В, Передреев А.К., Роженцов А.А., Хафизов Р.Г., Егошина И.Л., Леухин А.Н. 2-е изд.. испр. - М.: Физматлит, 2003. - 592 с. ISBN 5-9221-0374-1.
2. Ю. Г. Мартыненко, "Управление движением мобильных колёсных роботов" Московский государственный университет им. М. В. Ломоносова 2004.
3. Павловский В.Е., Евграфов В.В. "Синтез D2-гладких траекторий для мобильного робота с дифференциальным приводом", материалы научной школы-конференции «Мобильные роботы и мехатронные системы», М.. МГУ, 2004.
4. Плахов А. «Алгоритмы иерархического поиска пути в играх» на конференции разработчиков компьютерных игр 21 - 22 марта 2003 года URL:www.kriconf.ru/2003/index.php?type=thesis.
5. Роберт Седжвик "Алгоритмы на C++" Пер с англ. - М. ООО «И.Д. Вильямс» 2011-1056 с.