Научная статья

Original article

УДК 681.5

ИССЛЕДОВАНИЕ НАВИГАЦИОННОЙ СИСТЕМЫ НА БАЗЕ КОЛЕСНОГО МОБИЛЬНОГО РОБОТА MECANUM

RESEARCH ON NAVIGATION SYSTEM BASED ON MECANUM WHEELED MOBILE ROBOT

Ван Юй, магистрант, МГТУ им. Н.Э.Баумана. кафедра СМ7
Wang Yu, BMSTU, Department SM7, ulyssessci@gmail.com

Аннотация. С быстрым развитием робототехники в последние годы среда применения мобильных роботов становится все более и более разнообразной, а повышение автономности мобильных роботов становится важной темой исследований. В статье исследуется система навигации мобильных роботов. Традиционный дифференциальный мобильный робот имеет такие недостатки, как большой радиус поворота и негибкое действие, которые ограничивают его применение во многих областях, поэтому в этой статье в качестве механизма движения мобильного робота используется колесо Mecanum. В этой статье сначала анализируется кинематическая модель мобильного робота Mecanum, а также создается мобильный робот Mecanum и среда моделирования с использованием платформы моделирования GAZEBO. Планирование пути сочетает в себе алгоритм планирования глобального пути A* и алгоритм планирования локального пути DWA. Для повышения точности навигационной системы в навигационную систему добавлен алгоритм позиционирования AMCL. Результаты экспериментов показывают, что мобильный робот может хорошо выполнять задачу автономной навигации

целевой точки и может реализовать функцию автономного обхода препятствий.

Annotation. With the rapid development of robotics in recent years, the application environment of mobile robots has become more and more diverse, and improving the autonomy of mobile robots has become an important research topic. The article investigates the navigation system of mobile robots. The traditional differential mobile robot has disadvantages such as large turning radius and inflexible action, which limit its application in many areas, so this article uses the Mecanum wheel as the movement mechanism of the mobile robot. This article first analyzes the kinematic model of the Mecanum mobile robot and builds the Mecanum mobile robot and simulation environment using the GAZEBO simulation platform. Path planning combines the A* global path planning algorithm and the DWA local path planning algorithm. To improve the accuracy of the navigation system, the AMCL positioning algorithm has been added to the navigation system. The experimental results show that the mobile robot can perform well the task of autonomously navigating the target point, and can realize the autonomous obstacle avoidance function.

Ключевые слова: Мобильный робот, Колесо Mecanum, Навигация, Моделирование и симуляция

Keywords: Mobile robot, Mecanum Wheel, Navigation, Modeling and simulation.

Введение

В связи с быстрым развитием индустрии робототехники в последние годы увеличивается применение автономных мобильных роботов в сферах транспорта, логистики, социальных услуг и аварийно-спасательных служб [1]. Реализация навигационной функции автономных мобильных роботов в основном заключается в решении задач автономного позиционирования роботов в неизвестных средах, создании карт среды и планировании пути. Основное внимание в этой статье уделяется позиционированию и

планированию пути, поэтому для навигации используются предварительно созданные карты. Неизвестные препятствия будут появляться во время фактического движения робота. Таким образом, только алгоритм глобального планирования не может дать роботу хорошую стратегию предотвращения препятствий. Поэтому необходимо использовать алгоритмы локального планирования пути для обхода временных препятствий. Реализация навигационной функции мобильного робота требует позиционирования робота в окружающей среде, а точность позиционирования напрямую влияет на навигационный эффект. Таким образом, навигационная система, изучаемая в этой статье, включает глобальное планирование пути, локальное планирование пути и позиционирование робота. Колесный мобильный робот Mecanum более гибок, чем традиционный дифференциальный мобильный робот. В данной статье используется колесный мобильный робот Mecanum.

1. Модель робота и проектирование среды моделирования

1.1 кинематическая модель

Мобильный робот Mecanum может двигаться в любом направлении и обладает большей маневренностью, чем традиционный дифференциальный робот [2]. Для управления движением робота необходимо проанализировать модель его движения и установить прямую и обратную кинематические модели.

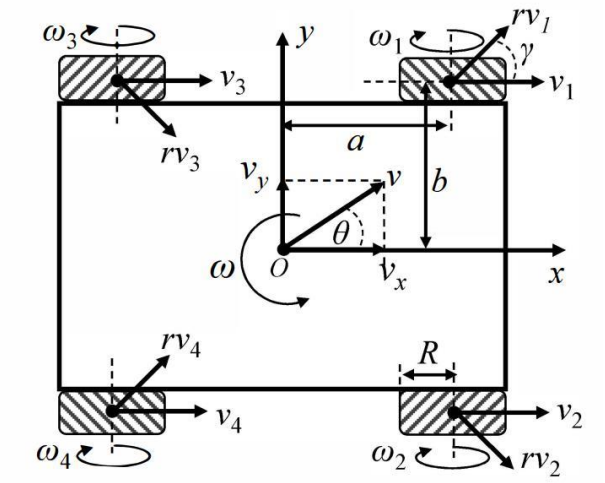


Рисунок 1- Кинематическая модель мобильной платформы Mecanum

Как показано на рисунке 1 [3], Радиус колеса R, Угловая скорость колеса ω_i, где $i \in \{1,2,3,4\}$. Скорость робота в направлении x и направлении y равна v_x、v_y, а скорость вращения равна ω. Расстояние между центром колеса и началом O системы координат шасси в направлении x равно a, а расстояние в направлении y равно b. Использовать a_i и b_i для представления координат центра колеса в направлениях x и y системы координат шасси соответственно, Линейная скорость колеса равна v_i, линейная скорость ролика на колесе равна rv_i, а γ представляет собой угол между линейной скоростью колеса и линейной скоростью ролика.

Для фактической скорости каждого колеса в системе координат шасси:

$$v_i + rv_i \cos\gamma_i = v_x - b_i\omega \tag{1}$$

$$rv_i \sin\gamma_i = v_y + a_i\omega \tag{2}$$

$$v_i = v_x - b_i\omega - \frac{v_y + a_i\omega}{\tan\gamma_i} \tag{3}$$

Поскольку $\tan\gamma_i = \{1, -1, -1, 1\}$, $a_i = \{a, a, -a, -a\}$,$b_i = \{b, -b, b, -b\}$, то:

$$\begin{cases} v_1 = v_x - v_y - a\omega - b\omega \\ v_2 = v_x + v_y + a\omega + b\omega \\ v_3 = v_x + v_y - a\omega - b\omega \\ v_4 = v_x - v_y + a\omega + b\omega \end{cases} \tag{4}$$

Преобразовать в форму угловой скорости:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{R} \begin{bmatrix} 1 & -1 & -(a+b) \\ 1 & 1 & (a+b) \\ 1 & 1 & -(a+b) \\ 1 & -1 & (a+b) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{5}$$

Приведенная выше формула представляет собой уравнение обратной кинематики, и скорость четырех колес может быть получена в соответствии со скоростью движения мобильного робота.

Точно так же скорость робота можно получить по скорости вращения четырех колес. Уравнение прямой кинематики:

$$\begin{cases} v_x = \frac{R}{4}(\omega_1 + \omega_2 + \omega_3 + \omega_4) \\ v_y = \frac{R}{4}(-\omega_1 + \omega_2 - \omega_3 + \omega_4) \\ \omega = \frac{R}{4(a+b)}(-\omega_1 + \omega_2 - \omega_3 + \omega_4) \end{cases} \quad (6)$$

1.2 Модель траектории движения

Предположим, что система координат шасси робота берет центр шасси в качестве начала координат, направление вдоль передней части колеса — это направление x, а боковое направление, перпендикулярное колесу, — это направление y. Для колесного шасси Mecanum имеется скорость v_x вдоль оси x и скорость v_y вдоль оси y, а также угловая скорость вращения ω.

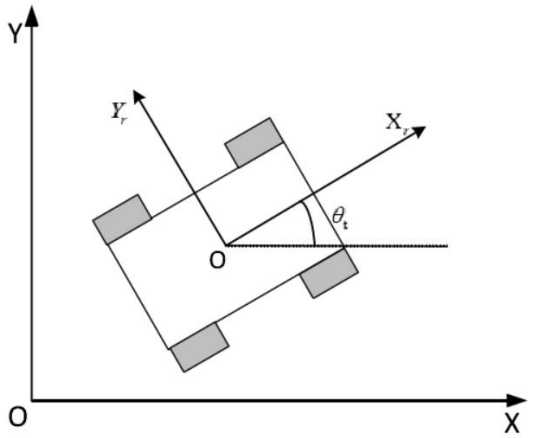


Рисунок 2- Модель расчета траектории колеса Mecanum

Проецирование смещения, создаваемого скоростью в направлении x в системе координат шасси робота в пределах Δt, в глобальную систему координат имеет:

$$\begin{cases} \Delta x_X = v_x \Delta t \cos\theta_t \\ \Delta y_X = v_x \Delta t \sin\theta_t \\ \theta_t = \theta_{t-1} + \omega \Delta t \end{cases} \quad (7)$$

Проецирование смещения, создаваемого скоростью в направлении y в системе координат шасси робота в пределах Δt, в глобальную систему координат имеет:

$$\begin{cases} \Delta x_Y = -v_y \Delta t \sin\theta_t \\ \Delta y_Y = v_y \Delta t \cos\theta_t \\ \theta_t = \theta_{t-1} + \omega \Delta t \end{cases} \quad (8)$$

Следовательно, формула расчета траектории колесного шасси Mecanum в глобальной системе координат:

$$\begin{cases} x_t = x_{t-1} + v_x \Delta t \cos\theta_t - v_y \Delta t \sin\theta_t \\ y_t = y_{t-1} + v_x \Delta t \sin\theta_t \, v_y \Delta t \cos\theta_t \\ \theta_t = \theta_{t-1} + \omega \Delta t \end{cases} \quad (9)$$

Одометр робота использует приведенную выше формулу для расчета положения робота, но существует кумулятивная ошибка, поэтому позиционирование робота не может быть точно достигнуто только с помощью одометра.

1.3 Построение модели робота

Настройка модели робота в GAZEBO включает в себя создание модели моделирования и загрузку контроллера. В этой статье файлы Xacro используются для создания моделей роботов. Как показано на рисунке 3, создана модель мобильного робота Mecanum. На робота устанавливаются радар, камера и другие датчики.



Рисунок 3 - Модель робота в GAZEBO

Для контроллера пакет функций ROS Control, предоставляемый ROS, используется для управления моделью робота в GAZEBO.

1.4 Среда моделирования

В этом документе используется программное обеспечение GAZEBO для создания смоделированной карты окружающей среды и заранее строится двумерная карта сетки с помощью 2D-радара для последующего планирования

пути и позиционирования. Как показано на рисунке 4, длина и ширина карты равны 20 м и 20 м.



Рисунок 4 - Карта среды моделирования

2. Проектирование навигационной системы

Процесс навигации робота в основном делится на два этапа: сначала в соответствии с двумерной картой сетки алгоритм глобального планирования A* используется для планирования оптимального пути от начальной точки до целевой точки, а затем локальный Алгоритм планирования DWA используется для управления движением робота и обхода препятствий. В процессе движения робота возникают ошибки, такие как ошибка одометра и ошибка управления движением, эти ошибки приводят к тому, что робот не может точно определить местоположение и точно перейти к целевой точке. Поэтому в навигационную систему добавлен алгоритм позиционирования AMCL, чтобы исправить эти ошибки и повысить точность позиционирования робота.

2.1 Структура навигационной системы

Навигационная система робота в основном включает в себя модуль планирования глобального пути, модуль планирования локального пути, модуль позиционирования, модуль преобразования координат tf, сенсорный модуль и т. д.

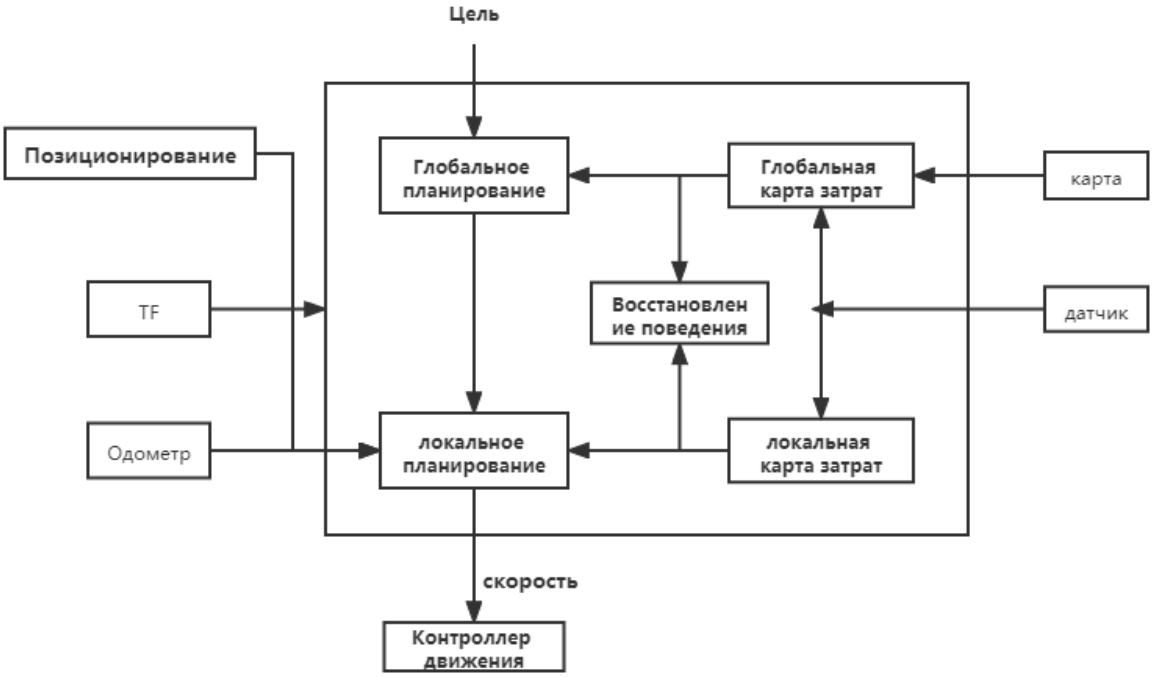


Рисунок 5 - Схема структуры навигационной системы

Как показано на рисунке 5：

- Модуль позиционирования используется для позиционирования на основе построенной сетки-карты, по собственным данным о пробеге робота и особенностям карты использует алгоритм Монте-Карло на основе фильтрации частиц (AMCL) для ремонта робота. Положение положительного робота на известной карте.
- Модуль преобразования координат используется для определения положения робота на карте и установления связи между данными датчика и статической картой.
- Модуль одометра используется для публикации информации об одометре в режиме реального времени.
- Сенсорный модуль используется для выдачи информации о лазерном сканировании в режиме реального времени, чтобы избежать препятствий в окружающей среде.
- Модуль планирования пути включает в себя глобальное, локальное планирование пути и локальную настройку позы. В соответствии с

глобальными и локальными картами стоимости алгоритм A* и алгоритм DWA используются для выполнения глобального и локального планирования пути.

- Модуль контроллера движения используется для получения информации о скорости, запланированной навигационным модулем, а затем для управления движением робота.
- Модуль карты используется для получения построенной растровой карты.

2.2 Алгоритм позиционировании AMCL

В навигационной задаче робота первое, что нужно сделать, это точно определить местоположение на известной карте, только точно определяя свое собственное положение на карте, навигационная задача может быть выполнена правильно. Позиционирование роботов заключается в использовании датчиков для определения собственных координат положения в мировой системе координат. Адаптивный алгоритм локализации Монте-Карло (Adaptive Monte Carlo Localization, AMCL) является одним из основных алгоритмов для решения проблемы перемещения роботов на известных картах [4]. Адаптивная локализация Монте-Карло — это вероятностная задача, в которой робот находит свою собственную позу посредством постоянного движения в двумерном пространстве и использует частицы для отслеживания своей позы на карте [5]. Адаптивный алгоритм Монте-Карло добавляет случайные частицы и выборку KLD (Kullback-Leibler Divergence) к традиционному алгоритму Монте-Карло (MCL) [6]. Добавляя случайные частицы для решения проблемы затухания частиц в таких сценариях, как похищение роботов, и повысить эффективность фильтрации частиц, регулируя размер сбора частиц.

2.2.1 Алгоритм MCL

Основная идея алгоритма управления Монте-Карло исходит из динамического программирования, целью которого является получение

оптимальной конечной стратегии посредством непрерывной итерации стратегии [7].

Этапы расчета MCL (Monte Carlo Localization) следующие:

1) Произвольно генерировать M частиц.
2) Моделирование движения частиц. На основе роя частиц в предыдущий момент алгоритм обновляет рой частиц через кинематическую модель робота и получает положение роя частиц в следующий момент.
3) По данным измерений датчика рассчитывается вес каждой частицы. Затем обновите каждый состояние частицы.
4) Повторная выборка для получения M частиц из всех частиц на основе новых значений веса.
5) Повторить вышеуказанные шаги непрерывно, чтобы добиться эффекта позиционирования.

2.2.2 Метод отбора проб KLD

Фильтр частиц использует среднее значение частиц перед выборкой для аппроксимации ожидаемого значения величины состояния (функция плотности апостериорной вероятности). Чем больше количество отобранных частиц, тем лучше это может отразить реальную ситуацию в теории. Однако вычислительная сложность возрастает с увеличением количества частиц [8]. Поэтому метод выборки KLD вводится в соответствии с Выборка статистических границ приблизительной массы для адаптивной корректировки количества частиц с течением времени [9].

2.3 Алгоритм глобального планирования A*

Алгоритм A* (A-Star) представляет собой алгоритм планирования глобального пути, основанный на картах сетки. Алгоритм A* представляет собой эвристический алгоритм поиска пути, в котором используется эвристическая функция для оценки стоимости прохождения точки до целевой точки, что сокращает время поиска и потребление вычислительных ресурсов.

Ядром алгоритма A* является оценка приоритета следующего узла n, и его функция такова:

$$f(n) = g(n) + h(n) \quad (10)$$

Где:

f(n) — это оценка следующего узла n, который является всеобъемлющим приоритетом узла чем меньше значение f(n), тем выше приоритет. Узел с наивысшим приоритетом является следующим пройденным узлом;

g(n) — стоимость расстояния между следующим узлом n и начальной точкой;

h(n) — его эвристическая функция, оценочная стоимость расстояния между следующим узлом n и целевой точкой, выбор h(n) определяет производительность алгоритма;

Шаги расчета алгоритма A* следующие:

1) Поместите начальную точку в Open List;
2) Определить, пуст ли Open List. Если он пуст, поиск не пройден. Если он не пуст, выбрать точку с наименьшим F в Open List в качестве текущего узла;
3) Удалить текущий узел из Open List и поместить его в Close List;
4) Обход смежных достижимых узлов текущего узла;
5) Повторять шаги с (2) по (4) до конца цикла.

2.4 Алгоритм локального планирования DWA

Основная идея алгоритма DWA (Dynamic Window Approach) состоит в том, чтобы произвести выборку в пространстве скоростей (v, ω), получить несколько выведенных траекторий в соответствии с моделью движения и каждым набором скоростей за фиксированное время, рассчитать стоимость каждой траектории и выбрать оптимальную. Траектория используется как линия движения мобильного робота. В разделе 1.2 мы получили траекторную модель робота. Чтобы завершить прогнозирование траектории мобильного робота, скорость (v, ω) колесной робота Mecanum также необходима для выборки.

Во-первых, мобильный робот ограничен максимальной скоростью и минимальной скоростью, заранее заданными им самим, поэтому диапазон скорости выборки составляет:

$$v_a = \{(v, \omega) | v \in [v_{min}, v_{max}], \omega \in [\omega_{min}, \omega_{max}]\} \quad (11)$$

Влияет на производительность двигателя робота с учетом максимального ускорения и замедления робота:

$$v_b = \{(v, \omega) | v \in [v_t - b_v \Delta t, v_t + a_v \Delta t], \omega \in [\omega_t - b_\omega \Delta t, \omega_t + a_\omega \Delta t]\} \quad (12)$$

Где, (a_v, a_ω) – максимальное ускорение; (b_v, b_ω) – максимальное замедление.

Кроме того, для обеспечения безопасности робота во время движения необходимо, чтобы мобильный робот останавливался до столкновения с дистанционным препятствием при условии максимального торможения, поэтому диапазон скоростей составляет:

$$v_b = \left\{(v, \omega) \middle| v \leq \sqrt{2dis(v, \omega) b_v}, \omega \leq \sqrt{2dis(v, \omega) b_\omega}\right\} \quad (13)$$

Где, $dis(v, \omega)$ — ближайшее расстояние траектории до препятствия при скорости дискретизации(v, ω).

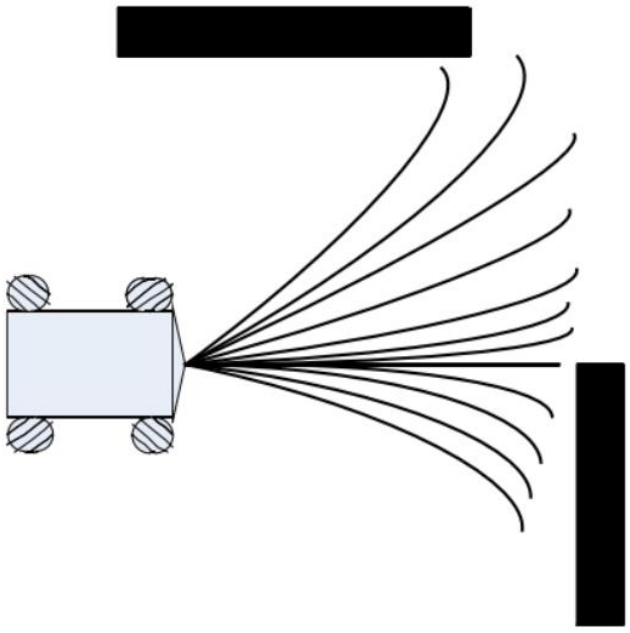


Рисунок 6 - Вычет траектории

Предел выборки скорости мобильного робота должен одновременно удовлетворять уравнениям (11), (12) и (13). В соответствии с моделью

движения колесной тележки Mecanum и выборкой скорости тележки (v, ω) можно вывести траекторию движения колесной тележки Mecanum на разных скоростях, как показано на рисунке 6.

Функция оценки траектории выглядит следующим образом:

$$F(v,\omega) = \eta\big(\alpha head(v,\omega) + \beta dist(v,\omega) + \gamma velocity(v,\omega)\big) \qquad (14)$$

Где,

Функция $head(v,\omega)$ используется для оценки азимута;

Функция $dist(v,\omega)$ используется для оценки расстояния между траекторией и ближайшим препятствием;

Функция скорости $velocity(v,\omega)$ используется для оценки скорости текущей траектории. η - постоянная нормировки.

Поиск траектории с наименьшими затратами — это локально спланированный путь для колесного робота Mecanum, чтобы избежать препятствий.

3. Анализ экспериментальных результатов

Эксперименты в основном включают позиционирование AMCL, навигацию по статической среде и навигацию по динамической среде. Поскольку эта статья не посвящена картографированию, используется предварительно созданная 2D карта затрат. Карта построена мной по алгоритму gmapping, как показано на рисунке 7. Черные части представляют препятствия, белые части представляют проходимые области, а серые части представляют неизвестные области. Карта используется для позиционирования робота и планирования пути. Разрешение карты 0.05, Пиксель карты 400*400.

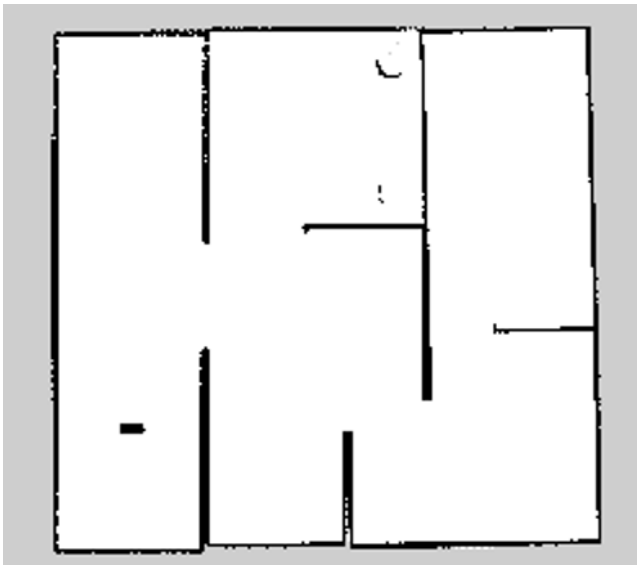


Рисунок 7 - 2D карта затрат

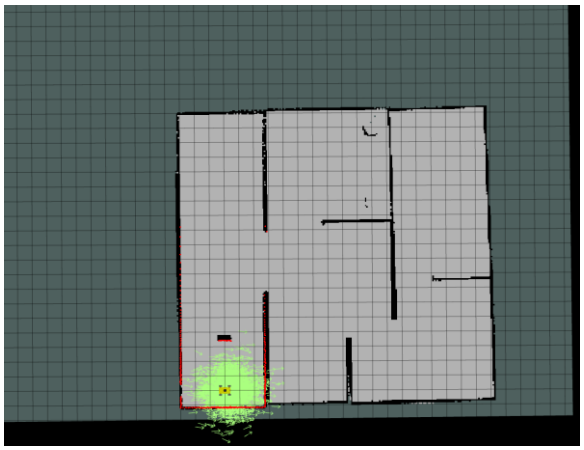
3.1 Эксперимент по позиционированию робота AMCL

Некоторые параметры, используемые в тесте позиционирования робота, показаны в таблице 1.

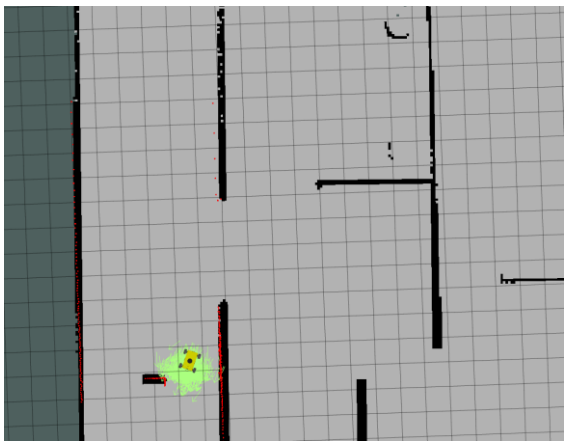
Таблица 1 Некоторые параметры, используемые навигацией робота

Скорость линии	Скорость поворота	Количество частиц	Лазерный диапазон
0.6m/s	0.6rad/s	300-3000	8m

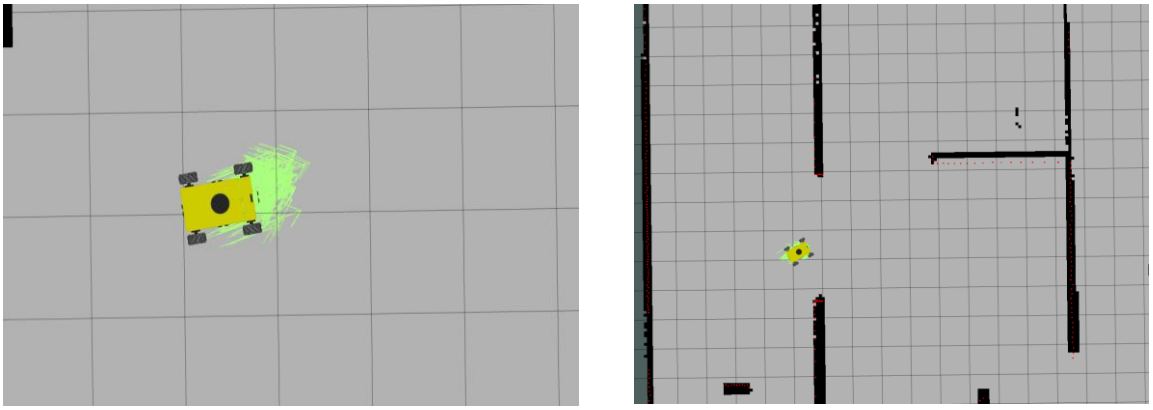
Процесс позиционирования робота показан на рисунке 8. Зеленая стрелка — это рой частиц, сгенерированный алгоритмом, а направление стрелки указывает направление движения робота. Красные точки — это положения границ объекта, отображаемые лидаром.



a) Робот в исходном положении



b) Двигаться около 2 м



c) Двигаться около 4 м d) Двигаться около 6 м

Рисунок 8 - Позиционирование робота AMCL

Как показано на рисунке 8-a, робот находится в исходном положении, вы можете видеть его в исходном положении. Частицы сильно разбросаны вокруг робота, а точность позиционирования низкая. Затем управлять движением робота (от рис. 8-a до рис. 8-d), видно, что частицы вокруг робота постоянно сходятся. Когда робот переместился примерно на 6 метров, частицы смогли точно отобразить позу робота (рис. 8d).

Точность позиционирования робота была протестирована в одном направлении, всего было собрано 180 наборов данных. Как показано на рисунке 9.

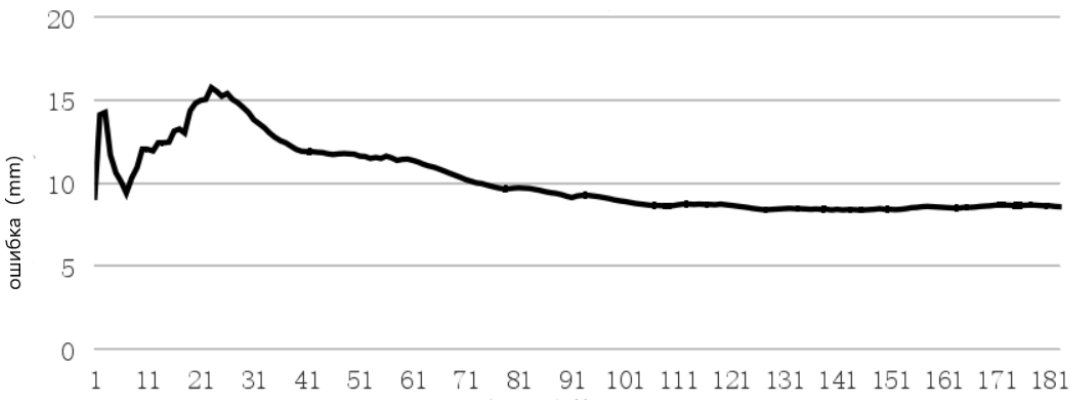


Рисунок 9 - Средняя ошибка однонаправленного позиционирования

В начале теста роботу задается начальная точка с большой ошибкой, а затем запускается тест позиционирования. На 25-й раз средняя ошибка достигла 15.76 мм. По мере прохождения теста точность постепенно

улучшается, а средняя ошибка падает до 10 мм после 73 прогонов, пока тест не будет завершен и средняя ошибка не составит около 8.58 мм.

Тест на точность позиционирования в одном направлении доказывает, что робот имеет лучшую повторяющуюся точность позиционирования. С увеличением времени работы точность робота также повышается, а также была доказана надежность алгоритма.

Чтобы проверить надежность точности позиционирования робота, был проведен тест на точность позиционирования с разных направлений до целевой точки. Всего было собрано 128 наборов данных, и результаты показаны на рисунке 10. Общая тенденция точности разнонаправленного позиционирования такая же, как и у однонаправленного позиционирования. В начале теста ошибка велика и незначительно колеблется. В девятом тесте максимальная средняя ошибка составляет 19.83 мм. После 30 тестов точность постепенно улучшается, и в конце концов ошибка сходится к 10,3 мм или около того.

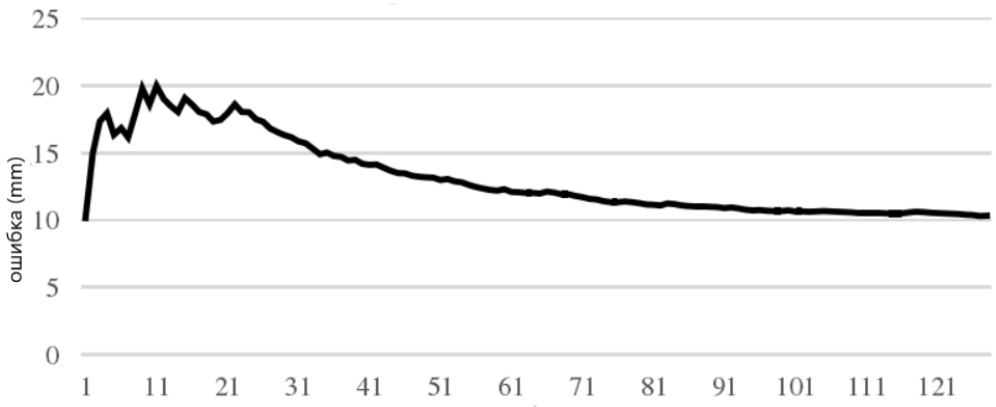


Рисунок 10 - Средняя ошибка многонаправленного позиционирования

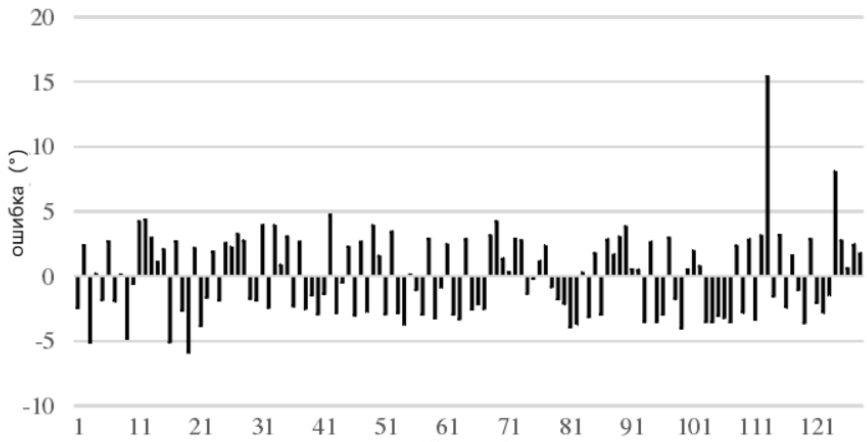


Рисунок 11 - Статистика угловых ошибок

Для проверки точности позиционирования в дополнение к положению выполняется также статистика углового отклонения при многонаправленном позиционировании. Как показано на рисунке 11, за исключением двух больших отклонений (-8° и +16°), все остальные ошибки угла позиционирования находятся в пределах 6°, а общая средняя ошибка угла составляет +0.91°, угловая ошибка в пределах ±4° составляет 85.9%.

3.2 Навигационные эксперименты в статической среде

Как показано на рисунке 12, цель установлена на статической карте (местоположение, показанное зеленой стрелкой). Затем планируется глобальный，оптимальный путь. Как видно из рисунка , путь обходит все препятствия.

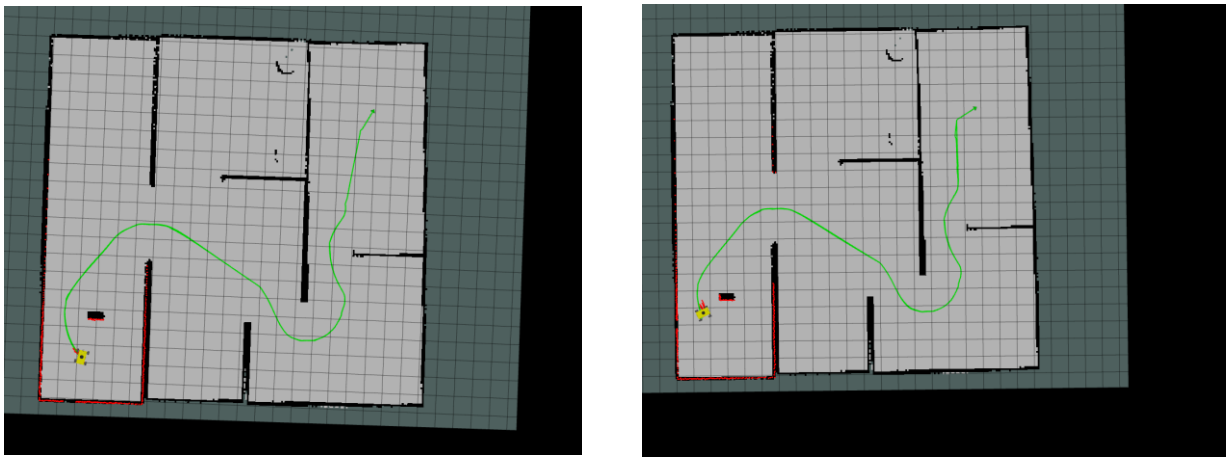


Рисунок 12 - поставить цель и планирование глобального пути

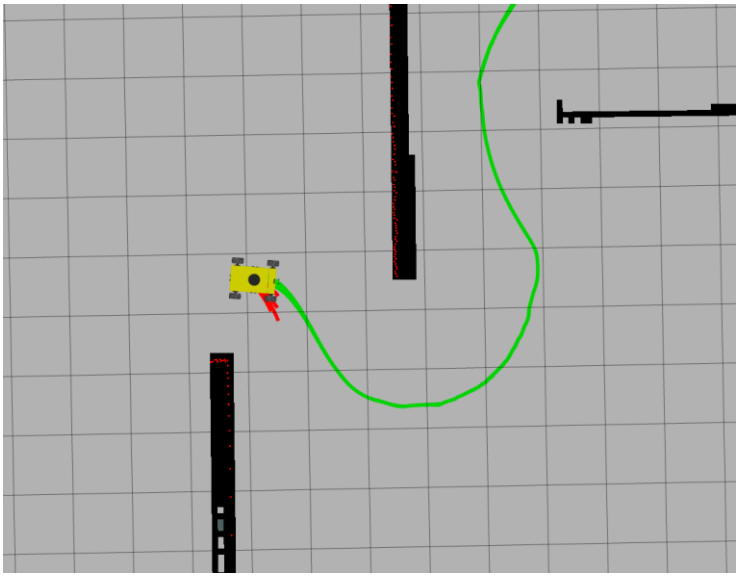


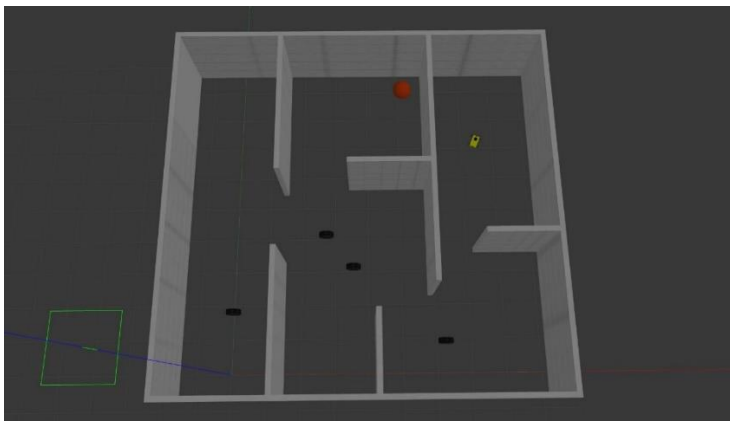
Рисунок 13 - Планирование локального пути

Как показано на рис. 13, красная линия — это планирование локального пути. Из-за препятствий впереди запланированный локальный путь немного отклоняется от глобального пути, но общее направление остается прежним.

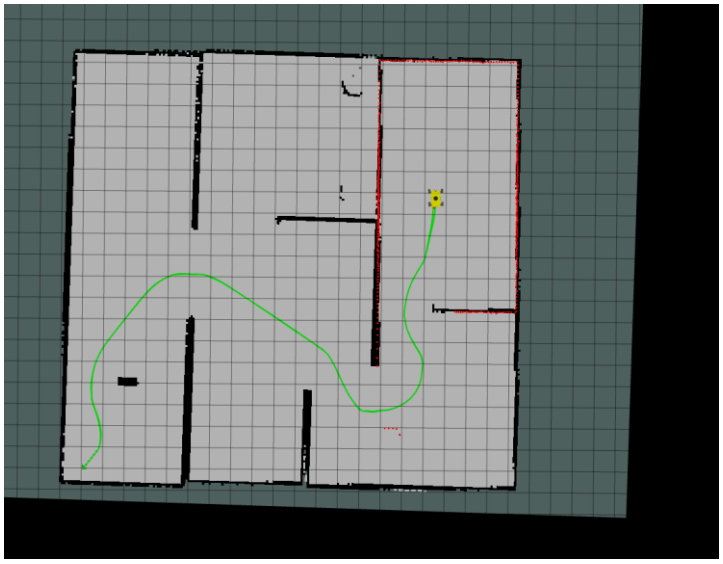
Экспериментальные результаты показывают, что робот может обходить препятствия на карте в течение всего процесса планирования пути, планировать оптимальный путь и, наконец, достигать целевой точки.

3.3 Навигационные эксперименты в динамических средах

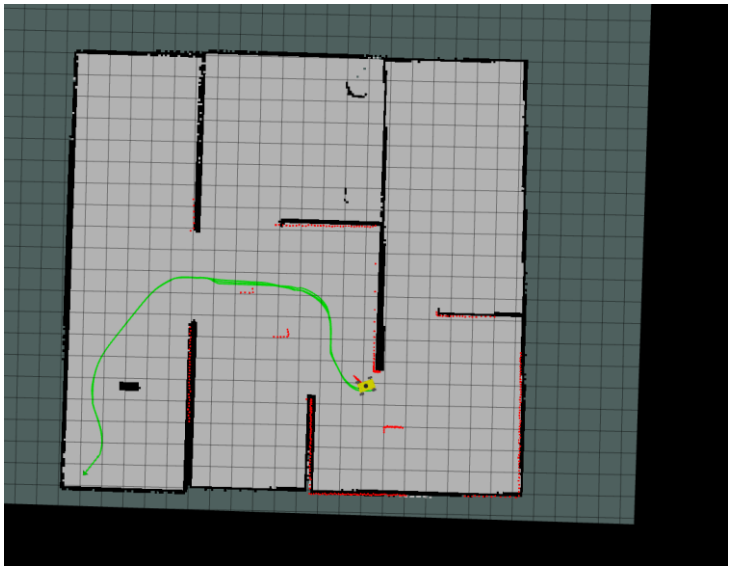
В навигационном эксперименте в динамической среде в процессе движения робота размещаются препятствия, так что запланированный роботом глобальный путь блокируется препятствиями. Как показано на рисунке 14(a).



a)



b)



c)

Рисунок 14 - Динамический обход препятствий

Как показано на рисунке 15(b), алгоритм глобального планирования A* планирует путь. Затем во время движения робота на пути расставляются препятствия. Когда робот перемещается в положение, показанное на рисунке 15(c), он обнаруживает препятствие и затем перепланирует новый путь. Этот новый путь позволяет избежать новых препятствий. Экспериментальные результаты показывают, что навигационная система может обеспечить динамическое избегание препятствий. После экспериментального анализа делается вывод, что использование алгоритма AMCL позволяет повысить

точность позиционирования робота, а погрешность находится в допустимых пределах.

4. Заключение

В данной статье исследуется навигационная система мобильного робота на базе колесного робота Mecanum. Анализируются кинематическая модель и модель траектории движения робота Mecanum. Модель робота и модель среды создаются в GAZEBO. Система навигации мобильного робота разработана путем объединения алгоритма глобального планирования пути A*, алгоритма планирования локального пути DWA и алгоритма локализации робота AMCL. Эксперимент показал, что использование алгоритма AMCL может повысить точность позиционирования робота, и результаты показывают, что ошибка находится в допустимом диапазоне. Навигационная система, объединяющая алгоритм планирования глобального пути и алгоритм планирования локального пути, может реализовать функцию динамического обхода препятствий. Колесный мобильный робот Mecanum обладает характеристиками гибкого движения, поэтому навигационная система на базе колесного мобильного робота Mecanum может применяться для различных сценариев работы.

Литература

1. Ван Ди, Ли Гуан, Ю Тэн, Ли Ренши, Исследование и моделирование системы навигации мобильных роботов на основе ROS [J], Журнал Северо-Китайского университета науки и технологий, 2021, 18 (04): 54-60.
2. Э. Маулана, М. А. Муслим и В. Хендрайаван, «Реализация инверсной кинематики четырехколесного мобильного робота с механическим приводом с использованием шаговых двигателей», Международный семинар 2015 г. по интеллектуальным технологиям и их приложениям (ISITIA), 2015 г., стр. 51-56.

3. Чжан Дэхао. Позиционирование в помещении и планирование пути мобильного робота с колесной конструкцией Mecanum [D]. Чжэцзянский университет, 2021 г.
4. Пэн Г., Чжэн В., Лу З. З. и др. Усовершенствованный алгоритм AMCL, основанный на совпадении лазерного сканирования в сложной и неструктурированной среде[J], 2018, 2018: 11-11.
5. A Cheong, MWS Lau, E Foo, et al. Разработка системы роботов-официантов[J]. IFAC PapersOnLine, 2016, 49(21): 681-686.
6. Fox D. Адаптация размера выборки в фильтрах твердых частиц с помощью KLD-sampling[J]. Международный журнал исследований робототехники, 2003, 22(12): 985-1003.
7. Ван Ю, Ву Д, Сейфзаде С и др. Алгоритм MCL на основе движущихся ячеек сетки для локализации мобильных роботов, Робототехника и биомиметика, 2009: 2445~2450.
8. Ху Чжанфан, Цзэн Линьцюань, Луо Юань, Луо Синь, Чжао Лимин.Адаптивный алгоритм локализации Монте-Карло, включающий информацию QR-кода.Компьютерные приложения, 2019, 39(04): 989~993.
9. Ли Т., Сун С., Саттар Т. П. Адаптация размера выборки в фильтрах частиц с помощью KLD-Пересэмплинг. Electronics Letters, 2013, 49(12): 740~742.
10. Шу Кайсян. Исследование и разработка трехмерной системы построения карт и навигации для мобильных роботов на основе изображений RGB-D [D]. Гуанчжоу: Южно-Китайский технологический университет. 2018: 42-59.

References

1. Wang Di, Li Guan, Yu Teng, Li Renshi. Research and Simulation of Mobile Robot Navigation System Based on ROS [J]. Journal of North China University of Science and Technology, 2021,18(04):54-60.

2. E. Maulana, M. A. Muslim and V. Hendrayawan, "Inverse kinematic implementation of four-wheels mecanum drive mobile robot using stepper motors," 2015 International Seminar on Intelligent Technology and Its Applications (ISITIA), 2015, pp. 51-56.
3. Zhang Dehao. Indoor positioning and path planning of mobile robot with Mecanum wheel structure[D]. Zhejiang University, 2021.
4. G. Peng, W. Zheng, Z. Z. Lu, et al. An improved AMCL algorithm based on Laser scanning match in a complex and unstructured environment[J]. 2018, 2018: 11-11.
5. A Cheong, MWS Lau, E Foo, et al. Development of a robotic waiter system[J]. IFAC PapersOnLine, 2016, 49(21): 681-686.
6. Fox D. Adapting the sample size in particle filters through KLD-sampling[J]. The international Journal of robotics research, 2003, 22(12): 985-1003.
7. Wang Y, Wu D, Seifzadeh S, et al. A moving Grid Cell Based MCL Algorithm for Mobile Robot Localization. Robotics and Biomimetics, 2009: 2445~2450.
8. Hu Zhangfang, Zeng Linquan, Luo Yuan, Luo Xin, Zhao Liming. Adaptive Monte Carlo Localization Algorithm Including QR Code Information. Computer Applications, 2019, 39(04): 989~993.
9. Li T, Sun S, Sattar T P. Adapting Sample Size in Particle Filters Through KLD-Resampling. Electronics Letters, 2013, 49(12): 740~742.
10. Shu Kaixiang. Research and Design of 3D Map Construction and Navigation System for Mobile Robots Based on RGB-D Images [D]. Guangzhou: South China University of Technology. 2018: 42-59.

© Ван Юй, Научно-образовательный журнал для студентов и преподавателей «StudNet» №3/2022.

Для цитирования: Ван Юй Исследование навигационной системы на базе колесного мобильного робота Mecanum// Научно-образовательный журнал для студентов и преподавателей «StudNet» №3/2022