

АЛГОРИТМЫ УПРАВЛЕНИЯ ДЛЯ КОМПЛЕКСА АВТОМАТИЧЕСКОГО ПИЛОТИРОВАНИЯ КОЛЕСНОЙ ПЛАТФОРМЫ

В.И. Рот

Научный руководитель: Е. С. Чердынцев
Томский политехнический университет
e-mail: vir7@tpu.ru

Введение

В настоящее время на высокотехнологичных производствах широко используются промышленные роботы, которые способны самостоятельно передвигаться, переносить грузы, поднимать их на заданную высоту и выполнять множество других операций, необходимых человеку. Все они работают по заранее разработанному алгоритму или под управлением оператора. Поэтому проблема создания мобильной системы автопилотирования, дополненной датчиками, является актуальной и требует разработки не только оборудования, но и комплекса программного обеспечения. По этой причине было решено разработать бюджетную систему автопилотирования на примере уменьшенной модели автомобиля [1].

Задачи

Необходимо было разработать автономную систему управления роботом, которая предназначена для обеспечения автономного движения в различных условиях. Общая цель для системы управления – изучение замкнутого пространства и автономное движение робота, а также управление роботом с помощью Android приложения.

Автономный робот – это робот, способный выполнять задачи в неструктурированных средах без постоянного человеческого контроля. Полностью автономный робот обладает следующими возможностями:

- Получение информации об окружающей среде.
- Работа в течение длительного времени без вмешательства человека.
- Включает в себя блоки, чтобы взаимодействовать с окружающей средой.

Разработка Android-приложения

Для упрощения работы с роботом разрабатывается Android-приложение, при помощи которого можно будет загружать исходные данные для робота и получать готовые карты, составленные роботом. Так же с приложения можно будет управлять работой робота, то есть запускать работу алгоритма, приостанавливать его, менять данные и т.д.

Алгоритмы работы робота

Для прохождения задается поле, по которому может перемещаться робот, в виде матрицы. Пример матрицы можете увидеть на рисунке. Размер матрицы может быть любым. Размер одной клетки матрицы соответствует 40см поля, так как габариты машины составляют 33см x 42см и для ее по-

ворота требуется много места. Для ручного тестирования алгоритма используем поле, показанное на рисунке 1.

```
0 0 0 0 0 0 0 0 0 3
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0
```

Рис. 1. Тестовая площадка для алгоритма

Размер данной матрицы составляет 10x10, что соответствует площадке 4x4 метра. 1 – начальная точка. 3 – конечная точка. 0 – неисследованное пространство. Скорость движения робота составляет 0,1 м/с. Путь робота в данном случае строится по кратчайшему направлению, то есть строится прямая от А до В, и робот по ней едет до точки. При прохождении пути роботом, единицей обозначается пройденный путь, то есть по которому проехал робот и он свободен, двойкой обозначается препятствие. При нахождении препятствия запускается алгоритм объезда, который показан в приложении. Его работа заключается в том, что при нахождении препятствия, он строит путь объезда. Для начала робот пытается проехать по правую сторону от препятствия, если не выходит, то он возвращается на то место, где было обнаружено препятствие и пытается объехать его слева. Если препятствие пройдено, робот так же по кратчайшему пути возвращается на траекторию, если опять на пути встречается препятствие, он так же его объезжает, но если препятствие объехать невозможно, то робот возвращается на точку, где было найдено препятствие и подает сигнал о том, что продолжить движение он не может. После прохождения пути матрица сохраняется и может быть выведена на экран ПК, либо стерта при перезаписи новой матрицы для прохождения. Пример матрицы, которая должна получиться в результате работы алгоритма можете увидеть на рисунке 2.

```
0 0 0 0 0 0 0 0 1 3
0 0 0 0 0 0 0 0 1 1
0 0 0 0 0 0 1 1 1 0
0 0 0 0 0 1 1 1 0 0
0 0 0 1 1 1 1 0 0 0
0 0 0 1 1 1 0 0 0 0
0 0 0 1 1 1 0 0 0 0
0 0 2 2 1 1 0 0 0 0
1 2 1 1 1 0 0 0 0
1 1 1 1 1 0 0 0 0
```

Рис. 2. Результат работы алгоритма прохождения от А до В

По завершении пути робот подает сигнал о достижении точки назначения. Далее в него можно загружать новые точки для прохождения пути.

Описание работы алгоритма

После завершения инициализации программы, загрузки карты и прокладывания кратчайшего маршрута, робот ожидает команды «Старт», для начала движения. После получения команды, робот начинает двигаться. Каждый раз при движении происходит проверка на то, достигли ли мы конечной точки, и есть ли на пути препятствие. Если достигли конечной точки, то подается сигнал о достижении нужной точки и работа алгоритма завершается, если нет, то продолжаем движение. Если найдено препятствие, то начинаем его объезд, если препятствия нет, то продолжаем движение. После выполнения объезда препятствия мы проверяем, нет ли дальше препятствий на нашем пути и полностью ли сработал алгоритм объезда. Если препятствий больше нет, то робот прокладывает путь, чтобы вернуться на изначально проложенный путь, то есть на самый короткий, и продолжает движение по этому пути. Если после объезда препятствия на пути есть еще одно, либо препятствие объехал робот не до конца, то проверяется, можем ли мы продолжить объезд. Если есть, с какой стороны можно продолжить объезд, то продолжаем маневр, если пути дальше нет, то подаем сигнал о невозможности завершения маршрута и заканчиваем алгоритм.

Описание работы алгоритма объезда препятствия

Объезд препятствия происходит по алгоритму проезда сначала справа, и если невозможно, то слева от препятствия. После инициализации программы, происходит проверка, какое расстояние осталось до препятствия. Если расстояние больше 20см, то робот подъезжает ближе к препятствию, если меньше, то начинает маневр объезда. Для начала пытаемся объехать справа, вывернув колеса в правую сторону и немного проехав вперед. Далее происходит проверка, может ли робот проехать дальше. Если да, то маневр объезда завершается и алгоритм заканчивает свою работу, если объезд справа не возможен, то пробуем объехать слева. Если слева объехать препятствие не получится, то алгоритм завершается, если возможность проезда есть, то завершаем маневр и завершаем работу алгоритма.

Описание работы алгоритма прохождения лабиринта

Для прохождения лабиринта задается поле, с указанием начального положения робота, по которому перемещается робот, в виде матрицы. Размер

матрицы может быть любым, в зависимости от размеров лабиринта. Размер матрицы может быть любым. Размер одной клетки матрицы соответствует 40см поля, так как габариты машины составляют 33см x 42см и для ее поворота требуется много места.

После завершения инициализации программы и загрузки карты, робот ожидает команды «Старт», для начала движения. Далее при продвижении вперед робот проверяет, достигли ли мы конца маршрута. Если да, то работа алгоритма заканчивается, если нет, то далее проверяем, не зашли ли мы в тупик. Если зашли в тупик, то возвращаемся на перекресток и едем по другому пути. Далее проверяем, есть ли поворот, если нет, то продолжаем движение прямо, если есть, то проверяем, есть ли правый поворот, так как его приоритет выше. Если есть правый поворот, то поворачиваем направо, если нет. То проверяем, есть ли путь прямо. Если есть, то едем прямо, если пути прямо нет, то поворачиваем налево. Алгоритм повторяется до нахождения выхода из лабиринта, после чего происходит отправка сигнала о завершении прохождения лабиринта и выход из программы. В зависимости от размера и сложности лабиринта, время его прохождения может быть большим.

Заключение

В итоге была разработана аппаратная платформа для робота, на котором можно проверять работоспособность различных алгоритмов автоматического пилотирования, а также Android приложение для управления роботом по средствам Bluetooth-соединения. В дальнейшем робот имеет возможность доработки, посредством добавления новых блоков.

Список использованных источников

1. Жимарши Ф. Сборка и программирование мобильных роботов в домашних условиях / Ф. Жимарши; пер. с фр. М. А. Комаров. – М.; ИТ Пресс, 2007. – 288 с.;
2. Белова, А. В. Создаем устройства на микроконтроллерах // А. В. Белова. – Санкт-Петербург: Наука и Техника, 2007. – 295 с. ;
3. Arduino + датчик HC-SR04 (ультразвуковой дальномер) // Электронный журнал [Электронный ресурс]. – URL: <http://academicfox.com/arduino-datchyk-hc-sr04-ultrazvukovoj-dalnomer/> (Дата обращения 21.04.2017).