

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра дискретной математики и
информационных технологий

Система интеллектуального управления роботом DaNI

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 421 группы
направления 09.03.01 – Информатика и вычислительная техника
факультета компьютерных наук и информационных технологий
Серебрякова Андрея Александровича

Научный руководитель
ассистент кафедры ДМиИТ

Н.Е. Тимофеева

Зав. кафедрой
доцент, к.ф.-м.н., доцент

Л.Б. Тяпаев

Саратов 2017

ВВЕДЕНИЕ

Робототехника – прикладная наука, занимающаяся разработкой автоматизированных технических систем.

Роботы и системы автоматизации повсеместно используются для решения широкого спектра задач и являются одними из наиболее важных компонентов промышленных и научных систем. Роботы позволяют облегчить выполнение сложных или опасных для жизни человека задач, например, робот-сапер, вспомогательные экзоскелеты. При этом роботы способны выполнять задачи намного точнее и быстрее повышая тем самым производительность. На данный момент существует большое количество типов роботов спроектированных для различных сфер деятельности человека (медицина, производство, развлечения, домашнее хозяйство и т.д.). Робототехника продолжает стремительно развиваться и является одним из важнейших направлений научно-технического прогресса, в котором проблемы механики и новых технологий соприкасаются с проблемами искусственного интеллекта. В связи с этим изучение робототехники и работы в области машинного зрения, распознавания образов и планирования траектории перемещения являются актуальными на данный момент.

Целью данной работы является разработка и программная реализация алгоритмов управления роботом DaNI, в частности задач «объезд препятствий» и «заезд в гараж».

При этом необходимо решить следующие задачи:

- изучить поколения развития робототехники;
- провести анализ существующих алгоритмов решения задачи «заезд в гараж»;
- изучить принципы работы датчиков робота и способы управления ими.
- разработать алгоритмы решения задач «объезда препятствий» и «заезд в гараж» для робота DaNI;

- реализовать программно разработанный алгоритм в среде графического программирования LabView.

Работа состоит из введения, четырех глав, заключения, списка литературы и приложений. Первая глава «Робототехника», вторая глава «Анализ алгоритмов парковки и заезда в гараж», третья глава «Аппаратные и программные средства разработки» и четвертая глава «Практическая реализация алгоритмов».

Основное содержание работы

1 Анализ алгоритмов парковки и заезда в гараж

На данный момент существует множество алгоритмов парковки и заезда в гараж, реализованных на различных роботах. Рассмотрим алгоритм, разработанный в рамках IX Международной научно-практической конференции. Решение задачи реализовано на базе робота Lego NXT. Робот обладает автомобильной колесной базой (передняя ось может вращаться относительно центра робота). Для выполнения данного алгоритма использовались следующие датчики: ультразвуковой сенсор, инфракрасный дальномер, энкодеры колес, а также установлены датчики смартфона (камера, электронный компас, гироскоп, акселерометр). С помощью ультразвукового датчика определяется свободное место для парковки и корректирует движение робота во время пробуксовки колес. Инфракрасный дальномер используется для предотвращения столкновения робота с объектами во время парковки, при движении назад. Остальной набор датчиков улучшает управляемость системы в целом и позволяет получать данные о текущем положении робота. Система обменивается данными со смартфоном, на котором установлен узел `rosjava`, через Bluetooth. Данный узел отвечает за управление двигателями робота. А смартфон соединен с компьютером через Wi-fi. На компьютере выполняется ядро ROS и узлы, необходимые для управления движением робота, управления радаром робота, составления карты препятствий и получения видео с камеры смартфона. При написании алгоритма использовался язык Python. Вначале работы алгоритма робот обнаруживает необходимое место для парковки. Затем двигается назад до тех пор, пока передние колеса робота не будут параллельны началу парковочного места. Далее, вычисляются возможные маршруты движения робота до конечной точки и среди них выбирается оптимальный. После определения оптимального маршрута робот движется до места парковки.

Также подобный алгоритм был реализован на базе робота Pioneer I. Данный робот имеет 8 фронтальных и 8 задних ультразвуковых датчиков расстояния, а также лазерный сенсор SICK. Данный робот имеет несколько версий отличающихся друг от друга набором датчиков и конструкцией. Главное отличие от предыдущей работы, является неподвижная колесная база, в связи с чем методы маневрирования робота будут несколько другими. В начале работы алгоритма робот ищет первый проем и проезжая его производит замер, если размеры парковочного места больше габаритов робота, то он перейдет к этапу парковки. Двигаясь к парковочному месту, робот постоянно считывает данные об окружении с помощью датчиков. Также процесс парковки корректируется с помощью лазерного сенсора, который направлен в стену, находящуюся на противоположной стороне от места парковки. В работе указано что, робот в 30% случаев задевает препятствия ограничивающие место парковки, это связано с неточностью установленных датчиков и с периодически возникающими ошибками лазерного сенсора.

В следующей работе, алгоритм парковки был реализован на базе робота ROboMObil. Данный робот предназначен для моделирования парковки автомобилей с искусственным интеллектом. Он обладает большим набором датчиков (камеры, дальномеры, ИК датчики, GPS). Также робот оснащен четырьмя независимыми колесами, которые способны поворачивать на угол от 35° в одну сторону и до 95° в другую. Робот поддерживает два вида управления: управления автомобилем вручную или удаленное управление и полностью автономное движение. В начале работы алгоритма робот обнаруживает потенциальные места для парковки с помощью алгоритма 3D реконструкции окружения. Затем робот определяет наиболее подходящее место для парковки из обнаруженных. На следующем этапе работы алгоритма, робот планирует траекторию маневра и движется к месту парковки используя, возможности колесной базы.

В отличие от ранее рассмотренных решений, данная система имеет более серьезную маневренность и больший набор высокоточных датчиков, что сильно упрощает решение задачи. Маневренность колес данного робота позволяет ему намного эффективнее достигать поставленной цели.

Рассмотренные выше алгоритмы показывают, что не существует универсального алгоритма парковки, так как при решении задачи данного типа важно учитывать конструктивную особенность робота, к которому будет применяться разработанный алгоритм (размеры робота, количество колес, возможность разворота колес). В основном, способ решения задачи определяет набор датчиков, имеющихся у робота в наличии. Задачи подобного рода часто решаются на базе платформ Lego NXT, Arduino и Raspberry Pi. В общем виде алгоритм решения данной задачи будет иметь следующие шаги:

- обнаружение места парковки или гаража;
- проверка соответствия размера места парковки/гаража и размера робота;
- расчет траектории движения;
- проверка оптимальности и корректности рассчитанной траектории;
- парковка/заезд в гараж.

2 Алгоритм парковки в гараж

Алгоритм реализует автоматический заезд робота в первый подходящий по размерам гараж. Робот должен начинать движение вдоль стены. Данный алгоритм не учитывает возможность потери стены роботом, поэтому важно расположить его изначально параллельно стене. Данную проблему можно решить установкой дополнительного ультразвукового датчика или датчика линии.

В общем виде траектория движения робота будет выглядеть следующим образом, как показано на рисунке 14.

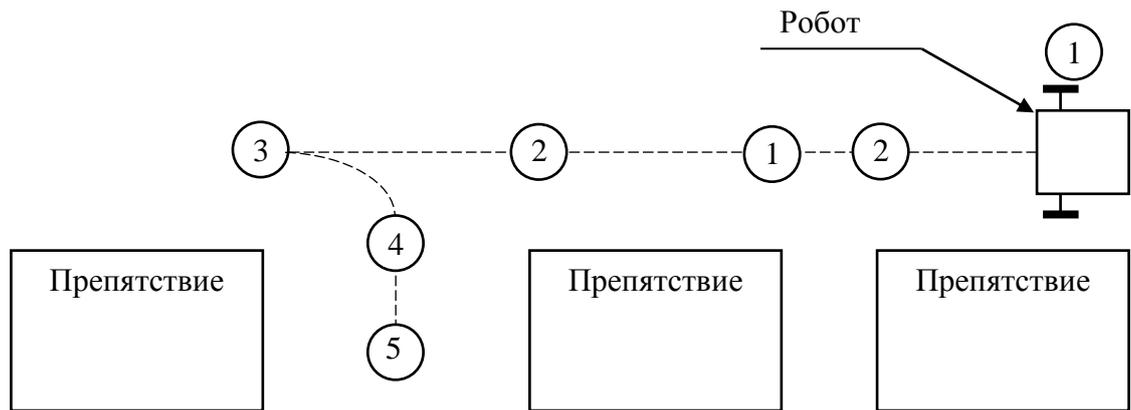


Рисунок 14 – Траектория движения робота при решении задачи парковки

Для начала работы алгоритма, необходимо разместить робота параллельно стене на небольшом расстоянии от нее (5-7 см.). Затем загрузить программу на робота и включить двигатели.

1. Поворот ультразвукового датчика в сторону стены на 90° .
2. Считывания расстояния до стены W .
3. Движение прямо со скоростью U .
4. Считывание текущего расстояния до стены W_c .
5. Если сканируемое расстояние не удовлетворяет условию $W + \Delta > W_c > W - \Delta$, то шаг 6, иначе шаг 4.
6. Считывание расстояния до стены W_g .
7. Запуск таймера t_g .
8. Вычисление пройденного расстояния $S = S + U * t_g$.
9. Если сканируемое расстояние не удовлетворяет условию $W + \Delta > W_g > W - \Delta$, то шаг 6, иначе шаг 10.
10. Если $S > D * 1.15$, где D – ширина робота, то переход к шагу 11, иначе переход к шагу 3.
11. Движение правым колесом, пока пройденное им расстояние S_r не будет удовлетворять условию $S_r > 0.5626$, где 0,5625 расстояние проходимое одним колесом робота для поворота на 90° .

12. Движение робота задним ходом на расстояние $S = \frac{1}{2} * L + 0.05$, где

L – длина робота.

Блок-схема алгоритма заезда в гараж изображена на рисунке 15.

3 Алгоритм объезда препятствий

Во время работы управляющей схемы используется четыре сигнала, принимающих значение 0 или 1. Два из них отвечают за левый и правый повороты, один за тупиковые ситуации и еще один отмечает отсутствие препятствий на пути. Для формирования данных сигналов следует провести предварительное преобразование входных данных, чтобы схема могла однозначно определять текущее состояние системы. В связи с этим данную задачу можно разбить на два этапа: предварительная подготовка данных для корректирующей схемы и корректировка движения.

В процессе выполнения алгоритма объезда препятствий считываются следующие данные: текущие скорости вращения колес (рад/с), дистанции до препятствий, а также угол поворота ультразвукового датчика расстояния. Угол поворота ультразвукового датчика изменяется от 65° до -65° , с шагом сканирования равным 4° . Очевидно, что область сканирования представляет собой сектор 130° .

С текущими параметрами будет просканировано 33 угла, обозначим их как A_n . Тогда углы A_{0-17} будут определять препятствия слева от робота, а углы A_{17-33} справа от робота. Обозначим расстояние, на котором робот начнет реагировать на препятствие как D . Так как расстояние между колесами равно 36 см., то параметр D должен быть больше, либо равен этому расстоянию иначе робот не успеет совершить маневр. Введем две булевых переменных L и R , которые будут передаваться на корректирующую схему в соответствии с условиями:

- Если хотя бы одно расстояние до препятствия, соответствующее углам A_{0-17} , меньше D , то подаем сигнал $L=1$, иначе $L=0$.
- Если хотя бы одно расстояние до препятствия, соответствующее углам A_{17-33} , меньше D , то подаем сигнал $R=1$, иначе $R=0$.

Скорости вращения колес подаются на блок корректировки напрямую.

Так как на схему подается два управляющих сигнала параллельно, то система может различать четыре состояния указанных в таблице 1.

Таблица 1 – Состояния различаемые системой распознавания препятствий

L	R	Состояние
0	0	Корректировка движения не требуется
1	0	Обнаружено препятствие слева.
0	1	Обнаружено препятствие справа.
1	1	Препятствие находится перед роботом.

В последней ситуации робот может находиться либо в тупике, либо перед стеной и направление поворота не может быть однозначно определено.

Обозначим эти состояния как CS_0, CS_1, CS_2, CS_3 .

Скорректированные скорости вращения колес вычисляются с помощью следующих формул:

$$Lv' = -b_1k(Lv - M_1)$$

$$Rv' = b_2k(Rv - M_2)$$

Где, Lv – скорость вращения левого мотора, Rv - скорость вращения правого мотора; k – коэффициент, влияющий на продолжительность корректировки.

Если схема получила на вход сигнал CS_2 , то одна из переменных b_1 или b_2 будет приравнена нулю, а вторая -1, иначе обе переменных равны 1.

1. Считывание массива углов A_i и массива дистанций до препятствий D_i , соответствующих этим углам.
2. Обнаружение препятствий O_i . Если в массиве D_i найдется элемент, значение которого будет меньше 0,4, то переход к шагу 3, иначе к шагу 1.
3. Определение положения препятствия относительно робота:
 - Если угол до препятствия находится в диапазоне 0° до 65° , то передача сигнала $L=1$;

- Если угол до препятствия находится в диапазоне 0° до -65° , то передача сигнала $R=1$;
4. Представление сигналов L и R в виде последовательности единиц и нулей и перевод в десятичную систему.
 5. Инициализация массива M из 4 целочисленных элементов, содержащих нули;
 6. Присвоение значения «1» элементу из массива M , находящемуся по индексу, полученному на шаге 4.
 7. Инициализация переменных $b_1 = 1$ и $b_2 = 1$.
 8. Если значение присвоено четвертому элементу в массиве, то $b_1 = 0$ или $b_2 = 0$.
 9. Вычисление новых скоростей моторов по формулам

$$Lv' = -b_1 k(Lv - M_1)$$

$$Rv' = b_2 k(Rv - M_2)$$

Где, Lv – скорость вращения левого мотора, Rv - скорость вращения правого мотора; k – коэффициент, влияющий на продолжительность корректировки.

ЗАКЛЮЧЕНИЕ

Разработанные алгоритмы объезда препятствий и парковки робота были проверены при различных начальных условиях и доказали свою работоспособность, но он не исключает полностью возможность столкновения или падения робота. Ультразвуковой датчик установлен таким образом, что робот не сможет засечь препятствие высотой менее 4 см. Также робот с текущим набором датчиков не сможет засечь резкий спуск или обрыв на участке. Указанные выше недостатки можно устранить с помощью установки дополнительного набора датчиков на систему.

В ходе выполнения работы были выполнены следующие задачи:

- изучены поколения развития робототехники;
- проведен анализ существующих алгоритмов решения задачи «заезд в гараж»;
- изучены принципы работы датчиков робота и способы управления ими.
- разработаны алгоритмы решения задач «объезда препятствий» и «заезд в гараж» для робота DaNI;
- реализованы программно разработанные алгоритмы в среде графического программирования LabView.

Таким образом, поставленные цель и задачи, бакалаврской работы были выполнены полностью.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Образовательная робототехника [Электронный ресурс]. URL:http://www.kurganrobot.ru/obrazovatel_nye_uslugi/osnovy_robototekhniki_5-6_klass/istoriya_robototekhniki/(дата обращения 01.05.2017). Загл. с экрана.
2. Википедия - свободная энциклопедия [Электронный ресурс]. URL:https://ru.wikipedia.org/wiki/Бэббидж,_Чарлз/ (дата обращения 01.05.2017). Загл. с экрана.
3. Робототехника [Электронный ресурс]. URL:<http://roboticslib.ru/books/item/f00/s00/z0000026/st004.shtml/>(дата обращения 04.05.2017). Загл. с экрана.
4. Портал искусственного интеллекта [Электронный ресурс]. URL:<http://neuronus.com/robo/47-teoriya/635-osnovy-robototekhniki.html/> (дата обращения 01.05.2017). Загл. с экрана.
5. Саловская А.А., Буйвал А.К. АЛГОРИТМ ПЛАНИРОВАНИЯ МАНЕВРА ПАРАЛЛЕЛЬНОЙ ПАРКОВКИ МОБИЛЬНОГО РОБОТА // Технические науки - от теории к практике: сб. ст. по матер. IX междунар. науч.-практ. конф. – Новосибирск: СибАК, 2012.
6. SMARTBOT Autonomous Parallel Parking Robot By David Johnson (2234574), Tim Miller (2132459), Huzefa Rangwala (3157226). Under the guidance of Professor Stergios Roumeliotis [Электронный ресурс]. URL:<https://pdfs.semanticscholar.org/9fdf/d2ef4f0a0bacc772a8de7fc1f1e4b191ad5e.pdf/>(дата обращения 14.05.2017). Загл. с экрана.
7. Alexander, Schaub. Autonomous Parking using a Highly Maneuverable Robotic Vehicle / Schaub. Alexander, Carlos Ramirez dela Cruz. Juan, Burschka. Darius. // Preprints of the 19th World Congress The International Federation of Automatic Control. – 2014.
8. National Instruments [Электронный ресурс] // Официальный сайт компании National Instruments. URL: <http://www.ni.com/> (дата обращения 21.05.2017). Загл. с экрана. Англ. рус.