

то же что и для трекинга агента. Разница лишь в том, что цель для задаётся на основе треклиста агента.

Выглядит это так:

1. Из треклиста агента выбирается очередное направление.
2. Выполняется вычисление и занесение в треклист пути толкателя до точки, находящейся в противоположном направлении от агента.
3. К треклисту толкателя добавляется выбранное направление агента.
4. Агент перемещается по выбранному направлению.
5. Из треклиста агента удаляется выполненное действие.
6. Если треклист агента пуст, то заканчиваем формирование треклиста толкателя. Иначе возвращаемся к пункту (1)

В результате в треклист запишется путь, который должен пройти толкатель для того, чтобы агент попал в заданную цель.

Например: (треклист для первого хода агента)

1	X			12	11	12
2	\$		@	10	11	
3	4		8	9	10	
4	5	6	7	8	9	
5	6	7		9	10	

Рисунок 6 – Путь толкателя к агенту

На рисунке выше символом «X» выделена цель толкателя, а серым отмечены позиции, через которые он должен пройти. Треклист толкателя в данной ситуации будет выглядеть следующим образом:

$Tracklist = \{Ю, Ю, 3, 3, С, 3, С, В, Ю\}$

В результате имеем алгоритмы преобразования игры сокобан в агентную систему, а также имеем алгоритмы для прозрачного управления игрой через сформированную агентную систему. Тем самым образовался более высокий уровень абстракции. Он позволит применять популярные мультиагентные подходы к задаче сокобан. Планируется провести имитационное моделирование и проверить эффективность разработанных алгоритмов.

УДК 004.896:621.865

Склипус Д. Б.

Научные руководители: доцент Дунец А.П., к.т.н., доцент Костюк Д.А.

РЕШЕНИЕ ТИПОВЫХ НАВИГАЦИОННЫХ ЗАДАЧ ПО ПРОГРАММИРОВАНИЮ АУТОНОМНОГО МОБИЛЬНОГО РОБОТА

В последнее время наблюдается увеличение доли бытового использования автономных мобильных роботов; благодаря широкому распространению дешевых и простых в реализации средств связи и портативных вычислительных систем для данного направления робототехники прогнозируется скачкообразный рост. Ведущие профильные вузы реагируют на наличествующий и ожидаемый в будущем спрос введением дополнительных учебных курсов.

Соответствующие лабораторные практикумы чаще всего строятся на базе различных виртуальных сред моделирования. Изучение программирования электронных устройств невозможно без практических экспериментов, и это особенно важно в случае управления автономной мобильной системой. Такой аппарат в своей физической реализации взаимодействует с широким спектром факторов и воздействий окружающей среды, подвержен ряду физических закономерностей, учет которых нехарактерен для систем мо-

делирования и никогда не встречается в виртуальном окружении в полном объеме. Т.о. разработка лабораторного мобильного стенда на базе программируемого микроконтроллера для решения учебных задач представляет несомненную практическую актуальность. Для данной цели нами разработана линейка автономных мобильных роботов с SRS локальным и дистанционным компьютерным управлением.

Для разработанной линейки SRS был сформулирован набор типовых навигационных задач, позволяющий на практике освоить технологию алгоритмизации управления и микроконтроллерного программирования автономного мобильного устройства. Список первичных задач включает неуправляемое движение с автоматическим обгибанием препятствий, движение в пределах фиксированной траектории, заданной черной линией, следование вдоль протяженного препятствия, а также варианты движения к конечному пункту на основе анализа изображения, полученного от веб-камеры.

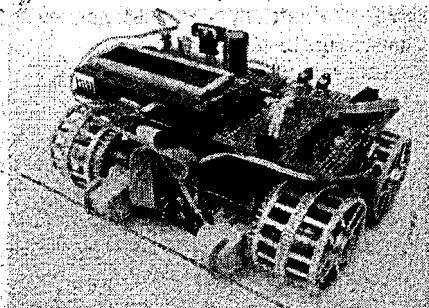


Рисунок 1 – Робот SimpleBot – первая модель в линейке SRS

Неуправляемое движение с объездом препятствий. Данная задача наиболее широко освещена тематическими информационными источниками, так как требует минимального аппаратного обеспечения и является простой в алгоритмизации. Алгоритм заключается в следующем: робот движется по прямой, а при обнаружении препятствия поворачивает на случайный угол и продолжает дальнейшее движение. Данный алгоритм был опробован на роботе SimpleBot, изображённом на рисунке 1, который являлся первой и наиболее простой моделью в линейке. Данный робот обладал тактильными датчиками в виде клавиш от стандартной клавиатуры ЭВМ [1].

Движение по чёрной линии. Эта задача – следующая по сложности реализации. Алгоритм обеспечивает движение робота по маршруту, заданному нанесенной на поверхность пола чёрной линией. Для устойчивого распознавания маршрута существуют геометрические и колориметрические требования к чёрной линии [2]. Для проведения практических экспериментов был специально разработан робот CDDR, изображённый на рисунке 2.

Робот обладает двадцатью датчиками чёрной линии и инерциальными тактильными сенсорами. Было решено не ограничиваться прямой линией, а добавить обработку перекрёстков и длинных кривых. Алгоритм основывается на последовательном преодолении перекрёстков, соответствующих описанным требованиям. Если робот движется вдоль левой границы чёрной линии, он повернёт налево на ближайшем перекрёстке, но если он двигался вдоль правой границы чёрной линии, то повернёт направо. Для исключения каких-либо поворотов и гарантированного прямолинейного движения робот движется

без отклонений вправо и влево. Данный алгоритм может быть применён для уже известных карт чёрной линии. Иначе при попытке двигаться в сторону, где нет чёрной линии (например, вперёд на Т-образном перекрёстке), направление движения робота будет не определено. Определение перекрёстков возможно одним из двух методов. Первый подразумевает сравнение со всеми возможными типами перекрёстков, число которых в нашем случае равняется 2^{20} . В связи с недопустимой трудоёмкостью данного процесса был выбран метод, подразумевающий объединение возможных типов перекрёстков в обобщённую модель, включающую в себя Х- и Т-образные перекрёстки, а также повороты. В рамках данного подхода была разработана функция для определения положения робота на перекрёстке. Входными данными для неё являются двумерный массив, описывающий состояние датчиков чёрной линии. Возвращаемое значение имеет логический тип: 1 обозначает нахождение робота на перекрёстке, а 0 – на участке чёрной линии. Оценка эффективности алгоритма проводилась по двум параметрам: безопасности и скорости.

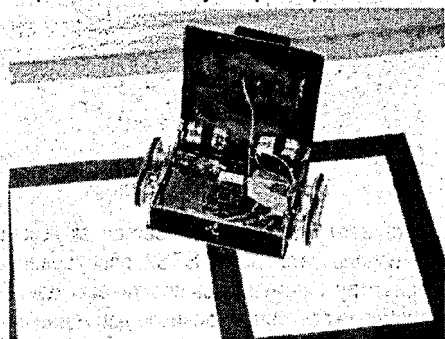


Рисунок 2 – Робот CDDR

На рисунке 3 представлено условное изображение траектории движения робота, задающей маршрут чёрной линии, а также площади, образованной пересечениями этих кривых. Безопасность оценивается по формуле $B = S/l$, где l – длина отрезка, а S – площадь. Чем меньше значение B , тем более безопасным является движение робота с точки зрения возможных столкновений и необратимой потери маршрута. Безопасность движения может быть также охарактеризована величиной граничного расстояния – максимального расстояния, на которое робот может удалиться от чёрной линии в рамках выполнения поставленной задачи. В большинстве случаев это расстояние не превышает ширины корпуса робота. Для определения площади S применен теоретический подход на основе использования интегрального исчисления.

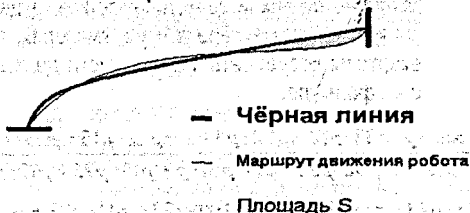


Рисунок 3 – Схематическое изображение маршрута робота, чёрной

линии и площади, образованной этими кривыми

Критерием оценки эффективности робота также является отношение скорости его движения с учетом задержек на перекрестках к скорости движения по прямой черной линии. В проведенных экспериментах это отношение составило 4 см/с / 5.5 см/с или 0.28. [2, 3]

Движение вдоль стенки. Данная задача является актуальной для роботов, движущихся в лабиринте. Она была решена за счёт двух подходов: контактного и бесконтактного. В первом случае применялся робот Robotmega, оборудованный восьмью тактильными датчиками по периметру робота. Робот изображён на рисунке 4.

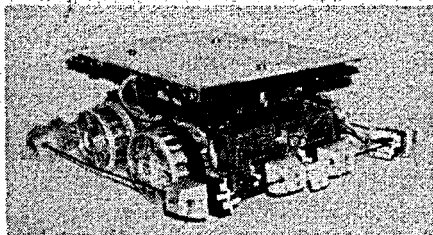


Рисунок 4 – Робот Robotmega

Второй подход предполагает использование бесконтактных пороговых датчиков препятствий. В этом случае используется робот SRS2, оборудованный восьмью датчиками, расположенными по периметру. Применялись практически одинаковые алгоритмы работы в обоих случаях. Robotmega показал устойчивое движение вдоль стенки; также робот успешно огибал углы с внешней и с внутренней стороны. В связи с дистанцией, на которой срабатывали бесконтактные датчики робота SRS2, робот аппроксимировал значительные участки пути, например, туннели. Решение данной задачи невозможно только за счёт алгоритмизации и требует изменения аппаратурной составляющей робота.

Движение из точки А в точку В на основе анализа видеоизображения. Одним из рассмотренных выше вариантов, пригодных для решения данной задачи, является движение по чёрной линии. Но более актуальны методы, позволяющие сохранять правильную траекторию движения без разметки. Нами были исследованы два способа движения. Один способ – это использование внешнего вычислительного центра и видеокмеры для наблюдения за полем действия робота и управление им. На рисунке 5 изображено схематическое расположение робота и камеры; для лучшего распознавания робота на его корпусе устанавливается яркий светодиод зелёного цвета. После обработки изображения и получения координат маркера x и y в пикселях, требуется провести их пересчёт в реальные декартовы координаты на плоскости движения робота. Пересчет выполняется по следующей формуле:

$$X = Z \cdot \frac{y \cdot p_{32} \cdot p_{13} - y \cdot p_{33} \cdot p_{12} + p_{22} \cdot p_{33} \cdot x - p_{22} \cdot p_{13} - p_{23} \cdot x \cdot p_{32} + p_{23} \cdot p_{12}}{-x \cdot p_{31} \cdot p_{22} - x \cdot p_{32} \cdot p_{21} - p_{11} \cdot p_{32} \cdot y + p_{11} \cdot p_{22} + p_{12} \cdot p_{31} \cdot y - p_{12} \cdot p_{21}}$$

$$Y = -Z \cdot \frac{-x \cdot p_{31} \cdot p_{23} + x \cdot p_{33} \cdot p_{21} - p_{11} \cdot y \cdot p_{33} + p_{11} \cdot p_{23} + p_{13} \cdot y \cdot p_{31} - p_{13} \cdot p_{21}}{-x \cdot p_{31} \cdot p_{22} - x \cdot p_{32} \cdot p_{21} - p_{11} \cdot p_{32} \cdot y + p_{11} \cdot p_{22} + p_{12} \cdot p_{31} \cdot y - p_{12} \cdot p_{21}}$$

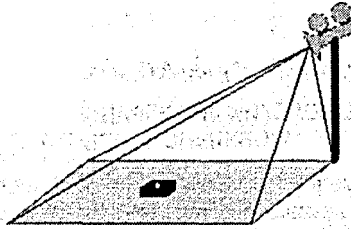


Рисунок 5 – Схематическое расположение робота и камеры наблюдения

Второй способ – это когда камера устанавливается на корпусе робота, и робот ищет цель самостоятельно, основываясь на параметрах датчиков и анализе видеопотока. В качестве цели было решено использовать яркий источник света (горящая свеча). На рисунке 6 изображены два окна программы, иллюстрирующие особенности определения расстояния до препятствия. Анализ расстояния основывается на площади светового пятна.

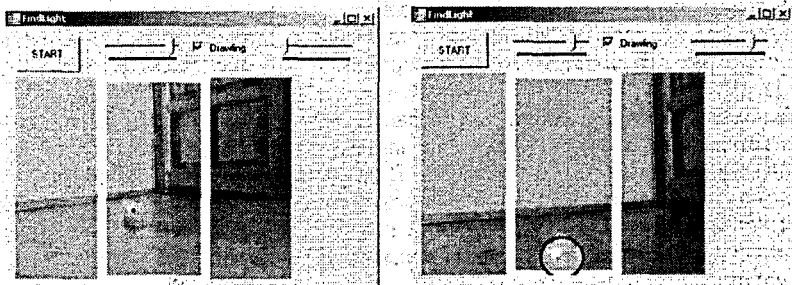


Рисунок 6 – Окна программы ведения робота на цель

Для этих экспериментов использовался робот SRS2, изображённый на рисунке 7.

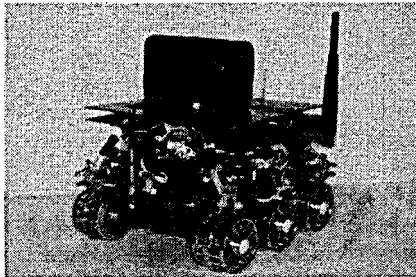


Рисунок 7 – Робот SRS2

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Склипус, Д.Б. Микроконтроллерная лаборатория: сб-к тезисов по секции «Информатика» // Междунар. конф. «VI Колмогоровские чтения». – Москва; 2006. – С. 26.
2. Склипус, Д.Б. Алгоритм движения по черной линии для автономного мобильного робота: сб-к курсных научных работ студентов и магистрантов. – Брест, 2009. – Ч. 1. – С. 122-125.
3. Склипус Д.Б. Робот SRS. http://www.prorobot.ru/nauka/Skipus_Robot_System.doc – Дата доступа 23/12/2008
4. Склипус, Д.Б. Автономный робот // <http://google.sites/skipusrobotssystem.com>. – Дата доступа 1/10/2009.