

УДК 004.8

Д.Ю. Бровкина, Т.А. Приходько

Донецкий национальный технический университет, Украина

**«РАЗРАБОТКА АЛГОРИТМА ОБХОДА ПРЕПЯТСТВИЙ ДЛЯ
МОБИЛЬНОГО УПРАВЛЯЕМОГО РОБОТА»****Аннотация**

Бровкина Д.Ю., Руководитель Приходько Т.А. «Разработка алгоритма обхода препятствий для мобильного управляемого робота». В данной статье представлена разработка алгоритма обхода препятствий мобильным управляемым роботом. Сначала выполнен обзор существующих подходов к его реализации. Приведена блок-схема разработанного алгоритма и прокомментирована возможность его использования на конкретной платформе.

Ключевые слова: алгоритм, обход препятствий, мобильный робот, нейронные сети, нечеткая логика.

Постановка задачи. Особенностью мобильных роботов является их автономность, возможность перемещаться независимо от каких-либо внешних стационарных устройств. Однако именно с этим связана одна из важнейших проблем разработки подобного устройства – разработка алгоритма движения мобильного робота в соответствии с выполняемыми им функциями. Именно эта задача стояла перед автором.

Анализ существующих алгоритмов обхода препятствий. Так как робототехника – стремительно развивающаяся отрасль в наше время, учеными было разработано множество способов реализации алгоритма, обеспечивающего функциональность главного «органа» мобильного робота – его навигации. Наиболее востребованными движениями робота являются движения, связанные с обходом объектов, и избежание столкновений с объектами в рабочем пространстве.

Алгоритмы, предназначенные для обхода роботом препятствий в трехмерном пространстве, можно разделить на несколько классов: гипотеза–тест; штрафная функция; метод скелетирования; нечеткая логика [3], нейронные сети, генетические алгоритмы. В данной работе будут рассмотрены наиболее интересные, по мнению автора, их варианты.

Метод гипотезы и теста состоит из трех основных шагов: предлагается гипотеза относительно пути-кандидата между начальной и конечной точками траектории движения мобильного робота; набор направлений вдоль этого пути тестируется на возможность столкновений; если столкновение оказывается возможным, то с целью определения пути обхода исследуется препятствие, которое может вызвать это столкновение. Весь процесс повторяется, пока не будет достигнута цель.

Второй класс алгоритмов обхода препятствий основывается на определении штрафной функции для конфигурации мобильного робота, с помощью которой кодируется наличие объектов.

Алгоритмы скелетирования сводят свободное пространство робота к одномерному представлению, для которого задача планирования пути становится проще. Такое представление с меньшим количеством измерений называется скелетом пространства конфигураций.

Нечеткий алгоритм определяется упорядоченным множеством нечетких инструкций, содержащих понятия, формализуемые нечеткими множествами. Все системы с нечеткой логикой функционируют по одному принципу: показания измерительных приборов: фазифицируются (превращаются в нечеткий формат), обрабатываются, дефазифицируются и в виде обычных цифровых сигналов подаются на исполнительные устройства.

Рассмотрим случай управления мобильным роботом, задачей которого является объезд препятствий. Введем две лингвистические переменные: ДИСТАНЦИЯ (расстояние от робота до препятствия) и НАПРАВЛЕНИЕ (угол между продольной осью робота и направлением к препятствию). Рассмотрим лингвистическую переменную ДИСТАНЦИЯ. Ее значения можно определить терминами ДАЛЕКО, СРЕДНЕ, БЛИЗКО и ОЧЕНЬ БЛИЗКО. Для физической реализации лингвистической переменной необходимо определить точные физические значения термов этой переменной. Пусть переменная ДИСТАНЦИЯ может принимать любые значения из диапазона от нуля до бесконечности. Согласно теории нечетких множеств, в таком случае каждому значению расстояния из указанного диапазона может быть поставлено в соответствие некоторое число от нуля до единицы, которая определяет степень принадлежности данного физического расстояния до того или другого термина лингвистической переменной ДИСТАНЦИЯ. Степень принадлежности определяется функцией принадлежности $M(d)$, где d - расстояние до препятствия. К примеру, при расстоянии до препятствия, равному 40 см, можно задать степень принадлежности к терму ОЧЕНЬ БЛИЗКО равную 0,7, а к терму БЛИЗКО - 0,3. Переменной НАПРАВЛЕНИЕ, которая принимает значения в диапазоне от 0 до 360 градусов, зададим термины ЛЕВО, ПРЯМО и ПРАВО. Теперь необходимо задать исходные переменные. В данном примере достаточно одной переменной, которую назовем РУЛЕВОЙ УГОЛ. Она может содержать термины: РЕЗКО ВЛЕВО, ВЛЕВО, ПРЯМО, ВПРАВО, РЕЗКО ВПРАВО. Связи между переменными заносятся в таблицу нечетких правил.

Таблица 1. – Таблица нечетких правил

Дистанция \ Направление	очень близко	близко	среднее	далеко
лево	резко вправо	резко вправо	вправо	прямо
прямо	резко влево	влево	влево	прямо
право	резко влево	резко влево	влево	прямо

Каждая запись в данной таблице соответствует своему нечеткому правилу, например "Если дистанция близко и направление правое, тогда рулевой угол резко влево".

Таким образом, мобильный робот с нечеткой логикой будет работать по следующему принципу: данные от сенсоров о расстоянии до препятствия и направление к нему будут фаззифицированы, обработаны согласно табличным правилам, дефаззифицированы, и полученные данные в виде управляющих сигналов поступают на приводы робота.

Основная идея челночного алгоритма обхода следующая: робот пытается обойти фигуру, совершая горизонтальные челночные движения от границы до границы. Робот, встретив препятствие, меняет строку либо переходит вверх или вниз в зависимости от состояния регистровой памяти. В процессе такого обхода фигура может быть обойдена не полностью ввиду того, что на пути робота встретилась другая фигура. И эта фигура будет препятствовать горизонтально-челночному движению. Робот «переключит свое внимание» на препятствие, уже пытаясь его обойти, придерживаясь той же стратегии обхода. При помощи установки флагов робот контролирует, была ли обойдена фигура полностью или нет. Также робот использует флаг при возврате к обойденной не полностью фигуре [2].

Для реализации алгоритма с помощью нейронных сетей используется нейронная сеть из двух слоев: 6 нейронов первого слоя - по одному на область и 3 выходных нейрона - по одному на выбор движения. Схематичное представление этой сети изображено на рисунке 1.

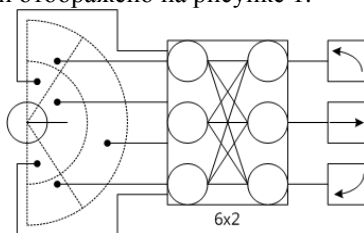


Рисунок 1. – Схематичное отображение нейронной сети

Главная особенность нейронной сети – это ее обучаемость. В нашем случае, обучение направлено на то, чтобы робот не врезался в препятствия. Если столкновение произошло, это значит, что распределение весов сети неправильно, значения весов пересчитываются. Для обучения системы управления робота используется обучение без учителя, основанное на методе Хебба, которое осуществляется на основе следующей формулы:

$$w_{ij}(t) = w_{ij}(t-1) + a y_i^{(n-1)} y_j^{(n)} \quad (1)$$

где $y_i^{(n-1)}$ – выходное значение нейрона i слоя $(n-1)$, $y_j^{(n)}$ – выходное значение нейрона j слоя n ; $w_{ij}(t)$ и $w_{ij}(t-1)$ – весовой коэффициент синапса, соединяющего эти нейроны, на итерациях t и $t-1$ соответственно; a –

коэффициент скорости обучения ($[0;0.3]$). Здесь и далее, для общности, под n подразумевается произвольный слой сети. При обучении согласно данному методу усиливаются связи между возбужденными нейронами. НС робота запоминает последний образ. Это необходимо в том случае, когда робот правильно реагирует на препятствие (не врезается) и отъезжает от препятствия, если после этого попытаться обучить сеть, то ничего не выйдет, т.к. отклики нейронов будут нулевые, поскольку и на входных нейронах будут нулевые сигналы [4].

Очень интересным алгоритмом осуществления движения мобильного робота является алгоритм, основанный на поведении мексиканских прыгающих бобов [5]. Этот алгоритм был разработан исследователями из Технологического института Джорджии (США). Уникальность так называемых прыгающих бобов в том, что у них почти нет вычислительных мощностей или сенсоров. Внутри боба мексиканского кустарника некрупная моль заселяет личинку, которая подвешивает себя на сети шёлковых нитках ко внутренней оболочке боба. Если Солнце нагревает боб, личинка может оказаться обезвоженной и погибнуть. Поэтому при нагревании бобы «прыгают» или «катятся», фактически становясь транспортными средствами личинок. В первом случае личинка удерживается на внутренней поверхности одним концом своего тела и ударяет по этой же поверхности другим концом. Чтобы катиться, она поступает как хомяк в колесе, только не в спортивных, а в транспортных целях: быстро перебирает ложноножками, стоя на месте внутри боба, и заставляет округлую оболочку перекатываться. Чем выше температура, тем выше прыгучесть, позволяющая двигаться быстрее перекатывания: при 45°C боб подпрыгивает 40 раз в минуту, перекатываясь (благодаря округлой форме) после прыжка. Алгоритм основывается на следующем: чтобы выбрать самое холодное место, делается прыжок №1. Если температура снизилась, начинаются прыжки на меньшую дистанцию или перекатывания. Если наоборот — снова следует прыжок на большую дистанцию. Если после прыжка №2 температура окажется выше, чем после прыжка №1, начинается движение назад, но не прыжком, а перекатыванием, с меньшей скоростью и т.д. Легче всего реализуется на нечеткой логике.

Разработка алгоритма. Робот, для которого разрабатывается алгоритм в данной работе, представляет собой платформу на 4 колесах, оснащенную пятью ультразвуковыми датчиками для измерения расстояния, платами Arduino и Raspberry Pi [6]. Также платформа оснащена wifi-адаптером и устройством для распознавания речи, которое хранит и распознает до 16 голосовых команд. Именно с помощью этих устройств и реализуется управление платформой. Способ размещения ультразвуковых сенсоров приведен на рисунке 2.

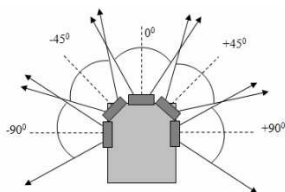


Рисунок 2. – Размещение ультразвуковых сенсоров на платформе

Также важной особенностью аппаратной реализации является устройство и управление движением. Каждое из четырех колес оснащено электродвигателем, однако к контроллеру их управления двигатели подключены попарно: левые отдельно, правые отдельно. Это необходимо для реализации поворотов. Для осуществления поворота запускается одна соответствующая пара двигателей. Таким образом, нельзя заранее предугадать, на какой угол повернется платформа при запуске двигателей на определенное количество секунд, так как скорость и угол перемещения будут очень зависеть от поверхности, по которой движется аппарат. Поэтому все движения согласно разработанному алгоритму будут осуществляться за одну секунду, то есть после каждого запуска двигателей за одну секунду будут выполняться соответствующие проверки.

При ручном управлении платформой с удаленного компьютера программное обеспечение не должно контролировать близость препятствий, так как аппарат находится под контролем пользователя и за его безопасность отвечает пользователь. Однако для обеспечения безопасности конструкции лучше все-таки установить контроль на чрезмерное сближение с преградой, чтобы, если пользователь вдруг отвлечется или не заметит преграды, робот не столкнулся с каким-либо прочным препятствием, а остановился, не доезжая до него.

Куда более важен алгоритм обработки голосовых команд. Для обеспечения управления будут использоваться следующие команды: «вперед», «вправо», «влево», «назад», «стоп». При командах «вправо» и «влево» осуществляется соответствующее движение, пока не будет дана следующая команда, то есть если не будет дана другая команда, робот будет ехать по кругу. При этом будет сохраняться контроль на чрезмерное сближение с преградой, как и при ручном управлении. При команде «назад» безопасность робота полностью лежит на пользователе, так как никаких датчиков, позволяющих фиксировать преграду, на задней стороне платформы нет.

Основная команда, обработку которой и осуществляет разработанный алгоритм, является команда «вперед». По этой команде робот должен двигаться вперед и при достижении преграды пытаться ее обойти и при возможности восстановить первоначальное направление движения.

При разработке алгоритма в первую очередь необходимо учесть тот факт, что платформа будет полностью управляема с удаленного компьютера, поэтому алгоритм системы ее управления должен просто обеспечивать ее

безопасность, то есть реализовывать обход препятствий. То есть перед роботом пока не стоит задача двигаться по какому-либо конкретному маршруту к цели, что весьма усложняло бы алгоритм, поэтому при разработке данного алгоритма нет необходимости использовать сложные теории (такие, как нейронные сети и нечеткая логика). Алгоритм представляет собой просто последовательную обработку показателей датчиков. То есть включает в себя некоторые элементы теории нечетких алгоритмов и элементы челночного алгоритма.

Блок-схема алгоритма приведена на рисунке 3.

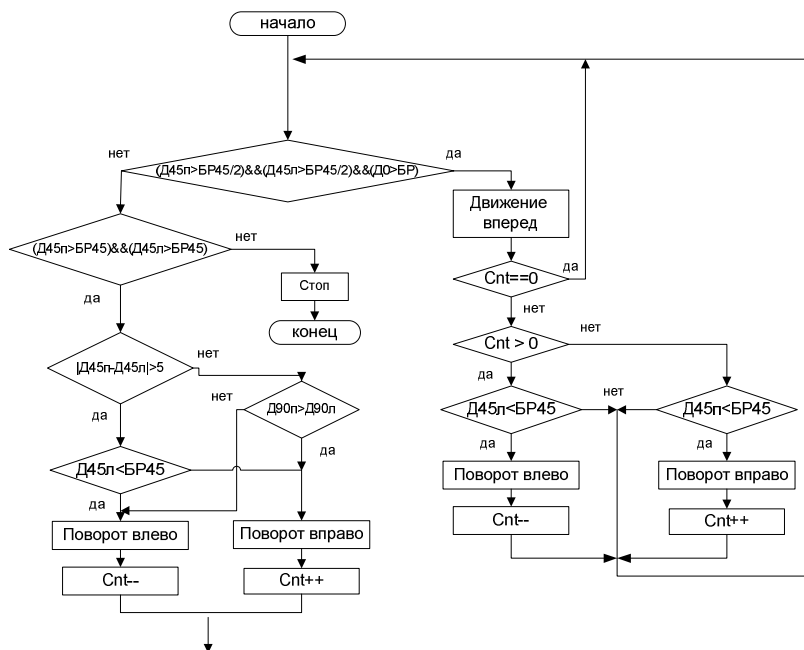


Рисунок 3 – Блок-схема алгоритма

Как уже говорилось выше, вершины «движение вперед», «поворот влево» и «поворот вправо» означают запуск двигателей на одну секунду. В блок-схеме приняты следующие сокращения: Д0, Д45л, Д45п, Д90л, Д90п – показания с соответствующих датчиков, БР, БР45 – безопасные расстояния с соответствующих сторон), Cnt – счетчик для возврата к первоначальному направлению, он наращивается при повороте вправо и уменьшается при повороте влево, при этом после каждого движения проверяется возможность восстановления первоначального направления проверкой этого счетчика на 0. После каждого движения алгоритм пытается приблизить значение счетчика к 0, то есть восстановить первоначальное направление. Основными критериями

для обхода препятствий являются показатели датчиков под углами 0, +45 и -45 градусов. Если показание главного «нулевого» датчика приблизилось к значению его безопасного расстояния, то обязательно нужно реализовать поворот. Для этого проверяются значения «45-градусных» датчиков. Если их значения приблизительно равны (что, к примеру, может произойти при приближении к стене под углом 90 градусов), то направление поворота выбирается в соответствии с показателями датчиков, расположенных под углами 90 градусов: где показатель больше, то есть больше свободного пространства, туда и осуществляется поворот. Если же значения «45-градусных» датчиков отличаются больше, чем на 5 см, то осуществляется поворот в сторону большего значения.

Первая основная вершина обеспечивает возможность прохождения робота через узкий проход, когда расстояния под углом 45 градусов меньше необходимого расстояния для поворота (для этого осуществляется проверка на половину безопасного расстояния), но значение «нулевого» датчика больше безопасного расстояния.

Если же робот заедет в тупик, то есть показатели трех основных датчиков будут меньше или равны соответствующим значениям безопасного расстояния, то робот осуществляет остановку.

Перспективные задачи. В перспективе планируется программная реализация и тестирование данного алгоритма на описанной платформе, а также установка GSM-модуля в конструкцию для возможности удаленного управления роботом при помощи клавиш телефона.

Список литературы

1. Юревич Е.И. «Основы робототехники». - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2005. - 416 с.: ил.
2. Килибарда Г., Кудрявцев В.Б., Ушчюмлич Ш. Независимые системы автоматов в лабиринте // Дискретная математика – Т. 15. – В. 3 – 2003.
3. Круглов В.В. Нечеткая логика и искусственные нейронные сети / Круглов В.В., Дли М.И., Голунов Р.Ю. – М.: Физматлит, 2001. – 224 с.
4. Нейронная сеть для обхода препятствий. Электронный ресурс. Режим доступа: <http://stswoon.blogspot.com/2010/05/blog-post.html>
5. Передвижение мексиканских прыгающих бобов. Электронный ресурс. Режим доступа: <http://www-old.me.gatech.edu/hu/Publications/Hu12-beans.pdf>
6. Бровкина Д.Ю., Приходько Т.А. Разработка радиоуправляемой сенсорной платформы Информатика та комп'ютерні технології / Збірка праць VIII міжнародної науково-технічної конференції студентів, аспірантів та молодих науковців – 18-19 вересня 2012 р., Донецьк, ДонНТУ. – 2012. У 2-х томах, Т. 2. – 334 с. 228-233