

УДК 681.5
EDN ZKNABI

Н. АЛСАЙЕД, А. Ю. КРАСНОВ

АЛГОРИТМЫ НАВИГАЦИИ САМОХОДНОГО КОЛЕСНОГО РОБОТА В ИЗМЕНЯЮЩЕЙСЯ ВНЕШНЕЙ ОБСТАНОВКЕ, ГАРАНТИРУЮЩИЕ ПРЕДУПРЕЖДЕНИЕ СТОЛКНОВЕНИЙ С ПОДВИЖНЫМИ И НЕПОДВИЖНЫМИ ПРЕПЯТСТВИЯМИ

В статье описывается инновационный метод автономной навигации самоходного колесного робота Pioneer 3-DX в условиях наличия как статических, так и динамических препятствий. Для управления роботом используется алгоритм искусственных потенциальных полей (ИПП), позволяющий рассчитывать безопасную траекторию, а также нейросеть, помогающая классифицировать области вероятной опасности. Три ультразвуковых датчика обеспечивают измерение расстояния для оценки опасности, что наряду с данными об относительной скорости и направлении дает возможность идентифицировать области повышенного (зона 1) и меньшего риска (зона 2). При обнаружении опасности система нечеткой логики обеспечивает эффективное предотвращение столкновения, регулируя скорость колес. Результаты моделирования, проведенного в среде MATLAB и V-REP, демонстрируют эффективность предложенного алгоритма по сравнению с альтернативными подходами для навигации автономных мобильных роботов в сложных условиях с разными уровнями риска. Показано, что алгоритм обладает такими преимуществами, как адаптивность, отказоустойчивость и надежность.

Ключевые слова: расчет траектории, навигация мобильных роботов, нейросеть, нечеткая логика, исключение столкновений.

1. Введение

Автономная навигация в незнакомом районе ставит перед мобильными роботами одну из наиболее сложных задач – идентификации локальных препятствий, перемещения среди них и, наконец, достижения заданной конечной точки. Расчет траектории и последующая навигация представляют собой две основные парадигмы в навигации мобильных роботов. В течение многих лет исследователи бьются над решением задач навигации и управления, связанных с мобильными роботами, с помощью различных методов, в том числе ИПП, генетических алгоритмов, нейросетей, компьютерного зрения и нечеткой логики. Эти подходы способствовали преодолению разнообразных навигационных препятствий, стоящих на пути мобильных роботов в реальной обстановке [1, 2]. Метод ИПП наиболее востребован в мобильной робототехнике для расчета безопасной траектории. Считается, что мобильный

Алсайед Нур. Аспирант, факультет систем управления и робототехники, Университет ИТМО (С.-Петербург). ORCID 0009-0006-6157-0877.

Краснов Александр Юрьевич. Кандидат технических наук, преподаватель, факультет систем управления и робототехники, Университет ИТМО. ORCID 0000-0001-6026-6706.

робот подвержен влиянию искусственной потенциальной силы, вырабатываемой искусственным потенциальным полем. Функция потенциала является скалярной и задействуется в методе ИПП [3, 4]. При этом цель и препятствия рассматриваются как заряженные поверхности, где накопленный потенциал прилагает к роботу виртуальную силу, с помощью которой тот движется в направлении цели, одновременно обходя препятствия. В [5] описывается навигационная стратегия, объединяющая понятие виртуальных препятствий и метод потенциальных полей для управления цилиндрическими мобильными роботами в незнакомой среде. Были проведены компьютерное моделирование и испытания, продемонстрировавшие эффективность стратегии и снижение локальных минимумов, связанных с методами потенциальных полей. В [6] представлена новая потенциальная функция, специально адаптированная к постоянно меняющимся окружающим условиям – присутствию подвижных препятствий и целей. Примечательно, что предложенная функция охватывает данные о координатах и скоростях роботов, препятствиях и целях, что повышает адаптивность метода ИПП к таким динамическим сценариям.

Нейросети представляют собой набор нелинейных, многослойных, параллельных регрессионных подходов для группирования, прогнозирования и обработки сигналов [7]. Чтобы повысить эффективность функционирования в реальном времени и точность позиционирования мобильных роботов внутри помещения, в [8] был задействован метод комплексирования данных, полученных от нейросети, позволяющий снизить отрицательное воздействие среды или погрешностей измерений. В [9] исследовался вопрос навигации автономного колесного робота при использовании метода ИПП, классификации зон опасности посредством нейросети и нечеткой логики для предотвращения столкновений с помощью ультразвуковых датчиков. Этот подход, сочетающий архитектуру адаптивной сети на основе системы нечеткого вывода (ANFIS) и алгоритм безопасных границ, продемонстрировал высокую эффективность решения задач навигации, в особенности при обходе изогнутых и неровных препятствий. В настоящей работе показано, что методы нечеткой логики, нейросетей и ANFIS позволяют существенно повысить точность навигации автономных мобильных роботов.

Метод нечеткой логики является наиболее действенным при обработке недостаточной информации, такой как неопределенность измерений и моделей процессов при навигации роботов [1]. Для достижения цели в полностью статической среде авторы статьи [10] разработали контроллер, базирующийся на принципах нечеткой логики, входными данными для которого стали три расстояния, а выходными – скорость левого и правого колеса.

В [11] исследовалось управление движением колесного робота Pioneer 3-DX с одновременным использованием алгоритмов ИПП и нечеткой логики для обхода препятствий. Подход доказал свою эффективность благодаря способности быстро адаптироваться к изменению условий окружающей среды и предотвращать столкновения с препятствиями за счет автоматического регулирования скорости при учете размеров робота и безопасных расстояний до объектов. В автономной навигации роботов это является значительным прогрессом. Работа [12] посвящена моделированию движения мобильного робота с обходом препятствий, навигация осуществлялась с использованием контроллера с нечеткой логикой и ультразвуковых датчиков. Моделирование, проводившееся в среде MATLAB при заданных 25 правилах обхода препятствий, продемонстрировало превосходные результаты в различных условиях окружающей сре-

ды. Применение алгоритма нечетких рассуждений Мамдани позволило значительно улучшить обход препятствий, что расширяет навигационные возможности робота.

В [13] исследовались способы комбинирования нейросетей и метода нечеткой логики. Поиск цели и обход препятствий выполнялись с помощью нейросети, обученной на наборе данных, выработанных двумя разными навигационными алгоритмами с использованием нечеткой логики.

В [14] этот подход был разделен на два этапа. На первом этапе для обхода препятствий был задействован нечеткий генетический алгоритм (Fuzzy-GA), предполагающий формирование одной или двух траекторий с учетом статической окружающей обстановки и вычисление расстояния и направления на препятствие. Целью является поиск для них оптимального диапазона. На втором этапе выполняется обход движущихся препятствий за счет контроля относительной скорости. Полученные результаты продемонстрировали высокую вычислительную эффективность метода планирования движения мобильного робота, объединяющего Fuzzy-GA и концепцию трех траекторий (Three Path Concept).

В [15] описывается новый метод FLACO, в соответствии с которым оптимальная траектория в дорожной сети определялась в рамках концепции протяженности виртуального маршрута. Метод позволял вырабатывать кратчайший виртуальный маршрут даже в непредсказуемых ситуациях, что открывает широкие возможности применения FLACO для навигации автономных объектов в непрерывно изменяющихся условиях.

В настоящей статье особое внимание уделяется реактивным алгоритмам, предусматривающим использование метода ИПП для определения начальной траектории с целью обхода движущихся препятствий. Обход препятствий, как движущихся, так и неподвижных, выполнялся с помощью нейронных сетей, классифицирующих зоны опасности, и метода нечеткой логики для контроля скорости.

Далее статья структурирована следующим образом: в разделе 2 дается краткое описание предлагаемого подхода. В разделе 3 рассматривается первый алгоритм для выработки траектории движения. Раздел 4 посвящен нейронной сети и наборам данных. В разделе 5 описывается алгоритм нечеткой логики для управления мобильным роботом. В разделе 6 представлены результаты проведенного моделирования. В разделе 7 они сравниваются с результатами других исследований. Наконец, в заключении приводятся выводы.

2. Предложенный подход

Целью предлагаемого подхода является построение универсального алгоритма для эффективного управления мобильными роботизированными системами в сложной окружающей обстановке при соблюдении мер безопасности. Алгоритм состоит из трех основных процессов: определения начальных условий окружающей среды и планирования траектории с помощью ИПП, мониторинга движущихся препятствий и их классификации посредством байесовской регрессионной нейросети (BRNN), обхода препятствий с использованием метода нечеткой логики.

Вначале с учетом данных лазерного датчика Нокьюо определяются заданные условия среды (обстановки), в том числе исходное положение мобильного робота и пункт назначения. На основе этой информации алгоритм ИПП вырабатывает оп-

тимальную траекторию движения робота в точку прибытия, включая обход препятствий. Эта траектория служит базой для последующей навигации.

Движущиеся препятствия непрерывно отслеживаются при помощи набора ультразвуковых датчиков, расположенных на левой, правой и передней сторонах робота. Данные о расстояниях, измеренных этими датчиками, а также об относительной скорости и направлениях на препятствия служат входными данными для классификатора BRNN, распределяющего их по категориям «безопасные» и «потенциально опасные».

Когда BRNN фиксирует опасное препятствие, система активизирует модуль нечеткой логики для его обхода. Как было сказано выше, алгоритм нечеткой логики регулирует скорость колес робота, что дает последнему возможность эффективно маневрировать и обходить препятствия, после чего робот возобновляет движение по траектории, заданной алгоритмом ИПП.

Предлагаемая архитектура (рис. 1) обеспечивает эффективную и безопасную навигацию мобильного робота в сложной окружающей обстановке. Объединение алгоритма ИПП, классификатора BRNN и метода нечеткой логики позволяет решить задачи навигации и соблюдать меры безопасности на протяжении всего пути.

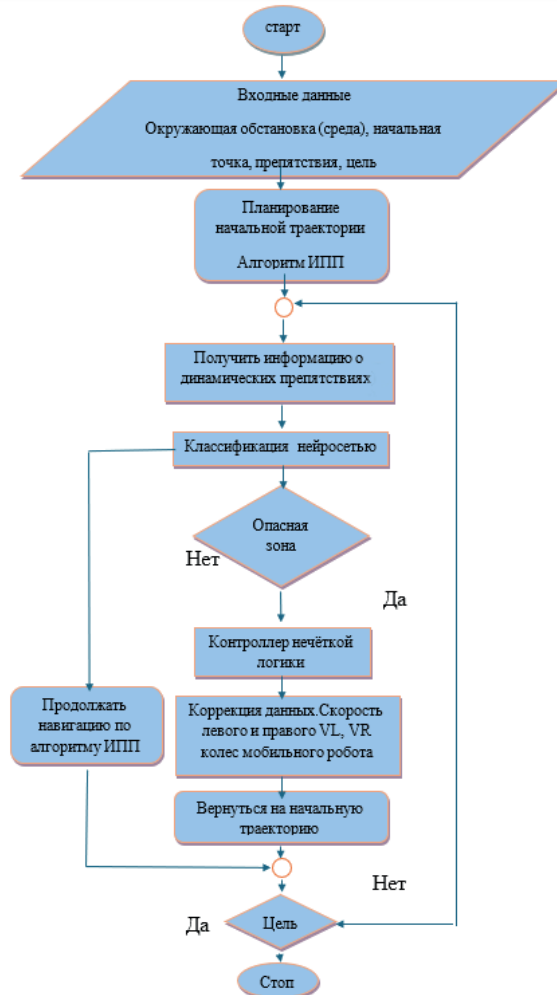


Рис. 1. Алгоритм объединения работы нейросети и нечеткой логики

3. Начальное планирование траектории

Планирование траектории – ключевой момент навигации автономного мобильного робота, который хорошо изучен в робототехнике. Несмотря на это, планирование траектории движения мобильных роботов все еще представляется сложной задачей. Робототехнические системы выбирают оптимальную или субоптимальную траекторию на основе таких факторов, как цена прохождения трассы или продолжительность пути. Траектория, состоящая из последовательности отрезков, позволяет роботу через точки поворота пройти путь от исходного положения до точки прибытия [23].

3.1. Искусственные потенциальные поля

Предполагается, что робот – это объект с точечной массой. Пусть q обозначает координаты положения робота в двумерном (2D) пространстве: $q = [x \ y]^T$, где x и y – местоположение мобильного робота в декартовой системе координат, при этом $q_{obs} = (x_{obs}, y_{obs})$ и $q_{goal} = (x_{goal}, y_{goal})$ – координаты препятствий и конечной точки соответственно.

Сила притяжения. Функции потенциального поля часто представляются в виде параболы, как показано на рис. 2. С учетом расстояния до цели эта функция растет квадратично [24]:

$$U_{att}(q) = \frac{1}{2} \zeta d^2(q, q_{goal}), \quad (1)$$

где ζ – коэффициент пропорциональности функции притягивающего потенциала. В терминах евклидова пространства расстояние между роботом q и точкой назначения q_{goal} обозначается как $d(q, q_{goal})$. При вычислении отрицательного градиента поля притягивающего потенциала силу притяжения, действующую на робота, можно определить следующим образом:

$$F_{att}(q) = -\nabla U_{att}(q) = -\zeta d(q, q_{goal}). \quad (2)$$

Отталкивающий потенциал. Расстояние до препятствий обратно пропорционально суммарной силе отталкивания всех расположенных рядом препятствий. Поле отталкивающего потенциала

$$U_{rep}(q) = \sum_i U_{repi}(q), \quad (3)$$

где i – количество препятствий, влияющих на окружающую робота среду и порождающих отталкивающий потенциал $U_{repi}(q)$ (рис. 3.)

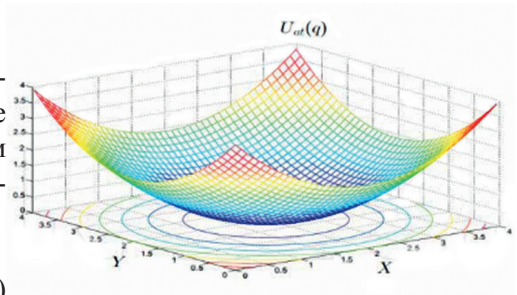


Рис. 2. Поле притягивающего потенциала

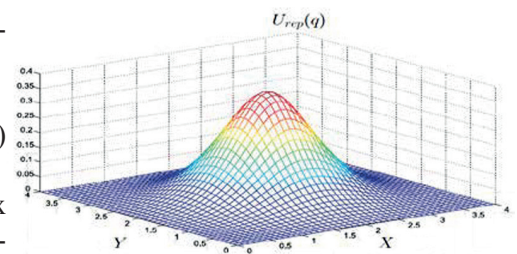


Рис. 3. Поле отталкивающего потенциала

Функция отталкивания имеет вид [25]:

$$U_{rep}(q) = \begin{cases} \frac{1}{2} \eta \left(\frac{1}{d(q, q_{obs})} - \frac{1}{d_0} \right)^2 & \text{при } d(q, q_{obs}) < d_0, \\ 0 & \text{при } d(q, q_{obs}) > d_0 \end{cases}, \quad (4)$$

где q_{obs} – местоположение препятствия; q – местоположение робота; d_0 – положительная постоянная, обозначающая область влияния препятствия; $d(q, q_{obs})$ – расстояние между роботом и препятствием; η – медленно меняющаяся постоянная константа, которая является коэффициентом пропорциональности функции отталкивающего потенциала. Силу отталкивания можно рассчитать как отрицательный градиент функции отталкивающего потенциала:

$$F_{rep}(q) = -\nabla U_{rep}(q) = \begin{cases} \eta \left(\frac{1}{d^*} - \frac{1}{d_0} \right) \frac{(q - q_{obs})}{d^3(q - q_{obs})} & \text{при } d^* < d_0, \\ 0 & \text{при } d^* > d_0 \end{cases}, \quad (5)$$

где $d^* = d(q, q_{obs})$.

Общее поле потенциала определяется как сумма притягивающего U_{att} и отталкивающего U_{rep} потенциалов. [26]. Математически суммарное поле потенциала можно записать как

$$U(q) = U_{att}(q) + U_{rep}(q). \quad (6)$$

Суммарная сила, приложенная к мобильному роботу, определяется путем вычисления отрицательного градиента общего поля потенциала, который представляет собой направление наискорейшего спуска, определяющего движение робота:

$$F(q) = -\nabla U(q) = -\nabla U_{att} - \nabla U_{rep}. \quad (7)$$

Сила, действующая на робота, складывается из сил притяжения и отталкивания:

$$F(q) = F_{att}(q) + F_{rep}(q), \quad (8)$$

где ∇U – вектор-градиент U в месте нахождения робота; F_{att} и F_{rep} – соответствующие векторы.

На начальном этапе работы задаются условия окружающей обстановки и с помощью алгоритма ИПП выработывается траектория. Точное измерение расстояний и широкое поле зрения, необходимые для оценки окружающей обстановки, обеспечиваются за счет применения лазерного датчика Нокуюо.

Данные датчика задействуются алгоритмом ИПП для построения потенциального поля в окружающей среде и ведения робота от исходной точки до конечной с обходом препятствий. Алгоритм ИПП распознает окружающую среду как последовательность притягивающих и отталкивающих сил, что позволяет при его использовании эффективно сочетать целенаправленное движение к точке назначения и обход препятствий. Благодаря тому, что лазерный датчик Нокуюо способен обнаруживать препятствия и точно вычислять расстояние до них, алгоритм выработывает такую оптимальную траекторию, чтобы минимизировать потенциальные столкновения и максимально повысить точность и безопасность навигации в сложной окружающей обстановке.

Несмотря на эффективность алгоритма ИПП при построении начальных траекторий, в ходе движения могут возникать трудности, обусловленные препятствиями сложной формы или их непрерывными изменениями. Для устранения этих ограничений мы и предлагаем использовать сочетание передовых методов – нечеткой логики и нейросетей. Эти методы расширяют возможности алгоритма ИПП, позволяя ему более эффективно управлять мобильными роботами в условиях неизвестной неизменной и постоянно меняющейся среды.

3.2. Создание модели робота

Задействованный в эксперименте робот Pioneer 3-DX оснащен двумя колесами и 16 ультразвуковыми датчиками для оценки окружающей обстановки. Кроме того, на корпусе робота установлен лазерный датчик Нокуюо, с помощью которого измеряются расстояния и реализуется траектория (рис. 4).

Кинематическая и динамическая модели робота Pioneer 3-DX, представленные на рис. 5, характеризуются неголономностью, обусловленной принципом вращающихся колес и их неспособностью двигаться в поперечном направлении. Робот имеет два ведущих колеса спереди и опорное колесико сзади для поддержания равновесия. Управление движением и ориентацией робота осуществляется посредством двух отдельных двигателей, которые вращают его колеса [27]:

$$q = [x, t, \theta]^T. \quad (9)$$

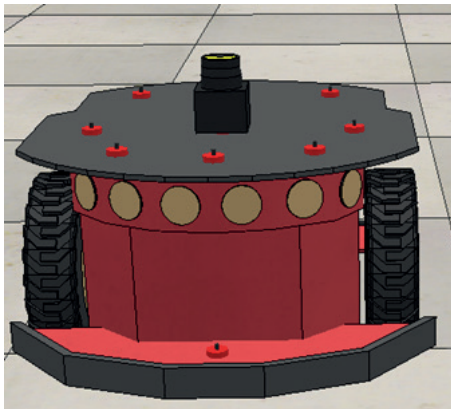


Рис. 4. Модель колесного мобильного робота Pioneer 3-DX

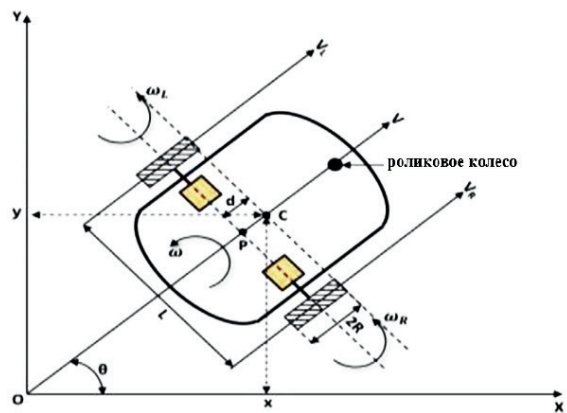


Рис. 5. Кинематическая и динамическая модели

Предположим, что ведущие колеса мобильного робота не подвержены проскальзыванию. Неголономная связь описывается уравнением

$$y \cos \theta - x \sin \theta = 0. \quad (10)$$

Соотношение линейной и угловой скорости колес записывается в виде уравнений:

$$V = \omega \cdot R, \quad (11)$$

$$V_r = \omega_r \cdot R, \quad (12)$$

$$V_l = \omega_l \cdot R, \quad (13)$$

$$\omega = \frac{V_r - V_l}{L}, \quad (14)$$

$$V = \frac{V_r + V_l}{2}. \quad (15)$$

Объединив уравнения (14) и (15), получим

$$\begin{bmatrix} V \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} V_r \\ V_l \end{bmatrix}. \quad (16)$$

Скорость движения мобильного робота по осям (x, y) и изменение угла поворота робота рассчитываем по формулам:

$$\begin{aligned} \frac{dx}{dt} &= \dot{x} = V \cdot \cos \theta, \\ \frac{dy}{dt} &= \dot{y} = V \cdot \sin \theta, \\ \frac{d\theta}{dt} &= \dot{\theta} = \omega. \end{aligned} \quad (17)$$

В результате получим кинематические уравнения, описывающие модель двухколесного мобильного робота:

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V \\ \omega \end{bmatrix}. \quad (18)$$

Подставим (14) и (15) в уравнения (16), (17) и (18):

$$\begin{aligned} \dot{x} &= \frac{R}{2}(\omega_r + \omega_l) \cdot \cos \theta, \\ \dot{y} &= \frac{R}{2}(\omega_r + \omega_l) \cdot \sin \theta, \\ \dot{\theta} &= \frac{R}{2}(\omega_r - \omega_l). \end{aligned} \quad (19)$$

Выразим кинематическую модель мобильного робота через систему уравнений:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} \cos \theta & \frac{R}{2} \cos \theta \\ \frac{R}{2} \sin \theta & \frac{R}{2} \sin \theta \\ \frac{R}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix}. \quad (20)$$

Движение робота [28] описывается формулой (11).

Управление мобильным роботом осуществляется изменением линейных скоростей V_r и V_l правого и левого колес, соответствующих их угловым скоростям ω_r и ω_l . Линейная и угловая скорости центра масс мобильного робота обозначаются V и ω соответственно.

4. Классификация препятствий и прогнозирование столкновений

Основываясь на данных ультразвуковых датчиков об измеренных расстояниях, скорости мобильного робота относительно препятствий и углах, задающих направления на препятствия, классифицируем участки по ходу движения робота на чрезвычайно опасные (зона 1) и не представляющие угрозы (зона 2). Препятствие идентифицируется как опасное в зависимости от близости к роботу и относительной скорости, с которой тот перемещается. Таким образом, неопасным будет считаться препятствие, находящееся далеко от робота. В табл. 1 показано распределение этих зон на основе эмпирических данных. Заметим, что скорость и расстояние между препятствием и мобильным роботом имеют наибольшее значение. При более высоких скоростях и на более близких расстояниях риск столкновения увеличивается, что отражено в табл. 1, при этом диапазон значений углов, задающих направления на препятствия, для каждой ячейки составляет от 180 до -180° .

Таблица 1

Классификация зон

Расстояние (м) \ Скорость (м/с)	0,2	0,5	0,7 и более
0,2	зона 1	зона 1	зона 2
0,3	зона 1	зона 2	зона 2
0,5	зона 1	зона 2	зона 2

4.1. Сбор данных

Ключевым аспектом при сборе данных является обучение нейросети, с тем чтобы правильно классифицировать препятствия, встречающиеся на пути робота.

На вход предлагаемой нейросети подается пять параметров и два – на выход. Три входных параметра – это расстояния до препятствий, измеренные передними, левым и правым датчиками робота Pioneer 3-DX соответственно, дальность действия которых изменяется от 0,1 до 1 м. Как показано на рис. 6, показания двух передних ультразвуковых датчиков (№3 и 4) используются для определения расстояния до объектов впереди, датчика №0 – до объектов слева, а датчика №7 – для объектов справа. Остальные два параметра – это скорость мобильного робота относительно препятствия и угол, задающий направление между ними.

Высокоточная навигация в зоне 1, определяемой как область на расстоянии от 0,2 до 0,4 м от обнаруженных препятствий, достигается при заданной скорости 0,4 м/с. При этих данных на пути, который проходит робот, отмечается 10 равноудаленных точек в рамках указанного диапазона расстояний от препятствий, что позволяет охватить всю зону. Кроме того, чтобы точно определить местоположение препятствия, в каждой точке тщательно измеряются тринадцать углов в диапазоне от -180 до 180° с шагом 30° . Результаты применения данного подхода представлены на рис. 7, где зеленым цветом обозначены сегменты, на которые была разделена траектория движения, а белыми стрелками – распределение углов. На завер-

шающем этапе алгоритма фиксируются результаты 1170 измерений, выполненных в рамках 13 сценариев (см. табл. 2).

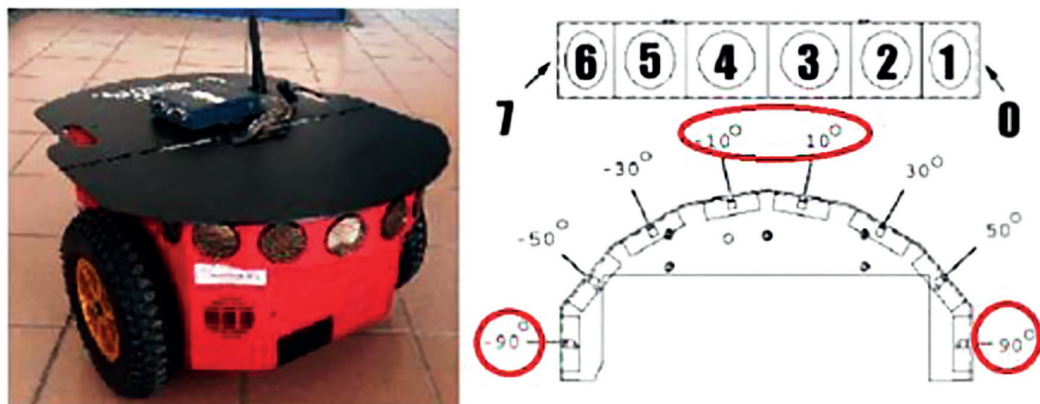


Рис. 6. Датчики, используемые в схеме обхода препятствий

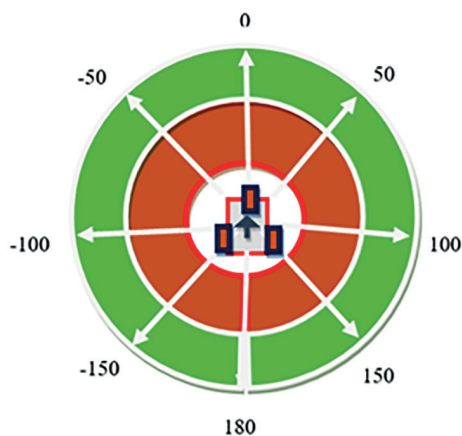


Рис. 7. Диапазон углов траектории

Процесс сбора данных можно упростить за счет объединения среды MATLAB и программы V-REP, известной также как CoppeliaSim – ведущей мировой платформы для 3D-моделирования в области робототехники. Оснащенная удобным программным интерфейсом (API), платформа V-REP позволяет пользователям моделировать целые роботизированные системы или их компоненты – датчики и приводы. Она обеспечивает интеграцию выполняемых роботом операций за счет возможности отдельно управлять моделируемыми объектами или моделями с помощью встроенных скриптов или удаленного клиента API в интегрированной среде обработки и архитектуре распределенного управления [29].

В ходе эксперимента для измерения дистанции до подвижных объектов, перемещающихся с заданными скоростями и по заданным траекториям, мы воспользовались датчиком, установленным на роботе Pioneer 3-DX в среде V-REP. По мере приближения этих объектов к роботу V-REP передает десять измерений в MATLAB для обучения нейросети. Таким образом, взаимодействие сред MATLAB и V-REP позволяет оптимизировать процесс сбора данных и задать параметры движения препятствий в среде моделирования.

4.2. Нейросети

Одна из первых моделей нейронных сетей прямого распространения – перцептрон Розенблатта – изначально была разработана для управления базовыми операциями [30]. С течением времени задачи усложнялись, объем данных увеличивался, в результате возникла необходимость в дополнительных слоях. Многослойные перцептроны получили широкое распространение [31]. Наличие нескольких слоев упрощает сложные процессы обучения нейросети, поскольку выходные данные со всех узлов одного слоя передаются на входы всех узлов следующего (рис. 8).

Однослойная нейросеть имеет ограничения из-за скрытых нейронов, не связанных напрямую с входными или выходными данными и неспособных по этой причине моделировать нелинейности. Учитывая, что большинству физических систем свойственна нелинейность, для моделирования нелинейных функций требуется обучение многослойного перцептрона. В современных исследованиях, посвященных искусственному интеллекту, в качестве основного способа обучения нейросети применяется метод обратного распространения ошибки, суть которого состоит в коррекции ошибок в обратном направлении по сети. Этот итеративный процесс позволяет минимизировать целевую функцию ошибки E_w путем корректировки весовых коэффициентов в направлении от выходного слоя ко входному при условии, что ошибка превышает заданный порог. Нейросеть состоит из трех слоев: входного – для получения обучающих данных, скрытого – для обработки и выходного – для выдачи решений [32]:

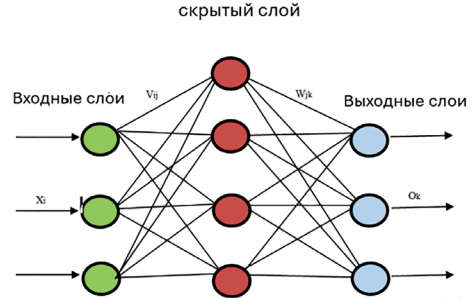


Рис. 8. Многослойный перцептрон

$$E_w = \frac{1}{2} \sum_{k=0}^k (t_k - O_k)^2, \quad (21)$$

где x_i – i -я входная величина; y_j – выходная величина j -го скрытого нейрона; O_k – выходная величина k -го выходного нейрона; t_k – искомое значение выходной величины.

При этом для корректировки весов скрытых слоев используется уравнение

$$V_{ij}(n+1) = V_{ij}(n) + \eta \times \delta_j(n) \times x_i(n), \quad (22)$$

где V_{ij} – весовой коэффициент от i -й входной величины к j -му скрытому нейрону; W_{kj} – весовой коэффициент от j -го скрытого нейрона к k -му выходному нейрону; η – скорость обучения; i – индекс для входных нейронов.

Сигнал ошибки, вырабатываемый j -м скрытым нейроном, обозначается символом δ_j . Следующее уравнение позволяет скорректировать веса в выходном слое:

$$W_{kj}(n+1) = W_{kj}(n) + \eta \times \delta_k(n) \times y_j(n). \quad (23)$$

Сигнал ошибки, вырабатываемый k -м выходным нейроном, обозначается как δ_k :

$$\delta_k(n+1) = (t_k - o_k) \times (1 - o_k) \times o_k. \quad (24)$$

С помощью δ_k можно вычислить δ_j :

$$\delta_j = (1 - y_j) \times y_i \times \sum_{k=0}^k \delta_k \times W_{jk} . \tag{25}$$

Процесс предусматривает итеративную корректировку весов нейронов скрытого и выходного слоев на основе определенных формул; при этом сигналы ошибок распространяются в обратном направлении от скрытых нейронов к выходному слою до тех пор, пока значение ошибки не окажется ниже заданного порога для каждой комбинации входных/выходных сигналов [33]. В отличие от обычных методов обратного распространения сигнала ошибки этот подход позволяет избежать медленной сходимости, характерной для градиентных алгоритмов первого порядка, и, таким образом, обеспечивает эффективное обучение. В нашем исследовании было проведено обучение нейросети прямого распространения для классификации опасных зон с применением обучающего алгоритма обратного распространения ошибок на собранных данных, приведенных в табл. 2.

Таблица 2

Коды зон

Зоны классификации	Выходное значение нейросети
зона 1	10
зона 2	01

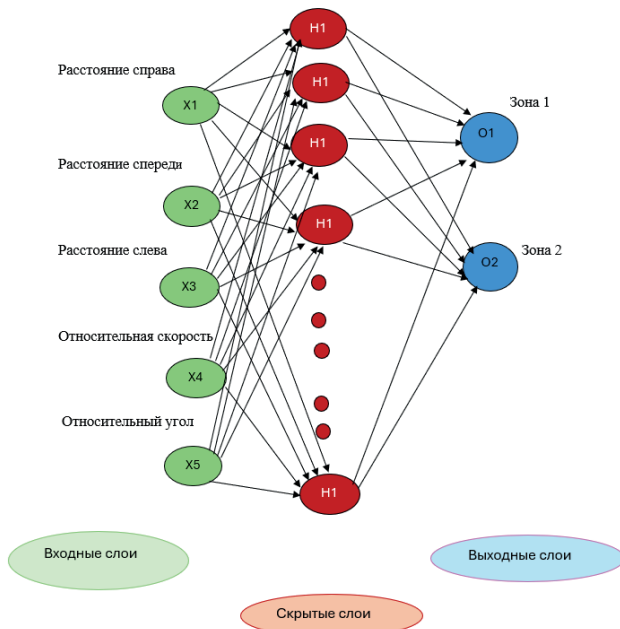


Рис. 9. Структура используемой нейросети

Чтобы упростить обучение и оценку сети, база данных разделена на две части: примерно 70% используется для обучения и 30% – для тестирования. Такое разделение повторяется на 100 эпохах в процессе обучения (количество эпох выбрано таким образом, чтобы достичь заданной точности классификации и избежать переобучения). Во время обучения нейросеть достигает впечатляющей точности 98,9% как по обучающим, так и по контрольным выборкам. Важно учитывать, что оптимальное количество нейронов и скрытых слоев зависит от характеристик обучающих выборок данных [34].

В результате процесса обучения точность существенно повышается, что видно из рис. 9 и табл. 3. Кроме того, матрицы неточностей сети, изображенные на рис. 10, наглядно демонстрируют эффективность работы сети при классификации препятствий в окружающей среде.

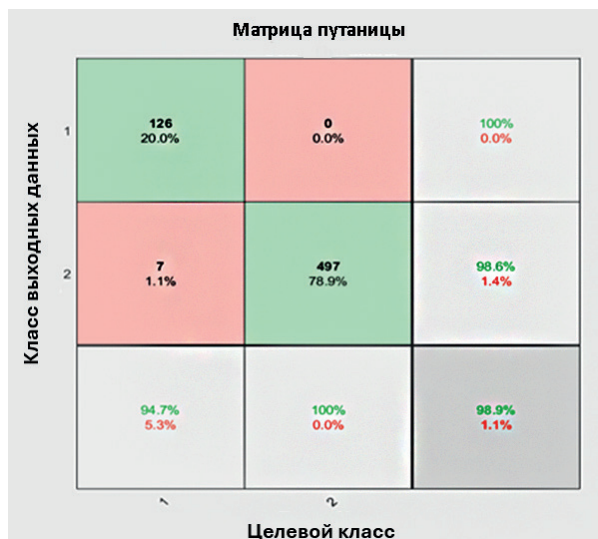


Рис. 10. Матрица неточностей нейросети

Таблица 3

Характеристики исследуемой нейросети

Параметр	Значение
Количество нейронов во входном слое	5
Количество нейронов в скрытом слое	10
Количество нейронов в выходном слое	2
Функция активации скрытого слоя	Байесовская регуляризация
Функция активации выходного слоя	Линейная
Количество интервалов	100
Скорость обучения нейросети	0,05
Точность	98,9%

5. Нечеткая логика

В 1960-х годах математик Лотфи Заде ввел понятие нечетких систем, предложив гибкий метод обоснования фактов с помощью нечетких экспертных знаний. В состав нечеткой системы входят такие элементы, как база знаний, механизм логического вывода, дефаззификатор и фаззификатор; она преобразует точные входные данные в нечеткие функции, оценивает их принадлежность к нечетким концепциям и задействует методы нечеткого логического вывода для принятия решений по управлению. Затем происходит процесс дефаззификации, в результате которого нечеткие значения управляющего сигнала преобразуются в реальные целочисленные выходные данные, которые легко применимы на практике [35].

Показания датчиков робота и расстояния до препятствий спереди, слева и справа представляются в виде трех пакетов входных данных, которые используются системой нечеткой логики: Forward Distance (расстояние спереди), Left Distance (рассто-

яние слева) и Right Distance (расстояние справа). Скорости правого и левого колес робота, обуславливающие скорость его движения, являются выходными данными системы. На рис. 11 показан процесс принятия решений. Нечеткая система использует три входных параметра – расстояния спереди, слева и справа, полученные от датчиков робота. Когда робот приближается к опасной зоне, идентифицированной нейросетью, нечеткая логическая система анализирует эти входные расстояния для определения соответствующих корректировок скоростей левого и правого моторов.



Рис. 11. Нечеткая логика системы

В рамках нечеткой логики реализованы три функции принадлежности для каждого множества входных данных, где каждое входное расстояние обозначается лингвистическими переменными: «близкое», «среднее», «далекое». Для нечетких выходных данных реализованы три функции принадлежности, для каждого блока выходных данных – следующие функции принадлежности для левого и правого двигателей: «реверс», «медленно», «средне», «быстро». У каждого блока входных данных имеются три функции принадлежности, следовательно, общее количество правил, необходимых для управления нечеткой системой, равно 27 (количество возможных комбинаций значений трех трехзначных входных параметров). На рис. 12 приведен пример с 27 правилами нечеткой логики, в том числе правила «ЕСЛИ – ТО», применяемые к контроллеру, для которого входными данными являются расстояния спереди (flc_FD), слева (flc_LD) и справа (flc_RD), а выходными – скорость левого (L_motor) и правого (R_motor) колесных двигателей робота для предотвращения столкновения.

```

1. If (flc_LeftDistance is far) and (flc_ForwardDistance is far) and (flc_RightDistance is far) then (L_motor is fast)(R_motor is fast) (1)
2. If (flc_LeftDistance is far) and (flc_ForwardDistance is far) and (flc_RightDistance is medium) then (L_motor is medium)(R_motor is fast) (1)
3. If (flc_LeftDistance is far) and (flc_ForwardDistance is far) and (flc_RightDistance is near) then (L_motor is slow)(R_motor is fast) (1)
4. If (flc_LeftDistance is far) and (flc_ForwardDistance is medium) and (flc_RightDistance is far) then (L_motor is slow)(R_motor is slow) (1)
5. If (flc_LeftDistance is far) and (flc_ForwardDistance is near) and (flc_RightDistance is far) then (L_motor is slow)(R_motor is reverse) (1)
6. If (flc_LeftDistance is far) and (flc_ForwardDistance is medium) and (flc_RightDistance is medium) then (L_motor is slow)(R_motor is medium) (1)
7. If (flc_LeftDistance is far) and (flc_ForwardDistance is medium) and (flc_RightDistance is near) then (L_motor is slow)(R_motor is fast) (1)
8. If (flc_LeftDistance is far) and (flc_ForwardDistance is near) and (flc_RightDistance is medium) then (L_motor is reverse)(R_motor is slow) (1)
9. If (flc_LeftDistance is far) and (flc_ForwardDistance is near) and (flc_RightDistance is near) then (L_motor is reverse)(R_motor is slow) (1)
10. If (flc_LeftDistance is medium) and (flc_ForwardDistance is far) and (flc_RightDistance is far) then (L_motor is fast)(R_motor is medium) (1)
11. If (flc_LeftDistance is medium) and (flc_ForwardDistance is far) and (flc_RightDistance is medium) then (L_motor is medium)(R_motor is medium) (1)
12. If (flc_LeftDistance is medium) and (flc_ForwardDistance is medium) and (flc_RightDistance is far) then (L_motor is fast)(R_motor is fast) (1)
13. If (flc_LeftDistance is medium) and (flc_ForwardDistance is medium) and (flc_RightDistance is medium) then (L_motor is medium)(R_motor is medium) (1)
14. If (flc_LeftDistance is medium) and (flc_ForwardDistance is medium) and (flc_RightDistance is near) then (L_motor is slow)(R_motor is fast) (1)
15. If (flc_LeftDistance is medium) and (flc_ForwardDistance is near) and (flc_RightDistance is far) then (L_motor is slow)(R_motor is reverse) (1)
16. If (flc_LeftDistance is medium) and (flc_ForwardDistance is near) and (flc_RightDistance is medium) then (L_motor is reverse)(R_motor is slow) (1)
17. If (flc_LeftDistance is medium) and (flc_ForwardDistance is near) and (flc_RightDistance is near) then (L_motor is reverse)(R_motor is slow) (1)
18. If (flc_LeftDistance is near) and (flc_ForwardDistance is far) and (flc_RightDistance is far) then (L_motor is fast)(R_motor is medium) (1)
19. If (flc_LeftDistance is near) and (flc_ForwardDistance is far) and (flc_RightDistance is medium) then (L_motor is medium)(R_motor is medium) (1)
20. If (flc_LeftDistance is near) and (flc_ForwardDistance is medium) and (flc_RightDistance is far) then (L_motor is fast)(R_motor is fast) (1)
21. If (flc_LeftDistance is near) and (flc_ForwardDistance is medium) and (flc_RightDistance is medium) then (L_motor is medium)(R_motor is medium) (1)
22. If (flc_LeftDistance is near) and (flc_ForwardDistance is medium) and (flc_RightDistance is near) then (L_motor is slow)(R_motor is fast) (1)
23. If (flc_LeftDistance is near) and (flc_ForwardDistance is near) and (flc_RightDistance is far) then (L_motor is slow)(R_motor is reverse) (1)
24. If (flc_LeftDistance is near) and (flc_ForwardDistance is near) and (flc_RightDistance is medium) then (L_motor is reverse)(R_motor is slow) (1)
25. If (flc_LeftDistance is near) and (flc_ForwardDistance is near) and (flc_RightDistance is near) then (L_motor is reverse)(R_motor is slow) (1)
26. If (flc_LeftDistance is near) and (flc_ForwardDistance is far) and (flc_RightDistance is far) then (L_motor is fast)(R_motor is medium) (1)
27. If (flc_LeftDistance is near) and (flc_ForwardDistance is far) and (flc_RightDistance is medium) then (L_motor is medium)(R_motor is medium) (1)

```

Рис. 12. Правила «ЕСЛИ – ТО»

6. Результаты и обсуждение

Одной из задач исследования было создание программного обеспечения для модели робота Pioneer 3-DX с использованием платформ V-REP и MATLAB. Разработанное приложение было смоделировано в среде V-REP, чтобы проанализировать поведение мобильного робота.

В ходе экспериментов начальная и конечная точки траектории были заданы заранее, а местонахождение препятствий оставалось для робота Pioneer 3-DX неизвестным. Расстояния до препятствий, вычисленные датчиками, служили входными данными для нейросети, а выходные данные представляли собой соответствующие значения скорости двигателей правого и левого колес мобильного робота. Нейросеть была разработана с использованием платформы MATLAB. Обнаружив, что мобильный робот вошел в опасную зону, нейросеть классифицирует препятствие и сообщает о возможном риске столкновения в том случае, если оно потенциально опасно. После этого начинает действовать нечеткая логика для определения правильной реакции мобильного робота, чтобы избежать столкновения. В первом эксперименте была выполнена оценка ситуации с двумя динамическими препятствиями. В частности (тест 1, рис. 13), была изучена ситуация, когда препятствие 1 (помечено желтым цветом) двигалось впереди мобильного робота с углом курса 0° . Когда препятствие оказалось в опасной зоне, скорости левого и правого колес мобильного робота (V_l и V_r) были скорректированы в соответствии с заданными нечеткими правилами. Одновременно препятствие 2 (помечено синим цветом) двигалось впереди робота, не создавая угрозы столкновения.

Согласно сценарию теста 2 оба препятствия следовали по траекториям, параллельным роботу. Соответственно, в этой ситуации зона была классифицирована как неопасная.

По сценарию теста 3 (рис. 14) препятствие 1 (помечено желтым цветом) было выявлено как опасное, а другое – нет. В этом случае мобильный робот изменял направление движения так, чтобы обойти опасное препятствие, а затем возвращался на свою первоначальную траекторию.

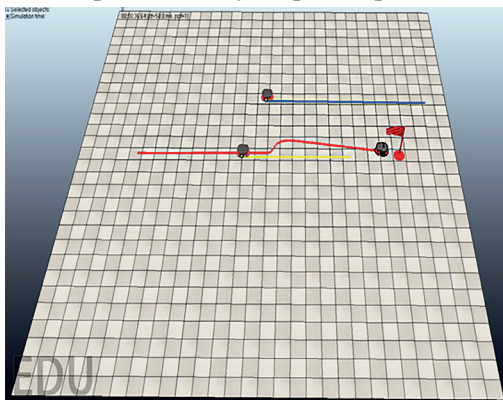


Рис. 13. Тест 1

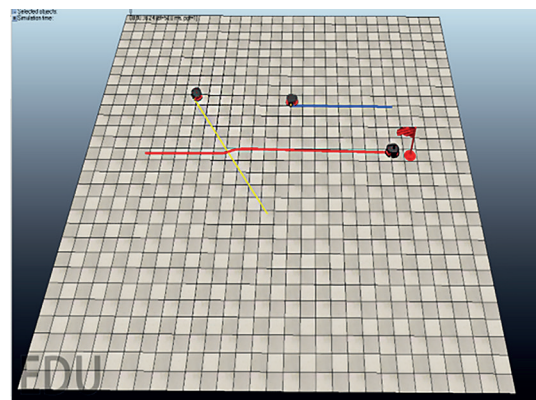


Рис. 14. Тест 3

В ходе теста 4 была рассмотрена ситуация, когда оба препятствия представляли опасность и, следовательно, требовалась корректировка скорости колес мобильного робота на основе принципов нечеткой логики. Мобильный робот успешно обходил первое препятствие, возвращался на первоначальную траекторию, а затем обходил второе препятствие и снова возвращался на свою траекторию.

В табл. 4 приведены полные сведения, включая начальную длину пути, вычисленную алгоритмом ИПП, измененную траекторию после обхода препятствия, а также скорости препятствий и мобильного робота, в том числе относительные.

Т а б л и ц а 4

Результаты тестов для эксперимента 1

	Тест 1	Тест 2	Тест 3	Тест 4
Начальная длина пути, рассчитанная с помощью алгоритма ИПП (м)	10,24	10,24	10,24	10,24
Время, требуемое для прохождения первоначальной траектории (с)	40,89	37,49	41,89	41,45
Длина по прямой от начальной до конечной точки (м)	10,24	10,24	10,24	10,24
Длина пути после уклонения от столкновений (м)	10,46	10,24	10,61	10,62
Скорость движения препятствия 1 (м/с)	0,15	0,22	0,16	0,21
Скорость движения препятствия 2 (м/с)	0,18	0,11	0,11	0,11
Скорость робота (м/с)	0,29	0,37	0,27	0,31
Относительная скорость препятствия 1 (м/с)	0,45	0,60	0,39	0,62
Относительная скорость препятствия 2 (м/с)	0,46	0,38	0,28	0,39
Увеличение длины пути (%)	2,2 %	0,2%	3,61%	3,75%

Во втором эксперименте в окружающее пространство было введено дополнительное третье препятствие, что увеличило потенциальный риск столкновения с роботом. В частности, в ходе теста 5 (рис. 15) исследовалась ситуация, когда во время маневрирования мобильного робота при обходе препятствия 2 (помечено желтым цветом) в критическую зону попало препятствие 3 (помечено синим цветом), и робот должен был изменить свою траекторию, чтобы избежать столкновения. Робот успешно обошел препятствие 2, одновременно уклонившись от препятствия 3.

В следующем сценарии (тест 6, рис.16) наряду с динамическими препятствиями появились статические, то есть условия движения усложнились. Мобильный робот должен был пройти от начальной точки до конечной при наличии статических и динамических препятствий, последние не были для него опасными, и он мог продолжать движение по своей исходной траектории без отклонений. В тесте 7 два препятствия (2 и 3) были определены как опасные, но робот эффективно их обходил, при этом препятствие 1 угрозы не представляло. Наконец, в тесте 8 был смоделирован сценарий, когда все три динамических препятствия создавали угрозу столкновения.

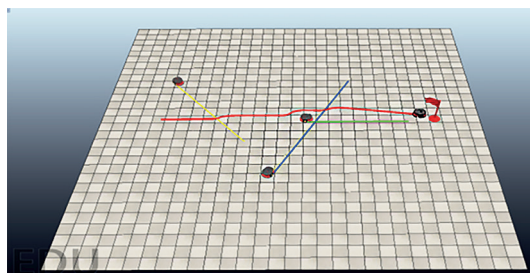


Рис. 15. Тест 5

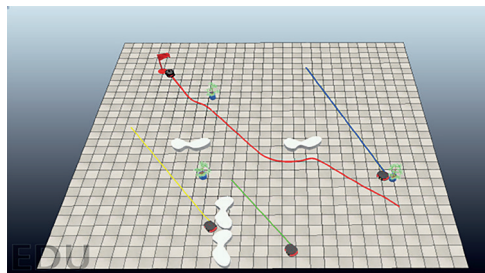


Рис. 16. Тест 6

Как показано в табл. 5, мобильный робот успешно обходил каждое из этих опасных препятствий, демонстрируя высокое качество навигации в сложных условиях.

Таблица 5

Результаты тестов для сценария 2

	Тест 5	Тест 6	Тест 7	Тест 8
Начальная длина пути, рассчитанная с помощью алгоритма ИПП (м)	10,76	14,09	13,67	13,67
Время, требуемое для прохождения первоначальной траектории (с)	40,84	42,09	46,04	48,94
Длина по прямой от начальной до конечной точки (м)	10,76	14,09	13,67	13,67
Длина пути после уклонения от столкновений (м)	11,23	14,39	14,62	14,70
Скорость движения препятствия 1 (м/с)	0,11	0,18	0,11	0,11
Скорость движения препятствия 2 (м/с)	0,21	0,21	0,11	0,11
Скорость движения препятствия 3 (м/с)	0,14	0,11	0,11	0,11
Скорость робота (м/с)	0,33	0,37	0,33	0,31
Относительная скорость препятствия 1(м/с)	0,46	0,54	0,35	0,35
Относительная скорость препятствия 2 (м/с)	0,51	0,59	0,24	0,31
Относительная скорость препятствия 3 (м/с)	0,48	0,48	0,27	0,35
Увеличение длины пути (%)	4,43%	2,10%	6,94%	7,54%

7. Сравнение с другими исследованиями

Проведен сравнительный анализ предложенного метода комбинирования нечеткой логики и нейросети (Fuzzy Logic-NN) с широко известными методами навигации мобильных роботов, включая нечеткий генетический алгоритм (Fuzzy-GA) и оптимизацию по алгоритму муравьиной колонии на основе нечеткой логики (FLACO) [14, 15]. Моделирование выполнялось по двум разным сценариям, в которых динамические препятствия имели ограниченную скорость. Результаты показали, что алгоритм, сочетающий нечеткую логику с использованием нейросети, позволял последовательно вычислять оптимальные траектории с минимальным временем обхода препятствий по сравнению с другими моделями. В частности, траектории, рассчитанные с помощью этого алгоритма, поддавались точной корректировке и обеспечивали эффективный обход различных препятствий на пути. Кроме того, в предложенной модели количественные показатели оптимизации длины траектории оказались лучше, чем у методов Fuzzy-GA и FLACO в рассмотренных сценариях (табл. 6). Полученные результаты позволяют сделать вывод об эффективности применения комбинации нечеткой логики и нейросети в реальном времени и непростой обстановке. Кроме того, алгоритм может служить надежным инструментом при решении задач планирования траектории в сложной динамической среде. Различные окружающие условия, в которых проводились эксперименты, представлены на рис. 17 [36], а на рис. 18 показана траектория, рассчитанная с помощью предложенного алгоритма. На рис. 17 начальная точка обозначена красным, конечная – зеленым, оранжевыми фигурами – подвижные препятствия, черными – неподвижные.

Таблица 6

Сравнительный анализ алгоритма, построенного на сочетании нечеткой логики и нейросети, и других известных подходов

Сценарий	Подход	Время (с)	Длина пути (см)	Увеличение длины пути (%)
1	Комбинация нечеткой логики и нейросети	22,69	104,54	1,08%
	Нечеткий генетический алгоритм	41,79	141,24	11%
	FLACO	31,47	129,64	3%
2	Комбинация нечеткой логики и нейросети	30,04	132,15	1,89%
	Нечеткий генетический алгоритм	81,31	219,96	10%
	FLACO	77,12	201,43	2%

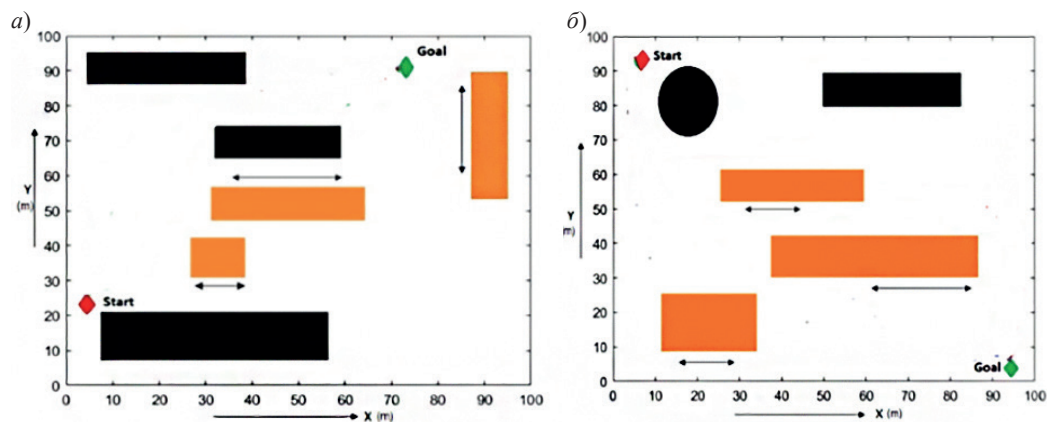


Рис. 17. Сценарии 1 (а) и 2 (б)

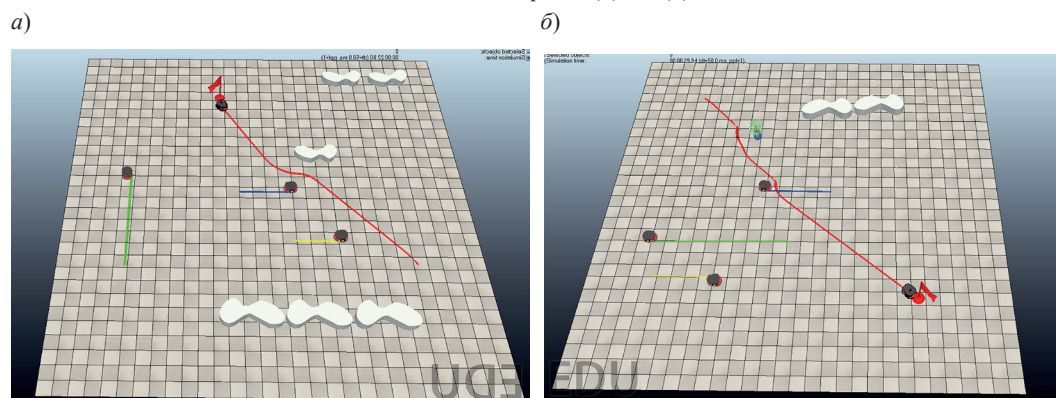


Рис. 18. Применение предложенного алгоритма в сценариях 1 (а) и 2 (б)

8. Заключение

В исследовании представлен комплексный подход к автономной навигации колесного робота Pioneer 3-DX в условиях статических и динамических препятствий.

Сочетая алгоритм искусственного потенциального поля, нейросеть для классификации зон опасности и ультразвуковые датчики, робот выполнял динамический расчет оптимальных маршрутов и определял различные уровни риска столкновения с препятствиями. Встроенный алгоритм нечеткой логики позволил эффективно маневрировать, обходя препятствия.

Результаты моделирования в средах MATLAB и V-REP подтвердили эффективность предлагаемого подхода. Сопоставительный анализ продемонстрировал отказоустойчивость и адаптивность системы. Несмотря на то что изначально метод разрабатывался для платформы Pioneer, его можно адаптировать и для других роботизированных систем, если они соответствуют требуемым условиям, то есть способны выполнять точные измерения, достоверно обнаруживать препятствия и имеют достаточные вычислительные ресурсы.

При этом может потребоваться корректировка алгоритма с учетом отличий конструкций роботов. Эффективность системы зависит от таких обязательных условий, как точность измерений и отсутствие пробуксовки колес. Нарушение этих условий может отрицательно повлиять на безопасность и успешный обход препятствий. Дальнейшие исследования должны быть направлены на разработку адаптивных алгоритмов, способных компенсировать неточности датчиков и пробуксовку колес в реальном времени и тем самым повысить надежность и устойчивость системы. Кроме того, хотя настоящее исследование относится к навигации в двухмерном пространстве (2D), описанные методики применимы и в трехмерном (3D). С этой целью алгоритмы необходимо адаптировать для расчета потенциальных полей и обработки дополнительных пространственных данных, тогда предложенный подход может быть задействован и в случае воздушных и подводных роботов, то есть получить расширенную функциональность.

С учетом ограничений, связанных с проверкой методики путем моделирования, дальнейшая работа будет включать испытания в реальных сложных условиях для подтверждения результатов моделирования. При этом будет выполнена оценка эксплуатационных показателей, а также сравнение реальных и расчетных результатов и итеративное уточнение алгоритмов на основе эмпирических данных. В целом сочетание алгоритмов ИПП, нейросети и нечеткой логики обеспечивает эффективную автономную навигацию в динамической обстановке. Разработанные методики являются универсальными и могут использоваться для различных робототехнических платформ, в том числе в задачах 3D-навигации, что свидетельствует об их значительном потенциале для широкого распространения.

ЛИТЕРАТУРА

1. **Faisal, M. et al.**, Fuzzy logic navigation and obstacle avoidance by a mobile robot in an unknown dynamic environment, *Int. J. Adv. Robot. Syst.*, 2013, vol. 10.
2. **Shitsukane, A. et al.**, Fuzzy logic sensor fusion for obstacle avoidance mobile robot, *IST-Africa Week Conf.*, 2018, pp. 1–8.
3. **Pimenta, L.C.A., et al.**, On computing complex navigation functions, *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, vol. 2005, pp. 3452–3457.
4. **Khatib, O.**, Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, *IJRR*, 1986, 5, 1, 90–98.
5. **Chengqing, L. et al.**, Virtual obstacle concept for local-minimum-recovery in potential-field based navigation, *Proc. IEEE Int. Conf. Robot. Autom.*, 2000, vol. 2, pp. 983–988.

6. Ge, S.S., Cui, Y.J., Dynamic motion planning for mobile robots using potential field method, *Auton. Robots.*, 2002, vol. 13, no. 3, pp. 207–222.
7. Shanmuganathan, S., Artificial neural network modelling: An introduction, *Stud. Comput. Intell.*, 2016, vol. 628, pp. 1–14.
8. Li, H. et al., A neural network approach to indoor mobile robot localization, *Proc. 19th Distrib. Comput. Appl. Bus. Eng. Sci.*, 2020, pp. 66–69.
9. Rao, A.M. et al., Navigation of non-holonomic mobile robot using neuro-fuzzy logic with integrated safe boundary algorithm, *Int. J. Autom. Comput.*, 2017, vol. 14, no. 3, pp. 285–294.
10. Batti, H., Jabeur, C.B., Seddik, H., Mobile Robot Obstacle Avoidance in labyrinth Environment Using Fuzzy Logic Approach, *Int. Conf. Control. Autom. Diagnosis, Proc.*, 2019.
11. Chelghoum, A., Wang, Q., Wang, K., Design and simulation of autonomous mobile robots obstacle avoidance system, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 2017, vol. 10092, pp. 165–180.
12. Khudaverdiyeva, M., Modeling of Mobile Robot With Obstacle Avoidance Using Fuzzy Controller, *Adv. Inf. Syst.*, 2022, vol. 6, no. 2, pp. 21–25.
13. Tan, R.J.C., Dadios, E.P., Motion planning of mobile robot using Fuzzy-GA method along with Three Path concept in dynamic environment, *J. Intell. Fuzzy Syst.*, 2018, vol. 35, no. 2, pp. 1445–1457, 2018, 1–6.
14. Singh, S.J. et al., Motion planning of mobile robot using Fuzzy-GA method along with Three Path concept in dynamic environment, *J. Intell. Fuzzy Syst.*, 2018, vol. 35, no. 2, pp. 1445–1457.
15. Song, Q. et al., Dynamic path planning for unmanned vehicles based on fuzzy logic and improved ant colony optimization, *IEEE Access. IEEE*, 2020, vol. 8, pp. 62107–62115.
16. Park, K., Zhang, N., Behavior-based autonomous robot navigation on challenging terrain: A dual fuzzy logic approach, *Proc. IEEE Symp. Found. Comput. Intell.*, 2007, pp. 239–244.
17. Suzuki, T., Takahashi, M., Obstacle avoidance for autonomous mobile robots based on position prediction using fuzzy inference, *6th Int. Conf. Informatics Control. Autom. Robot. Proc.*, 2009, vol. 2 RA, pp. 299–304.
18. Antonelo, E.A. et al., Intelligent autonomous navigation for mobile robots: Spatial concept acquisition and object discrimination, *Proc. IEEE Int. Symp. Comput. Intell. Robot. Autom.*, 2005, pp. 553–557.
19. Melingui, A. et al., Adaptive navigation of an omni-drive autonomous mobile robot in unstructured dynamic environments, *IEEE Int. Conf. Robot. Biomimetics*, 2013, pp. 1924–1929.
20. Farag, K.K.A., Shehata, H.H., El-Batsh, H.M., Mobile robot obstacle avoidance based on neural network with a standardization technique, *J. Robot.*, 2021, vol. 2021.
21. Alves, R.M.F., Lopes, C.R., Obstacle avoidance for mobile robots: A Hybrid Intelligent System based on Fuzzy Logic and Artificial Neural Network, *IEEE Int. Conf. Fuzzy Syst.*, 2016, pp. 1038–1043.
22. Wang, C. et al., A real-time obstacle avoidance strategy for safe autonomous navigation of intelligent hospital beds in dynamic uncertain environments, *Australas. Conf. Robot. Autom.*, 2013, pp. 4382–4387.
23. Liu, H., *Robot Systems for Rail Transit Applications*. Chapter 1 – Introduction, Ed. Elsevier, 2020, pp. 1–36, doi: 10.1016/B978-0-12-822968-2.00001-2.
24. Arambula Cosío, F., Padilla Castañeda, M.A. Autonomous robot navigation using adaptive potential fields, *Math. Comput. Model.*, 2004, vol. 40, № 9–10, pp. 1141–1156.
25. Liu, Y. et al., Path Planning for Mobile Robot Based on Improved Artificial Potential Field Method, *Proc. China Autom. Congr.*, 2023, pp. 4757–4762.
26. Li, G. et al., An efficient improved artificial potential field based regression search method for robot path planning, *IEEE Int. Conf. Mechatronics Autom.*, 2012, pp. 1227–1232.
27. Handayani, A.S. et al., Application of type-1 and type-2 fuzzy logic controller for the real swarm robot, *Int. J. online Biomed. Eng.*, 2019, vol. 15, no. 6, pp. 83–98.
28. Nurmaini, S., Chusniah, C., Differential drive mobile robot control using variable fuzzy universe of discourse, *Proceeding Int. Conf. Electr. Eng. Comput. Sci. Sustain. Cult. Herit. Towar. Smart Environ. Better Futur*, 2017, pp. 50–55.
29. Tursynbek, I., Shintemirov, A., Modeling and Simulation of Spherical Parallel Manipulators in CoppeliaSim (V-REP) Robot Simulator Software, *Int. Conf. Nonlinearity, Inf. Robot.*, 2020.
30. Rosenblatt, F., The perceptron: a probabilistic model for information storage and organization in the brain, *Psychology Review*, 1958, vol. 65, no. 6, pp. 386–408.
31. Spooner, J.T., Maggiore, M., Ordóñez, R., Passino, K.M., *Stable Adaptive Control and Estimation for Nonlinear Systems Neural and Fuzzy Approximator Techniques*, New York. 2002.
32. Wu, D. et al., Application of Bayesian regularization back propagation neural network in sensorless measurement of pump operational state, *Energy Reports*, 2022, vol. 8, pp. 3041–3050.

33. Singh, N.H., Thongam, K., Neural network-based approaches for mobile robot navigation in static and moving obstacles environments, *Intell. Serv. Robot.*, 2019, vol. 12, no. 1, pp. 55–67.
34. Uzair, M., Jamil, N., Effects of Hidden Layers on the Efficiency of Neural networks, *Proc. 23rd IEEE Int. Multi-Topic Conf.*, 2020, pp. 1–6.
35. Yosif, Z.M., Mahmood, B.S., Saeed, S.Z., Artificial Techniques Based on Neural Network and Fuzzy Logic Combination Approach for Avoiding Dynamic Obstacles, *J. Eur. des Syst. Autom.*, 2022, vol. 55, no. 3, pp. 339–348.
36. Sangeetha, V. et al., A fuzzy gain-based dynamic ant colony optimization for path planning in dynamic environments, *Symmetry* (Basel), MDPI AG, 2021, vol. 13, no. 2, pp. 1–22.

Alsayed, N. and Krasnov, A.Yu. (ITMO University, Saint-Petersburg)

Advanced Control Algorithms for Dynamic Environment Navigation and Obstacle Avoidance, *Giroskopiya i Navigatsiya*, 2024, vol. 32, no. 3, pp. 105–125.

Abstract. This study proposes an autonomous navigation approach for the Pioneer P3-DX Autonomous Wheeled Robot (AWR) in environments containing both static and dynamic obstacles. The robot utilizes the Artificial Potential Field (APF) algorithm for path calculation, while a neural network aids in zone classification. Three ultrasonic sensors provide distance measurements for hazard assessment. These measurements, along with relative velocity and angle data, aid in identifying regions of elevated risk (Zone 1) and those of lesser risk (Zone 2). Upon hazard detection, fuzzy logic facilitates effective collision avoidance by adjusting wheel velocities. Simulation results conducted in MATLAB and V-REP demonstrate the approach's efficacy in navigating diverse obstacles, showcasing its adaptability and resilience compared to alternative algorithms. This research introduces an innovative methodology for autonomous mobile robot navigation, emphasizing its reliability and efficiency in traversing intricate environments with varying risk levels.

Key words: path planning, mobile robot navigation, neural network, fuzzy logic, obstacle avoidance.

Материал поступил 13.05.2024