

УПРАВЛЕНИЕ АГЕНТОМ В ДИНАМИЧЕСКОЙ СРЕДЕ НА ОСНОВЕ СИСТЕМЫ НЕЧЁТКОГО ВЫВОДА

Шульпин С. М.¹

(Нижегородский государственный университет
им. Н.И. Лобачевского, Нижний Новгород)

Рассматривается задача управления мобильным агентом в динамическом неизвестном окружении. Для решения этой задачи предлагается использовать регулятор на основе системы нечёткого вывода. Для настройки параметров регулятора используется генетический алгоритм. Также в работе предложена модификация генетического алгоритма, позволяющая настраивать не только параметры функций принадлежности лингвистических переменных, но и их тип. Это позволяет провести более тонкую настройку регулятора и, как следствие, улучшить качество системы. Эффективность предложенного подхода была подтверждена при помощи компьютерной симуляции в разработанной виртуальной среде. Полученный регулятор был использован в физическом эксперименте с мобильным агентом, реализованным на основе Arduino Uno, в среде с динамическими и статическими препятствиями.

Ключевые слова: планирование путей, генетический алгоритм, нечёткая логика.

1. Введение

Развитие беспилотного транспорта и расширение сфер применения систем автономных роботов приводит к появлению множества новых задач, которые требуют исследования. Одной из наиболее важных является построение эффективных систем навигации автономных роботов в динамическом или неизвестном окружении. Цели навигации могут быть различными, и в её процессе может быть решен целый ряд проблем, таких как позиционирование в пространстве, построение карты неизвестного окружения, построение безопасных и эффективных траекторий движения, исследование местности, проведение поисково-спасательных работ и другие.

Система навигации может состоять из двух подсистем, описанных в [8]: задача первой — глобальное планирование путей

¹ Стенан Михайлович Шульпин, аспирант (shulpinstm@gmail.com).

и соответствие поставленным целям, второй — локальное планирование, взаимодействие автономного робота с окружением и обработка данных с датчиков. При этом выделяется два подхода для решения задачи планирования путей: статическое и динамическое. Статическое планирование чаще применяется в задачах с известной конфигурацией пространства, например, глобальное планирование на карте. При этом статическое планирование может быть применено и в условиях неизвестного окружения. Но такой подход требует хранения и обновления внутреннего представления навигационной карты окружения, а также такие алгоритмы имеют высокую временную и вычислительную емкость, что является критичным в условиях динамического окружения [15, 21]. Алгоритмы динамического планирования способны отыскать путь даже при наличии неопределенности или в окружении с динамическими объектами с неизвестными законами движения [4, 9, 22].

Решение задач, связанных с управлением мобильными агентами, функционирующими в динамической среде, все чаще основывается на применении технологий искусственного интеллекта. Предлагается использовать компьютерное зрение, распознавание препятствий, нейронные сети. В статье [7] предложен подход на основе рекуррентной нейронной сети, осуществляющий управление мобильным агентом. В [18] предложен алгоритм реального времени на основе глубоких нейронных сетей. Такие методы позволяют добиться работы системы в режиме реального времени, так как не требуют перестройки всего маршрута при изменении окружения. Эти подходы обладают хорошей способностью к обобщению для новых окружений и препятствий. Но при этом необходимо подготавливать данные вручную или разрабатывать дополнительные алгоритмы для получения данных, предназначенных для обучения нейронных сетей. Также выбор архитектуры нейронной сети требует аккуратности и часто осуществляется эмпирически. Большое число нейронов скрытого слоя увеличивает как время обучения сетей, так и сложность применения таких подходов [4, 7, 18]. Для решения задачи планирования маршру-

та в неизвестном окружении предлагается использовать и методы обучения с подкреплением. Эксперименты показывают, что агенты способны успешно достичь локальной целевой позиции в неизвестной динамической среде. Для таких методов сложностью является уменьшение размера таблицы действий и выбор функции для подсчета их полезности [9].

В данной работе предлагается использовать регулятор на основе системы нечёткого вывода. Алгоритмы, работающие с использованием нечеткой логики, не требуют построения всего маршрута и порождают регуляторы в виде обратной связи по состоянию. Это позволяет использовать такой метод даже при условии отсутствия первоначальной информации об окружении. Эффективность использования подхода на основе нечеткой логики зависит от решения задачи определения оптимальных функций принадлежности. Для поиска параметров регуляторов часто применяются методы машинного обучения. С учетом особенностей планирования маршрутов в данной задаче логично использовать генетический алгоритм для поиска конфигураций, показывающих наилучшие результаты, что было предложено в работах [5, 11]. Использование такого подхода не требует сбора обучающих данных.

Дизайн системы нечеткого вывода основывается на экспертных знаниях в прикладной области и, в связи с этим, системы, основанные на нечеткой логике, имеют высокую репрезентативность и более простую архитектуру по сравнению с нейронными сетями. В то же время такие методы не теряют универсальности и гибкости. Для более тонкой регулировки системы нечеткого вывода в настоящей работе предлагается модификация генетического алгоритма, позволяющая настраивать как параметры функций принадлежности лингвистических переменных, так и их тип.

2. Модель динамики агента

В качестве мобильного агента могут быть использованы различные виды роботов с разным набором сенсоров и характером движения. В данной работе используется классический мобиль-

ный агент, представляющий собой платформу с двумя ведущими колесами, вращение которых обеспечивается двумя электродвигателями постоянного тока. Схематичное изображение агента представлено на рис. 1, здесь AB — стержень, соединяющий колеса, v_1 и v_2 — линейные скорости. При рассмотрении задачи с движением без проскальзывания математическая модель такого агента может быть описана следующей системой уравнений:

$$(1) \quad \begin{cases} x_{k+1} = x_k - \tau \frac{R}{2} (\omega_{r,k} + \omega_{l,k}) \sin \varphi_k, \\ y_{k+1} = y_k + \tau \frac{R}{2} (\omega_{r,k} + \omega_{l,k}) \cos \varphi_k, \\ \varphi_{k+1} = \varphi_k + \tau \operatorname{arctg} \frac{R(\omega_{r,k} - \omega_{l,k})}{2l}, \end{cases}$$

здесь x_k и y_k — координаты агента в k -й момент дискретного времени, φ_k — его ориентация в пространстве, τ — интервал дискретизации, R — радиус колеса, l — половина длины стержня, соединяющего колёса, $\omega_{r,k}$ и $\omega_{l,k}$ — угловые скорости правого и левого колёс.

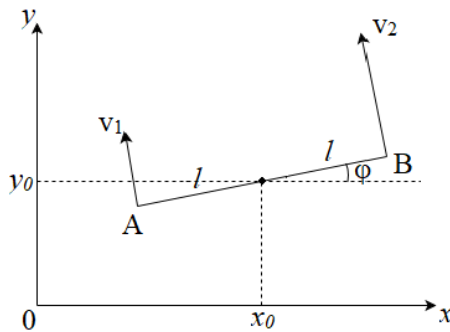


Рис. 1. Схема движения агента

Используемая модель оборудована пятью датчиками расстояния, равномерно распределенными в передней части платформы, так что третий датчик направлен вперед. Данные, поступающие с датчиков расстояния, являются основой для планирования движения в условиях динамического окружения.

Обозначим через d_k значение дистанции, возвращаемое k -м датчиком робота, $k = \overline{1, 5}$. Требуется синтезировать регулятор, основанный на системе нечеткого вывода, вида:

$$(2) \quad \begin{pmatrix} \omega_l \\ \omega_r \end{pmatrix} = \Theta(x, y, \varphi; d_1, d_2, d_3, d_4, d_5),$$

обеспечивающий движение агента в динамической среде с минимальным числом столкновений.

3. Регулятор на основе нечёткой логики

В данной работе предлагается использовать систему нечёткого вывода. Работа такой системы состоит из трёх этапов: фаззификации, нечёткого вывода и дефаззификации (см. рис. 2).

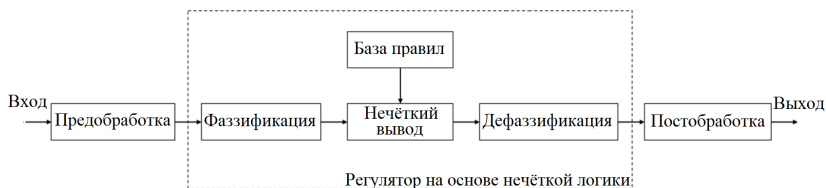


Рис. 2. Процесс работы регулятора на основе нечёткой логики

На этапе фаззификации происходит активация функций принадлежности и определение степени принадлежности всех входных числовых переменных. Результатом являются степени истинности всех элементарных логических высказываний, входящих в посылки базы продукционных правил. Степень принадлежности числовой переменной в нечёткой логике характеризуется набором функций принадлежности, или термов, определяющих соответствие числовых параметров заданным нечётким множествам. Формально нечёткое множество можно описать как множество пар $T = \{ \langle x, \mu_T(x) \rangle | x \in U \}$, где x — элемент универсума U , $\mu_T(x)$ — функция принадлежности нечёткого множества T . Несколько нечётких множеств связываются между собой

в одну лингвистическую переменную, в зависимости от значения которой активируется та или иная функция принадлежности. Формально лингвистическая переменная может быть выражена набором $\langle x, T, U, G, M \rangle$, где x — имя переменной, T — нечёткие множества, G — синтаксические правила, позволяющие порождать новые термы, M — семантические правила, задающие функции принадлежности для термов, образованных с помощью G .

Также нечёткое множество может быть расширено с помощью дополнительного модификатора, определяющего степень функции и изменяющего её определенным образом. Например:

$$(3) \quad \text{очень } \mu_T(x) = \mu_T^2(x).$$

Чаще всего используются типовые функции принадлежности, которые могут быть разбиты на две группы: кусочно-линейные и непрерывно-дифференцируемые. К кусочно-линейным относятся треугольная и трапецевидная функции принадлежности. К непрерывно-дифференцируемым относятся функция обобщенного колокола, сигмоидальная функция, функция принадлежности Гаусса, S-образная, Z-образная функции.

В качестве входных лингвистических переменных в данной работе используются значения дистанции d_k , которые принимают нечёткие значения «близко» (Б) и «далеко» (Д).

На этапе нечёткого логического вывода осуществляется переход от нечётких условий к нечетким заключениям о требуемом выходе системы, на основе нечёткой информации и заданной базы продукционных правил вида IF...AND...THEN.... Процесс нечёткого вывода состоит из агрегирования подусловий, активации и композиции подзаключений. Правила, используемые в работе, представлены в таблице 1.

Процедура дефаззификации заключается в приведении нечёткого множества выходной переменной к чёткому значению. В качестве выходных лингвистических переменных в данной работе используются значения угловых скоростей левого и правого колёс ω_l и ω_r . Выходные переменные принимают нечёткие значения «высокая отрицательная» (ВО), «умеренная отрицательная» (УО), «низкая отрицательная» (НО), «низкая положитель-

Таблица 1. База продукционных правил

| Входные переменные | | | | | Выходные переменные | |
|--------------------|-------|---------|-------|-------|---------------------|------------|
| d_1 | d_2 | d_3 | d_4 | d_5 | ω_l | ω_r |
| Б | | | | | УП | УО |
| | Б | | | | УП | УО |
| | | Б | Д | | УП | УО |
| | | очень Б | Д | | НП | НО |
| | Д | Д | Д | | очень ВП | очень ВП |
| | Б | Б | Б | | ВП | НО |
| | Д | очень Б | | | НО | НП |
| | Д | Б | | | УО | УП |
| | | | Б | | УО | УП |
| | | | | Б | УО | УП |

ная» (НП), «умеренная положительная» (УП) и «высокая положительная» (ВП).

Алгоритм вывода Мамдани, используемый в работе, описывает последовательное применение представленных процедур [13].

Поведение и свойства регулятора, основанного на нечёткой логике, зависят от выбора базы продукционных правил и функций принадлежности нечётких множеств (термов) лингвистических переменных. Для настройки параметров регулятора предлагается использовать модифицированный генетический алгоритм как эффективный метод обучения без учителя.

4. Адаптация параметров регулятора

Генетический алгоритм – это адаптивный метод поиска для решения задач оптимизации, основанный на механизме наследования и естественного отбора. В привычном понимании (см. рис. 3) генетический алгоритм состоит из следующих этапов [1]:

- 1) генерация начальной популяции хромосом;
- 2) вычисление приспособленности для каждой особи;
- 3) селекция;

- 4) рекомбинация;
- 5) мутация;
- 6) генерация нового поколения.

Этапы 2–6 выполняются пока не будет достигнуто заданное качество регулятора.



Рис. 3. Общая схема работы генетического алгоритма

В данной работе при помощи генетического алгоритма предлагается настраивать параметры функций принадлежности $\mu_{type}(x, \lambda)$, при этом предлагается модификация общепринятого подхода. Будем кодировать информации о форме и наборе параметров функции принадлежности при помощи «нечёткого гена» $G_{T_k} = \langle type_k, \lambda_{k1} \dots \lambda_{kn} \rangle$, где $type_k$ – переменная, отвечающая за форму функции принадлежности (треугольная, трапециевидная и т.д.), а λ – вектор параметров. В этом случае «нечёткая хромосома» представляет собой список «нечётких генов»:

$$Y = \{G_{T_k}, k = \overline{1, M}\}.$$

Использование «нечёткой хромосомы» позволяет более тонко настроить параметры и таким образом улучшить сходимость генетического алгоритма. Для этого были разработаны два вида оператора мутации: мутация типа функции принадлежности и мутация её параметров. Мутация типа функции принадлежности нечёткого терма позволяет менять вид функций внутри одного класса эквивалентности. Мутация такого рода является затухающей мутацией, т.е. вероятность её применения уменьшается со временем работы генетического алгоритма и достижением приемлемых значений функции приспособленности популяции. Оператор мутации параметров функции принадлежности осуществляет их тонкую настройку и работает на всем протяжении времени работы генетического алгоритма.

Для каждого типа функции принадлежности можно задать формулы перевода специфичных параметров типа в обобщенные и обратно. За счет этого происходит преобразование типа функции принадлежности, используемое в мутации.

Качество хромосомы определяется на основе её приспособленности, которая соответствует расстоянию, пройденному агентом в окружении с динамическими препятствиями до первого столкновения.

5. Обучение регулятора в виртуальной среде

Для проведения симуляции была разработана виртуальная тестовая среда, в которой случайным образом генерируется окружение со статическими и динамическими препятствиями. Тестовая среда позволяет настраивать количество статических и динамических препятствий, их размеры и параметры самого агента. Также возможно использование шаблонных участков пути для обучения маневрированию в специфичных условиях. Система симуляции движения была разработана на основе «движка» Unity (см. рис. 4). Под случайностью понимается положение, размеры и количество статических препятствий. При генерации динамических препятствий случайным образом выбираются координаты

начала и конца отрезка, вдоль которого они осуществляют перемещение, а также значение постоянной скорости их движения, которое сравнимо со скоростью самого агента.

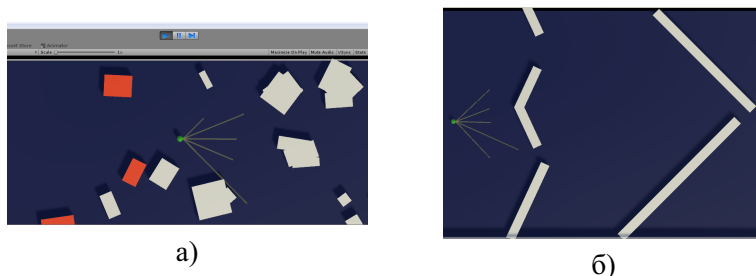


Рис. 4. Изображение конфигурации участка пути: а) случайная, б) шаблонная

Для возможности обучения была создана сцена, позволяющая запускать одновременную симуляцию движения нескольких агентов. Цель агента, управляемого регулятором, – достичь бесконечно удалённой цели, избежав столкновений с окружающими объектами. Каждую итерацию происходит последовательно генерация новых особей, их параллельный старт, определение лучших особей на основе пройденного расстояния до первого столкновения, применение операторов генетического алгоритма. Данная сцена позволяет настраивать количество агентов, число итераций и другие параметры обучения.

Изначально термы задаются функцией принадлежности случайной формы, которая в дальнейшем с помощью операторов мутации может эволюционировать в более подходящее представление. Область значений переменной «расстояние» была выбрана от 0 до 15, переменной «скорость» – от -10 до 10 . При обучении проводилось два вида экспериментов:

1. Нечеткие значения ВО, УО, НО, ВП, УП, НП лингвистической переменной «скорость» задавались независимыми функциями принадлежности.

2. Значения ВП, УП, НП вычислялись на основе ВО, УО, НО, соответственно, исходя из условия симметрии относительно

нуля.

Результаты экспериментов были схожи, поэтому был выбран второй подход.

Обучение параметров регулятора происходило в течение 500 эпох при одновременной симуляции движения 100 виртуальных агентов. Использовались следующие настройки генетического алгоритма: вероятность мутации — 0,3, для рекомбинации выбирались 100 пар родительских особей, являющихся победителями с точки зрения пройденного расстояния среди 50 выбранных случайно агентов, далее лучшие 10% исходной популяции переносились в новую без фильтрации, а среди остальных участников предыдущей популяции и среди особей, полученных в результате рекомбинации, проводился турнирный отбор до тех пор, пока размер нового поколения не равнялся 100.

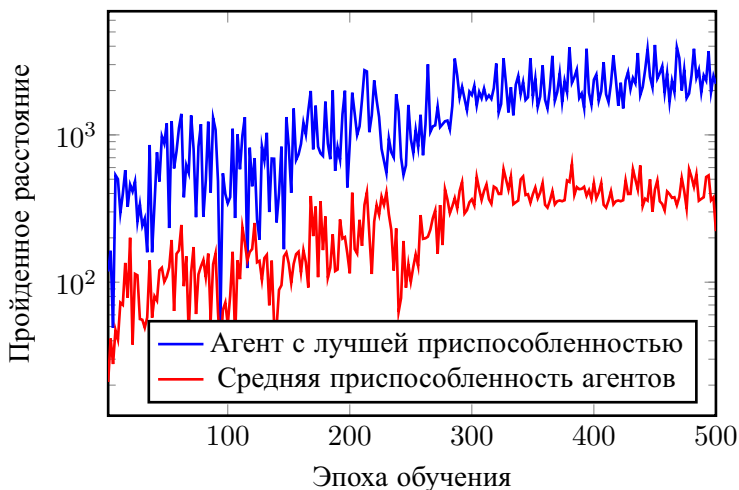
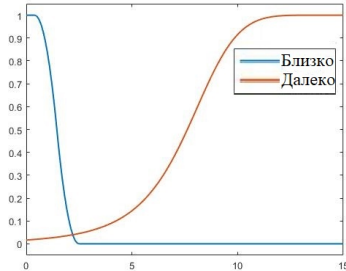


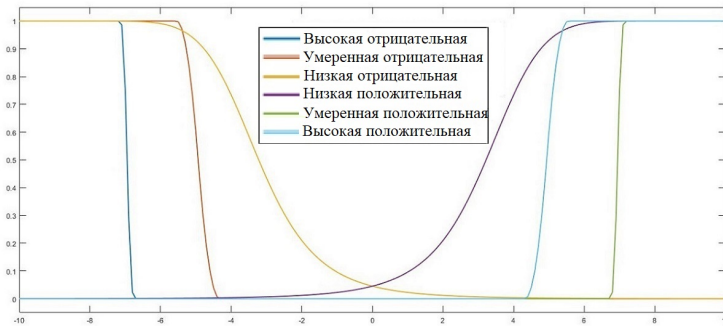
Рис. 5. Зависимость приспособленности особей от числа проведенных эпох обучения

Работа генетического алгоритма представлена на рис. 5. Как видно из графика, особи с высокой степенью приспособленности начали появляться после 200 эпох обучения, но с 350 по 500 эпоху обучения не было явного улучшения приспособленности, что 226

свидетельствует о том, что алгоритм обучения завершил настройку параметров. Также на графике видно, что разброс значения приспособленности между разными эпохами достаточно велик, это связано с высокой степенью случайности итогового статического и динамического окружения.



а)



б)

Рис. 6. Полученные функции принадлежности для лингвистических переменных: а) расстояние, б) угловая скорость вращения

На рис. 6 приведены полученные лингвистические переменные с их терминами.

Для сравнения была также проведена адаптация параметров

регулятора, предложенного в [5], с теми же настройками генетического алгоритма. В отличие от используемого в данной статье подхода отсутствовала мутация типа функций принадлежности и использовалась только треугольная форма. На рис. 7 можно заметить, что значение целевой функции после обучения для такого регулятора получилось меньше. Таким образом, применение мутации типа функции принадлежности оправдано.

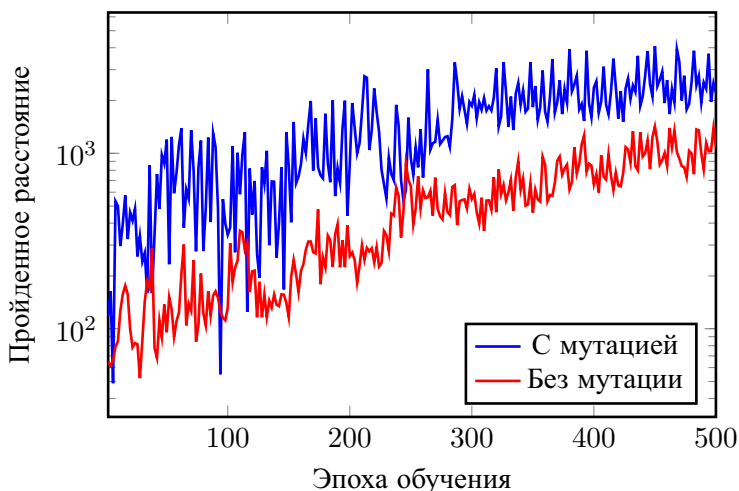


Рис. 7. Зависимость лучшей приспособленности особей от числа проведенных эпох обучения для обучения с и без мутации типа функции принадлежности

6. Физический эксперимент

Полученный после обучения в виртуальной среде регулятор был использован при проведении эксперимента в реальном окружении с физической моделью.

Физическая модель была реализована на основе микроконтроллера Arduino Uno, который был дополнен платформой, двумя двигателями постоянного тока, пятью ультразвуковыми датчиками и двумя модулями для обмена данными XBee Series 2.

Управление вращением моторов происходит с помощью ШИМ сигнала. Такой вид управления полностью соотносится с виртуальной моделью и полученным регулятором. Расстояния до препятствий измеряются с помощью ультразвуковых датчиков. Чтобы иметь актуальную информацию о расстояниях до препятствий в любой момент, опрос датчиков происходит постоянно. Чтобы не нарушать работу основной программы, измерение времени работы ультразвукового датчика осуществляется с помощью механизма прерываний.

Связь агента с компьютером осуществляется через последовательный интерфейс обмена данных (UART). Эта коммуникация происходит через два модуля XBee S2, один из которых настроен как координатор, а второй – как конечное устройство. В ходе экспериментов было обнаружено, что полный цикл обмена данных осуществляется за 300 мс, поэтому модули XBee, Arduino и серверное приложение были перенастроены так, что время обмена данными было уменьшено до 45 мс.

Для упрощения работы была принята следующая цель навигации: двигаться прямо, избегая при этом столкновений с препятствиями в неизвестном окружении. Данная постановка задачи соответствует описанному в [8] делению системы управления на две части. Описанная выше система отвечает за локальное планирование маршрута, тогда как глобальное может быть реализовано посредством алгоритмов, работающих с известным статическим окружением. Например, может быть использован алгоритм на основе вероятностных дорожных карт RRT* (Rapidly-exploring Random Trees), описанный в [14].

В физическом эксперименте были использованы представленные на рис. 6 функции принадлежности для определения лингвистических переменных. В регуляторе для этой модели была сделана поправка на масштаб системы.

За основу проведения физического эксперимента было взято исследование из статьи [5]. Была проведена серия экспериментов с разными конфигурациями расположения статических препятствий, неизвестных заранее агенту, на площадке размером

153 см × 287 см при размерах мобильного агента 16 см × 23 см. На рис. 8 схематично представлена одна из конфигураций.

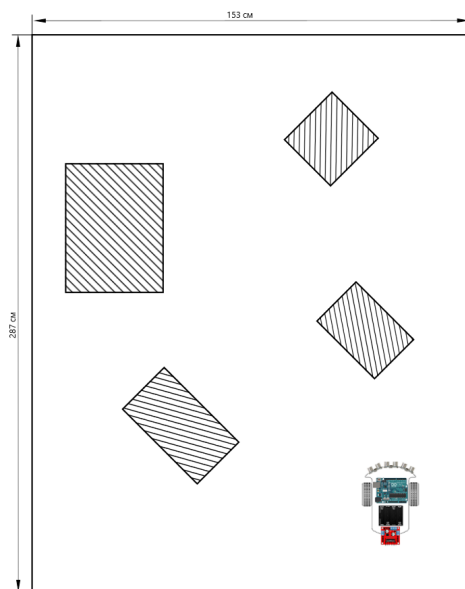


Рис. 8. Схематическая конфигурация окружения для проведения экспериментов

В таблице 2 приведены результаты экспериментов по запуску агента в неизвестном статическом окружении с различной средней скоростью движения и плотностью расположения препятствий. Испытания проводились в течение 5 минут, для каждой совокупности параметров вычислялось среднее количество столкновений среди десяти запусков.

При добавлении в окружение динамического препятствия число столкновений не меняется при условии, что скорость движения препятствия ниже скорости движения мобильного агента. При увеличении скорости движения динамического препятствия на первый план выходит запаздывание, существующее в системе. Таким образом, скорости передачи сигнала становится недостаточно для принятия своевременного решения.

Таблица 2. Зависимость среднего числа столкновений при десяти запусках от скорости движения и расстояния между препятствиями

| Средняя скорость движения, см/сек | Среднее расстояние между препятствиями, см | Усредненное количество столкновений при десяти запусках |
|-----------------------------------|--|---|
| 27 | 40 | 0,3 |
| 27 | 30 | 1,1 |
| 35 | 40 | 0,4 |
| 35 | 30 | 1,5 |
| 43 | 40 | 0,4 |
| 43 | 30 | 4,8 |

Результаты, полученные в физической симуляции, сравнимы с результатами в статье [5]. При этом скорость движения мобильного робота, а следовательно, требования к скорости реакции системы, были выше, что говорит о лучшем качестве регулятора.

7. Заключение

Предложенный в работе подход к синтезу регуляторов позволяет строить законы управления при недостаточных экспертных знаниях в области его применения. Подход обеспечивает сходимость параметров регулятора к оптимальным.

Для достижения целей работы был синтезирован регулятор, основанный на системе нечеткого вывода и позволяющий управлять мобильным агентом в динамически изменяющемся окружении. С помощью компьютерной симуляции был настроен регулятор, а именно вид и параметры функций принадлежности, характеризующие лингвистические переменные, входящие в базу продукционных правил. Результаты обучения системы были продемонстрированы на физическом эксперименте на сконструиро-

ванной физической модели с учетом специфики её управления. Эти результаты показывают работу алгоритма в условиях системы реального времени.

В ходе обучения было получено множество «достаточно хороших» регуляторов с точки зрения выбранной целевой функции. Также было замечено, что с достижением определенного порога качественного улучшения характеристик движения агента не происходило. Это может быть связано со случайным характером выбранной системы моделирования препятствий. Для возможного улучшения системы в дальнейшем стоит рассмотреть более точные математические системы для моделирования движения агента. А также более сложный вид функций приспособленности, позволяющий учитывать не только пройденное расстояние, но и качество построенных траекторий, например, время нахождения в окрестности препятствия. Это позволит решить проблемы, связанные с приспособляемостью агентов к определенным видам препятствий.

Результаты данной работы и программный комплекс, представленный в ней, могут быть использованы для исследования возможности применения аналогичного подхода для решения задач управления. Полученный механизм после небольших преобразований, связанных с описанием предметной области, может быть применен к многим другим задачам управления.

Литература

1. ПАНЧЕНКО Т.В. *Генетические алгоритмы* / Под ред. Ю.Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. – 87 с.
2. ПЕГАТ А. *Нечеткое моделирование и управление* / Пер.с англ. – М.: БИНОМ. Лаборатория знаний, 2009. – 798 с.
3. CHEN L., MANTEGH I., HE T., XIE W. *Fuzzy Kinodynamic RRT: a Dynamic Path Planning and Obstacle Avoidance Method* // Int. Conf. on Unmanned Aircraft Systems. – 2020. – P. 188–195.

4. CHEN Y., LIANG J., WANG Y. *Autonomous mobile robot path planning in unknown dynamic environments using neural dynamics* // Soft Computing. – 2020 – Vol. 24. – P. 13979–13995.
5. HASSANZADEH I., SADIGH S.M. *Path planning for a mobile robot using fuzzy logic controller tuned by GA* // 6th Int. Symposium on Mechatronics and its Applications. – 2009. – P. 1–5.
6. JANG J.-S.R., SUN C. *Neuro-Fuzzy Modeling and Control* // Proc. of the IEEE. – 1995. – No. 83. – P. 378–406.
7. KO B., CHOI H., HONG C., KIM J., KWON O.C., YOO C. *Neural network-based autonomous navigation for a homecare mobile robot* // IEEE Int. Conf. on Big Data and Smart Computing. – 2017. – P. 403–406.
8. LATOMBE J.C. *Robot Motion Planning* // The Kluwer International Series in Engineering and Computer Science. – 1990.
9. LEI X., ZHANG Z., DONG P. *Dynamic Path Planning of Unknown Environment Based on Deep Reinforcement Learning* // Journal of Robotics. – 2018 – Vol. 24. – P. 1–10.
10. LI J., WANG J., PENG H., ZHANG L., HU Y., SU H. *Neural fuzzy approximation enhanced autonomous tracking control of the wheel-legged robot under uncertain physical interaction* // Neurocomputing. – 2020. – Vol. 410. – P. 342–353.
11. LIU Q., LU Y.-G., XIE C.-X. *Optimal genetic fuzzy obstacle avoidance controller of autonomous mobile robot based on ultrasonic sensors* // IEEE Int. Conf. on Robotics and Biomimetics. – 2006. – P. 125–129.
12. LUVIANO-CRUZ D., YU W. *Path planning of multi-agent systems in unknown environment with neural kernel smoothing and reinforcement learning* // Neurocomputing. – 2017. – Vol. 233. – P. 34–42.
13. MAMDANI E. *Applications of fuzzy algorithms for control of a simple dynamic plant* // Proc. of the IEEE. – 1974. – No. 31. – P. 1585–1588.

14. MOON C., CHUNG W. *Kinodynamic Planner Dual-Tree RRT (DT-RRT) for Two-Wheeled Mobile Robots Using the Rapidly Exploring Random Tree* // IEEE Trans. on Industrial Electronics. – 2015. – Vol. 62, No. 2. – P. 1080–1090.
15. NINOMIYA K. *Planning approaches to constraint-aware navigation in dynamic environments* // Computer Animation and Virtual Worlds. – 2015. – Vol. 2, No. 26. – P. 119–139.
16. OLEIWI B.K., MAHFUZ A., ROTH H. *Application of Fuzzy Logic for Collision Avoidance of Mobile Robots in Dynamic-Indoor Environments* // 2nd Int. Conf. on Robotics, Electrical and Signal Processing Techniques. – 2021. – P. 131–136.
17. PATIAL A., MANDALIA D., NANDOSKAR N., HALDANKAR G.T. ET AL. *FIS based autonomous navigation system* // 8th Int. Conf. on Computing, Communication and Networking Technologies. – 2017. – P. 1–5.
18. REN Z., LAI J., WU Z., XIE S. *Deep neural networks-based real-time optimal navigation for an automatic guided vehicle with static and dynamic obstacles* // Neurocomputing. – 2021. – Vol. 443. – P. 329–344.
19. SAHLOUL S., BENHALIMA D., REKIK C. *Comparative study of hybrid fuzzy logic methods for mobile robot navigation in unknown environments* // 19th Int. Conf. on Sciences and Techniques of Automatic Control and Computer Engineering. – 2019. – P. 170–175.
20. SANGEETHA V., RAVICHANDRAN K.S. *A Modified Fuzzy A* Based Inference System for Path Planning in an Unknown Environment* // 2nd Int. Conf. on Trends in Electronics and Informatics. – 2018. – P. 181–186.
21. VAN TOLL W.G., COOK A.F., GERAERTS R. *A navigation mesh for dynamic environments* // Computer Animation and Virtual Worlds. – 2012. – Vol. 6, No. 23. – P. 535–546.
22. VAN DEN BERG J., FERGUSON D., KUFFNER J. *Anytime path planning and replanning in dynamic environments* // Proc. IEEE Int. Conf. on Robotics and Automation. – 2006. – P. 2366–2371.

23. WANG M., LIU J. *Fuzzy logic based robot path planning in unknown environment* // Int. Conf. on Machine Learning and Cybernetics. – 2005. – Vol. 2. – P. 813–818.
24. XU F., WANG S.-C., YANG W. *Local logic optimization algorithm for autonomous mobile robot based on fuzzy logic* // Chinese Control And Decision Conference. – 2018. – P. 4160–4165.

AGENT CONTROL IN A DYNAMIC ENVIRONMENT BASED ON FUZZY INFERENCE SYSTEM

Stepan Shulpin, Lobachevsky State University of Nizhny
Novgorod, Nizhniy Novgorod, graduate student
(shulpinstm@gmail.com).

Abstract: The paper considers the task of control a mobile agent in a dynamic unknown environment. To solve this problem, it is proposed to use a neuro-fuzzy controller. To adjust the parameters of the controller, a genetic algorithm is used. The paper also proposes a modification of the genetic algorithm that allows to configure not only the parameters of the membership functions of linguistic variables, but also their type. This allows to fine-tune the controller and, as a result, improve the quality of the system. The effectiveness of the proposed approach was confirmed by computer simulation in the developed virtual environment. The resulting controller was used in a physical experiment with a mobile agent implemented on the basis of Arduino Uno, in an environment with dynamic and static obstacles.

Keywords: path planning, genetic algorithm, fuzzy logic.

УДК 681.1

ББК 32.816

DOI: 10.25728/ubs.2022.100.10

*Статья представлена к публикации
членом редакционной коллегии Г.А. Угольником.*

Поступила в редакцию 28.12.2021.

Дата опубликования 30.11.2022.