

Программная реализация задания поведения автономного мобильного робота с биоинспирированной управляющей системой

В. В. Иванова

СПбГЭТУ «ЛЭТИ», ЦНИИ РТК

overtpace@gmail.com

А. М. Корсаков

ЦНИИ РТК

a.korsakov@rtc.ru

Аннотация. В данной работе была рассмотрена программная реализация задания поведения автономного мобильного робота с биоинспирированной управляющей системой. Управляющая система представляла собой конечный набор элементов системы принятия решений (СПР), приоритетность и параметры которых определялись пользователем.

Ключевые слова: сегментная спайковая модель нейрона; биоинспирированная управляющая система; программная реализация

I. ОБЗОР ЛИТЕРАТУРЫ

A. Удаленное управление мобильными роботами

Удаленное управление мобильными роботами реализуется во многих областях [1], при этом отдельное место занимает осуществление удаленного управления в сложных условиях. Так исследование космического пространства, ликвидация последствий катастроф, разведка полезных ископаемых, обезвреживание взрывоопасных устройств [2] проводятся в условиях неблагоприятной для человека, но допустимой для функционирования автономного интеллектуального агента среды. В данной работе спайковая нейронная сеть использовалась в приложении для задания поведения автономного мобильного робота.

В общем случае решается задача обеспечения выполнения роботом некоторых действий: например, в [3] рассматривался алгоритм обхода препятствий в неструктурированной среде, а в [4] производилось управление промышленным роботом для дуговой сварки.

B. Обеспечение адаптивности мобильного робота к условиям окружающей среды

В создании системы удаленного управления автономным роботом применяются технологии, связанные с искусственными нейронными сетями, которые обладают адаптивными свойствами и высокоскоростной параллельной обработкой данных [5].

Спайковые нейронные сети (SNN – Spiking Neuron Network) могут использоваться в качестве контроллеров роботов [6] благодаря своим преимуществам по сравнению с классическими искусственными

нейронными сетями: лучшей устойчивости к шуму и меньшими энергозатратами. Так, в [7] приведен пример, как SNN выступает в качестве оператора манипулятора, а в [8] SNN взаимодействует с контроллером для решения задачи сопровождения цели.

В данной работе спайковая нейронная сеть использовалась в приложении для задания поведения автономного мобильного робота.

Необходимость создания программной реализации вызвана потребностью в визуализации настроек управляющей системы и возможности сохранять и загружать созданные структуры системы управления мобильным роботом. Проверка работоспособности приложения проводилась на робототехнической платформе при использовании удаленного рабочего стола.

II. ИНСТРУМЕНТАРИЙ

A. Робототехническая платформа

В качестве основы для автономного робота (рис. 1), включающей в себя колесную платформу, контроллер и аккумуляторный блок, была выбрана робототехническая платформа «Omegabot».

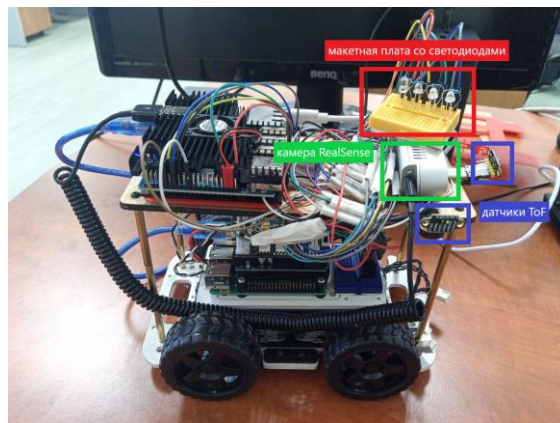


Рис. 1. Внешний вид робототехнической платформы

В качестве вычислительного устройства использовался одноплатный компьютер Orange Pi 5 Plus, а связь между ним и моторами реализована за счет платы Arduino Uno с прошивкой библиотекой FirmataExpress.

На платформе располагалась пара связанных между собой и компьютером макетных плат, на которых были установлены светодиоды разных цветов. При

Работа выполнена в рамках выполнения государственного задания Минобрнауки России №075-00553-25-00 от 25.12.2024 «Исследование методов организации навигации роботов в сложных недетерминированных средах на базе теории когнитивного представления пространства» (FNRG-2025-0007 1024050200005-9-1.2.1;2.2.2)

детектировании той или элемента СПР зажигалась определенная последовательность светодиодов.

Также с трех сторон робототехнической платформы было прикреплено по дальномеру ToF (Time of Flight), а на верхней части платформы располагалась камера. С определенным временным интервалом данные с камеры поступали на вход нейронной сети модели yolo8s, которая детектирована наличие и расположение цели, в качестве которой выступала бутылка с водой.

В. Описание приложения

Приложение предназначено для задания поведения автономного мобильного робота с биоинспирированной управляющей системой. Приложение позволяет добавлять элементы СПР из существующего набора, менять их приоритет, настраивать угловую и линейную скорости, выбирать способ кодирования, а также тестировать модель поведения, созданную пользователем.

В качестве базового элемента для классов различных элементов СПР в структуре конфигурации выступает сегментная спайковая модель нейрона (CSNM – Compartmental Spiking Neuron Model) [9].

III. РЕАЛИЗАЦИЯ

А. Биоинспирированная система управления автономным мобильным роботом

Структуру системы элементов СПР можно представить в виде совокупности независимых блоков (рис. 2), из которых собиралась система управления. При этом приоритет выполнения каждого из элементов зависел от того, какое место он занимал в структуре.



Рис. 2. Система управления автономным мобильным роботом

Для запуска модулей ситуаций, соответствующих детектированию препятствия сбоку, требовались показания только с одного дальмера, для углов и преграды спереди – с двух, а детектирование отсутствия какой-либо препятствий – с каждого дальмера.

В. Функциональные возможности приложения

Приложение визуализирует настройки управляющей системы и обеспечивает возможность сохранения и загрузки созданных пользователем структур системы управления мобильным роботом. Внешний вид главного окна приведен на рис. 3.

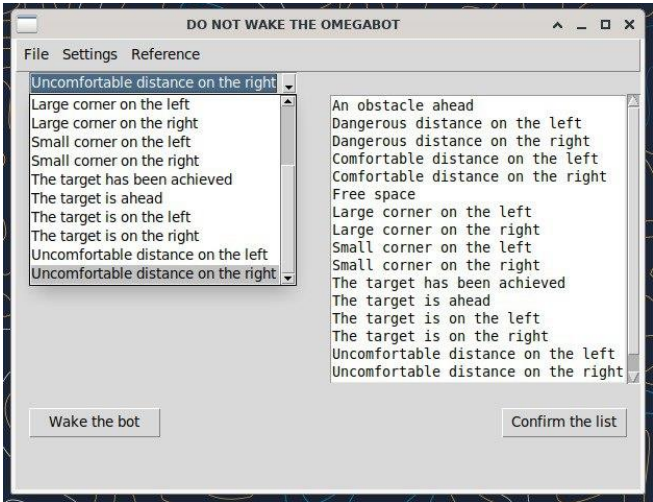


Рис. 3. Главное окно приложения

В главном окне через выпадающий список можно задать структуру системы СПР, которая отображается в текстовом поле. При загрузке пользовательская конфигурация также отображается в текстовом окне.

Перед запуском управляющего скрипта осуществляется подтверждение текущих настроек. Запуск скрипта с заданными настройками осуществляется по кнопке «Wake the bot».

Окно настройки скоростей представлено на рис. 4. Путем перемещения ползунков, соответствующим угловой и линейной скорости, пользователь задает общую скорость движения робота.

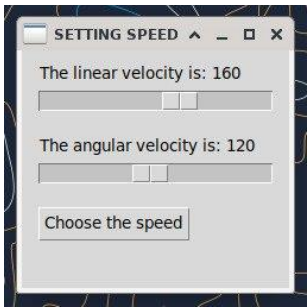


Рис. 4. Окно настройки скоростей

Окно настройки способа кодирования временных паттернов для каждого блока приведено на рис. 5.

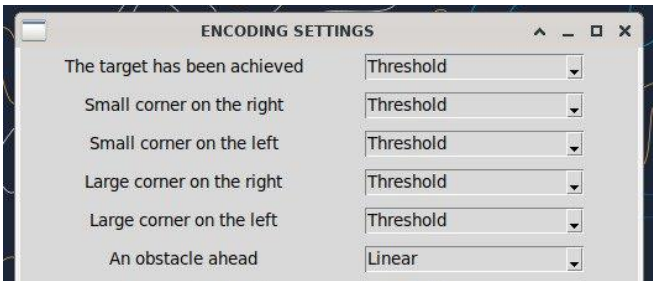


Рис. 5. Фрагмент окна настройки способа кодирования

Пользователь имеет возможность выбрать, линейным (Linear) или пороговым (Threshold) будет перевод полученных от сенсоров данных во временные задержки, подающиеся на вход нейронов.

Построение конфигурации осуществляется по умолчанию, вручную или с помощью робота. Способ кодирования по умолчанию представляет собой линейное кодирование для блока, соответствующему ситуации «Barrier» и пороговое кодирование для всех остальных блоков. Этот способ кодирования можно задать вручную или нажав на кнопку «Save» в новом окне настроек конфигурации.

Конфигурация по умолчанию может быть задана только в том случае, когда способ кодирования задан по умолчанию. Тогда при выборе данного способа задания параметров конфигурации пользователь видит соответствующее информационное окно. В противном случае окно сообщений отображает ошибку конфигурации и предлагает поменять способ кодирования.

В случае, когда конфигурация задается вручную, пользователь самостоятельно вписывает подобранные значения диапазонов срабатывания ситуации в нужные поля настройки (рис. 6). Для блоков, соответствующим детекции препятствиям, достаточно заполнить только верхнюю часть параметров, а для блоков, соответствующих детекции стен – вторую.

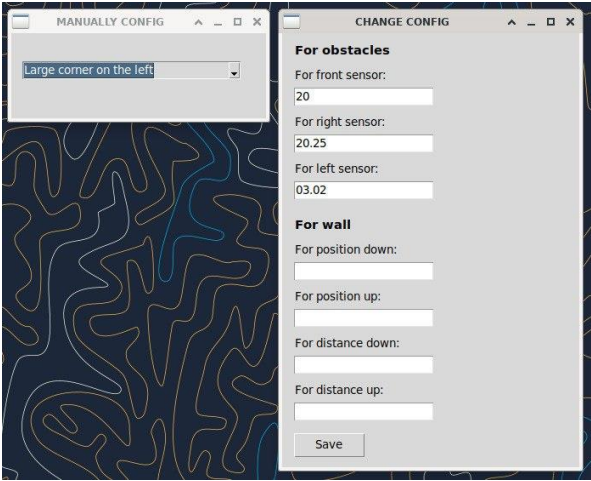


Рис. 6. Окно настройки конфигурации вручную

В качестве возможных значений выступают положительные числа (целые или с «плавающей точкой»), как показано на рис. 6).

В том случае, если настройка конфигурации определяется с помощью робота, то робот сначала вручную помещается в ситуацию, которой он обучается (рис. 7).



Рис. 7. Настройка конфигурации «Barrier» и «Small corner on the right»

После нажатия кнопки «Learn!» в окне настройки отображаются значения диапазонов срабатывания, получаемые от сенсоров (рис. 8).

При необходимости пользователь может изменить местоположение робота и запустить процесс обучения повторно.

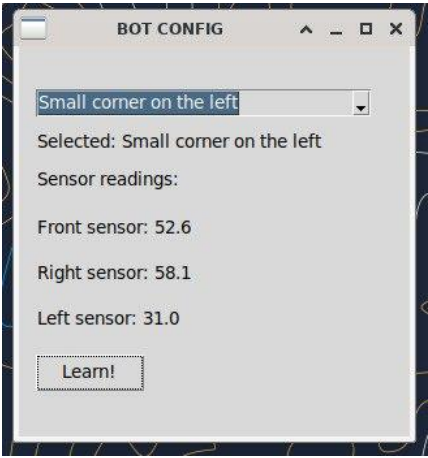


Рис. 8. Окно настройки конфигурации с помощью робота

Возможность настройки конфигурации вручную или с помощью робота возможна только для тех модулей ситуаций, в которых присутствует получение данных с датчиков.

С. Проведение экспериментов

Были протестированы все функциональные возможности приложения: настраивались структура конфигурации, линейная и угловая скорости, способ кодирования, а также возможности сохранения и загрузки выполняемых файлов.

Наиболее показательные результаты были получены в ходе эксперимента, в котором конфигурация поведения задавалась с помощью робота.

В создаваемой конфигурации были задействованы все существующие модули ситуаций. Данные для конфигурации, полученные вручную, отличались от тех, которые задавались по умолчанию (табл. 1). Показания, которые различались более, чем на 3 см, выделены шрифтом.

ТАБЛИЦА I. Отличия в данных конфигурации

Конфигурационные данные		
Модуль конфигурации	Данные по умолчанию	Данные заданы вручную
Small corner on the right	40/40 см	39.4/37.5 см
Small corner on the left	40/40 см	39.7/41.1 см
Large corner on the right	56.5/56.5 см	56.0/54.7 см
Large corner on the left	56.5/56.5 см	53.4/53.9 см
Dangerous distance on the right	10 см	13.5 см
Dangerous distance on the left	10 см	12.3 см
Comfortable distance on the right	10 см	13.6 см
Comfortable distance on the left	10 см	12.4 см
Uncomfortable distance on the right	20 см	18.1 см
Uncomfortable distance on the left	20 см	20.1 см

После настройки конфигурации робототехническая платформа была перенесена в простейший созданный лабиринт (рис. 9). Красной стрелкой на рисунке отмечено направление, в котором осуществляется движение.

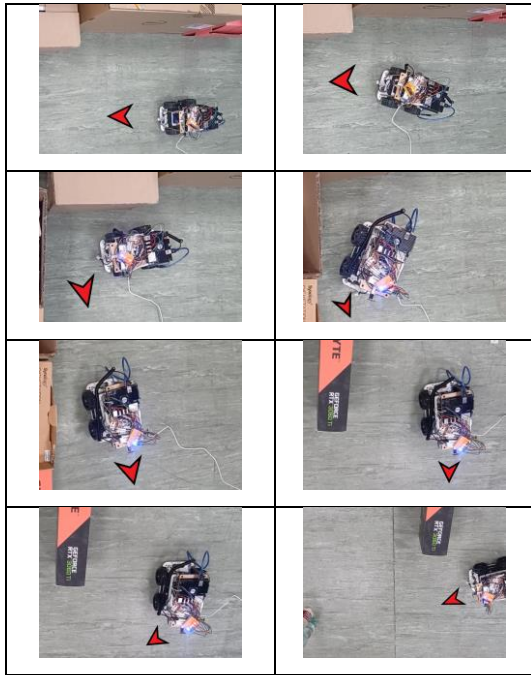


Рис. 9. Движение робота в лабиринте

Задача робота заключалась в том, чтобы добраться до цели, двигаясь на комфортном расстоянии от стены (рис. 10), не врезаясь в препятствия.

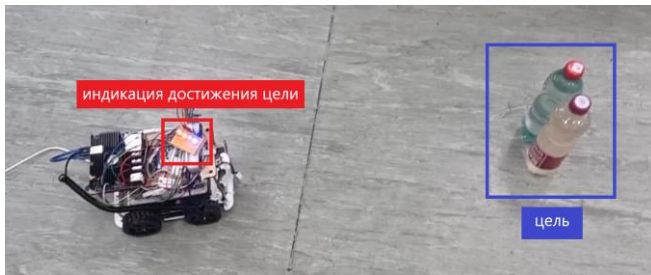


Рис. 10. Достижение цели

После успешного достижения цели настройки задания поведения робота были сброшены до значений по умолчанию и скрипт был запущен повторно, после чего проводилось сравнение выполнения задания.

При настройке параметров конфигурации вручную поведение робота незначительно отличалось от того, которое тот показывал при настройках конфигурации по

умолчанию, но в обоих случаях поставленная перед роботом цель была достигнута.

IV. ЗАКЛЮЧЕНИЕ

В работе было приведено описание программной реализации задания поведения автономного мобильного робота с биоинспирированной управляющей системой.

Функции приложения включали в себя возможности настройки структуры конфигурации, линейной и угловой скоростей, способ кодирования, а также сохранения и загрузки выполняемых файлов.

На основе вышеизложенного были сделаны следующие выводы.

- Приложение позволило визуализировать настройки задания поведения мобильного робота.
- Все функциональные возможности приложения были успешно протестированы.

СПИСОК ЛИТЕРАТУРЫ

- [1] W. Sun, C. Liu and J. Zhu, "A Remote Controlled Mobile Robot Based on Wireless Transmission," 2018 2nd IEEE Advanced Information Management, Communication, Electronic and Automation Control Conference (IMCEC), Xi'an, China, 2018, pp. 2173-2176.
- [2] Gürçam, Kenan & Bayram, Atilla. (2015). Wireless Remote Control of a Mobile Robot. International Journal of Scientific Research in Information Systems and Engineering (IJSRISE). 1. 51-56.
- [3] Gyula Mester Intelligent Mobile Robot Motion Control in Unstructured Environments // Acta Polytechnica Hungarica Vol. 7, No. 4, 2010. 153.
- [4] Żółkiewski, S., & Galuszka, K. (2015). Remote Control of Industry Robots Using Mobile Devices. Advances in Intelligent Systems and Computing, 323-332.
- [5] Jin, L., Li, S., Yu, J., & He, J. (2018). Robot manipulator control using neural networks: A survey. Neurocomputing, 285, 23-34.
- [6] Wang, X., Hou, Z.-G., Zou, A., Tan, M., & Cheng, L. (2008). A behavior controller based on spiking neural networks for mobile robots. Neurocomputing, 71(4-6), 655-666.
- [7] A. Bouganis and M. Shanahan, "Training a spiking neural network to control a 4-DoF robotic arm based on Spike Timing-Dependent Plasticity," The 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, Spain, 2010, pp. 1-8
- [8] Cao, Z., Cheng, L., Zhou, C. et al. Spiking neural network-based target tracking control for autonomous mobile robots. Neural Comput & Applic 26, 1839-1847 (2015).
- [9] Bakhshiev A., Demcheva A., Stankevich L. CSNM: The Compartmental Spiking Neuron Model for Developing Neuromorphic Information Processing Systems // International Conference on Neuroinformatics, (2021) pp. 327-333. DOI: 10.1007/978-3-030-91581-0_43