

СИСТЕМА УПРАВЛЕНИЯ НАЗЕМНЫМИ МОБИЛЬНЫМИ РОБОТАМИ

Н. В. Азарнов¹, С. Д. Кашицын², А. П. Баринов³, А. Е. Ефимов⁴, П. О. Михайлов⁵^{1,2,3,4} Научно-производственное предприятие «Исток» имени Шокина, Фрязино, Московская обл., Россия² МИРЭА – Российский технологический университет (филиал в г. Фрязино), Фрязино, Московская обл., Россия⁵ Научно-производственное предприятие «Радиотехника», Москва, Россия⁵ Московский государственный университет имени М. В. Ломоносова, Москва, Россия¹ nvazarnov@istokmw.ru, ² sedkashitsyn@istokmw.ru, ³ apbarinov@istokmw.ru,⁴ aeefimov@istokmw.ru, ⁵ pm@nppt.ru

Аннотация. *Актуальность и цели.* Рассматриваются основные аспекты построения систем пространственной ориентации и позиционирования наземных мобильных роботов с применением алгоритмов детектирования объектов. Целью работы является исследование возможности применения нейросетевых алгоритмов детектирования объектов в режиме реального времени для управления технологической транспортной платформой, выполняющей различные операции вне зданий. *Материалы и методы.* Применение в робототехнике специализированного программного обеспечения, созданного на базе программных продуктов и сред для разработки. *Результаты и выводы.* Представлен анализ наиболее подходящих нейросетевых алгоритмов детектирования объектов в режиме реального времени, позволяющих повысить качество детектирования объектов, делая системы более эффективными и точными.

Ключевые слова: робототехника, наземные мобильные роботы, адаптивные роботы, пространственная ориентация, позиционирование, навигация, управление

Для цитирования: Азарнов Н. В., Кашицын С. Д., Баринов А. П., Ефимов А. Е., Михайлов П. О. Система управления наземными мобильными роботами // Измерение. Мониторинг. Управление. Контроль. 2025. № 3. С. 67–72. doi: 10.21685/2307-5538-2025-3-8

CONTROL SYSTEM FOR GROUND MOBILE ROBOTS

N.V. Azarnov¹, S.D. Kashitsyn², A.P. Barinov³, A.E. Efimov⁴, P.O. Mikhailov⁵^{1,2,3,4} Research and Production Enterprise "Istok" named after A.I. Shokin, Fryazino, Moscow region, Russia² MIREA – Russian Technological University (Fryazino branch), Fryazino, Moscow region, Russia⁵ Research and Production Enterprise "Radiotechnika", Moscow, Russia⁵ Lomonosov Moscow State University, Moscow, Russia¹ nvazarnov@istokmw.ru, ² sedkashitsyn@istokmw.ru, ³ apbarinov@istokmw.ru,⁴ aeefimov@istokmw.ru, ⁵ pm@nppt.ru

Abstract. *Background.* Considers the main aspects of constructing spatial orientation and positioning systems for ground-based mobile robots using object detection algorithms. The aim of the work is to study the possibility of using neural network object detection algorithms in real time to control a technological transport platform that performs various operations outside buildings. *Materials and methods.* Application in robotics of specialized software created on the basis of software products and environments for development in robotics. *Results and conclusions.* Presents an analysis of the most suitable neural network algorithms for detecting objects in real time, which improve the quality of object detection, making the systems more efficient and accurate.

Keywords: robotics, ground mobile robots, adaptive robots, spatial orientation, positioning, navigation, control

For citation: Azarnov N.V., Kashitsyn S.D., Barinov A.P., Efimov A.E., Mikhailov P.O. Control system for ground mobile robots. *Izmerenie. Monitoring. Upravlenie. Kontrol' = Measuring. Monitoring. Management. Control.* 2025;(3):67–72. (In Russ.). doi: 10.21685/2307-5538-2025-3-8

Введение

Развитие робототехники и автономных систем является одним из приоритетных направлений научно-технологического развития Российской Федерации. В этом контексте особую

актуальность приобретает разработка эффективных систем управления наземными мобильными роботами, способными выполнять задачи в условиях динамичной и неструктурированной среды. Одной из наиболее актуальных задач в этой области является разработка и внедрение систем управления мобильными роботами, которые становятся важным инструментом цифровой трансформации промышленности.

В рамках данной работы рассматриваются актуальные задачи, связанные с разработкой и совершенствованием алгоритмов детектирования объектов в режиме реального времени. Основное внимание уделяется исследованию нейросетевых методов, включая модели семейства YOLO и алгоритмы SLAM, таких как RTAB-Map. Эти технологии позволяют роботам точно идентифицировать окружающие объекты и строить карты пространства даже в условиях наличия динамических препятствий.

Важным аспектом исследования является оптимизация вычислительных ресурсов автономных платформ. На примере мобильного робота X3 на базе Raspberry Pi 4B анализируются компромиссы между точностью детектирования и производительностью системы. Особое внимание уделяется практическим аспектам внедрения, включая проблемы проскальзывания колес, коррекцию траектории движения и адаптацию к изменяющимся условиям окружающей среды.

Под объектом исследования будем предполагать логистические внутрицеховые операции. В качестве предмета исследования принят мобильный транспортный робот, исследование создания которого проводится в рамках развития «Передовой инженерной школы СВЧ-электроники» на базе АО «НПП "Исток" им. Шокина».

Принципы построения и архитектура систем управления роботом

Ниже представлены результаты тестирования системы локализации и навигации автономного колесного робота. В качестве опытного образца был использован робот X3 производителя Yahboom.

В составе робота имеются:

- RGB камера (разрешение 640×480, частота кадров 30);
- RGBD камера (разрешение 640×480, частота кадров 30);
- LiDAR A1, однолучевой лидар со сканированием на 360° (диапазон расстояний от 0,15 до 12 м) и частотой сканирования 10 Гц.

В качестве вычислительного центра робота выступает одноплатный компьютер Raspberry Pi 4B, обладающий:

- четырехъядерным процессором Cortex-A72;
- оперативной памятью 4 Гб LPDDR4-3200;
- USB-портами для подключения к периферийным устройствам (камера, лидар).

Был проведен анализ качества и скорости работы алгоритмов локализации: RTAB-Map (RGBD+LiDAR режим).

Также был проведен анализ нейросетевых алгоритмов детектирования объектов в режиме реального времени, работающих в различных средах разработки:

- среда Darknet, DNN OpenCV (модели YoloV3tiny, YoloV4tiny, YoloV4);
- среда PyTorch (модели YoloV5nano, YoloV8nano).

Тестирование всех алгоритмов проводилось на базе системы Ubuntu 18.04 с программным пакетом для создания роботизированных систем ROS1-melodic. В настоящее время ROS1 считается устаревшим пакетом, ему на смену пришел ROS2, в котором реализованы более удобные методы создания программ и который делает систему робота более защищенной к сбоям в работе отдельных программ. Еще одним достоинством ROS2 является наличие и постоянное пополнение готовыми программными пакетами с открытым исходным кодом.

Алгоритм RTAB-Map

Картирование местности осуществлялось с помощью связки сенсоров RGBD, LiDAR. На рис. 1 представлена трехмерная карта помещения. Стоит отметить, что алгоритм потребляет меньшее количество вычислительных ресурсов робота, нежели чем ORB-SLAM2: во время записи карты уровень загрузки CPU составлял от 80 до 90 %, при использовании ORB-SLAM2 – 100 %.

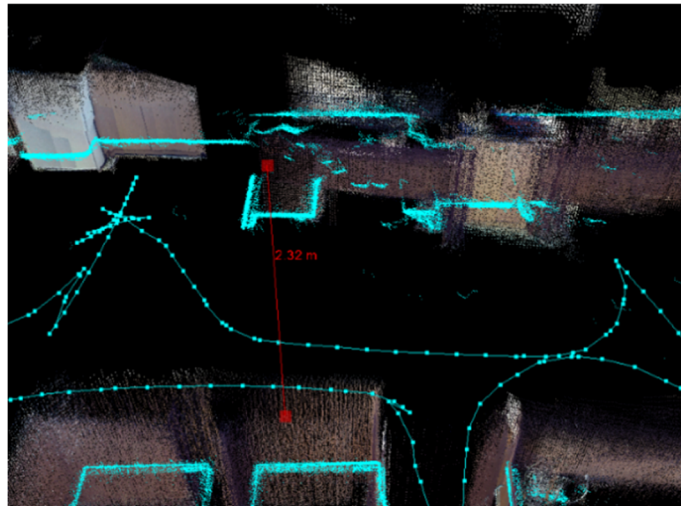


Рис. 1. Карта помещения, на которой отражена траектория движения робота в процессе картирования

Каждому узлу на траектории движения соответствует RGB-изображение и D-изображение, которые записываются в отдельную директорию пакета. На рис. 1 красным отрезком отмечено расстояние 2,32 м.

Для навигации на улице на открытой территории более целесообразно использовать не трехмерную карту местности, а двумерную. В методе RTAB-Мар двумерная карта строится на основе двух источников данных: проекции трехмерной карты на двумерную плоскость и данных с лидара. Алгоритм позволяет записать двумерную карту местности.

В пакете RTAB-Мар реализован функционал для осуществления навигации по заранее записанной карте. Для навигации необходимо совершить несколько действий: указать точку на карте, в которой находится робот и указать точку назначения. В результате робот начинает передвижение по маршруту.

Качество навигации по карте все равно невысокое, так как колеса робота проскальзывают по полу при поворотах и энкодеры колес накручивают пробег, что с точки зрения программы, означает поворот робота на определенный угол, хотя в действительности он повернулся на меньший угол. Эта проблема может быть решена подбором оптимальной скорости робота, при которой проскальзывание удастся нивелировать.

В режиме навигации робот избегает препятствий: если перегородить роботу дорогу, он в течение короткого времени (от 0,5 до 1 с) меняет траекторию движения, чтобы объехать препятствие.

В графическом интерфейсе навигации робота видно, что алгоритм четко разделяет статическую (заранее записанную) и динамическую карту местности. В настройках алгоритма можно изменять радиус «летальности» динамических препятствий и скорость затухания «летальности». «Летальность» показывает то, как сильно возрастает опасность сближения с препятствием при сокращении расстояния до него. Радиус «летальности» обозначает расстояние до препятствия, на котором у точек на карте вес летальности становится максимальным (робот избегает такие точки карты). Соответственно, вне радиуса летальности этот вес плавно уменьшается (скорость уменьшения можно настраивать).

Экспериментально были протестированы летальности с радиусом 12 и 4 см. Установлено, что одним из плюсов данного алгоритма является запись только статических объектов в карту при картировании местности. Если при записи карты мимо робота проходит человек, то алгоритмы фильтруют точки карты, которые соответствуют человеку, и они не попадают в карту местности.

Детектор Darknet и DNN OpenCV

На рис. 2 показаны графики процесса обучения моделей Yolo v3 tiny и Yolo v4 tiny для детектирования фонарей-меток, а также показано количество кадров в секунду, которое удалось получить при детектировании объектов в режиме реального времени. В дальнейшем предполагается использование предварительно обученных моделей, взятых из открытых источников.

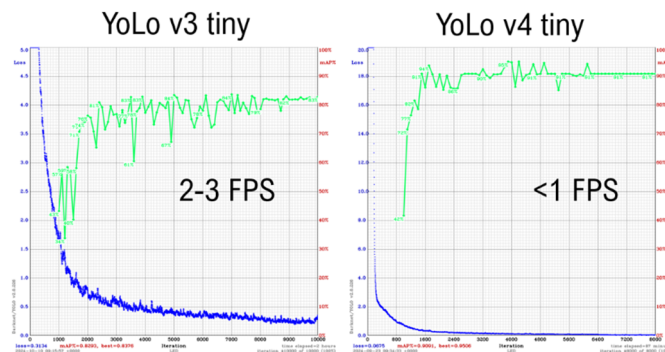


Рис. 2. Графики обучения легковесных моделей Yolo v3 tiny и Yolo v4 tiny в среде Darknet

На рис. 3 представлены экспериментальные результаты детектирования фонарей-меток. Красным прямоугольником показано местоположение фонаря на картинке, зеленым – днища фонаря. Числа указывают на достоверность детектирования: чем ближе к единице, тем лучше. Синим прямоугольником отмечен светодиод. Детектируя изменения яркости в синем прямоугольнике, робот может определять частоту мерцания фонаря и, таким образом, определять свое местоположение в пространстве.

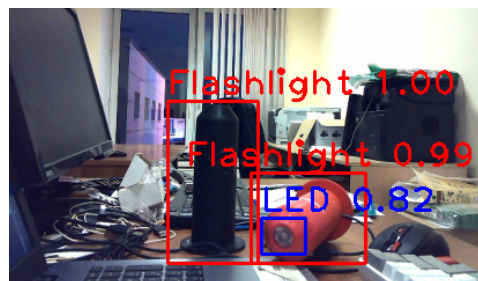


Рис. 3. Демонстрация детектирования сразу нескольких фонарей-меток

Детектор в среде PyTorch

На рис. 4 представлены результаты работы по внедрению PyTorch детекторов в робот. Прямоугольниками отмечаются объекты – выводятся их названия. Оценка достоверности детектирования «Модель» позволяет не только выводить графическую информацию о детектируемых объектах, но и передавать местоположение этих объектов в поле зрения в виде массива чисел, обозначающих границы прямоугольников, внутри которых эти объекты расположены. Эти данные можно использовать, чтобы избежать столкновений робота с объектами: чем больше размер прямоугольника, тем ближе объект находится к роботу.

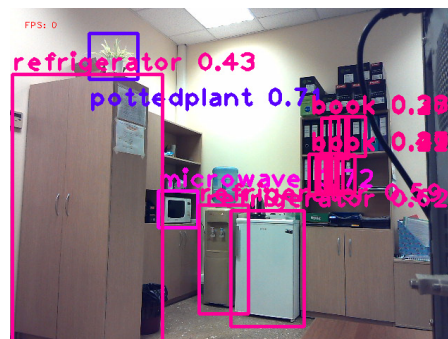


Рис. 4. Результат работы детектора Yolo v5 small PyTorch

Важной особенностью детекторов на основе среды PyTorch является практически непрерывное сопровождение детектируемых объектов на изображении, что является достоинством в контексте применения в автономных роботизированных системах.

Быстродействие окружения PyTorch не уступает быстродействию Darknet + DNN OpenCV. Поскольку у Raspberry Pi 4B довольно ограниченные возможности по осуществлению вычислений, задержка между соседними кадрами составляет порядка 2300 мс для Yolo v5 small

и 1100 мс для Yolo v5 nano. Эти модели отличаются количеством параметров. Соответственно, чем больше модель, тем лучше качество детектирования, но меньше скорость работы.

В табл. 1, 2 приведена сравнительная информация, которая показывает время обработки одного кадра (последний столбец) и информацию о модели: размер входного изображения в пикселях, качество предсказаний (стремимся к единице), количество параметров модели в миллиардах, количество операций с плавающей точкой.

Таблица 1

Сравнение различных моделей nano-сегмента

Модель	Размер изображения	mAP _{0.5:0.95} (val)	Количество параметров, 10 ⁹	Количество FLOPS, 10 ⁹	PyTorch ΔT, мс
Yolov5nu	640×640	34,3	2,6	7,7	1100
Yolov8n	640×640	37,3	3,2	8,7	1100
YoloX-nano	416×416	25,8	0,91	1,08	–

Здесь mAP – качество детектирования объекта, которое выражается в том, как хорошо прямоугольник для объекта совпадает с реальным местоположением объекта. Также в таблице отражено количество операций с плавающей точкой в секунду, которое указывает на вычислительную нагрузку на системы робота. Для модели YoloX-nano эксперименты еще не были проведены в силу возникших ошибок с ПО робота.

Таблица 2

Сравнение различных моделей small-сегмента

Модель	Размер изображения	mAP _{0.5:0.95} (val)	Количество параметров, 10 ⁹	Количество FLOPS, 10 ⁹	PyTorch ΔT, мс
Yolov5su	640×640	49,0	9,1	24,0	2300
Yolov8s	640×640	44,9	11,2	28,6	2900
YoloX-tiny	416×416	32,8	5,06	6,45	–

Видно, что при переходе от nano- к small-моделям длительность детектирования объектов на одном кадре изменяется от 1100 до 2900 мс.

Заключение

В ходе работ были проведены тесты нейросетевых алгоритмов детектирования объектов в режиме реального времени, работающих в различных средах разработки. Каждая из моделей отличается количеством параметров: чем больше модель, тем лучше качество детектирования, но меньше скорость работы. В ближайшее время при использовании более мощных вычислительных платформ появится возможность применения еще более сложных и «тяжелых» моделей в мобильных наземных роботах. Это станет важным шагом, который поможет значительно повысить качество детектирования объектов, делая системы более эффективными и точными. В данный момент ведутся работы по разработке и внедрению таких моделей, что, несомненно, откроет новые горизонты в области обработки данных и обучения машин.

Список литературы

1. Болтянский В. Г., Гамкредидзе Р. В., Понtryгин Л. С. Теория оптимальных процессов. I. Принцип максимума // Изв. АН СССР. Сер. математическая. 1960. Т. 24, вып. 1. С. 3–42.
2. Макаров И. М., Лохин В. М., Манько С. В., Романов М. П. Искусственный интеллект и интеллектуальные системы управления. М. : Наука, 2006. 333 с.
3. Юревич Е. И. Управление роботами и робототехническими системами. СПб., 2000. 171 с.

References

1. Boltyanskiy V.G., Gamkrelidze R.V., Pontryagin L.S. Theory of optimal processes. I. The maximum principle. *Izv. AN SSSR. Ser. Matematicheskaya* = Proceedings of the USSR Academy of Sciences. The series is mathematical. 1960;24(1):3–42. (In Russ.)
2. Makarov I.M., Lokhin V.M., Man'ko S.V., Romanov M.P. *Iskusstvennyy intellekt i intellektual'nye sistemy upravleniya* = Artificial intelligence and intelligent control systems. Moscow: Nauka, 2006:333. (In Russ.)
3. Yurevich E.I. *Upravlenie robotami i robototekhnicheskimi sistemami* = Control of robots and robotic systems. Saint Petersburg, 2000:171. (In Russ.)

*Информация об авторах / Information about the authors***Никита Владимирович Азарнов**

инженер,
Научно-производственное предприятие
«Исток» имени А. И. Шокина
(Россия, Московская обл., г. Фрязино,
ул. Вокзальная, 2а)
E-mail: nvazarnov@istokmw.ru

Nikita V. Azarnov

Engineer,
Research and Production Enterprise
"Istok" named after A.I. Shokin
(2a Vokzalnaya street, Fryazino,
Moscow region, Russia)

Сергей Дмитриевич Кашицын

администратор проектов,
Научно-производственное предприятие
«Исток» имени А. И. Шокина
(Россия, Московская обл., г. Фрязино,
ул. Вокзальная, 2а);
преподаватель,
МИРЭА – Российский технологический
университет (филиал в г. Фрязино)
(Россия, Московская обл., г. Фрязино,
ул. Вокзальная, 2а)
E-mail: sedkashitsyn@istokmw.ru

Sergey D. Kashitsyn

Project administrator,
Research and Production Enterprise
"Istok" named after A.I. Shokin
(2a Vokzalnaya street, Fryazino,
Moscow region, Russia);
lecturer,
MIREA – Russian Technological University
(Fryazino branch)
(2a Vokzalnaya street, Fryazino,
Moscow region, Russia)

Анатолий Павлович Барinov

архитектор программной системы,
Научно-производственное предприятие
«Исток» имени А. И. Шокина
(Россия, Московская обл., г. Фрязино,
ул. Вокзальная, 2а)
E-mail: apbarinov@istokmw.ru

Anatoly P. Barinov

Software architect,
Research and Production Enterprise
"Istok" named after A.I. Shokin
(2a Vokzalnaya street, Fryazino,
Moscow region, Russia)

Артём Евгеньевич Ефимов

младший программист,
Научно-производственное предприятие
«Исток» имени А. И. Шокина
(Россия, Московская обл., г. Фрязино,
ул. Вокзальная, 2а)
E-mail: aeefimov@istokmw.ru

Artem E. Efimov

Junior software developer,
Research and Production Enterprise
"Istok" named after A.I. Shokin
(2a Vokzalnaya street, Fryazino,
Moscow region, Russia)

Павел Олегович Михайлов

инженер-программист,
Научно-производственное предприятие
«Радиотехника»
(Россия, г. Москва, 5-й Донской пр., 15, стр. 11);
аспирант,
Московский государственный университет
имени М. В. Ломоносова, Москва, Россия
(Россия, г. Москва, ул. Ленинские горы, 1, стр. 2)
E-mail: pm@nppt.ru

Pavel O. Mikhailov

Software engineer,
Research and Production Enterprise
"Radiotechnika"
(build. 11, 15 5th Donskoy passage, Moscow,
Russia);
postgraduate student,
Lomonosov Moscow State University
(build. 2, 1 Leninskie Gory street, Moscow, Russia)

Авторы заявляют об отсутствии конфликта интересов /

The authors declare no conflicts of interests.

Поступила в редакцию/Received 11.03.2025

Поступила после рецензирования/Revised 10.04.2025

Принята к публикации/Accepted 12.05.2025