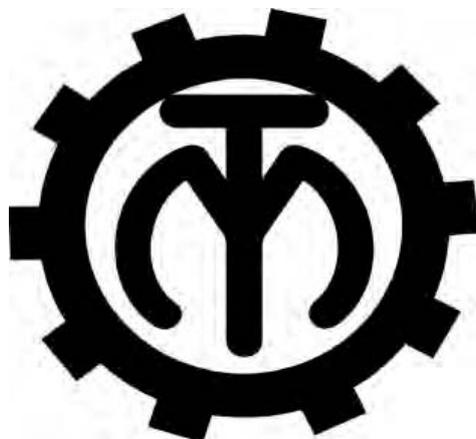


МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Технология машиностроения»

АВТОНОМНЫЕ РОБОТЫ И КИБЕРФИЗИЧЕСКИЕ СИСТЕМЫ

*Методические рекомендации к лабораторным работам
для студентов специальности
7-06-0714-02 «Инновационные технологии в машиностроении»
очной и заочной форм обучения*



Могилев 2026

УДК 65.011.56
ББК 32.965
А22

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Технология машиностроения» «29» декабря 2025 г.,
протокол № 7

Составители: канд. техн. наук, доц. В. М. Шеменков;
С. П. Шишов

Рецензент канд. техн. наук, доц. А. С. Федосенко

Методические рекомендации к выполнению лабораторных работ по дисциплине «Автономные роботы и киберфизические системы» для студентов специальности 7-06-0714-02 «Инновационные технологии в машиностроении» очной и заочной форм обучения.

Учебное издание

АВТОНОМНЫЕ РОБОТЫ И КИБЕРФИЗИЧЕСКИЕ СИСТЕМЫ

Ответственный за выпуск	М. А. Рабыко
Корректор	А. А. Подошевка
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л . Тираж 26 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2026

Содержание

1 Лабораторная работа № 1. Изучение мобильного робота Robotino	4
2 Лабораторная работа № 2. Программирование мобильных роботов.....	7
3 Лабораторная работа № 3. Изучение методов навигации мобильных роботов.....	12
4 Лабораторная работа № 4. Изучение коллаборативного робота TM12..	17
5 Лабораторная работа № 5. Изучение системы технического зрения коллаборативного робота TM12.....	20
6 Лабораторная работа № 6. Моделирование движения роботов.....	22
Список литературы.....	24

1 Лабораторная работа № 1. Изучение мобильного робота Robotino

Цель работы: изучить компоновку, принцип работы основных узлов и систему управления мобильного робота Robotino.

1.1 Основные теоретические положения

Промышленные мобильные роботы состоят из разнообразных подсистем, выполняющих различные функции. В таблице 1.1 схематически представлены такие подсистемы и подфункции.

Таблица 1.1 – Подсистемы и подфункции промышленного робота

Подсистема	Подфункция
Кинематика	Установление пространственного отношения между деталью/инструментом и технологическим устройством
Привод	Преобразование и передача необходимой энергии на все оси движения
Кодовый датчик перемещения	Измерение положения и скорости для отдельных осей движения
Управление без обратной связи / с обратной связью	Сохранение, регулирование и контроль запрограммированной последовательности. Регулировка и управление числом оборотов и скоростями
Датчики	Измерение образца физических переменных и определение положения
Механизмы захвата	Захват заготовки, фиксация положения заготовки во время перемещения

Мобильный робот Robotino (рисунок 1.1) предназначенся, главным образом, для обучения, он выполнен модульно; все технические компоненты (электроприводы, датчики, камера) можно отключить от робота и изучить отдельно.



Рисунок 1.1 – Внешний вид робота Robotino

Общая информация по роботам со всенаправленным приводом.

Основным элементом всенаправленного привода является так называемое разнонаправленное колесо или ролик, также известное как всенаправленное колесо. Эти колеса обычно имеют форму бочонка и крепятся на поворачивающейся поверхности основного колеса, их оси вращения расположены под прямым углом к оси вращения основного колеса (рисунок 1.2).

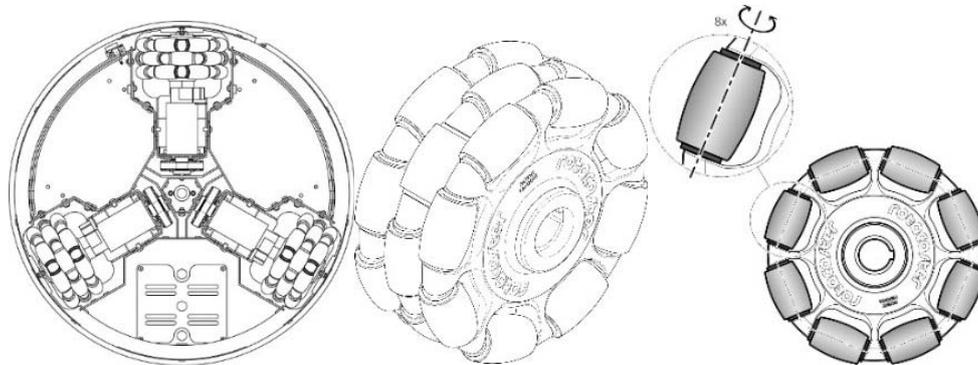


Рисунок 1.2 – Конструкция всенаправленного привода

Разнонаправленные колеса могут активно приводиться в движение двигателем, а также пассивно катиться в сторону на встроенных в колесо роликах. Роботу не надо отворачиваться от контролируемого объекта; он сохраняет свою линию визирования. В случае использования симметричного привода колеса (разнонаправленные колеса или ролики) установлены под углом 120° друг к другу. На всенаправленном приводе колеса приводятся в движение так, чтобы все они вращались вместе.

Движение тел.

Robotino может двигаться в любом направлении: вперед, назад, вбок, по кругу с сохранением или без сохранения линии визирования. Следовательно, обладает тремя степенями свободы – двумя поступательными и одной вращательной.

Система безопасности мобильного робота Robotino.

Датчик защиты от столкновения Robotino состоит из так называемой чувствительной кромки (рисунок 1.3). Эта чувствительная кромка состоит из полимерного профиля разной формы со встроенной переключающей полостью. Внутри полости располагаются две отдельные проводящие зоны, которые замыкаются накоротко при надавливании на чувствительную кромку и генерируют сигнал для блока оценки. Чувствительная кромка Robotino работает по принципу тока холостого хода, поэтому возможно обнаружение потенциального разрыва кабеля и остановка Robotino.

Ток холостого хода – это электрический ток, который непрерывно протекает в цепи, даже если она неактивна. Если ток не протекает, то это происходит из-за разрыва кабеля или повреждения чувствительной кромки.

Чувствительные кромки преимущественно применяются в технологии обеспечения безопасности для предотвращения получения персоналом травм или повреждения оборудования или материалов в результате дробления или

срезания. В медицинской технике они применяются на диагностическом оборудовании, излучателях и подвижных защитных кожухах для защиты пациентов и персонала. Также они применяются в дверях лифтов, автобусов и потолочных окнах с электроприводом для защиты пальцев. Бампер доступен при контакте, подает сигнал и в основном применяется для остановки Robotino в случае столкновения.

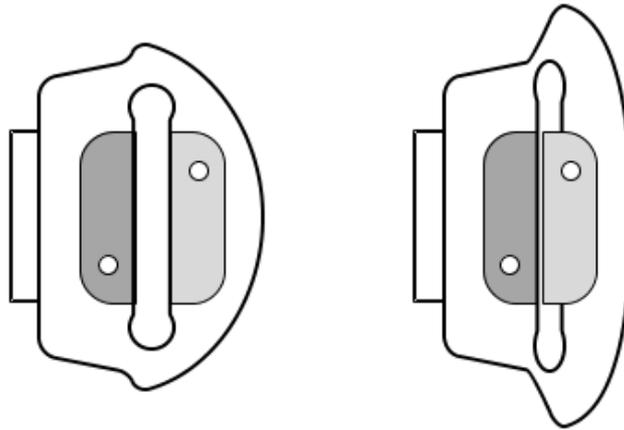


Рисунок 1.3 – Датчик защиты от столкновения

Задание

- 1 Изучить компоновку, принцип работы основных узлов и систему управления мобильного робота Robotino.
- 2 Составить отчет о выполнении лабораторной работы.
- 3 Ответить на контрольные вопросы.

Контрольные вопросы

- 1 Опишите принцип работы всенаправленного привода. Как расположены оси роликов относительно оси основного колеса?
- 2 Сколько степеней свободы имеет платформа Robotino и какие именно?
- 3 Объясните принцип «тока холостого хода» в системе безопасности (чувствительной кромке) робота. Почему это важно для обнаружения разрыва кабеля?
- 4 Какие программные среды официально поддерживаются для управления и симуляции Robotino?
- 5 Что такое одометрия в мобильной робототехнике и каковы основные причины накопления ошибок при использовании энкодеров?
- 6 В чем заключается разница между голономными и неголономными мобильными платформами?

2 Лабораторная работа № 2. Программирование мобильных роботов

Цель работы: изучить систему управления и программирование различных функций мобильного робота Robotino.

2.1 Основные теоретические положения

Robotino – это технический продукт Festo Didactic. Робот предназначен в основном для обучения, исследования и демонстрации мобильных роботизированных устройств. Robotino работает под управлением Linux (Ubuntu), но также может работать и на операционной системе Windows.

С Robotino можно взаимодействовать с помощью следующих трех стандартных программ: Robotino SIM, Robotino Factory, Robotino View. Существует возможность использования другого программного обеспечения для управления мобильной платформой, которое также официально поддерживается. Как правило, стандартные программы включают: программирование в LabVIEW; программирование в MATLAB; операционную систему программирования роботов (ROS) и др. Robotino также поддерживает средства прямого управления им: через Robotino Factory, интерфейс браузера и через вспомогательный контроллер, или через программные продукты, упомянутые ранее.

Для того чтобы управлять Robotino через Robotino View, программа должна быть напрямую подключена к роботу. После подключения к Robotino через правильный IP-адрес можно загрузить созданную программу в систему. После загрузки Robotino View не нужно повторно подключать к системе, чтобы программа могла управлять роботом. Также после этого программа будет доступна через веб-интерфейс.

Robotino View содержит элементарные элементы управления для прямого управления Robotino без сложного программирования, как показано на рисунке 2.1. Этого можно добиться, подключив систему привода Robotino к панели управления в работающей программе. Также возможно подключить внешнее устройство управления и назначить его элементы для прямого управления.

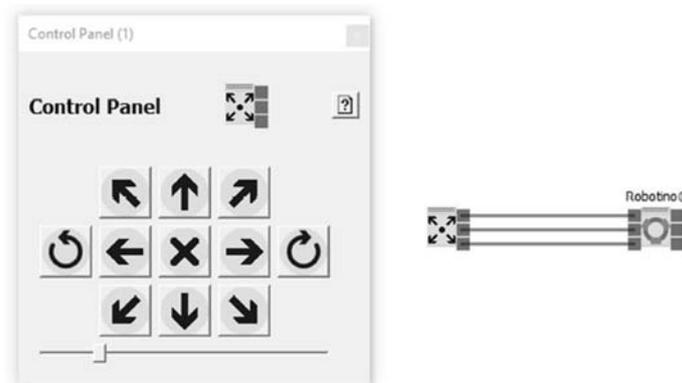


Рисунок 2.1 – Прямое управление Robotino через Robotino View

Запуск программы осуществляется нажатием зеленой кнопки воспроизведения, показанной на рисунке 2.2, вверху рядом с кнопкой сохранения. Текущую подпрограмму можно запустить, нажав вторую кнопку воспроизведения рядом с кнопкой паузы. Это полезно для проверки работы отдельных частей программы без прохождения полного цикла работы.

Чтобы остановить программу внешнего управления, необходимо нажать кнопку остановки. Кнопка остановки расположена в том же ряду, что и кнопки запуска на рисунке 2.2.

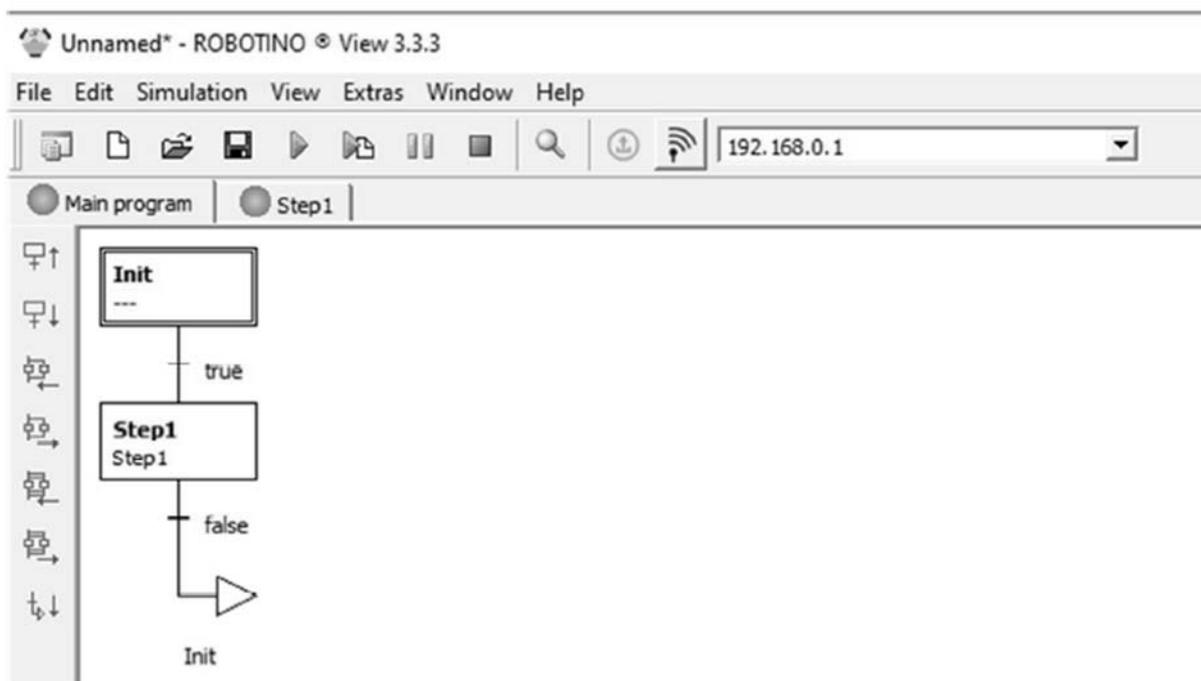


Рисунок 2.2 – Основная программа и панель инструментов Robotino View

2.2 Программирование мобильного робота Robotino в среде симулятора

Robotino SIM – это программа для Windows, предназначенная для 3D-моделирования Robotino в заранее заданной виртуальной экспериментальной среде. Имитационная модель Robotino включает в себя геометрическую модель с тремя всенаправленными ведущими колесами, камерой, девятью датчиками расстояния и цифровым датчиком в защитной планке шасси.

2.2.1 Программирование перемещения.

Папка Drive system содержит функциональные блоки для управления системой привода Robotino.

Блоки двигателей (Motor). Каждый из блоков дает доступ к управлению одним из двигателей мобильного робота Robotino (рисунок 2.3). Номер двигателя, к которому предоставляется доступ, указан на самом блоке (рисунок 2.4).

Управление скоростью каждого двигателя осуществляется с помощью ПИД-регулятора.

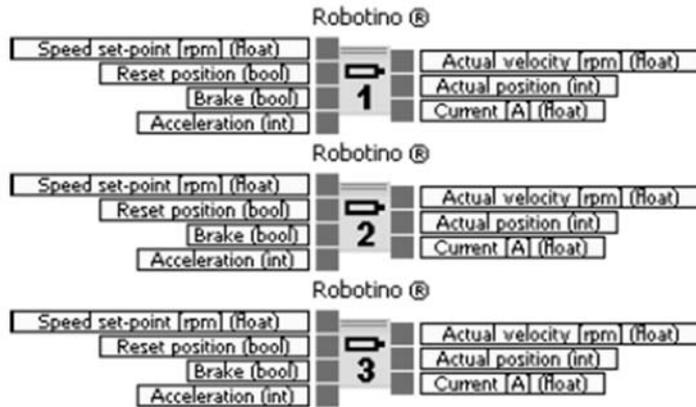


Рисунок 2.3 – Блоки двигателей Robotino

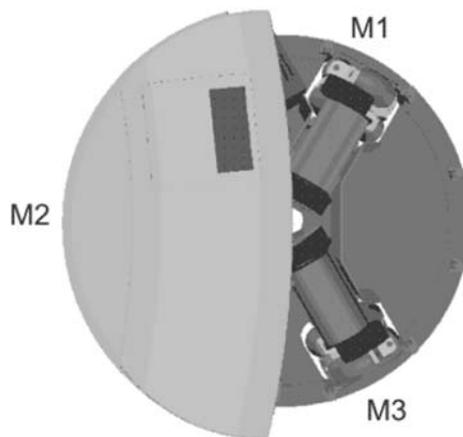


Рисунок 2.4 – Схема расположения двигателей

Функциональный блок Omnidrive. В блоке Omnidrive рассчитываются заданные скорости вращения для двигателей 1–3 в направлениях x и y на основании заданной скорости. Кроме того, фактическая скорость в направлении x и y , а также фактическая скорость вращения рассчитываются из фактической скорости (рисунок 2.5).

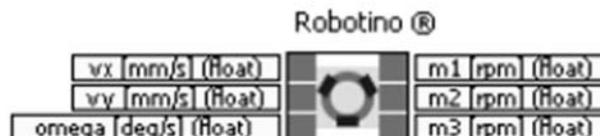


Рисунок 2.5 – Функциональный блок Omnidrive

У Robotino две поступательные степени свободы, движение по оси x и y для движения прямо вперед и боком (рисунок 2.6). Кроме того, возможно вращение вокруг оси z .

Входы и выходы модуля omni заняты соответствующим образом.

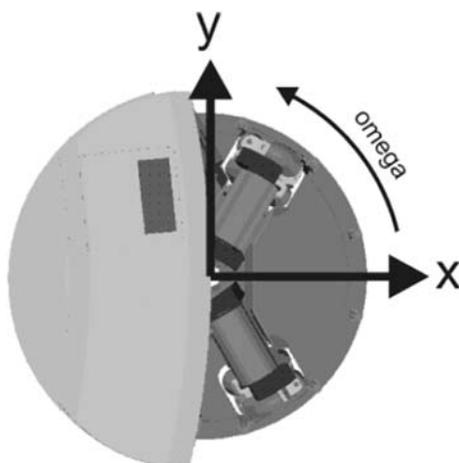


Рисунок 2.6 – Локальная система координат Robotino

Входы:

v_x_float – заданная скорость в направлении x в локальной системе координат Robotino, мм/с;

v_y_float – заданная скорость в направлении y в локальной системе координат Robotino, мм/с;

ω_float – заданная скорость вращения, град/с.

Выходы:

$m1_float$ – заданное значение скорости двигателя 1, мин^{-1} ;

$m2_float$ – заданное значение скорости двигателя 2, мин^{-1} ;

$m3_float$ – заданное значение скорости двигателя 3, мин^{-1} .

Направление (координаты) и заданная скорость определяются, вводятся и рассчитываются на входе функционального блока *omnidrive* и преобразовываются в заданные значения для соответствующей скорости двигателя (см. рисунок 2.6).

Заданные скорости для трех двигателей выводятся на выходе функционального блока *omnidrive*. Эти заданные скорости позволяют реализовать соответствующие фактические скорости в направлении x и y или скорость вращения.

По ключевому слову *omnidrive* в справке Robotino View можно найти информацию о том, как в результате преобразования вектора заданной скорости и заданной скорости вращения получить заданную скорость отдельных двигателей.

2.2.2 Программирование распознавания цвета.

Существуют следующие графические блоки для распознавания цветов Robotino:

 (Robotino/Image system/Camera) – блок, подключающий камеру робота;

 (Function block library/Image processing/Segmenter) – блок, присваивающий выбранным цветам порядковый номер;

 (Function block library/Image processing/Segment Extractor) – блок поиска текущего цвета сегмента и получения данных о координатах его центра;

 (Function block library/Image processing/Image Information) – блок выдачи ширины и высоты принимаемого изображения с камеры робота.

2.2.3 Программирование функций камеры.

Камера предоставляет прямое изображение происходящего, зафиксированное на данный момент времени. Камера подключена к надстройке управления через один из шести USB-разъемов. Функциональный модуль Camera (Камера) в Robotino View позволяет получать доступ к изображениям с веб-камеры, установленной на Robotino.

Настройки разрешения можно задавать в Robotino device dialogue (диалоговом окне устройства Robotino). В поле списка справа можно выбрать одно из возможных разрешений (320×240 , 640×480 , 800×600). В случае с распознаванием цветов обычно предпочтительнее более низкое разрешение, т. к. в этом случае изображение содержит меньше различных данных о цвете. Это позволяет упростить для программного обеспечения задачу распознавания связанных цветовых участков.

Robotino View предлагает большой выбор функциональных блоков для обработки изображения (рисунок 2.7).

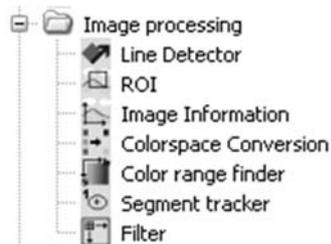


Рисунок 2.7 – Функциональные блоки для обработки изображения

Задание

- 1 Согласно заданию, выданному преподавателем, составить управляющую программу для робота Robotino в среде симулятора.
- 2 Составить отчет о выполнении лабораторной работы.
- 3 Ответить на контрольные вопросы.

Контрольные вопросы

- 1 Перечислите три стандартные программы для работы с Robotino и поясните их назначение (SIM, Factory, View).
- 2 Каким образом осуществляется связь между средой Robotino View и физическим (или виртуальным) роботом?
- 3 Какие функциональные блоки используются для получения доступа к конкретному двигателю робота и как определяется его номер?
- 4 Опишите назначение и основные входы/выходы блока Omnidrive. Какие параметры движения он рассчитывает?
- 5 По какому принципу работает ПИД-регулятор?

3 Лабораторная работа № 3. Изучение методов навигации мобильных роботов

Цель работы: изучить методы навигации мобильных роботов.

3.1 Основные теоретические сведения

Существует три основных подхода к навигации.

1 Навигация с картой.

Классический подход, требующий построения карты окружающей среды (карты занятости, графа, карты стоимости) и решения задачи локализации робота на этой карте. Позволяет находить глобально оптимальные пути.

Преимущества:

- глобальная оптимальность. Позволяет находить наилучший (кратчайший, самый быстрый, самый безопасный) путь с точки зрения всей карты, а не текущего окружения;
- высокая предсказуемость и надежность. Поведение робота детерминировано и предсказуемо, т. к. известна вся геометрия пространства;
- эффективное планирование. Можно использовать сложные, вычислительно затратные алгоритмы планирования (A, D, RRT) на этапе предварительной подготовки;
- интеллектуальное поведение. Легко реализовать сложные сценарии: посещение нескольких точек, выбор альтернативных маршрутов, оптимизация по нескольким критериям (расстояние, энергия, время);
- относительно низкие требования к вычислениям в режиме реального времени. Основная нагрузка (построение плана) выполнена заранее. В реальном времени требуется лишь коррекция локализации и слежение за маршрутом.

Недостатки:

- зависимость от карты. Критически требует точной, актуальной карты. Не работает в неизвестной или сильно изменяющейся среде;
- проблема локализации. Требуется решения сложной задачи точного определения своего положения на карте (часто с помощью дополнительных сенсоров – лидаров, камер, UWB-меток), особенно после сбоя или в повторяющихся местах;
- негибкость к изменениям. Любое непредвиденное препятствие (поставленный стул, закрытая дверь, человек), не нанесенное на карту, может полностью заблокировать выполнение плана;
- трудоемкость подготовки. Создание и поддержание точной метрической карты (особенно крупных объектов) – отдельная сложная задача.

2 Бескартная (реактивная) навигация.

Робот действует на основе непосредственных данных с датчиков без построения внутренней модели мира. Оптимизирует путь локально, но не глобально.

Преимущества:

- простота и надежность реализации. Алгоритмы (Bug-алгоритмы, метод

потенциальных полей) просты для понимания и кодирования;

- высокая скорость реакции. Минимальная задержка между сенсорным входом и управляющим воздействием. Отлично подходит для избегания внезапных, динамичных препятствий;

- низкие вычислительные затраты. Не требует хранения и обработки карты, сложных вычислений;

- робастность в динамичной среде. Эффективно работает в условиях, где окружение постоянно меняется (например, среди движущихся людей);

- не требует предварительных знаний. Готов работать «из коробки» в любой новой среде.

Недостатки:

- локальная оптимальность (ловушки). Легко попадает в локальные минимумы (тупики, циклические движения). Не может «видеть» обходной путь, если прямой заблокирован;

- отсутствие глобального планирования. Неспособен выбрать стратегически лучший маршрут. Может привести к длинным, неэффективным или вообще невозможным траекториям в сложных средах;

- ограниченность поведения. Не может выполнять сложные задачи, требующие памяти о среде (вернуться в точку старта, построить эффективный маршрут обхода нескольких точек);

- нестабильность на открытых пространствах. Методы вроде потенциальных полей могут вызывать колебания при отсутствии четких ориентиров.

3 SLAM (Simultaneous Localization and Mapping) – Одновременная локализация и построение карты.

SLAM решает две взаимосвязанные задачи одновременно: построение карты неизвестной среды и определение собственного положения робота внутри этой карты.

Преимущества:

- полная автономность в неизвестной среде. Главное преимущество. Не требует никакой априорной информации. Робот может начать работу «с чистого листа»;

- универсальность. Построенную карту можно сразу использовать для глобального планирования, как в первом подходе;

- адаптивность. Может учитывать изменения в среде (в той или иной степени), обновляя карту;

- решает фундаментальную проблему. Преодолевает циклическую зависимость «для локализации нужна карта, для карты нужна локализация».

Недостатки:

- высокая вычислительная сложность. Реализация (особенно оптимизационных и визуальных методов) требует значительных ресурсов CPU/GPU;

- проблема накопления ошибки и замыкания петель. Ошибки одометрии накапливаются, карта «плывет». Критически важна корректная идентификация ранее посещенных мест (loop closure), которая сама по себе является сложной задачей;

- требовательность к сенсорам и окружению. Для надежной работы нужны качественные и часто дорогие датчики (лидары, камеры глубины). Может плохо работать в лишенных особенностей (гладкие белые стены) или сильно динамических средах;

- сложность отладки и настройки. Ошибки в SLAM-системе часто трудно диагностировать, т. к. они влияют и на карту, и на локализацию одновременно;

- неявная форма представления карты. Результатом часто является сложный граф поз или облако точек, требующее дополнительной обработки для понятного человеку представления.

Таким образом, для производственных сред, таких как складские и заводские помещения, наиболее рационально использовать методы навигации с картой, т. к. они идеально соответствуют ключевым требованиям этих объектов: обеспечивают максимальную точность и предсказуемость перемещений в условиях стабильной, заранее известной планировки; позволяют интегрировать виртуальные зоны безопасности и оптимизировать логистические потоки на основе единой цифровой модели; требуют меньших вычислительных ресурсов и затрат на внедрение по сравнению с SLAM, избегая при этом фатальных недостатков реактивной навигации, таких как локальные ловушки и отсутствие глобального планирования.

3.2 Методы навигации с картой

3.2.1 Дистанционное преобразование.

Дистанционное преобразование (Distance Transform) представляет собой волновой алгоритм, который строит карту расстояний от всех точек свободного пространства до целевой позиции, минимизируя метрику длины пути с учётом препятствий. Его функционирование основано на итеративном распространении волнового фронта от цели, присваивающем каждой ячейке стоимость, равную минимальному количеству шагов до цели; последующая навигация сводится к градиентному спуску робота по этому полю к ячейке с нулевым значением. Ключевыми особенностями метода являются гарантированное нахождение глобально оптимального (кратчайшего) маршрута и значительные вычислительные затраты на этапе предварительного планирования. При этом сформированный план жёстко привязан к конкретной цели и требует полного пересчёта при её изменении.

3.2.2 Алгоритм D.*

D* является развитием графовых методов поиска пути, адаптированным для работы в условиях неполной или изменяющейся информации о среде. Он оперирует обобщённой картой стоимости прохождения ячеек, а не бинарной сеткой занятости, и на этапе инициализации строит оптимальный граф направлений к цели. Основное отличие заключается в механизме инкрементного перепланирования, который позволяет локально корректировать граф при обнаружении изменений в стоимости ячеек (например, появлении новых

препятствий), что существенно снижает вычислительные затраты по сравнению с полным перестроением плана. Таким образом, данный алгоритм сочетает возможность работы с *planned*-моделями проходимости и эффективную адаптацию к динамическим изменениям карты, оставаясь, однако, зависимым от фиксированной цели.

3.2.3 Метод вероятностных дорожных карт.

Метод вероятностных дорожных карт (Probabilistic Roadmap, PRM) основан на вероятностной выборке конфигурационного пространства для построения упрощённой сети допустимых путей (дорожной карты). На этапе планирования в свободном пространстве случайным образом генерируются узлы, которые затем соединяются прямыми беспрепятственными отрезками, образуя связный граф. На этапе запроса стартовая и целевая точки присоединяются к ближайшим узлам этого графа, а оптимальный маршрут между ними находится с помощью алгоритмов поиска на графах. Ключевыми особенностями PRM являются разделение дорогостоящего предварительного планирования и быстрого запроса пути, а также независимость построенной дорожной карты от конкретных начальной и конечной точек. Однако метод носит стохастический характер, может порождать несвязные графы и плохо учитывает узкие проходы из-за низкой вероятности их выборки.

3.2.4 Планировщик на решетке.

Планировщик на решетке (Lattice Planner) функционирует в конфигурационном пространстве робота, явно учитывая его кинематические ограничения. Алгоритм использует дискретный набор элементарных управляющих воздействий (примитивов движения), таких как дуги заданной кривизны, для построения решетки достижимых состояний путём рекурсивного применения этих примитивов из начальной конфигурации. Поиск пути в полученном графе сводится к задаче нахождения маршрута между узлами, соответствующими начальному и целевому состояниям. Основное достоинство метода заключается в генерации заведомо выполнимых для неголономной платформы траекторий, учитывающих её ориентацию, при этом гладкость и оптимальность пути напрямую зависят от богатства набора используемых примитивов и введённой стоимости различных манёвров.

3.2.5 Быстро исследующее случайное дерево.

Быстро исследующее случайное дерево (Rapidly-Exploring Random Tree, RRT) – это вероятностный алгоритм, который строит дерево достижимых конфигураций, последовательно расширяя его в сторону случайно выбранных точек в пространстве состояний. На каждом шаге выбирается случайная конфигурация, находится ближайшая к ней вершина в существующем дереве и для робота моделируется движение с выбранным управляющим воздействием в течение фиксированного интервала времени; достигнутое состояние добавляется в дерево как новая вершина, если путь свободен. Данный метод эффективно исследует высокоразмерное конфигурационное пространство, учитывая

нелинейную динамику системы, и способен находить выполнимые траектории в сложных сценариях, однако не гарантирует оптимальности найденного решения, а его результативность сильно зависит от параметров алгоритма и стохастической природы процесса выборки.

Сравнение методов навигации с картой представлено в таблице 3.1.

Таблица 3.1 – Сравнение методов навигации с картой

Метод	Distance Transform	D*	PRM	Lattice Planner	RRT
Достоинства	Простота навигации – гарантия нахождения пути	Эффективная работа в динамической среде	Разделение предварительной обработки и запросов	Генерация гладких, кинематически выполнимых траекторий	Быстрое исследование пространства
Недостатки	Непригодность для динамических условий, привязанность к цели	Зависимость от фиксированной цели	Стохастичность, плохая работа в узких проходах	Высокая вычислительная сложность при увеличении разрешения	Вероятностный характер, неоптимальность пути

Задание

- 1 Согласно заданию, выданному преподавателем, произвести моделирование траектории движения мобильного робота в среде Matlab.
- 2 Составить отчет о выполнении лабораторной работы.
- 3 Ответить на контрольные вопросы.

Контрольные вопросы

- 1 В чем заключаются основные преимущества и недостатки навигации на основе заранее построенной карты?
- 2 Какую фундаментальную проблему решает алгоритм SLAM?
- 3 Объясните проблему «локальных минимумов» в реактивной навигации.
- 4 Чем алгоритм D* отличается от классических методов поиска пути в условиях изменяющейся среды?
- 5 В чем особенности метода вероятностных дорожных карт (PRM)?
- 6 Что такое «замыкание петли» (Loop Closure) в SLAM и почему оно критично для точности карты?
- 7 Какие типы карт используются в современной робототехнике?

4 Лабораторная работа № 4. Изучение коллаборативного робота TM12

Цель работы: изучить устройство и программирование коллаборативного робота TM12.

4.1 Основные теоретические сведения

Коллаборативный робот, также известный как кобот, представляет собой высокопроизводительную роботизированную руку со встроенными интеллектуальными функциями для достижения прямого взаимодействия и скоординированной работы с операторами-людьми. Конструкция кобота предназначена для обеспечения прямого взаимодействия робота-человека с общим пространством или там, где люди и роботы находятся в непосредственной близости. И как таковые, движения роботов должны обеспечивать безопасность людей в непосредственной близости.

Робот TM12 состоит из манипулятора робота и его блока управления (включая пульт управления), как показано на рисунке 4.1.



Рисунок 4.1 – Основные компоненты коллаборативного робота TM12

4.2 Программирование коллаборативного робота TM12

Программирование коллаборативного робота TM12 осуществляется в программной среде TMflow, которая позволяет создавать управляющие программы для роботов через графический интерфейс, представленный на рисунке 4.2.

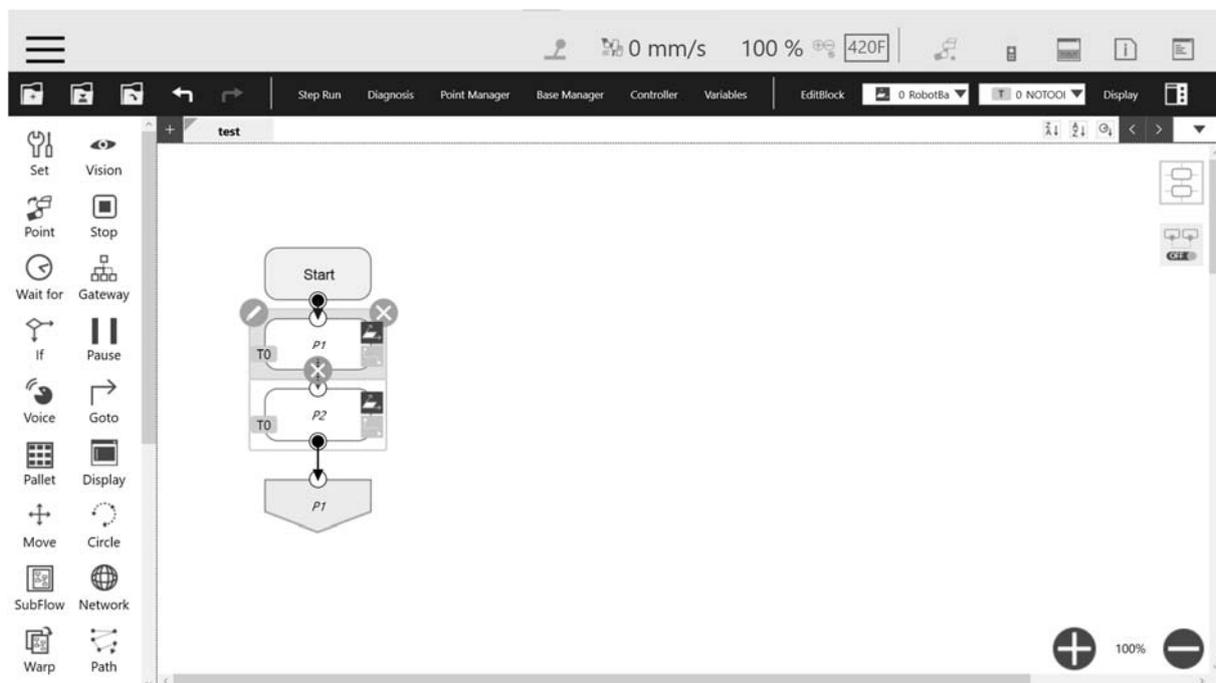


Рисунок 4.2 – Страница редактора проекта TMflow

В среде TMflow программирование построено на визуальном соединении функциональных блоков (нодов) в последовательности (поток) для выполнения задачи. Основные группы блоков и их назначение описаны в таблице 4.1.

Таблица 4.1 – Основные группы блоков в TMflow

Категория блоков	Основное назначение	Применение	Пример
1	2	3	4
Управление движением	Перемещение робота в точку, по траектории, управление инструментом	Pick-and-place (взять и поместить), сборка, сварка	Move, Waypoint, Path
Логика и алгоритмизация	Организация ветвления, циклов и пауз в программе	Создание сложной логики, ожидание сигнала, повторение действий	If/Else, Loop, Wait
Работа со встроенным зрением	Реакция на усилие и точное управление контактом	Визуальное позиционирование, проверка качества, чтение текста/кодов	Vision
Контроль усилия	Реакция на усилие и точное управление контактом	Полировка, сборка с посадкой	Force Control, Compliance

Окончание таблицы 4.1

1	2	3	4
Ввод/вывод и связь	Взаимодействие с внешними датчиками и другим оборудованием	Координация с конвейером, получение сигналов, отправка данных	Communication, Listen
Служебные и математические	Выполнение расчетов, преобразование данных, управление переменными	Математические операции, конвертация координат, логические вычисления	Set
Скриптовые узлы	Вставка пользовательского кода для сложной логики, недоступной через визуальные блок	Сложные вычисления, работа с массивами данных, нестандартная логика	TMScript

Задание

- 1 Согласно заданию, выданному преподавателем, составить управляющую программу для робота TM12.
- 2 Составить отчет о выполнении лабораторной работы.
- 3 Ответить на контрольные вопросы.

Контрольные вопросы

- 1 Дайте определение коллаборативному роботу (коботу). Чем он принципиально отличается от обычного промышленного манипулятора?
- 2 Назовите основные функциональные группы блоков в среде программирования TMflow.
- 3 Какие режимы работы системы технического зрения предусмотрены для робота TM12?
- 4 Какие четыре типа взаимодействия человека и робота определены стандартом ISO 10218?
- 5 Что такое «прямое обучение» (Hand-guiding) и как оно реализуется в коботах?

5 Лабораторная работа № 5. Изучение системы технического зрения коллаборативного робота TM12

Цель работы: изучить алгоритм настройки системы технического зрения коллаборативного робота TM12.

5.1 Основные теоретические положения

Система технического зрения (СТЗ) – аппаратно-программный комплекс, предназначенный для получения изображений объектов материального мира, их обработки, анализа и получения данных для управления исполнительными устройствами (например, промышленным роботом).

Промышленный робот с СТЗ – роботизированная система, способная адаптировать свои действия на основе информации об изменяющейся рабочей среде, получаемой посредством видеодатчиков.

Работу любой СТЗ следует рассматривать как последовательный конвейер обработки следующих данных.

1 *Получение изображения.* Исполнительное устройство – видеодатчик (камера). Фиксируется двумерная или трёхмерная проекция сцены. На этом этапе критически важны условия освещения и характеристики оптики.

2 *Предварительная обработка.* Цель – улучшение изображения для дальнейшего анализа. Применяются фильтрация (устранение шумов), коррекция контраста и яркости и геометрические преобразования.

3 *Извлечение признаков.* Изображение преобразуется в набор формализованных признаков: контуры (края), углы, области однородного цвета или текстуры, ключевые точки. На этом этапе происходит переход от массива пикселей к абстрактному описанию.

4 *Анализ и интерпретация.* На основе выделенных признаков решается целевая задача:

- обнаружение: «Есть ли объект в кадре?»;
- распознавание/классификация: «К какому классу принадлежит объект?»;
- измерение: «Каковы геометрические параметры объекта?»;
- позиционирование: «Каковы координаты и ориентация объекта?»;
- принятие решения и вывод результатов. Результат анализа (например, координаты или вердикт «брак») преобразуется в команду или сигнал для системы управления роботом (контроллера).

5.2 Система технического зрения робота TM12

Система технического зрения TM12 представляет собой законченный аппаратно-программный комплекс, встроенный в конструкцию робота. Он состоит из следующих основных компонентов.

Аппаратная часть – встроенная камера, интегрированная в кисть (запястье)

робота (в базовой комплектации – 5 мегапикселей).

Программная часть – интеллектуальное ПО TMvision для обработки изображений, работающее в среде TMflow. Включает функции распознавания образов, позиционирования объектов, чтения штрих-кодов и 2D-меток.

Помимо встроенного модуля камеры, робот TM12 также может быть подключен к внешней камере. Потенциальное использование и применение камеры с роботом TM показано на рисунке 5.1 с поддержкой трех режимов работы:

- 1) *Eye-In-Hand* – работа со встроенной камерой;
- 2) *Eye-To-Hand* – работа при подключении к совместимому внешнему устройству;
- 3) *Upward-looking Camera* – работа со встроенной камерой, позиционирование уже захваченной детали.

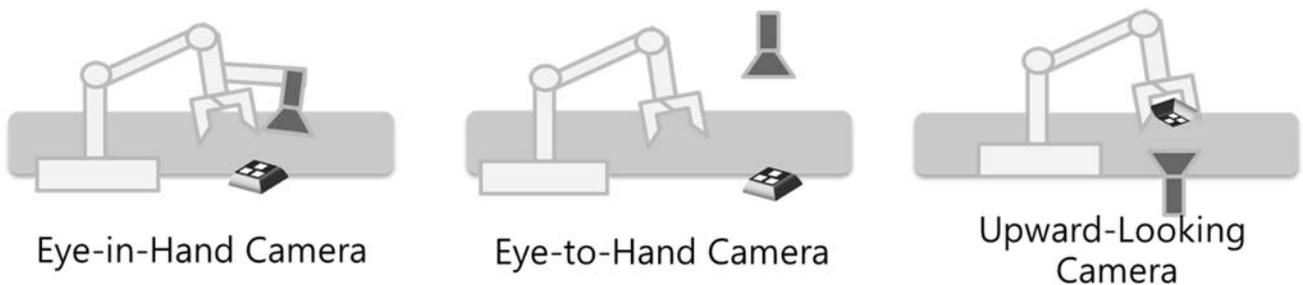


Рисунок 5.1 – Использование камеры в работе TM

Задание

- 1 Согласно заданию, выданному преподавателем, составить управляющую программу для робота TM12 с использованием системы технического зрения.
- 2 Составить отчет о выполнении лабораторной работы.
- 3 Ответить на контрольные вопросы.

Контрольные вопросы

- 1 Опишите этапы стандартного конвейера обработки данных в СТЗ: от получения изображения до принятия решения.
- 2 Почему для задач распознавания цвета в Robotino View часто рекомендуют использовать низкое разрешение камеры?
- 3 В чем разница между конфигурациями Eye-In-Hand и Eye-To-Hand?
- 4 Что такое калибровка камеры (внутренняя и внешняя) и зачем она нужна для точного захвата деталей роботом?
- 5 Объясните разницу между классическими алгоритмами обработки изображений (например, сопоставление с шаблоном) и методами на основе глубокого обучения (нейросетей).
- 6 Какую роль играет освещение в обеспечении стабильной работы промышленной системы технического зрения?

6 Лабораторная работа № 6. Моделирование движения роботов

Цель работы: изучить принципы моделирования движения промышленных и коллаборативных роботов.

6.1 Основные теоретические положения

6.1.1 Классификация роботов-манипуляторов.

Роботы-манипуляторы – это стационарные роботы с фиксированным основанием, работающие в ограниченном рабочем пространстве. Их основная задача – манипулирование объектами с помощью конечного эффектора (инструмента, схвата).

По структурной организации различаются следующие роботы-манипуляторы.

Последовательные манипуляторы. Цепь из жёстких звеньев и приводных сочленений (шарниров). Наиболее распространённый тип, например, шести-осевой промышленный робот.

Параллельные манипуляторы. Инструмент приводится в движение несколькими кинематическими цепями (ногами), соединёнными параллельно. Обладают высокой жесткостью и точностью.

SCARA-роботы. Жёсткие в вертикальном направлении и податливые в горизонтальной плоскости, оптимальны для сборки.

Портальные роботы. Двигаются по надземным рельсам, обеспечивая большую рабочую зону.

6.1.2 Прямая и обратная задачи кинематики.

Прямая задача кинематики – это задача определения положения и ориентации конечного звена робота по известным обобщённым координатам (углам/перемещениям в сочленениях). Положение каждого звена относительно предыдущего описывается однородной матрицей преобразования размерностью 4×4 , объединяющей вращение и перемещение. Положение и ориентация конечного звена вычисляется как последовательное перемножение матриц преобразования для всех звеньев и сочленений, от основания к инструменту.

Обратная задача кинематики – это задача, заключающаяся в нахождении набора обобщённых координат, которые обеспечат заданную позу конечного звена робота или инструмента. Эта задача сложнее и часто не имеет единственного решения.

Существуют следующие методы решения обратной задачи кинематики.

1 Аналитическое решение. Существует для манипуляторов с определённой геометрией, чаще всего для шестиосевых роботов со сферическим запястьем. Позволяет вывести явные формулы для вычисления углов поворота звеньев. Решения могут различаться по кинематической конфигурации (правый/левый, локоть вверх/локоть вниз).

2 Численное решение. Применяется для роботов любой структуры (с меньшим или большим числом степеней свободы, без сферического запястья). Задача сводится к итеративному подбору углов сочленений, минимизирующему ошибку между текущей и желаемой позой. Реализуется, например, методом Ньютона или с использованием псевдообратной матрицы Якоби.

6.2 Планирование движения

Для выполнения полезной работы робот должен двигаться по заданной *гладкой траектории* между точками.

Гладкая траектория – это главный, физически реализуемый закон движения из точки А в точку Б, который гарантирует непрерывность (отсутствие скачков), как минимум, скорости, а желательно, и ускорений звеньев манипулятора для обеспечения безопасности, точности и долговечности роботизированной системы.

Системы планирования движения роботов-манипуляторов можно разделить на следующие виды.

1 *Движение в пространстве сочленений (Joint-Space Motion)*. Траектория строится как интерполяция между начальной и конечной конфигурациями в пространстве обобщённых координат.

2 *Движение в декартовом пространстве (Cartesian Motion)*. Траектория задаётся как прямая линия (или иная заданная кривая) в рабочем пространстве робота. Для каждой точки траектории решается обратная кинематическая задача.

Задание

1 Согласно заданию, выданному преподавателем, смоделировать движение робота-манипулятора в среде Matlab.

2 Составить отчет о выполнении лабораторной работы.

3 Ответить на контрольные вопросы.

Контрольные вопросы

1 Сформулируйте прямую и обратную задачи кинематики. Какая из них технически сложнее?

2 В каких случаях применяется численное решение обратной задачи кинематики вместо аналитического?

3 Что такое «гладкая траектория» и какие параметры движения должны быть непрерывны для её обеспечения?

4 Опишите метод преобразования Денавита – Хартенберга (D–H) для описания геометрии манипулятора.

5 Что такое сингулярность (особое положение) манипулятора и почему движения робота вблизи точек сингулярности могут быть опасны?

6 Как матрица Якоби используется в расчетах скоростей звеньев робота?

Список литературы

- 1 **Иванов, А. А.** Основы робототехники : учеб. пособие / А. А. Иванов. – 2-е изд., испр. – М. : ИНФРА-М, 2022. – 223 с.
- 2 **Малафеев, С. И.** Теория автоматического управления / С. И. Малафеев, А. А. Малафеева. – М. : Академия, 2014. – 378 с.
- 3 **Corke, P.** Robotics, Vision and Control: Fundamental Algorithms in MATLAB® / P. Corke. – 2nd ed. – Berlin : Springer, 2017. – 693 p.
- 4 **Герман-Галкин, С. Г.** Matlab & Simulink. Проектирование мехатронных систем на ПК : учеб. пособие / С. Г. Герман-Галкин. – СПб. : КоронаВек, 2017. – 368 с. : ил.
- 5 **Тимохин, А. Н.** Моделирование систем управления с применением MatLab : учеб. пособие / А. Н. Тимохин, Ю. Д. Румянцев ; под ред. А. Н. Тимохина. – М. : ИНФРА-М, 2021. – 256 с.
- 6 **Булгаков, А. Г.** Промышленные роботы. Кинематика, динамика, контроль и управление : монография / А. Г. Булгаков, В. А. Воробьев. – М. : СОЛОН-Пресс, 2020. – 484 с.