

УДК 531.36

МАЛОГАБАРИТНЫЙ ЧЕТЫРЕХНОГИЙ ШАГАЮЩИЙ РОБОТ НА БАЗЕ БЕСКОЛЛЕКТОРНЫХ МОТОРОВ

© 2025 г. В. М. Буданов^a, В. А. Данилов^{a, *}, Д. В. Капытов^a, К. В. Климов^a

^aНИИ механики МГУ, Москва, Россия

*e-mail: vldanilov90@gmail.com

Поступила в редакцию 05.09.2024 г.

После доработки 23.09.2024 г.

Принята к публикации 16.10.2024 г.

Описываются механика, электроника и алгоритмы управления новым четырехногим шагающим роботом. Робот предназначен для сферы образования и исследовательской деятельности. Он имеет сравнительно малые массогабаритные показатели и построен с учетом модульности для упрощения производства и сборки. Робот приводится в движение с помощью бесколлекторных двигателей. Для их вращения используется векторное управление. В качестве базовой походки выбрана “рысь”. Алгоритм поддержания равновесия корпуса основан на методе управления точкой нулевого момента с прогнозированием. Формирование желаемой точки нулевого момента происходит с помощью отслеживания пересечения проекций концов диагонально расположенных ног. Программное обеспечение робототехнической платформы имеет полностью открытый исходный код. Для проверки работоспособности всех компонентов робота проведен ряд экспериментов как в виртуальной среде, так и на аппаратной платформе. Результаты экспериментов демонстрируют эффективность робота при ходьбе в разных направлениях с максимальной скоростью 1.2 м/с и грузоподъемностью вплоть до 7 кг, что практически равно собственному весу.

Ключевые слова: четырехногий робот, динамическая ходьба, бесколлекторный двигатель, точка нулевого момента

DOI: 10.31857/S0002338825030146, EDN: BHCUSF

MORS: BLDC BASED SMALL SIZED QUADRUPED ROBOT

V. M. Budanov^a, V. A. Danilov^{a, *}, D. V. Kapytov^a, K. V. Klimov^a

^aInstitute of Mechanics of Lomonosov Moscow State University, Moscow, Russia

*e-mail: vldanilov90@gmail.ru

The article describes the mechanical design, electronics and control software of the new quadruped robot MORS. The robot is intended for education and research fields. It is relatively small and lightweight and has modular design to simplify production and assembly. The robot is driven by brushless DC motors governed by the Field-Oriented Control. The trot gait is chosen as the basic one. The locomotion control algorithm is based on Zero Moment Point Preview Control method. The desired zero moment point is generated by tracking the intersection of projections of diagonally opposite legs. Software of the robotic platform is completely open sourced. To test the performance of all robot components, a number of experiments were conducted both in simulation and on the hardware platform. The experimental results demonstrate the efficiency of the robot walking in different directions with velocity up to 1.2 m/s and a load capacity of up to 7 kg, which is almost equal to its own weight.

Keywords: quadruped robot, dynamic gait, BLDC motor, zero moment point

Введение. Шагающие роботы представляют значительный потенциал для применения в реальном мире. Несмотря на то, что колесный и гусеничный транспорт остается более эффективным в плане стоимости и скорости передвижения, шагающие машины имеют преимущество при передвижении по неровной местности, сыпучим грунтам и в особенности в городах, где большая часть инфраструктуры предназначена для человека. Лестницы, пороги, бордюры,

щели и неубранный снег являются существенными преградами для колесных платформ. Возможно, именно шагающие роботы скоро станут частью нашей повседневной жизни.

Интенсивные исследования по механике и управлению шагающими роботами начались в 60–70-е гг. прошлого века. Этой темой активно занимались в НИИ механики МГУ [1, 2], ИПМ им. М.В. Келдыша [3], Массачусетском технологическом институте (MIT) [4] и Токийском технологическом институте [5].

В настоящее время научные организации и коммерческие предприятия по всему миру занимаются развитием проблематики четырехногих шагающих роботов и в последние годы добились серьезных успехов. Специалисты из MIT, разработавшие робота Mini Cheetah, смогли добиться максимальной скорости в 3.9 м/с и числа Фруда [6], равного 5.1 при длине ног в 0.4 м [7]. При этом робот способен передвигаться по пересеченной местности, сохраняя устойчивость даже при резкой смене типа поверхности. Также он может выполнять акробатические трюки (сальто, прыжок с поворотом и т.д.), оптимизируя свою траекторию в реальном времени [8]. Его предшественник Cheetah 2 обладал меньшей устойчивостью, при этом мог развивать скорость 6.4 м/с с числом Фруда 7.1 [9]. В Итальянском институте технологий ведут разработку серии шагающих роботов Ну Q. В последней версии робот НуQReal способен не только устойчиво передвигаться по неровным поверхностям, но и имеет высокую грузоподъемность. Во время экспериментов робот тянул небольшой пассажирский самолет весом 3.3 т [10]. Робот RaiBo, разработанный в Корейском институте передовых технологий, передвигается по сыпучему грунту с помощью контроллера, использующего обучение с подкреплением. Во время испытаний он перемещался по песчаному пляжу со скоростью 3 м/с [11]. В Швейцарской высшей технической школе Цюриха также ведутся работы по созданию шагающих роботов. Их робот ANYmal показывает хорошие результаты в плане проходимости и преодоления неровной поверхности [12]. В работе [13] использовано управление с прогнозирующей моделью MPC (model predictive control) для управления как локомоцией, так и манипулятором, прикрепленном к телу робота. Bjelonic и др. [14] также использовали MPC, но уже для управления роботом с закрепленными на ногах колесами. Его улучшенная версия – робот Waggu имеет облегченную кинематику, но при этом способен нести грузы весом до 90 кг [15]. Помимо университетов, четырехногими шагающими роботами занимаются и коммерческие организации. Роботы компаний Boston Dynamics [16], Unitree [17], Xiaomi [18] и Deep Robotics [19] уже имеются в продаже. К сожалению, по ним нет научных публикаций, так как вся информация закрыта, и можно опираться только на демонстрационные видео и информацию с сайтов разработчиков.

Большой проблемой в данный момент является малое количество специалистов, способных создавать шагающих роботов. Работа над такими проектами требует широких знаний и умений в различных областях, таких как механика, электроника, программирование, математика, теория автоматического управления. Хотя вышеперечисленные роботы обладают превосходной проходимостью, устойчивы к внешним возмущениям, выполняют акробатические трюки и перевозят большие веса, вся их внутренняя программная часть закрыта для пользователя. Это не только вызывает сложности при обучении студентов, но и затрудняет разработку собственных методов управления локомоцией.

В этой статье авторы представляют четырехногую шагающую платформу МОРС (малый образовательный робот-собака) (рис. 1). Цель данной разработки – дать возможность изучать устройство шагающих роботов и алгоритмы управления ими. Платформа полностью открыта на всех уровнях [20], что также дает возможность разрабатывать под нее собственные методы управления от взаимодействия с датчиками до навигации и пользовательских приложений. В существующих на данный момент открытых шагающих робототехнических платформах [21, 22] используются сервоприводы с коллекторными двигателями. В роботе МОРС, как и в промышленных роботах [16–19], установлены бесколлекторные моторы, дающие большие перспективы в плане увеличения скорости передвижения и грузоподъемности.

При разработке авторы ставили перед собой следующие задачи:

1) построить алгоритм управления, который обеспечивает равновесие робота, когда только две ноги находятся в фазе опоры;

2) при резком изменении направления движения алгоритм гарантирует быстрое, но вместе с тем достаточно плавное изменение траектории движения ног, другими словами, при движении не должно быть рывков;

3) робот имеет небольшие габариты (помещается на столе) для облегчения работы студентов и исследователей при условии, что в нем установлены бесколлекторные двигатели и компьютер с архитектурой процессора x86;

4) робот легко собирается и разбирается для облегчения ремонта;



Рис. 1. Фотография робота МОРС.

5) платформа имеет возможность легкой установки дополнительного оборудования (различные камеры, датчики и подвижные элементы);

6) электроника является модульной для облегчения замены и интеграции нового аппаратного обеспечения;

7) платформа имеет открытый исходный код систем верхнего уровня.

1. Описание системы. Описываемый робот представляет собой вторую версию четырехногого робота, разрабатываемого в НИИ механики МГУ [23]. Оба робота имеют по три электропривода на каждой ноге. Но по сравнению с предыдущей, у новой версии значительно меньшие массогабаритные показатели. Длина ноги составляет 0.295 м, общий вес робота – 7.35 кг. По габаритам он напоминает собаку среднего размера. Для достижения такого низкого показателя массы конструкция робота сделана из алюминия и карбона, а в приводах использованы бесколлекторные двигатели. Вся внутренняя электроника представляет собой модульные элементы, которые можно внедрять в другие проекты. Такая модульность позволяет как ускорить сборку, так и упростить процесс обучения студентов. Чтобы сделать робот применимым в различных сценариях, на верхней части корпуса расположены места для крепления дополнительного оборудования, такого как вспомогательные бортовые компьютеры, датчики и видеокамеры.

1.1. Конструкция робота. При разработке робота МОРС авторы ориентировались на создание максимально компактной и легкой конструкции в сравнении с другими мировыми аналогами, но с сохранением полной функциональности и всех степеней свободы.

Конструкция робота изготовлена так, чтобы его можно было легко собирать и разбирать, удобно работать с внутренним оборудованием и устанавливать дополнительное. Компонентная схема робота изображена на рис. 2, а его габаритные размеры представлены на рис. 3. Драйверы двигателей собраны в кассетные сборки в перешейках робота. Остальная электроника смонтирована модулями на единой плате, которая может быть целиком извлечена из корпуса. Рама туловища имеет жесткие несущие боковые стенки и дает возможность работать с роботом без крышки, закрывающей отсек компьютера. Верхняя крышка выполняет декоративную роль. Часть верхней крышки представляет собой ровное “окно”, сделанное из оргстекла методом лазерной резки. Эта панель позволяет легко создавать крепления для дополнительного оборудования и при этом всегда может быть заменена новой. Кроме того,



Рис. 2. Компоновочная схема робота.

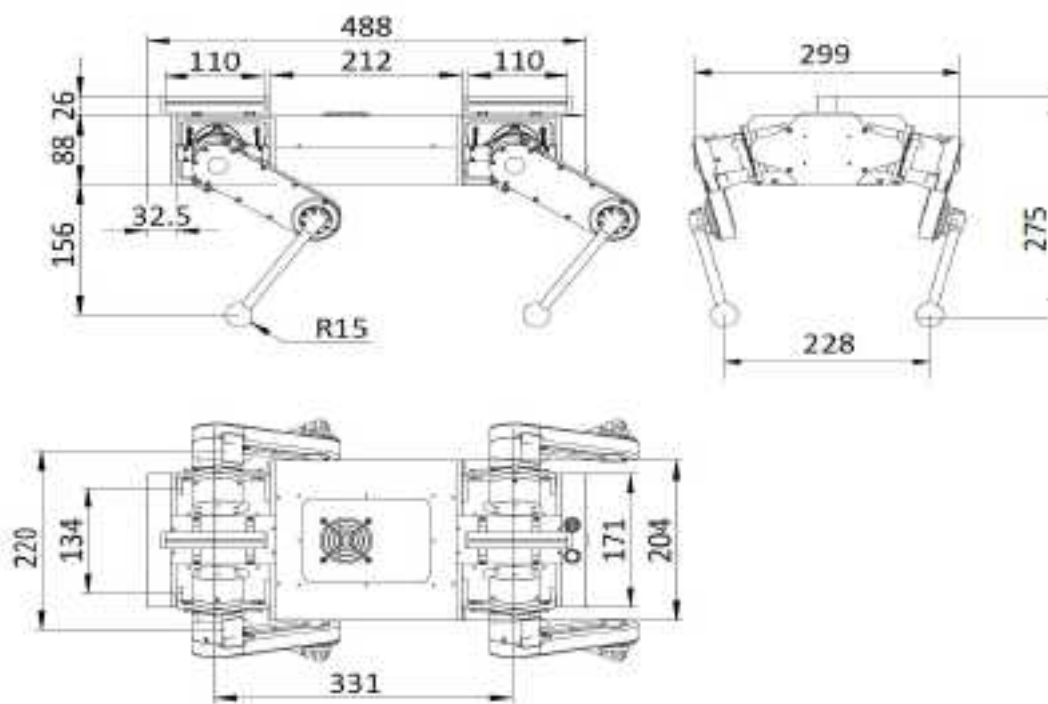


Рис. 3. Габариты робота.

над перешейками размещены две стандартные направляющие из конструкционного профиля, к которым можно прикреплять различное оборудование.

Нога робота имеет три степени свободы, что дает возможность управления стопой относительно корпуса в трехмерном пространстве. 3D-модель ноги с указанием основных узлов показана на рис. 4. Наиболее нагруженный коленный сустав имеет дополнительную понижающую ременную передачу в соотношении 2:1, что увеличивает момент в суставе и снижает нагрузку на привод во время фазы опоры. Такая конструкция позволяет расположить привод



Рис. 4. Основные узлы ноги робота.

колена соосно приводу плеча, что заметно уменьшает момент инерции ноги относительно привода плеча и привода бедра, ведет к улучшению управляемости и энергоэффективности системы. Дополнительным плюсом такой конструкции является отсутствие ограничения подвижности голени. Это дает возможность управлять ногой в двух режимах: коленями вперед и коленями назад. Голень может быть легко заменена, поскольку не имеет никаких кабелей и датчиков. Это удобно для изучения и оптимизации перемещения по различным типам поверхностей. Каждое сочленение имеет следующие диапазоны вращения: плечо — 190° (ограничено корпусом робота), бедро — 320° (ограничено длиной кабеля), колено — без ограничений.

Робот оснащен литий-ионной аккумуляторной батареей с выходным напряжением 25.2 В, номинальной емкостью 4500 мАч и пиковым током разряда 45 А. Масса батареи составляет 500 г. Она устанавливается в нижней части робота и закрепляется винтами. Конструктивно батарея представляет собой плату с держателями аккумуляторов типа 18650. Голени ног робота выполнены из карбоновых трубок, все остальные детали — из алюминия методом фрезеровки. Их общий вес равен 2550 г. Все детали проектировались так, чтобы минимизировать их массу. Общая масса робота составляет 7350 г, что на 18.3% меньше, чем масса робота Mini Cheetah [24]. Более того, это самый маленький робот на базе бесколлекторных моторов из всех известных авторам.

1.2. Э л е к т р о п р и в о д. Робот МОРС приводится в действие 12 одинаковыми приводами. В состав каждого привода входят бесколлекторный мотор постоянного тока, два абсолютных энкодера, управляющая электроника и планетарный редуктор с передаточным отношением 10:1 и сравнительно невысоким трением. Благодаря этому связь между моментом на валу двигателя и моментом на выходном валу привода близка к линейной. Такой вид привода имеет характеристики, присущие прямому приводу — высокую скорость реакции, способность вращаться под действием внешних сил, устойчивость к ударным нагрузкам. В рамках настоящей работы авторы называют такой привод псевдопрямым. Чтобы робот удовлетворял требованиям компактности, привод проектировался так, чтобы обеспечить его минимальные размеры и массу. Модель конструкции привода показана на рис. 5.

В разработанной конструкции редуктор вынесен за пределы статора. Это увеличило высоту привода, однако общий размер и вес заметно снизились: 210 г против 480 г у робота Mini Cheetah [25]. Все приводы в роботе имеют одинаковые характеристики: диаметр статора составляет 43 мм, высота — 10 мм, количество пар полюсов — 14. Модуль зуба планетарного редуктора равен 0.4 мм. Солнечная шестерня имеет 12 зубьев, три шестерни-сателлита — по 48 зубьев каждая, коронная шестерня имеет 108 зубьев. Из-за сильного облегчения приводов и отсутствия в них опорных подшипников на входе редуктора выходной вал не позволяет выдерживать необходимые радиальные ударные нагрузки, когда робот переходит из фазы переноса в фазу опоры. Для решения этой проблемы каждое плечевое звено сконструировано с дополнительным радиальным поддерживающим подшипником (рис. 4).

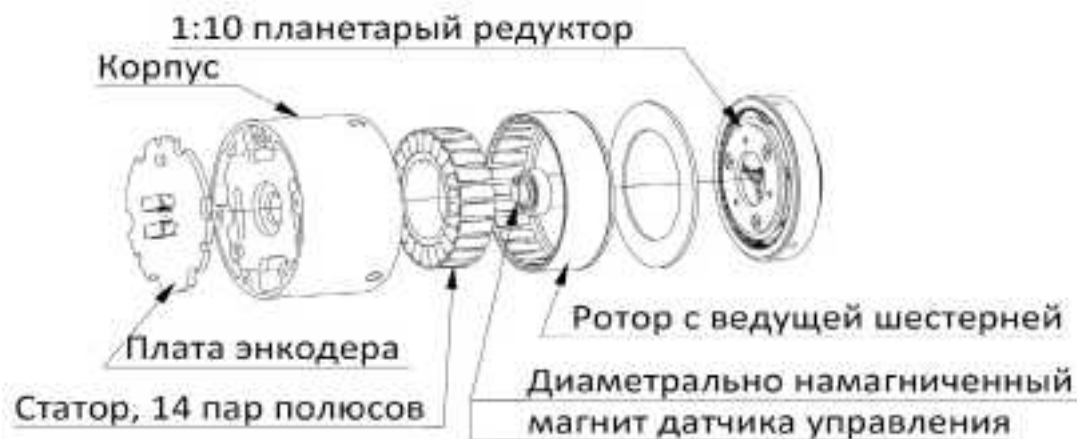


Рис. 5. Основные узлы электропривода.

Бесколлекторный мотор и планетарный редуктор находятся в общем корпусе. Абсолютный энкодер, установленный на валу двигателя, из-за наличия редуктора не может однозначно определить положение выходного вала привода. Обычный подход к решению данной проблемы заключается в складывании робота перед запуском в определенную конфигурацию так, чтобы вывод каждого из приводов находился в заранее известном механическом положении (в пределах одного сектора) [25], что накладывает определенные ограничения на удобство работы с роботом. Чтобы избежать эти трудности, каждый привод МОРСа оборудован выходом SPI (serial peripheral interface) для подключения внешнего абсолютного энкодера, расположенного за выходным редуктором. Плечевой и коленный приводы оборудованы энкодером, основанным на эффекте Холла, а привод бедра – индуктивным энкодером AS5715R.

Блок управления приводом спроектирован на основе микроконтроллера STM32G474, 3-фазного драйвера бесколлекторных моторов TI DRV8328 и абсолютного энкодера на эффекте Холла AS5047. Контроллер способен генерировать фазный ток до 17 А при питающем напряжении до 30 В. Устройство обменивается данными с бортовым компьютером по шине FDCAN на скорости до 8 Мбит/с, при этом используется протокол OpenCyphal [26].

Главной особенностью псевдопрямого привода является возможность точно регулировать момент на его выходном валу, контролируя ток мотора. Для поддержания заданного момента применяется метод векторного управления [27]. ПИ-регулятор (пропорционально-интегральный регулятор) тока получает желаемые значения из ПД-регулятора (пропорционально-дифференциального регулятора) позиции вала. Оба регулятора рассчитываются одновременно с частотой 40 кГц. Математические функции, управляющие коммутацией обмоток (преобразование Парка, преобразование Кларка и т.д.) взяты из работы [25]. Данные абсолютного энкодера фильтруются линейным фильтром Калмана [28]. Для ускорения расчета тригонометрических функций, нужных для работы регуляторов, применяется ускоритель STM32 CORDIC. Прошивка, скомпилированная с флагом оптимизации O3, использует ~65% процессорного времени для всех вычислений, связанных с управлением мотором.

1.3. Э л е к т р о н и к а. Электроника робота выполнена модульной в том смысле, что каждое электронное устройство обладает единственной главной функцией и реализовано в виде отдельной печатной платы. Такой подход упрощает включение нового аппаратного обеспечения в конструкцию робота. Приоритетным является использование готовых электронных устройств везде, где это возможно. Электронная схема МОРС изображена на рис. 6.

Для упрощения разработки силовой электроники робота (контроллеры приводов, контроллер силового питания PowerBoard) применен модульный подход. Это означает, что каждое из таких устройств состоит из нескольких печатных плат, спаянных вместе. Центральная печатная плата, называемая VBCore, содержит универсальный набор компонентов, которые должны быть представлены в каждом устройстве: микроконтроллер STM32G4, понижающий импульсный преобразователь напряжения на 5 В, линейный преобразователь на 3.3 В и трансивер шины FDCAN. Центральный модуль припаивается на функциональную печатную плату, которая содержит компоненты, определяющие назначение устройства. Например, контроллер бесколлекторного мотора несет на себе микросхему 3-фазного драйвера мотора, силовые

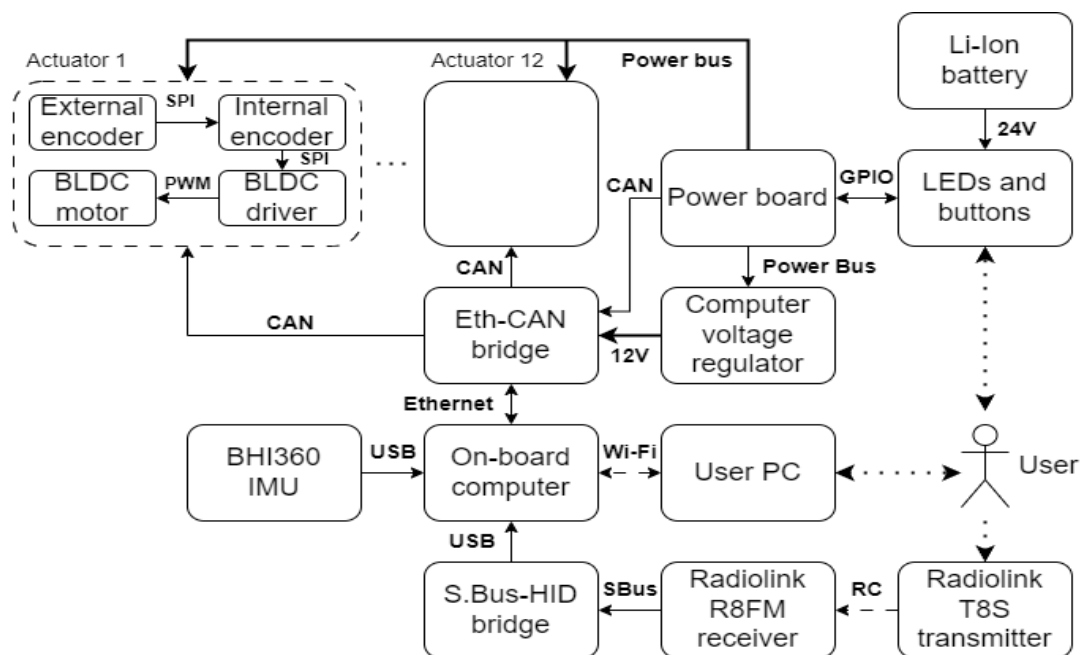


Рис. 6. Схема взаимодействия электронных блоков.

транзисторы, датчики тока и т.д. Такой подход упрощает разработку силовых электронных устройств и способствует использованию общих компонентов программного обеспечения. Центральные модули совместимы со средой программирования Arduino IDE, что позволяет студентам-первокурсникам пользоваться тем же аппаратным обеспечением, что и выпускникам. С технологической точки зрения печатная плата центрального модуля достаточно сложна и производится на заводе, но функциональные платы сравнительно просты и могут быть спаяны вручную. В результате такого подхода студентам не нужен большой опыт разработки аппаратного обеспечения, чтобы создавать сложные робототехнические устройства. Это позволяет им сэкономить время и сфокусироваться на разработке алгоритмов управления.

Для передачи данных между силовыми устройствами используются шесть независимых шин FDCAN. Для подключения к этим шинам обычного бытового компьютера на архитектуре x86 разработано специальное устройство: мост Ethernet – FDCAN. Он основан на двух микроконтроллерах: STM32H723 (ведущий) и STM32G474 (ведомый). Ведущий микроконтроллер обеспечивает работу Ethernet и трех шин FDCAN, в то время как ведомый обрабатывает еще три шины FDCAN. Программная поддержка Ethernet реализована с помощью библиотеки lwIP [29]. Бортовой компьютер применяет протокол LCM (lightweight communications and marshalling) [30] для передачи данных с приоритетом реального времени (управляющие команды для приводов) и OpenCyphal для сообщений с более низким приоритетом (связь с платой питания, настройка рабочих параметров приводов и т.д.). Поскольку передача сообщений LCM реализована на протоколе UDP multicast, такие сообщения принимаются Ethernet – FDCAN мостом напрямую по локальной сети робота. Для доставки данных с более низким приоритетом специальная программа на компьютере обеспечивает сбор сообщений, предназначенных приводам, записывает их в UDP-пакет и отправляет на мост для дальнейшей передачи по шине FDCAN. Время передачи сообщений реального времени составляет менее 1 мс.

Система питания робота обладает распределенной архитектурой. Каждый потребитель энергии (приводы, бортовой компьютер и т.д.) оборудован собственными преобразователями напряжения и питается от общей нестабилизированной шины питания. Шина питается от устройства PowerBoard, обладающего следующими функциями:

1. Мультиплексирование трех независимых источников питания (в обычной реализации – двух батарей и одного источника внешнего питания) в основную шину питания робота. Чтобы не допустить превышения рабочего напряжения шины при работе подключенных приводов в режиме генерации энергии (когда они приводятся в движение под действием внешних сил), каждый батарейный вход допускает обратный ток.

2. Защита батарей от глубокого разряда, превышения рабочего тока, короткого замыкания силовой шины.
3. Ограничение стартового тока.
4. Передача состояния системы питания робота на бортовой компьютер через шину FDCAN.
5. Реализация базового пользовательского интерфейса – кнопок, переключателей, индикаторов.
6. Реализация аварийного выключателя питания.
7. PowerBoard обеспечивает рабочий ток до 30 А и пиковый – более 50 А, при рабочем напряжении до 50 В.

Робот MOPC оборудован модулем инерциальной навигации, основанным на микросхемах Bosch BHI360 (3-осевые акселерометр и гироскоп) и Bosch BMM350 (3-осевой магнитометр). Модуль передает данные на бортовой компьютер по шине USB с использованием стандарта USB HID Sensor при частоте до 400 Гц. Для ручного управления робот имеет радиоприемник, работающий по стандарту Futaba S. Bus. Специальная печатная плата на микроконтроллере STM32F103 с помощью ПО Universal-RC-Joystick firmware [31] преобразует данные с приемника в сообщения USB HID Gamepad и отправляет их на бортовой компьютер по шине USB.

1.4. Архитектура программного обеспечения (ПО). В роботе установлен бортовой компьютер с процессором Intel Alder Lake N100 и 16 ГБ оперативной памяти. В качестве операционной системы используется Ubuntu 20.04. Все высокоуровневое ПО, такое как управление локомоцией и модули взаимодействия с внешними устройствами, работает под управлением ROS (robotic operating system) [32]. ROS Master запущен непосредственно на бортовом компьютере. ПО написано на языках Python и C++. Бортовой компьютер работает в режиме точки доступа Wi-Fi, и, подключившись к нему с любого внешнего ПК, можно через сеть ROS отправлять задающие команды или получать текущее состояние робота, тем самым распределив нагрузку между вычислительными устройствами. Подобным образом проводились эксперименты с построением карт местности и навигацией. Обмен с низкоуровневой частью осуществляется посредством протокола реального времени LCM. Структурная схема архитектуры ПО представлена на рис. 7. В этой схеме блоки с закругленными краями обозначают ROS-модули, прямоугольники – аппаратные блоки, тонкие стрелки показывают

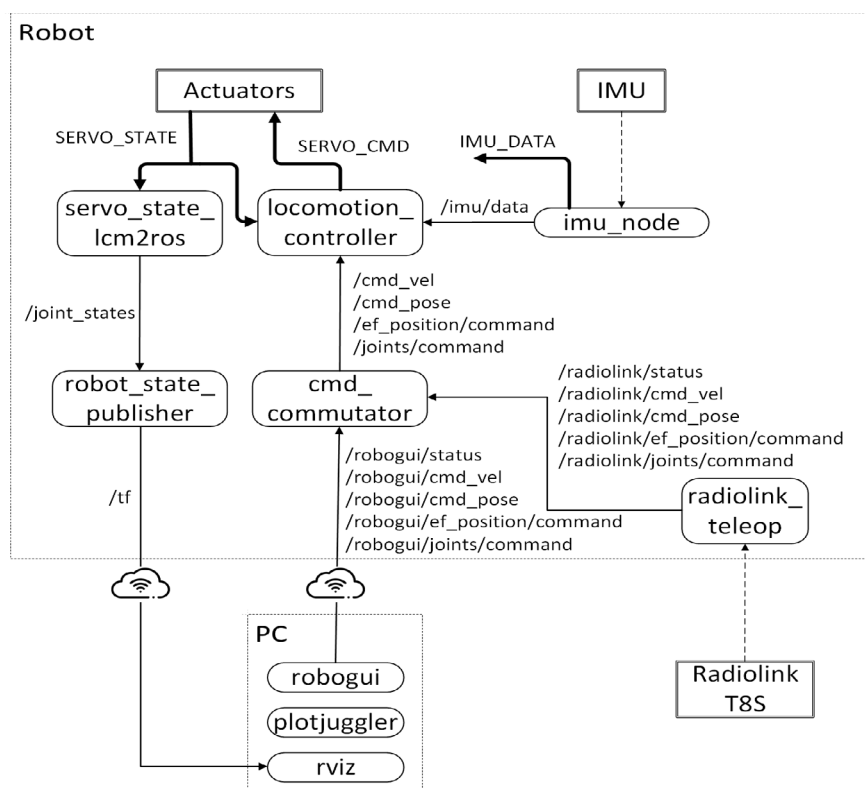


Рис. 7. Архитектура ПО.

взаимодействие через ROS, надписи с символом “/” – ROS-топики, жирные стрелки – взаимодействие посредством LCM, надписи – названия LCM-каналов.

2. Управление локомоцией. Описывается реализация основных компонентов системы управления локомоцией. По сравнению с предыдущей версией [23], в роботе МОРС алгоритм существенно изменен. Структурная схема контура управления показана на рис. 8. Каждый блок этой схемы разработан с учетом модульности, для того чтобы была возможность замены его блоком с другим алгоритмом без вмешательства в остальную часть системы управления. На вход контроллера управления локомоцией оператор или высокоуровневое ПО подают команды с желаемыми линейными и угловыми скоростями. Эти команды затем используются для планирования траектории точки нулевого момента ZMP (zero moment point) корпуса робота в фазе опоры и положения следовых точек в фазе переноса.

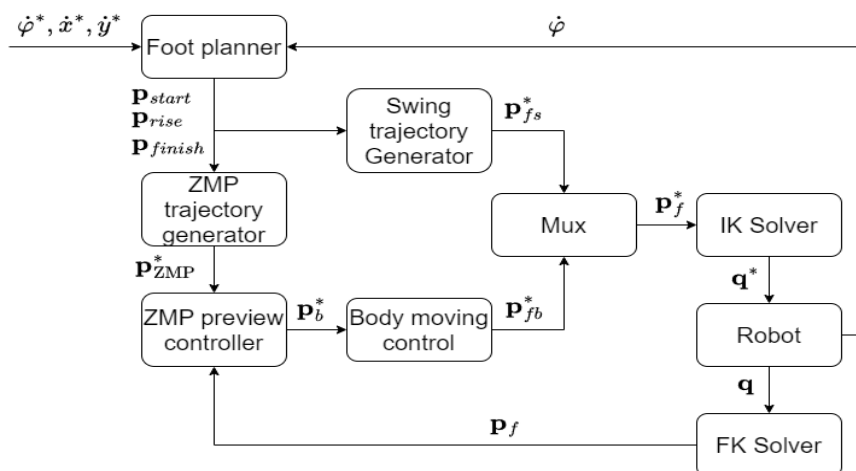


Рис. 8. Структурная схема алгоритма управления локомоцией.

Стабилизация корпуса в горизонтальном положении осуществляется с помощью алгоритма управления точкой нулевого момента с прогнозированием (ZMP preview control) [33]. Этот алгоритм широко применяется для управления двуногими роботами [34–36]. Но для управления четырехногими роботами его почти никто не использовал. Одной из немногих реализаций указанного алгоритма является походка иноходью [37]. Эта походка выбрана в силу прямой аналогии формирования траектории для точки нулевого момента с двуногими роботами, так как в каждый момент времени в воздухе находятся только обе правые или обе левые ноги. В текущей работе решено реализовать управление с помощью этого алгоритма для походки рысью, применяя собственный способ формирования желаемой траектории для ZMP. Во время фазы переноса строится дугообразная траектория на основе опорных точек \mathbf{p}_{start} , \mathbf{p}_{rise} и \mathbf{p}_{finish} . На выходе алгоритм производит преобразование желаемых положений стоп и корпуса в углы поворота сочленений путем решения обратной задачи кинематики.

В сравнении с предыдущей версией алгоритма [23] неизменными остались блок обратной кинематики и метод расчета следовых точек. За одним лишь исключением, что теперь на каждой итерации производится расчет массива из N_L будущих следовых точек при допущении, что желаемая скорость и направление движения останутся неизменными. Начало базовой системы координат робота находится в геометрическом центре корпуса. Ось Ox расположена вдоль продольной оси и направлена в сторону головы робота, ось Oz – вертикально вверх. Ось Oy выбрана так, чтобы система координат являлась правосторонней.

2.1. Формирование траектории стопы в фазе переноса. Формирование траектории движения стопы робота в фазе переноса относительно геометрического центра корпуса в трехмерном пространстве осуществляется с помощью блока Swing Trajectory Generator. В этом блоке сначала рассчитываются координаты опорных точек, а затем строится плавная траектория, соединяющая эти точки.

Расчет опорных точек производится на основании входной информации о высоте шага S_H , длине шага S_L , частоте шага ν и желаемых скоростях v_x^*, v_y^*, ω_z^* . Сначала рассмотрим расчет опорных точек для прямолинейного движения:

$$\mathbf{p}_{start1} = \begin{bmatrix} -\frac{S_L}{2} & 0 & 0 \end{bmatrix}^T, \quad (2.1)$$

$$\mathbf{p}_{finish1} = \begin{bmatrix} \frac{S_L}{2} & 0 & 0 \end{bmatrix}^T, \quad (2.2)$$

$$\mathbf{p}_{rise1} = \begin{bmatrix} 0 & 0 & S_H \end{bmatrix}^T, \quad (2.3)$$

где \mathbf{p}_{start1} — точка, в которой оканчивается фаза опоры и начинается фаза переноса, \mathbf{p}_{rise1} — точка наивысшего подъема ноги, $\mathbf{p}_{finish1}$ — конечная точка фазы переноса.

Затем добавим возможность линейного движения вдоль поперечной оси, дополнив эти опорные точки. Для этого определим угол ψ (рис. 9), равный арктангенсу отношений компонент желаемой скорости по осям Ox и Oy :

$$\psi = \text{atan} \frac{v_y^*}{v_x^*}.$$

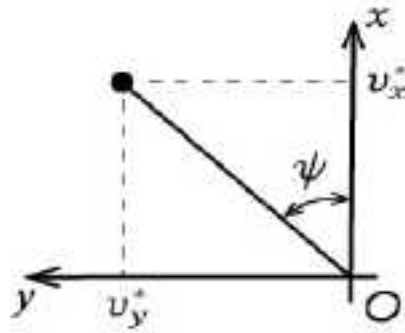


Рис. 9. Направление движения робота.

Сформируем поворотную матрицу для данного угла и умножим ее на первоначальные опорные точки (2.1)–(2.3):

$$\mathbf{p}_{start2} = \mathbf{R}_z(\psi) \mathbf{p}_{start1} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -\frac{S_L}{2} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -\frac{S_L}{2} \cos \psi \\ -\frac{S_L}{2} \sin \psi \\ 0 \end{bmatrix}.$$

Аналогично рассчитываются точки \mathbf{p}_{rise2} и $\mathbf{p}_{finish2}$.

Поворот вокруг вертикальной оси производится относительно определенного радиуса r . Для осуществления поворота необходимо представить положение ног, как показано на рис. 10. Модификация точек для поворота осуществляется с помощью следующих уравнений [38]:

$$\gamma = d \text{atan} 2(h, 2r + d b),$$

$$\xi = d \text{atan} 2(h, 2r - d b),$$

$$\mathbf{p}_{start3} = \begin{bmatrix} \cos \sigma_{\gamma\xi} & -\sin \sigma_{\gamma\xi} & 0 \\ \sin \sigma_{\gamma\xi} & \cos \sigma_{\gamma\xi} & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}_{start2},$$

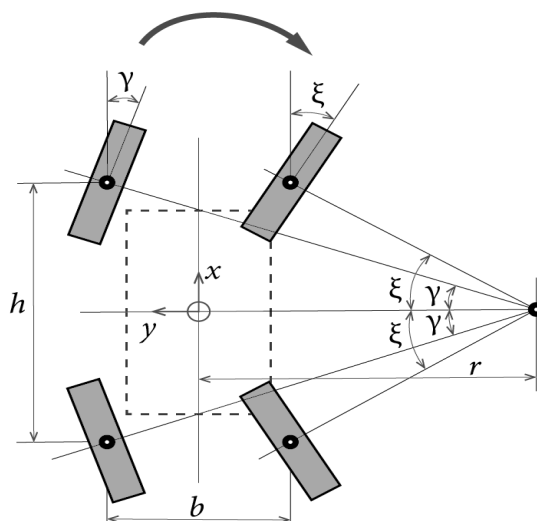


Рис. 10. Осуществление поворота робота.

где d – направление поворота (-1 – против часовой стрелки, 1 – по часовой стрелке), $\sigma_{\gamma\xi}$ – угол γ или ξ . Если нога правая, то $\sigma_{\gamma\xi} = \gamma$, если левая, то $\sigma_{\gamma\xi} = \xi$. Для точек \mathbf{p}_{rise2} и $\mathbf{p}_{finish2}$ расчет снова выполняется аналогичным образом. Финальным шагом является смещение каждого вектора в соответствии с базовым положением каждой стопы:

$$\mathbf{p}_{start} = \mathbf{p}_{start3} + \mathbf{p}_{idle},$$

$$\mathbf{p}_{rise} = \mathbf{p}_{rise3} + \mathbf{p}_{idle},$$

$$\mathbf{p}_{finish} = \mathbf{p}_{finish3} + \mathbf{p}_{idle},$$

где \mathbf{p}_{idle} – вектор базового положения стопы относительно центра масс тела робота, когда робот стоит.

Саму траекторию можно описать многочленом третьей степени $p(t)$, где t – время, пройденное с начала фазы переноса. Далее будем рассматривать построение траектории на примере одной оси координат. Расчет траектории для других двух ведется аналогичным образом. Сначала выберем ограничения, накладываемые на функцию $p(t)$:

$$t_f = \frac{1}{2\nu},$$

$$p(0) = p_0, \quad (2.4)$$

$$p(t_f) = p_f, \quad (2.5)$$

$$\dot{p}(0) = \vartheta, \quad (2.6)$$

$$\dot{p}(t_f) = 0, \quad (2.7)$$

где t_f – длительность фазы переноса, ν – частота шага, ϑ – линейная скорость конца стопы, $p_0 = p_{start}$ при подъеме ноги, $p_0 = p_{rise}$ при опускании, $p_f = p_{rise}$ при подъеме и $p_f = p_{finish}$ при опускании.

Уравнения (2.4) и (2.5) обусловлены выбором начальных и конечных следовых точек. Ограничение (2.6) необходимо для того, чтобы при резкой смене следовых точек функция не имела разрыва. Последнее ограничение (2.7) нужно для обеспечения плавного касания ноги с поверхностью в конце фазы переноса. Поскольку необходимо задавать положение и скорость

стопы робота во время движения, используется многочлен третьей степени для формирования траектории. Сам многочлен и его производная по времени имеют следующую форму:

$$p(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3, \quad (2.8)$$

$$\dot{p}(t) = a_1 + 2a_2 t + 3a_3 t^2, \quad (2.9)$$

где a_1, a_2, a_3 — искомые коэффициенты многочлена.

Чтобы получить коэффициенты a_i , совместим уравнения (2.8) и (2.9) с ограничениями (2.4)–(2.7):

$$p(0) = a_0, \quad (2.10)$$

$$p(t_f) = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3, \quad (2.11)$$

$$\dot{p}(0) = a_1, \quad (2.12)$$

$$0 = a_1 + 2a_2 t_f + 3a_3 t_f^2. \quad (2.13)$$

Далее решим полученную систему уравнений относительно коэффициентов a_i :

$$a_0 = p(0),$$

$$a_1 = \dot{p}(0),$$

$$a_2 = \frac{3}{t_f^2} (p(t_f) - p(0)) - \frac{2}{t_f} \dot{p}(0),$$

$$a_3 = -\frac{2}{t_f^3} (p(t_f) - p(0)) + \frac{1}{t_f^2} \dot{p}(0).$$

Имея уравнения (2.10)–(2.13) и зная коэффициенты a_i , можно построить генератор траектории, работающий в реальном времени. На каждом цикле работы алгоритма производится расчет точки $p(t)$ при помощи уравнений (2.10)–(2.13). Коэффициенты a_i рассчитываются в двух случаях: в самом начале фазы опоры и при изменении точки p_f в процессе работы. Вторым вариант необходим для плавного изменения траектории, когда оператор внезапно меняет скорость, направление движения или высоту шага.

На рис. 11 изображена траектория движения стопы в фазе переноса, формируемая построенным алгоритмом при движении робота вдоль продольной оси. Отличительной особенностью алгоритма является то, что он способен плавно менять формируемую траекторию при скачкообразном изменении значений высоты шага и вектора желаемой скорости. На рис. 12 показано, как будет меняться траектория при скачкообразной смене точки p_{finish} в разные моменты времени в проекции на ось Ox .

2.2. Модель робота. Локомоция может быть представлена как движение точек ZMP и центра масс (ЦМ) туловища. Во время быстрой ходьбы с постоянно меняющейся желаемой скоростью и направлением движения робот будет находиться в устойчивом положении, когда именно положение ZMP, а не ЦМ находится внутри опорного многоугольника. Если ZMP выйдет за его пределы, то робот потеряет равновесие. Траекторию ходьбы можно получить из желаемой ZMP путем вычисления допустимого положения ЦМ. Динамика туловища робота во время ходьбы может быть аппроксимирована с помощью модели тележки на столе. Она представляет собой тележку массой m , которая располагается на столе и способна свободно по нему двигаться.

Модель тележки на столе предполагает некоторые допущения и упрощения в ее динамике [33]:

- 1) масса точечная и сконцентрирована в геометрическом центре тележки;
- 2) ножка стола не имеет массы;
- 3) трение между столом и тележкой отсутствует.

Второе упрощение на первый взгляд сильно огрубляет систему, но, учитывая, что по большей части именно в корпусе сосредоточена вся масса робота, такое упрощение может быть допустимым.

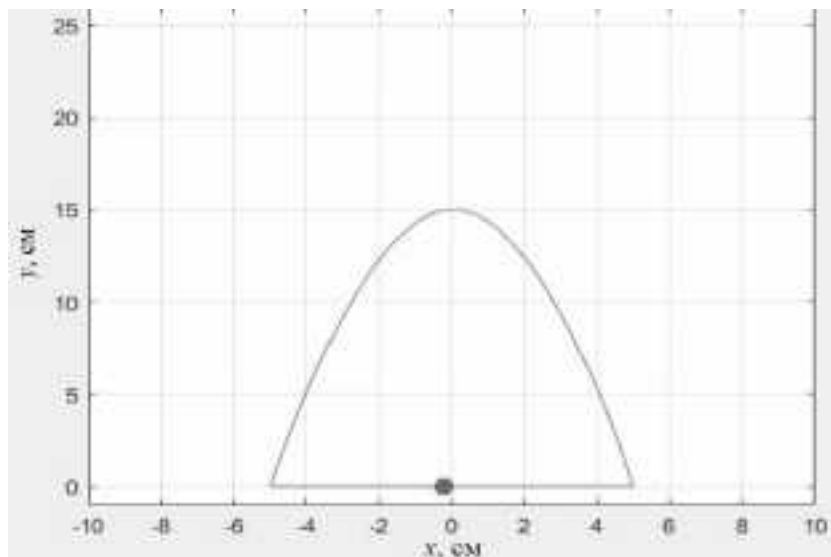


Рис. 11. Траектория стопы робота во время ходьбы в проекции на вертикальную плоскость Oxz .

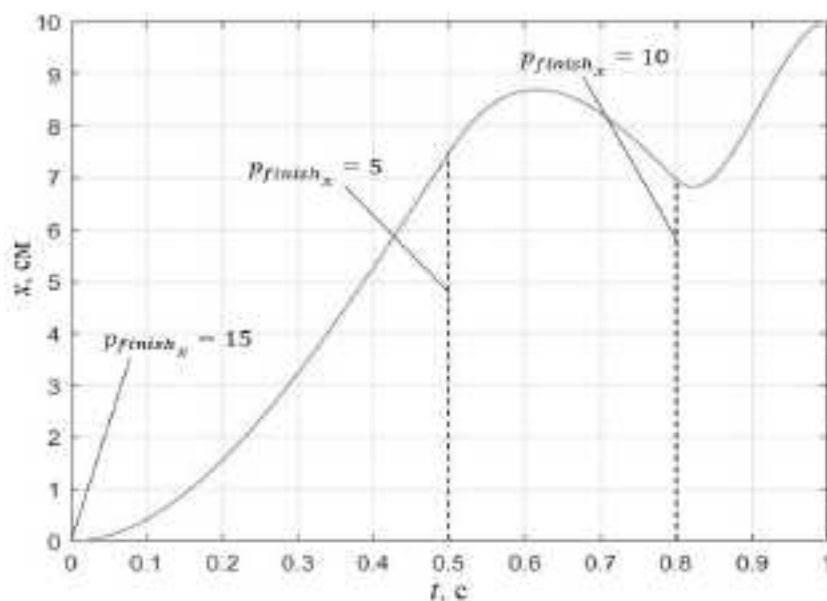


Рис. 12. Формирование плавной траектории стопы при скачкообразном изменении конечной точки p_{finish} в проекции на ось Ox . Пунктиром обозначены моменты времени, когда происходило изменение точки p_{finish} .

На рис. 13 графически представлено описание модели. Как можно заметить, основание ножки стола слишком мало, чтобы сохранить положение стола и тележки в равновесии. Однако если тележка будет иметь достаточно большое ускорение в определенном направлении, стол сможет оставаться в вертикальном положении. В момент равновесия ZMP как раз будет располагаться на горизонтальной проекции ножки стола.

Для робота, способного перемещаться вдоль продольной и поперечной осей, используются две модели тележки на столе: по модели на каждую ось. Уравнения динамики выглядят следующим образом:

$$\ddot{x} = \frac{g}{h}x - \frac{1}{mh}\tau_x, \quad (2.14)$$

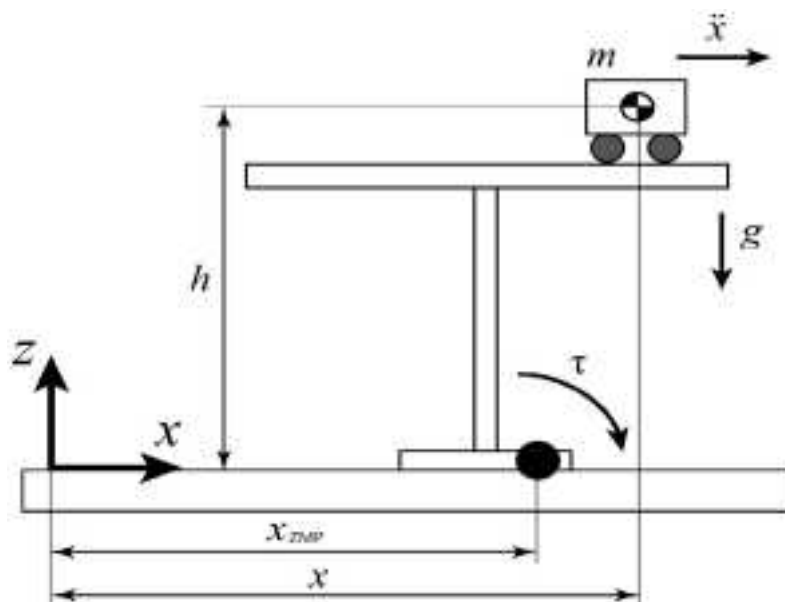


Рис. 13. Модель тележки на столе.

$$y = \frac{g}{h} y - \frac{1}{mh} \tau_y, \quad (2.15)$$

где m – масса тележки, g – ускорение свободного падения, τ_x и τ_y – моменты силы тяжести относительно ZMP вокруг осей Ox и Oy соответственно.

Пользуясь преобразованиями, описанными в работе [34], уравнения (2.14), (2.15) приобретают следующий вид:

$$\begin{aligned} x_{ZMP} &= x - \frac{h}{g} \ddot{x}, \\ y_{ZMP} &= y - \frac{h}{g} \ddot{y}, \end{aligned} \quad (2.16)$$

где x_{ZMP} и y_{ZMP} – координаты ZMP на опорной поверхности.

2.3. Формирование желаемой траектории ZMP. В настоящей работе отличительная особенность алгоритма управления ZMP preview control от других реализаций для четырехногих роботов заключается в формировании желаемой траектории ZMP.

На рис. 14 показано движение робота в проекции на горизонтальную плоскость при походе рысью. Серые окружности обозначают ноги в фазе опоры, они соединены черным

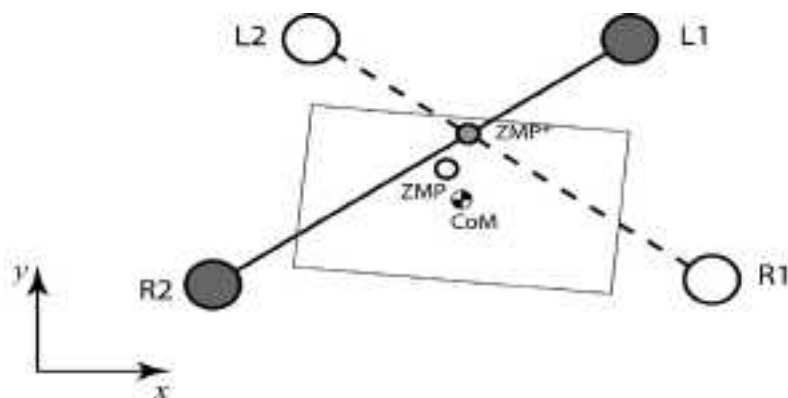


Рис. 14. Проекция робота на горизонтальную плоскость во время движения походкой рысь.

опорным отрезком. Белые окружности показывают ноги в фазе переноса, которые соединены пунктирным отрезком, выколота точка – текущее положение ZMP. На пересечении вышеописанных отрезков серая точка обозначает желаемое положение ZMP, а точка CoM – ЦМ корпуса робота. Во время ходьбы она постоянно перемещается вдоль текущего опорного отрезка. Если робот будет перемещаться вдоль продольной оси, то траектория ZMP примет зигзагообразную форму.

Таким образом, чтобы получить желаемое положение ZMP, нужно найти точку пересечения отрезков L2-R1 и L1-R2. Для этого рассмотрим систему уравнений двух прямых на плоскости:

$$\begin{cases} ax_{ZMP}^* + by_{ZMP}^* + c = 0, \\ dx_{ZMP}^* + ey_{ZMP}^* + f = 0, \end{cases} \quad (2.17)$$

где $a = y_{R1} - y_{L2}$, $b = x_{L2} - x_{R1}$, $c = x_{R1}y_{L2} - x_{L2}y_{R1}$, $d = y_{L1} - y_{R1}$, $e = x_{R2} - x_{L1}$, $f = x_{L1}y_{R2} - x_{R2}y_{L1}$.

Решив систему уравнений (2.17) относительно неизвестных x_{ZMP}^* , y_{ZMP}^* , получим выражения:

$$\begin{cases} x_{ZMP}^* = \frac{-by_{ZMP}^* - c}{a}, \\ y_{ZMP}^* = \frac{af - dc}{db - ae}. \end{cases}$$

В реальности для работы контроллера нужно вычислять координаты будущих положений ZMP необходимое для прогноза число раз, используя желаемые положения стоп робота. На рис. 15 показана формируемая траектория при движении вдоль продольной оси. На этом графике черные окружности показывают стопы ног в фазе опоры, серые – в фазе переноса, серые с пунктиром – прогнозируемые точки на следующем шаге. Толстая кривая обозначает уже построенную траекторию ZMP, а тонкая – прогнозируемую.

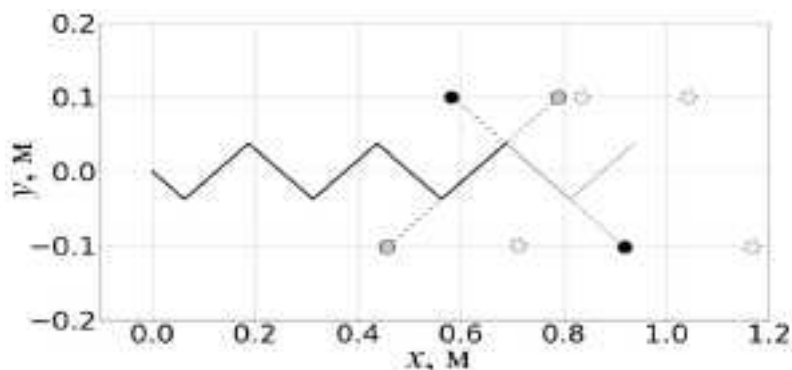


Рис. 15. Генерируемая желаемая траектория ZMP в проекции на горизонтальную плоскость.

2.4. ZMP preview control. Перед тем как синтезировать регулятор, представим динамическую модель тележки на столе в пространстве состояний. Здесь и далее преобразования ведутся на примере движения вдоль оси x . Введем прежде всего новую переменную u_x как производную от ускорения:

$$\frac{d}{dt} \ddot{x} = u_x.$$

Теперь запишем уравнение (2.29) в матричной форме:

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x, \quad (2.18)$$

$$x_{\text{ZMP}} = \begin{bmatrix} 1 & 0 & -\frac{h}{g} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix}. \quad (2.19)$$

Для координаты y_{ZMP} , система определяется аналогичным образом. Далее преобразуем систему уравнений (2.18) и (2.19) в дискретную форму:

$$X(k+1) = AX(k) + Bu(k), \quad (2.20)$$

$$p(k) = CX(k), \quad (2.21)$$

где

$$X(k) = \begin{bmatrix} x(k\Delta t) \\ \dot{x}(k\Delta t) \\ \ddot{x}(k\Delta t) \end{bmatrix}, p(k) = x_{\text{ZMP}}(k\Delta t), A = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} \frac{\Delta t^3}{6} \\ \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & -\frac{h}{g} \end{bmatrix}.$$

При помощи полученных выше уравнений (2.20) и (2.21) можно описать алгоритм управления, целью которого является достижение заданного положения ZMP с минимальной ошибкой. Алгоритм использует не только текущее состояние системы, но и прогнозируемую информацию о ее будущем состоянии. Впервые такой регулятор был описан в 1966 г. в работе [39]. Версия дискретного регулятора, решающего оптимальную линейно-квадратичную задачу и применяющего будущие состояния для ММО-систем (multiple input multiple output), реализована в 1985 г. [33].

Для реализации регулятора проведем дополнительное преобразование. Введем инкрементное состояние $\Delta X(k) = X(k) - X(k-1)$. Тогда состояние системы будет выглядеть следующим образом: $\tilde{X} = \begin{bmatrix} p(k) \\ \Delta X(k) \end{bmatrix}$, а уравнения динамики примут вид:

$$\tilde{X}(k+1) = \tilde{A}\tilde{X}(k) + \tilde{B}u(k),$$

$$p(k) = \tilde{C}\tilde{X}(k),$$

где $\tilde{A} = \begin{bmatrix} 1 & CA \\ 0 & A \end{bmatrix}$, $\tilde{B} = \begin{bmatrix} CB \\ B \end{bmatrix}$, $\tilde{C} = [1 \ 0 \ 0 \ 0]$.

Зная желаемое положение ZMP $p^*(i)$, выразим функционал качества для задачи оптимального управления следующим образом:

$$J = \sum_{i=k}^{\infty} Q_e (p^*(i) - p(i))^2 + \Delta X^T Q_x \Delta X + R \Delta u(i)^2, \quad (2.22)$$

где $p(i)$ – текущее положение ZMP, $p^*(i)$ – желаемое положение ZMP, Q_e , Q_x , R – симметричные неотрицательно определенные матрицы весовых коэффициентов размером 3×3 .

Структурная схема алгоритма управления представлена на рис. 16. Закон управления, минимизирующий функционал (2.22), запишем как:

$$u(k) = -K_e \sum_{i=0}^k (p^*(i) - p(i)) - K_x X(k) - \sum_{i=1}^{N_L} G(i) p^*(k+1), \quad (2.23)$$

где N_L – “дальность” прогнозирования будущей желаемой ZMP, K_e , K_x , $G(i)$ – весовые коэффициенты.

Согласно теореме 1 из работы [33], весовые коэффициенты K_e , K_x , $G(i)$ в законе управления (2.23) можно получить, решив дискретное алгебраическое уравнение Риккати относительно неотрицательно определенной матрицы \tilde{P} :

$$\tilde{P} = \tilde{A}^T \tilde{P} \tilde{A} - \tilde{A}^T \tilde{P} \tilde{B} (R + \tilde{B}^T \tilde{P} \tilde{B})^{-1} \tilde{B}^T \tilde{P} \tilde{A} + \tilde{Q}.$$

Тогда формулы для весовых коэффициентов K_e , K_x принимают вид:

$$\begin{bmatrix} K_e & K_x \end{bmatrix} = (R + \tilde{B}^T \tilde{P} \tilde{B})^{-1} \tilde{B}^T \tilde{P} \tilde{A} = \tilde{K}.$$

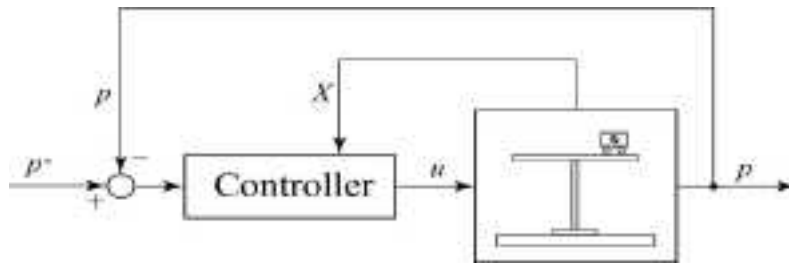


Рис. 16. Структурная схема системы управления ZMP preview control.

Коэффициенты $G(i)$ можно получить рекурсивно с помощью уравнений:

$$G(i) = (R + \tilde{B}^T \tilde{P} \tilde{B})^{-1} \tilde{B}^T \tilde{X}(i-1),$$

$$\tilde{X}(i) = (\tilde{A} - \tilde{B} \tilde{K})^T \tilde{X}(i-1).$$

Результат моделирования системы показан на рис. 17 для исходных значений $Q_e = 1, Q_x = 0, R = 10^{-6}, N_L = 160$ при частоте выполнения алгоритма 300 Гц (один временной шаг равен 0.00416 с). Значения прогнозирующего коэффициента $G(i)$, которые рассчитываются на N_L тактов алгоритма вперед, изображены на рис. 18. Можно наблюдать, что величина

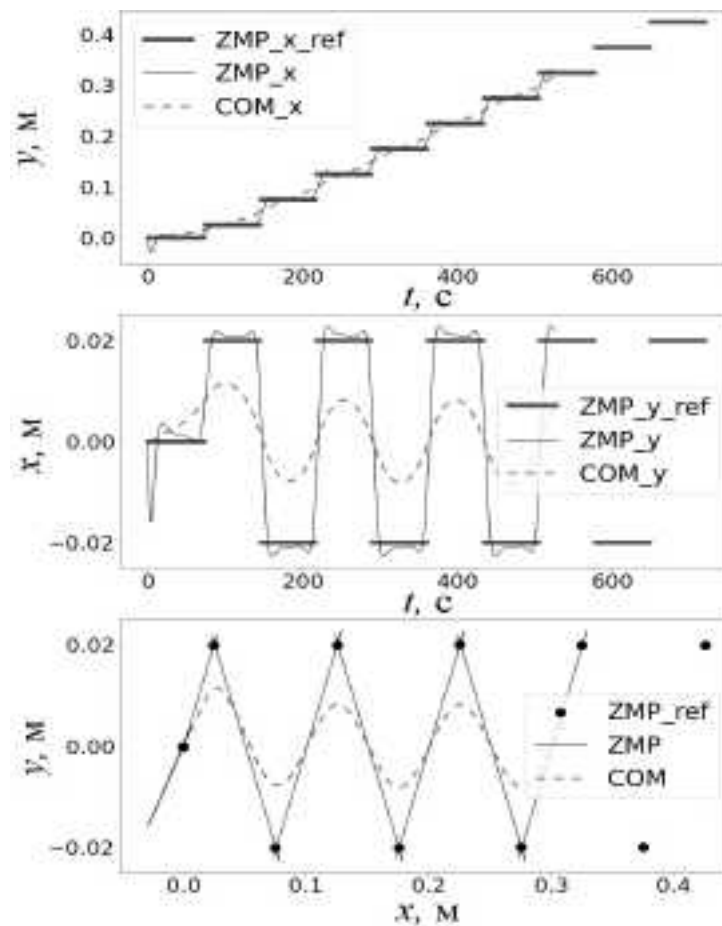


Рис. 17. Полученные в результате моделирования траектории ZMP и ЦМ для предполагаемого движения вдоль продольной оси.

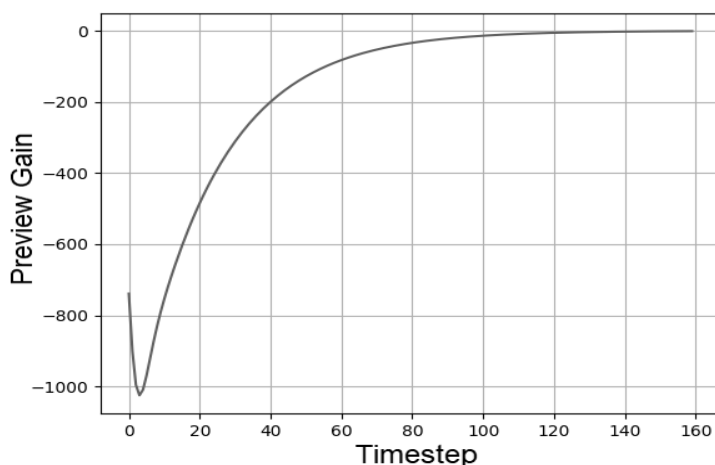


Рис. 18. Изменение коэффициента $G(i)$ в зависимости от прогнозируемого количества временных шагов.

коэффициента быстро стремится к нулю и уже на сотом шаге оказывается пренебрежительно малой. Это означает, что влияние прогнозирующей составляющей при $N_L > 100$ несущественно и для экономии вычислительных ресурсов необязательно выбирать большие значения N_L .

3. Результаты испытаний робота. Приводятся результаты испытаний робототехнической платформы с приведенным выше алгоритмом управления.

3.1. М о д е л и р о в а н и е. Перед проверкой алгоритма на аппаратной платформе авторы провели испытания в среде симуляции PyBullet [40]. Модель робота описана с помощью языка разметки URDF. На рис. 19 представлена ходьба робота при симуляции в среде PyBullet.

В течение всех испытаний робот перемещался по гладкой горизонтальной поверхности. Параметры алгоритма приведены в табл. 1.

Первый тест заключается в проверке способности алгоритма управления обеспечить устойчивое передвижение робота вдоль продольной оси. При скорости 0.4 м/с симуляция длилась 5 с. На рис. 20 изображена скорость ЦМ во время симуляции.

Второй тест связан с движением робота вдоль продольной оси с постоянно увеличивающейся скоростью от 0 до 2 м/с. Симуляция длится 15 с. Алгоритм показал свою способность

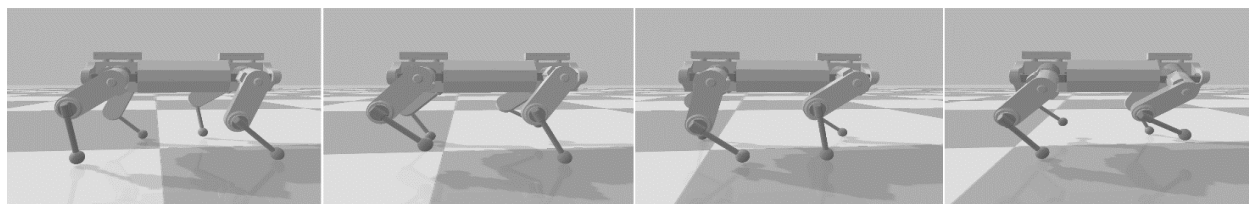


Рис. 19. Графическая визуализация в среде PyBullet ходьбы робота походкой рысь.

Таблица 1. Параметры алгоритма управления локомоцией

Параметр	Значение	Параметр	Значение
N_L	100	S_H , м	0.04
Q_x	0	h , м	0.19
Q_e	0.5	Период одной итерации алгоритма, с	0.0033
R	10^{-4}	Смещение ЦМ по Ox , м	-0.03
ν , Гц	3	Смещение ЦМ по Oy , м	0.0

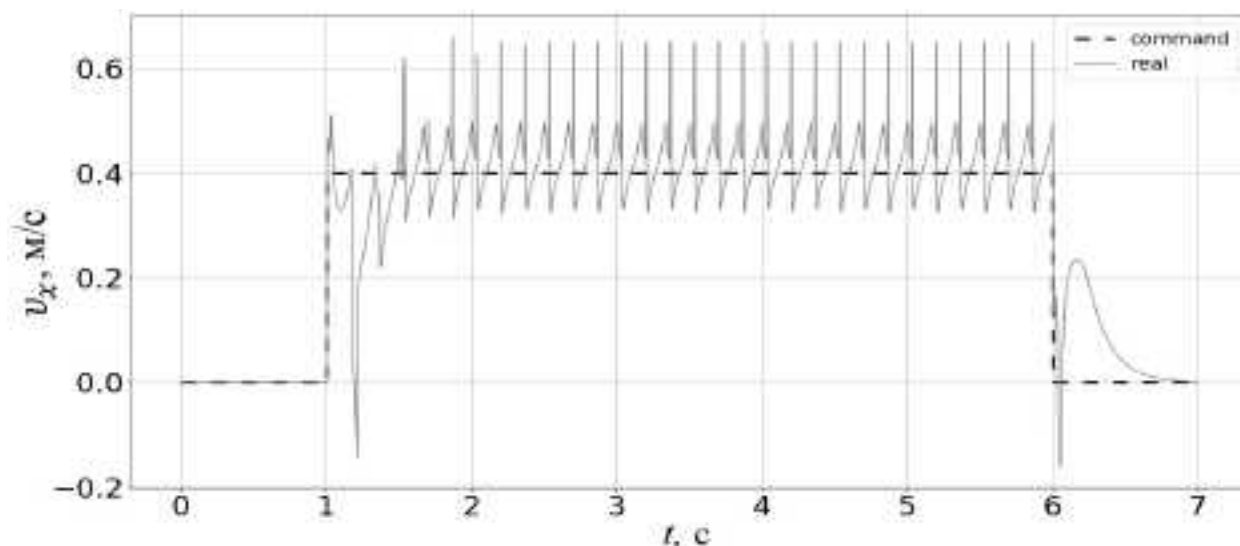


Рис. 20. Первый тест в виртуальной среде. Движение робота вдоль продольной оси с желаемой скоростью $v_x = 0.4$ м/с.

обеспечить локомоцию робота походкой рысь при сохранении равновесия в диапазоне скоростей от 0 до 1.2 м/с. Как видно на рис. 21, при желаемых скоростях выше, чем 1.2 м/с, робот начинает вести себя неустойчиво.

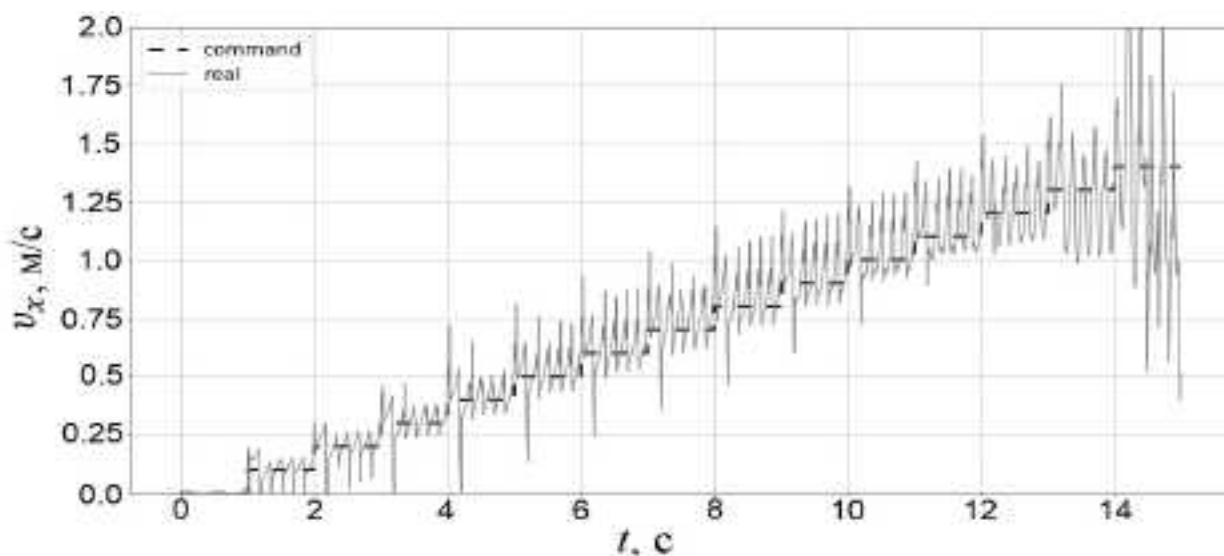


Рис. 21. Второй тест в виртуальной среде. Движение робота при постоянно увеличивающейся желаемой скорости v_x .

Целью третьего теста является проверка способности робота двигаться в разных направлениях вдоль осей Ox и Oy . Для этого формируется следующая траектория: робот движется вдоль оси Ox со скоростью 0.5 м/с в течение 10 с, затем вдоль оси Oy со скоростью 0.3 м/с в течение 10 с и в конце происходит движение назад по диагонали со скоростью $v_x = -0.5$ м/с, $v_y = -0.3$ м/с. На двух верхних графиках рис. 22 рассмотрены скорости робота v_x и v_y . На нижнем графике изображена траектория положения ЦМ робота, полученная в результате проверки.

Последний тест заключается в проверке способности робота поворачиваться вокруг вертикальной оси. Движение осуществляется в течение 5 с со скоростью $v_x = 0.5$ м/с с радиусом поворота 1 м. Рис. 23 демонстрирует результаты проведенного испытания.

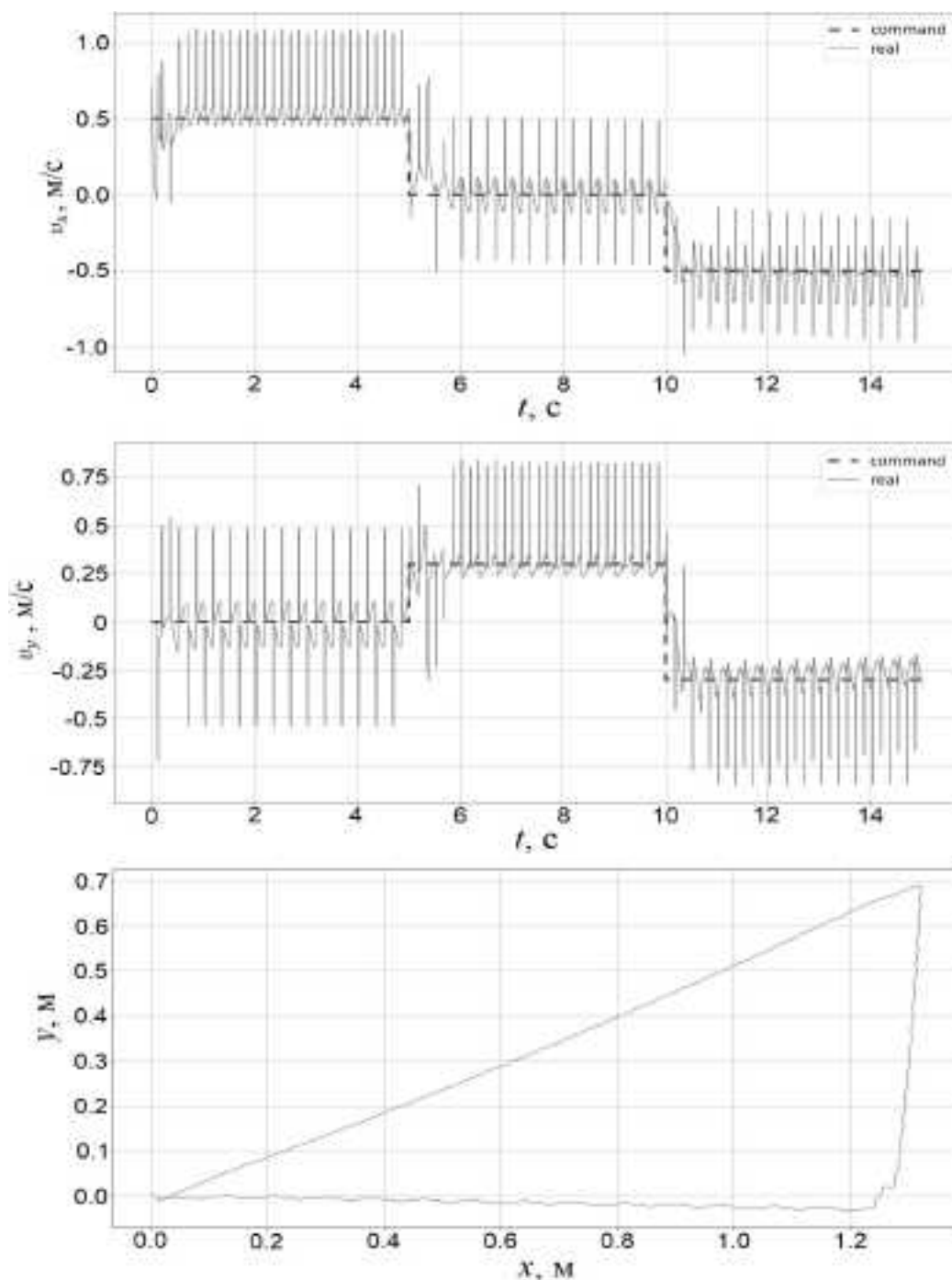


Рис. 22. Третий тест в виртуальной среде. Движение робота при меняющихся значениях желаемых скоростей v_x и v_y

Все испытания в симуляторе показали положительный результат. Во всех случаях робот двигался вдоль желаемого направления, сохраняя устойчивость.

3.2. Н а т у р н ы е и с п ы т а н и я. Во время тестов аппаратной платформы использовались те же параметры алгоритма, что и в тестах с имитационной моделью (табл. 1). Робот передвигался по плоской поверхности. Алгоритм ZMP preview control оказался способен справиться с задачей устойчивого перемещения робота для походки рысью: в каждый момент времени две диагонально противоположные ноги находятся в воздухе и корпус не теряет равновесие.

Так как у авторов нет возможности оценить положение робота в пространстве, то во время испытаний оценивалась только скорость перемещения робота. Поэтому на аппаратной

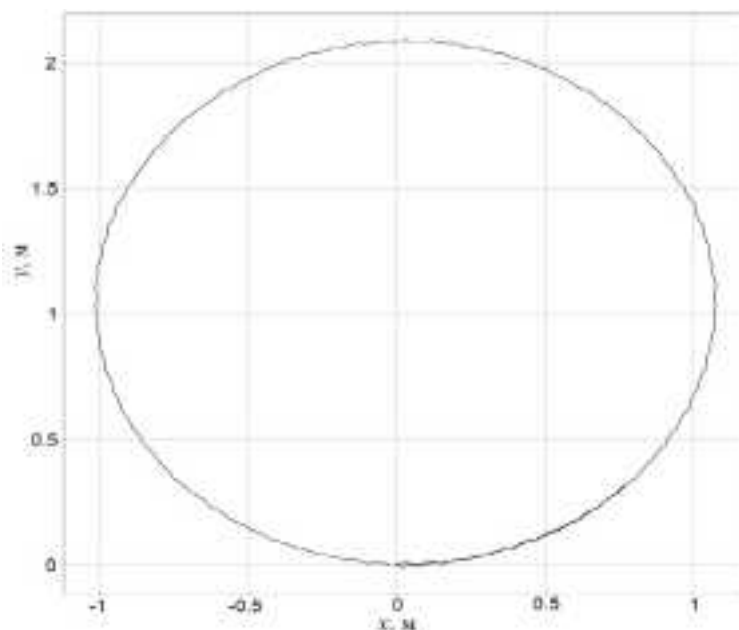


Рис. 23. Четвертый тест в виртуальной среде. Траектория ЦМ робота во время проверки способности робота поворачивать вокруг вертикальной оси.

платформе проводились первый и второй тесты из предыдущей главы. Оценка скорости производилась с помощью камеры Intel Realsense T265. Результаты можно видеть на рис. 24, 25.

Для оценки эффективности ходьбы применялись две метрики, широко используемые для оценки шагающих роботов: число Фруда (Fr) и энергетическая эффективность (cost of transport (CoT)). CoT – это безразмерная величина, показывающая, сколько энергии надо затратить для того, чтобы переместить 1 кг на 1 м [41]. Она находится по формуле:

$$CoT = \frac{UI}{mgv},$$

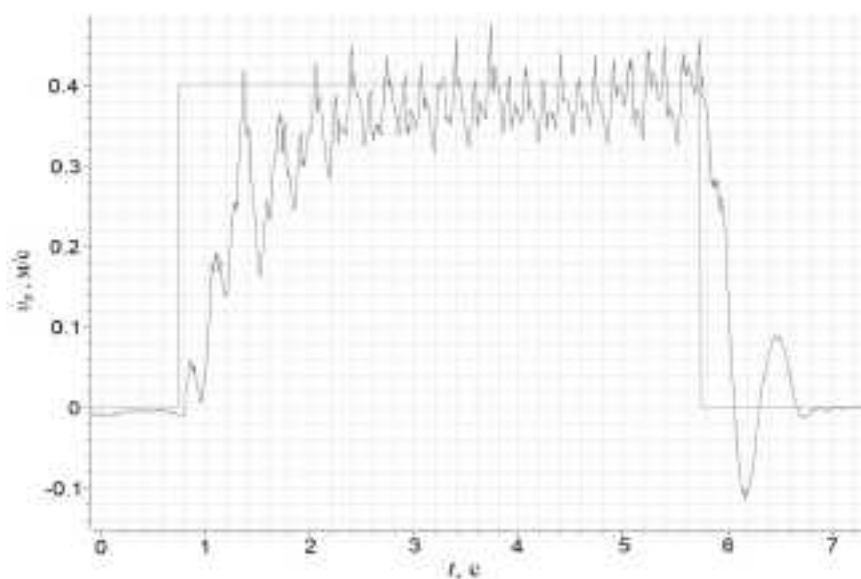


Рис. 24. Первый тест аппаратной платформы. Движение робота вдоль продольной оси с желаемой скоростью $v_x = 0.4$ м/с.

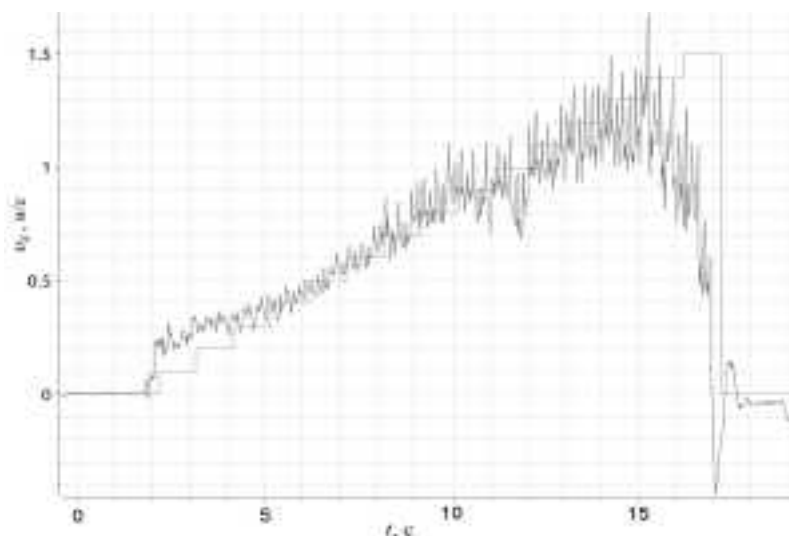


Рис. 25. Второй тест аппаратной платформы. Движение робота при постоянно увеличивающейся желаемой скорости v_x .

где U — напряжение батареи, I — потребляемый ток, m — масса робота, g — ускорение свободного падения, v — скорость робота.

Число Фруда — другая безразмерная величина, характеризующая динамическое подобие независимо от размера робота [6]:

$$Fr = \frac{v^2}{gL},$$

где L — длина ноги робота.

Дополнительно проходили испытания робота на грузоподъемность и время автономной работы. В процессе этих испытаний на спину робота устанавливались гири (рис. 26). Проводились две проверки: во время ходьбы и при стоянии на месте. Тест на время автономной работы выполнялся следующим образом: устанавливался полностью заряженный аккумуля-



Рис. 26. Робот во время испытаний на грузоподъемность.

тор, а затем робот двигался по кругу радиусом 2 м со скоростью 1 м/с до полной разрядки аккумулятора.

Робот успешно справился со всеми испытаниями. На графике результатов второго теста (рис. 25) можно заметить, что аппаратная платформа достигает такой же максимальной скорости, как и во время проверки в симуляторе (рис. 21). Отсюда можно сделать вывод о близости модели к реальному роботу. Итоговые результаты всех испытаний приведены в табл. 2.

Таблица 2. Результаты испытаний робота

Параметр	Значение
Максимальная скорость, м/с	1.2
Fr	0.52
CoT	1.3 при скорости 0.9 м/с
Максимальная грузоподъемность (ходьба), кг	5.1
Максимальная грузоподъемность (стояние), кг	7.0
Время автономной работы, мин	65

Заключение. Рассматривается новый четырехногий шагающий робот МОРС. Из всех существующих на данный момент — это самый легкий и малогабаритный шагающий робот на базе бесколлекторных моторов. Благодаря модульному принципу построения электроники, он прост в обслуживании и не требует специфических знаний при ремонте. В электроприводах применяется векторное управление для обеспечения плавности вращения и низкого энергопотребления. Конструкция ноги выбрана таким образом, чтобы можно было как менять кинематическую схему робота, так и использовать разные виды походок во время движения. Алгоритм управления ходьбой ZMP preview control адаптирован для организации передвижения четырехногих роботов. Он позволяет поддерживать равновесие корпуса при походке рысью, когда в опоре в каждый момент времени находятся лишь две ноги. Программное обеспечение систем высокого уровня имеет открытый исходный код. Разработанная платформа и ее открытый программный интерфейс позволяют изучать робототехнику и проводить дальнейшие исследования в направлении развития алгоритмов управления ходьбой, навигации и применения шагающих роботов в реальном мире.

Так как основным преимуществом шагающих машин является высокая проходимость, то построенный алгоритм управления предполагается усовершенствовать. Планируется включить методы оценки контакта с поверхностью, возможность преодоления препятствий, оценки положения и скорости робота в пространстве, возможность реагировать на внешние возмущения, а также двигаться более плавно и устойчиво, меняя походки в зависимости от типа поверхности и желаемой скорости передвижения. Авторы собираются улучшить конструкцию электроприводов, сделав их модульными, тем самым упростив обслуживание робота. Чтобы использовать алгоритмы управления, учитывающие полную динамику робота, требуется знать максимальное количество характеристик электродвигателя. Для решения этой проблемы планируется разработка стенда проверки и получения как статических, так и динамических характеристик двигателей.

СПИСОК ЛИТЕРАТУРЫ

1. Гурфинкель В.С., Гурфинкель Е.В., Девянин Е.А., Ефремов Е.В., Жихарев Д.Н., Ленский А.В., Шнейдер А.Ю., Штильман Л.Г. Макет шестиногого шагающего аппарата с супервизорным управлением // Исследование робототехнических систем. М.: Наука, 1981.
2. Гришин А.А., Житомирский С.В., Ленский А.В., Формальский А.М. Управление ходьбой двуногого пятизвенного механизма // Изв. АН. ТиСУ 1999. № 6. С. 142–152.
3. Охочимский Д.Е., Голубев Ю.Ф. Механика и управление движением автоматического шагающего аппарата. М.: Наука, 1984.
4. Raibert M.H., Tello E.R. Legged Robots that Balance. Cambridge: The MIT Press, 1986.
<https://doi.org/10.1109/MEX.1986.4307016>
5. Hirose S., Umetani Y. Some Consideration on a Feasible Walking Mechanism as a Terrain Vehicle // Proc. to 3rd RoManSy Sympos. Udine: Elsevier, 1978.

6. *Vaughan C.L., O'Malley M.J.* Froude and the Contribution of Naval Architecture to our Understanding of Bipedal Locomotion // *Gait & Posture*. 2005. № 21 (3). P. 350–362.
<https://doi.org/10.1016/j.gaitpost.2004.01.011>
7. *Margolis G., Yang G., Paigwar K., Chen T., Agrawal P.* Rapid Locomotion via Reinforcement Learning // *Intern. J. Robotics Research*. 2024. № 43 (4). P. 572–587.
<https://doi.org/10.1177/02783649231224053>
8. *Garsia G., Griffin R., Pratt J.* Time-Varying Model Predictive Control for Highly Dynamic Motions of Quadrupedal Robots // *Intern. Conf. on Robotics and Automation (IROS)*. IEEE Press, 2021.
<https://doi.org/10.1109/ICRA48506.2021.9561913>
9. *Park H.W., Wensing P.M., Kim S.* High-speed Bounding with the MIT Cheetah 2: Control Design and Experiments // *The Intern. J. Robotics Research*. 2017. № 36 (2). P. 167–192.
<https://doi.org/10.1177/0278364917694244>
10. *Semini C., Barasuol V., Focchi M., Boelens C., Emara M.E., Casella S. et al.* Brief Introduction to the Quadruped Robot HyQReal // *Intern. Conf. on Robotics and Intelligent Machines*. Rome: IRIM, 2019.
11. *Choi S., Ji G., Park J., Kim H., Mun J., Lee J.H., Hwangbo J.* Learning Quadrupedal Locomotion on Deformable Terrain // *Science Robotics*. 2023. V. 8. № 74.
<https://doi.org/10.1126/scirobotics.ade2256>
12. *Rudin N., Hoeller D., Bjelonic M., Hutter M.* Advanced Skills by Learning Locomotion and Local Navigation End-to-End // *Intern. Conf. on Robotics and Automation (IROS)*. IEEE Press, 2022.
<https://doi.org/10.1109/IROS47612.2022.9981198>
13. *Sleiman J.P., Farshidian F., Minniti M.V., Hutter M.* A Unified MPC Framework for Whole-Body Dynamic Locomotion // *IEEE Robotics and Automation Letters*. 2021. № 6. P. 4688–4695.
<https://doi.org/10.1109/LRA.2021.3068908>
14. *Bjelonic M., Grandia R., Harley O., Galliard C., Zimmermann S., Hutter M.* Whole-Body MPC and Online Gait Sequence Generation for Wheeled-Legged Robots // *Intern. Conf. on Robotics and Automation (IROS)*. IEEE Press, 2021.
<https://doi.org/10.1109/IROS51168.2021.9636371>
15. *Valsecchi G., Rudin N., Nachtigall L., Mayer K., Tischhauser F., Hutter M.* Barry: A High-Payload and Agile Quadruped Robot // *IEEE Robotics and Automation Letters*. 2023. № 8(11). P. 6939–6946.
<https://doi.org/10.1109/LRA.2023.3313923>
16. Робот Spot фирмы Boston Dynamics. URL: <https://bostondynamics.com/products/spot/>
17. Робот B2 фирмы Unitree. URL: <https://m.unitree.com/b2/>
18. Робот Cyberdog фирмы Xiaomi. Unitree. URL: <https://www.mi.com/global/discover/article?id=2069>
19. Deep Robotics: официальный сайт. URL: <https://www.deeprobotics.cn/en>
20. Документация робота МОРС. URL: <https://voltbro.gitbook.io/robot-sobaka-mors/>
21. *Kau N., Bowers S.* Stanford Pupper: A Low-Cost Agile Quadruped Robot for Benchmarking and Education // *ArXiv*. abs/2110.00736. 2021.
22. *Mudalige N.D.W., Zhura I., Babataev I., Nazarova E., Fedoseev A., Tsetserukou D.* HyperDog: An Open-Source Quadruped Robot Platform Based on ROS2 and Micro-ROS // *Intern. Conf. on Systems, Man, and Cybernetics (SMC)*. Prague, 2022. P. 436–441.
<http://doi.org/10.1109/SMC53654.2022.9945526>
23. *Danilov V., Diane S.* CPG-Based Gait Generator for a Quadruped Robot with Sidewalk and Turning Operations. *Robotics in Natural Settings* // *CLAWAR. Lecture Notes in Networks and Systems*. 2022. № 530. P. 276–288. 2023.
https://doi.org/10.1007/978-3-031-15226-9_27
24. *Katz B., Di Carlo J., Kim S.* Mini Cheetah: A Platform for Pushing the Limits of Dynamic Quadruped Control // *Intern. Conf. on Robotics and Automation (ICRA)*. Montreal: IEEE. 2019. P. 6295–6301.
<http://doi.org/10.1109/ICRA.2019.8793865>
25. *Katz B.G.* A Low Cost Modular Actuator for Dynamic Robots: Master Thesis. Massachusetts Institute of Technology, 2018.
26. Прокол информационного обмена OpenCyphal. Официальный сайт. URL: <https://opencyphal.org>
27. *Rekioua T., Tabar F.M., Le Doeuff R.* A New Approach for the Field-Oriented Control of Brushless, Synchronous, Permanent Magnet Machines // *Fourth Intern. Conf. on Power Electronics and Variable-Speed Drives*. № 324. London, 1990. P. 46–50.
28. *Bellini A., Bifaretti S., Costantini S.* A Digital Speed Filter for Motion Control Drives with a Low Resolution Position Encoder // *Automatika: Časopis za Automatiku, Mjerenje, Elektroniku, Računarstvo i Komunikacije*. Zagreb, 2003. V. 44. № 1–2. P. 67–74.
29. *Dunkels A.* Design and Implementation of the lwIP TCP/IP Stack. Stockholm: Swedish Institute of Computer Science, 2001. V. 2. № 77.
30. *Huang A.S., Olson E., Moore D.C.* LCM: Lightweight Communications and Marshalling // *IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems*. Taipei: IEEE, 2010. P. 4057–4062.
31. ПО Universal RC Joystick. URL: <https://github.com/Cleric-K/Universal-RC-Joystick>
32. *Quigley M., Gerkey B., Conley K., Faust J., Foote T., Leibs J. et al.* ROS: An Open-Source Robot Operating System // *ICRA Workshop on Open Source Software*. 2009. V. 3. № 3.2. P. 5.
33. *Katayama T., Ohki T., Inoue T., Kato T.* Design of an Optimal Controller for a Discrete-time System Subject to Previewable Demand // *Intern. J. Control*. 1985. V. 41. № 3. P. 677–699.

- <https://doi.org/10.1080/00207185089611561985>
34. *Kajita S., Kanehiro F., Kaneko K., Fujiwara K., Harada K., Yokoi K., Hirukawa H.* Biped Walking Pattern Generation by Using Preview Control of Zero-Moment Point // IEEE Intern. Conf. on Robotics and Automation. Taipei: 2003. V. 2. P. 1620–1626.
<https://doi.org/10.1109/ROBOT.2003.1241826>
 35. *Huang W., Chew C.M., Zheng Y., Hong G.S.* Pattern Generation for Bipedal Walking on Slopes and Stairs // 8th IEEE-RAS Intern. Conf. on Humanoid Robots. Daejeon: IEEE, 2008. P. 205–210.
<https://doi.org/10.1109/ICHR.2008.4755946>
 36. *Kovalev A., Pavliuk N., Krestovnikov K., Saveliev, A.I.* Generation of Walking Patterns for Biped Robots Based on Dynamics of 3D Linear Inverted Pendulum // Intern. Conf. on Interactive Collaborative Robotics. Istanbul: Springer International Publishing, 2019. P. 170–181.
<https://doi.org/10.1177/1729881417749672>
 37. *Akbas T., Eskimez S.E., Ozel S., Adak O.K., Fidan K.C., Erbaturo K.* Zero Moment Point Based Pace Reference Generation for Quadruped Robots via Preview Control // 12th IEEE Intern. Workshop on Advanced Motion Control (AMC). Sarajevo: IEEE Press, 2012. P. 1–7.
<https://doi.org/10.1109/AMC.2012.6197116>
 38. *Lee J.H., Park J.H.* Turning Control for Quadruped Robots in Trotting on Irregular Terrain // Proc. 18th Intern. Conf. on Advances in Robotics, Mechatronics and Circuits. Santorini, 2014. P. 303–308.
 39. *Sheridan T.B.* Three Models of Preview Control // IEEE Transactions on Human Factors in Electronics. 1966. № 2. P. 91–102.
<https://doi.org/10.1109/THFE.1966.232329>
 40. Физический симулятор PyBullet. Официальный сайт. URL: <http://pybullet.org/>
 41. *Nishii J.* An Analytical Estimation of the Energy Cost for Legged Locomotion // J. Theoretical Biology. 2006. V. 238. № 3. P. 636–645.
<https://doi.org/10.1016/j.jtbi.2005.06.027>