УДК 004.932

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ МОДУЛЬНЫХ СИСТЕМ УПРАВЛЕНИЯ РОБОТАМИ НА БАЗЕ ROBOT OPERATING SYSTEM

Фаткулин В.А. – студент 2 курса направления подготовки
“Программная инженерия”
Гагарин Ю.Е. – Научный руководитель, к.т.н., доцент, заведующий кафедрой
Калужский филиал Московского Государственного Технического
Университета имени Н.Э. Баумана
Россия, г. Калуга

Современные задачи робототехники требуют внимания к адаптивности, надёжности и масштабируемости программного обеспечения. В условиях динамически изменяющейся среды – будь то складские помещения, парки, улицы или производственные цеха – робот должен не только точно локализоваться и планировать движение, но и эффективно взаимодействовать с другими системами, включая сенсоры, управляющие модули и облачные сервисы. В этом контексте особую роль играет архитектура программной платформы, лежащей в основе управления.

Целью данной работы является анализ методов математического моделирования модульных систем управления роботами на базе ROS 2, а также выявление их преимуществ и ограничений при решении прикладных задач в науке, образовании и производстве. Особое внимание уделяется тому, как абстрактные математические представления – кинематические и динамические модели, алгоритмы фильтрации, оптимизации и предиктивного управления – реализуются в виде модульных компонентов экосистемы ROS 2 и как они влияют на эффективность и надёжность автономных систем.

Архитектурные особенности ROS. Одной из наиболее распространённых и развитых экосистем для разработки роботизированных систем является Robot Operating System (ROS). Версия ROS 1, появившаяся в конце 2000-х, стала стандартом в академической робототехнике благодаря открытости, модульности и большому сообществу разработчиков. Однако её архитектурные ограничения – отсутствие поддержки реального времени, отсутствие безопасности, сложности с распределённым развёртыванием и зависимость от централизованного управляющего узла – значительно сужали возможности промышленного применения.

Эти недостатки стали основным стимулом для создания ROS 2 – новой версии платформы, в которой принципиально переработаны механизмы коммуникации и управления. Ключевым нововведением стало использование стандарта Data Distribution Service (DDS) в качестве транспортного уровня. Это позволило внедрить гибкие настройки качества обслуживания (QoS). В совокупности эти изменения сделали ROS 2 пригодной не только для исследований

и образования, но и для внедрения в реальные промышленные и коммерческие решения [1].

ROS 2 построена на принципах модульности и распределенности. Каждая функция – будь то чтение данных с датчика, построение карты или генерация управляющих команд – реализуется в виде независимого узла, который может быть запущен на любом вычислительном устройстве в сети. Это решает проблему ROS 1, потому что теперь команды идут непосредственно в устройства, минуя центральный узел.

Основой коммуникации служит стандарт DDS, обеспечивающий надёжную передачу данных. Благодаря настройкам QoS, разработчик может задавать такие параметры, как надёжность доставки, длительность хранения сообщений, максимальное время доставки и другие. Критически важные команды (например, экстренная остановка) могут передаваться с гарантированной доставкой и минимальной задержкой, в то время как видеопоток может использовать упрощённый режим передачи, где важна скорость, а не полная целостность данных [1].

Также в ROS 2 реализована возможность композиции узлов – запуска нескольких логических компонентов в одном процессе. Это снижает накладные расходы на межпроцессное взаимодействие и особенно полезно при работе с большими объёмами сенсорных данных, например, в системах одновременной локализации и построения карты (SLAM) [2].

Математическое моделирование поведения роботов. Математическое моделирование в ROS 2 не сводится к написанию кода – оно реализуется через специализированные пакеты, приводящие сложные алгоритмы в виде готовых к использованию компонентов. Основу любого робота составляет его поведенческая модель, описывающая, как он перемещается в пространстве, как реагирует на управляющие воздействия и как взаимодействует с окружающей средой.

Особое значение в современных системах управления приобретает предиктивное управление на основе модели (Model Predictive Control, MPC). В отличие от классических регуляторов, MPC учитывает не только текущее состояние робота, но и прогноз его поведения на несколько шагов вперёд. При этом явно задаются ограничения – например: максимальные скорости, ускорения или зоны, которые нельзя посещать. Это делает управление более безопасным и эффективным, особенно в динамически изменяющейся среде.

Установка MPC в ROS 2 происходит через специализированные пакеты, которые принимают математическое описание модели робота и генерируют оптимальные управляющие сигналы. Такой подход особенно востребован в задачах, где требуется высокая точность маневрирования, например, в логистике или медицинской робототехнике [3].

Интеграция сенсоров и датчиков. Современные роботы оснащаются множеством сенсоров: лидарами, инерциальными измерительными блоками (IMU), камерами, GPS и ультразвуковыми датчиками. Каждый из них

предоставляет информацию о разных аспектах окружающей среды и состоянии робота, но с разной точностью, частотой и надёжностью.

Для объединения этих данных применяются методы мультисенсорного слияния. Наиболее распространёнными являются фильтры Калмана и их нелинейные модификации. Эти алгоритмы позволяют компенсировать недостатки отдельных сенсоров за счёт их взаимного усиления.

В ROS 2 такая интеграция реализована в пакете robot_localization, который позволяет одновременно обрабатывать данные с нескольких источников и гибко настраивать параметры погрешностей для каждого датчика [4].

Навигация и планирование траекторий. Современные платформы ROS 2 включают развитый фреймворк навигации, в частности Navigation2 (Nav2). В составе Nav2 доступны несколько алгоритмов локального планирования, каждый из которых основан на своём подходе к оптимизации движения:

1. DWA (Dynamic Window Approach) – хорошо понимает допустимую скорость, отлично определяет траекторию, но не всегда обеспечивает плавность движения.

2. TEB (Timed Elastic Band) – формулирует задачу как минимизацию «энергии» временной траектории с учётом препятствий, времени и динамических ограничений. Результат – плавные и предсказуемые движения.

3. Hybrid A* – расширяет классический алгоритм поиска пути за счёт учёта скоростных ограничений робота, что делает его особенно полезным для неповоротливых роботов.

Все эти методы опираются на математические модели, описывающие предпочтения системы через функции стоимости: предпочтение более коротким путям, избегание резких поворотов, минимизация времени в пути и т.д. Экспериментальные исследования, в том числе описанные в работах [4] и [5], показывают, что TEB и Hybrid A* лучше справляются с задачами в сложных средах, таких как склады или городские улицы.

Облачные вычисления и распределённое управление ROS 2 активно интегрируется с облачными платформами, что открывает возможности для облачной робототехники – концепции, впервые предложенной ещё в 2015 году [6]. Суть подхода заключается в переносе ресурсоёмких задач – таких как обучение нейросетей, построение глобальных карт или анализ больших объёмов данных – на удалённые серверы.

Такая архитектура позволяет использовать мощности облачных центров для обработки, в то время как на процессор робота ложатся простые задачи.

Методология эксперимента. Для оценки эффективности математического моделирования в ROS 2 была проведёна серия экспериментов в симуляционной среде. В качестве базовой платформы использовался робот TurtleBot3 в симуляторе Gazebo с поддержкой ROS 2 Jazzy, как описано в руководствах [5].

Эксперимент включал следующие этапы:

1. Создание модели робота: была разработана полная программная модель робота с описанием кинематики, динамики и сенсоров – лидара RPLIDAR и инерциального измерительного блока (IMU).

2. Настройка системы навигации: в составе Navigation2 были реализованы три сценария движения с использованием DWA, TEB и Hybrid A*.

3. Проведение испытаний: в контролируемой среде с динамическими и статическими препятствиями робот перемещался между точками.

4. Оценка результатов: учитывались время выполнения задачи, плавность траектории (по числу резких манёвров), точность локализации и общая длина пройденного пути.

5. Анализ и сравнение: полученные данные сопоставлялись с результатами, описанными в литературных источниках [2, 3].

Все компоненты системы запускались в изолированных контейнерах Docker для нескольких запусков. Коммуникация между узлами настраивалась с учётом требований к надёжности и задержкам – критически важные команды ставились первым приоритетом, потоковые данные – вторым.

Результаты показали, что алгоритмы TEB и MPC обеспечивают более плавные и энергоэффективные траектории, особенно в условиях ограниченного пространства. DWA продемонстрировал наилучшую скорость реакции, но чаще совершал резкие коррекции. Точность локализации с использованием комбинации лидара и IMU оказалась на уровне современных автономных систем.

ROS 2 представляет собой зрелую и гибкую платформу для разработки модульных систем управления роботами. Её архитектура, основанная на принципах распределённости, безопасности и гибкой настройки качества обслуживания, делает её пригодной как для научных исследований, так и для промышленного применения.

Математическое моделирование в рамках ROS 2 охватывает широкий спектр задач – от описания физического поведения робота до интеграции сенсорных данных и планирования траектории. Все эти компоненты реализуются в виде независимых, совместимых модулей, благодаря чему, систему можно запускать множество раз.

Таким образом, ROS 2 обеспечивает продвинутую систему для разработки автономных роботов, способных эффективно функционировать в условиях динамической среды, при правильном выборе сенсоров и алгоритмов.

Список литературы:

1. Macenski S., Foote T., Gerkey B., Lalancette C., Woodall W. Robot Operating System 2: Design, architecture, and uses in the wild // Science Robotics. – 2022. – Vol. 7, № 66. – P. eabm6074. – URL: https://www.science.org/doi/10.1126/scirobotics.abm6074

2. Steve Macenski. ROS 2 SLAM Toolbox: Official Repository and Tutorials [Электронный ресурс] // GitHub. – 2025. – URL: https://github.com/SteveMacenski/slam_toolbox (дата обращения: 15.11.2025).

3. Rapid Deployment of Model Predictive Control for Robotic Systems // IEEE Transactions on Robotics. – 2024. – Vol. 40. URL: https://ieeexplore.ieee.org/document/10505632

4. SLAM-based Robot Localization and Navigation Algorithm // Applied Water Science. – 2024. – Vol. 14, № 3. – P. 45–59. URL: https://link.springer.com/article/10.1007/s13201-024-02183-6

5. Addison A. Building a Map of the Environment Using SLAM – ROS 2 Jazzy [Электронный ресурс]. – 2025. – URL: https://automaticaddison.com/building-a-map-of-the-environment-using-slam-ros-2-jazzy (дата обращения: 17.11.2025).

6. Kehoe B., Patil S., Abbeel P., Goldberg K. Cloud Robotics and Automation // IEEE Transactions on Automation Science and Engineering. – 2015. – Vol. 12, № 2. URL: https://ieeexplore.ieee.org/document/7006734