## **4.3 Алгоритм программного обеспечения**

Алгоритм функционирования включает в себя иерархическую систему управления из 4 уровней функционирования робототехнической платформой. Тем самым подразделяется на:

1. интеллектуальный уровень;
2. стратегический уровень;
3. тактический уровень;
4. исполнительный уровень;
   1. мехатронная система.

Интеллектуальный уровень системы отвечает за стратегическое планирование и контроль оператором. С этого уровня идет постановка задачи и установки для функционирования платформы, проверка достигнутых результатов после выполнения задач, а также анализ собранных данных внешними системами.

Стратегический уровень, здесь происходит планирование и принятие решение, необходимых для выполнения задач, к задачам относится перемещение, а также одновременную локализацию и построение карты (SLAM). Задача перемещения включает в себя:

1. постановку конечной точки, к которой должен двигаться робот;
2. получение команды от устройства, исходящей от оператора или на основании работы SLAM;
3. построение глобальной карты местности;
4. использование глобальной траектории;

SLAM разделяется на следующие пункты:

1. построение глобальной карты местности;
2. сохранение карты местности;
3. фильтрация шумов на карте;
4. отправка результатов картографирования, для дальнейшего использования.

Тактический уровень, выполняет детальное планирование ближайшего окружения и отдает команды, направленные на взаимодействие с окружающей средой. Внутри тактического уровня протекают процессы:

1. создания детализированной карты, в зоне досягаемости основных датчиков;
2. сбора данных с таких датчиков как лидар и камера;
3. определения и отправки, управляющих команд на исполнительный уровень;
4. получения и обработки данных с исполнительного уровня.

Исполнительный уровень отвечает за выполнение команд и прямое управление движением, внутри этого уровня протекают такие процессы как:

1. сбора данных с оставшихся датчиков: энкодер, инерциальный датчик, датчик расстояния, дозиметр;
2. поддержания точного контроля над движением (ПИД-регулятор);
3. генерации управляющих сигналов, предназначенных для управления двигателями платформы (актуаторы системы);
4. транспортировки, собранных данных, на тактический и стратегический уровни;

Мехатронная система, несет ответственность за непосредственное выполнение команд через моторы, т.е. приводит платформу в движение.

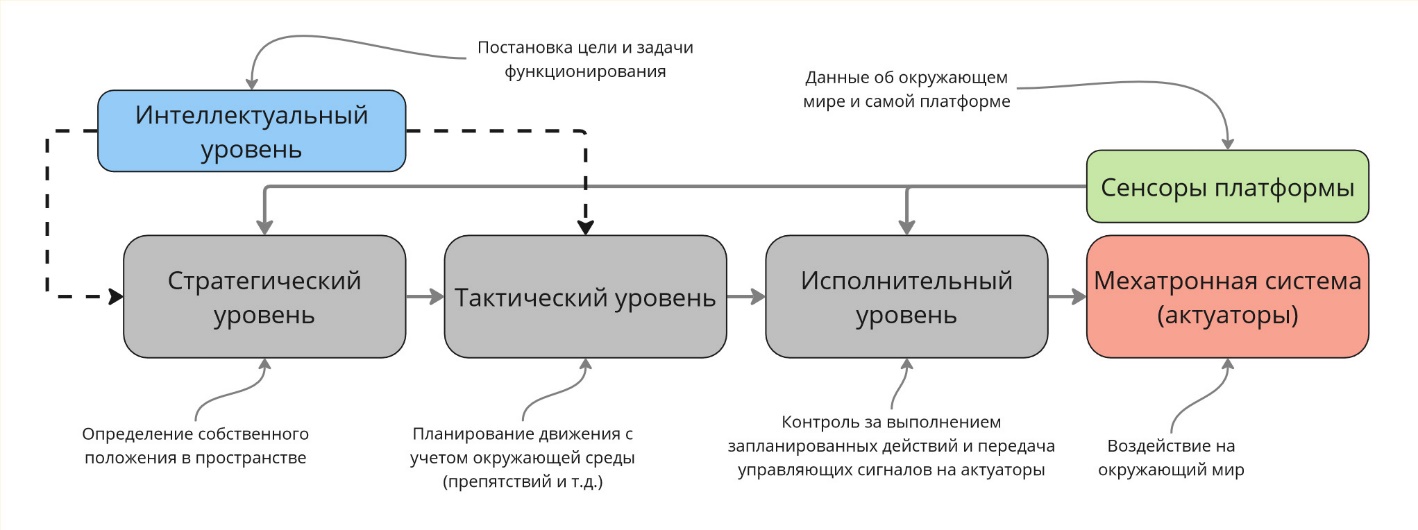


Рисунок 31–Упрощенная структура управления платформой

Иерархическая система управления позволяет робототехническому решению адаптироваться к изменениям окружающей среды. Версия структуры упрощенного управления системой, отображена на рисунке 31. Схема дает представление основных процессах работы платформы.

**4.3.1 Используемые методы в алгоритме функционирования устройства**

В качестве инструмента и основной рабочей средой был использован робототехнический фреймворк Robot Operating System (ROS). А именно 2-ую версию системы ROS2 Foxy, большого значения не имеет какой выпуск использовать ROS2, единственное чтобы поддерживался и был доступен для разработки.

Вполне очевидно, что большая часть робототехнических решений связанно с операционными системами (ОП), использующими ядро Linux, также и ROS полноценно функционирует на ОП Ubuntu, являющейся типичной средой для робототехника.

Коротко зачем и для чего нужен ROS. ROS (операционная система робота) предоставляет библиотеки и инструменты, помогающие разработчикам программного обеспечения создавать софт для роботов. Он обеспечивает абстракцию оборудования, драйверы устройств, библиотеки, визуализаторы, передачу сообщений, управление пакетами и многое другое, т.е. большое число полезных инструментов собрано в одной среде.

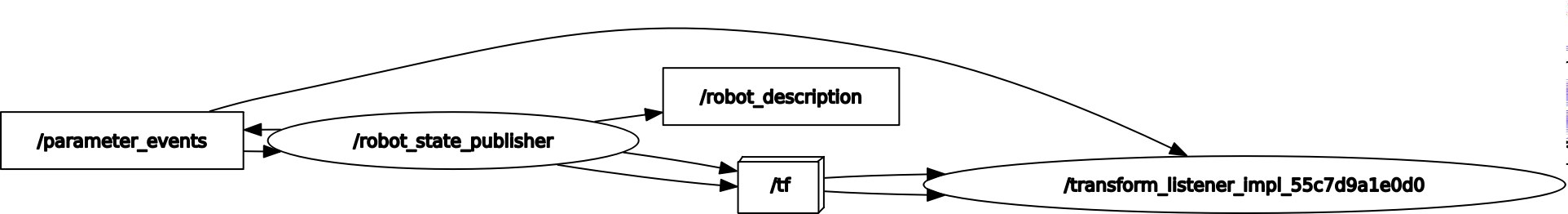
ROS2 построен на основе микросервисной архитектуры, позволяющей легко масштабировать системы и интегрировать различные компоненты. В ROS есть два основных компонента, которые напрямую будут использоваться при формировании иерархической системы управления платформой, это node и topics. Node являются основными рабочими единицами, представляющие в себе отдельные процессы, способные взаимодействовать между собой при помощи средств коммуникации ROS. Topics используются для обмена сообщениями между функционирующими node в асинхронном режиме т.е. не находиться в очереди выполнения задач. Каждый topic имеет свой уникальный тип и формат сообщений, что обеспечивает точность и целостность потока связи. ****

Рисунок 32 – Node и Topics

Рисунок 32 представляет схематичное изображение взаимодействия двух node (овальные фигуры) между собой, под средством существующих topics (прямоугольные фигуры).

### **4.3.2 Общий алгоритм функционирования программы**

Общий алгоритм функционирования MVP изображен на рисунке 33. Алгоритм состоит из нескольких этапов, начиная с процесса инициализации системы, заканчивая выполнением задач в выбранном режиме. Схема начинается с блока "Начало", который представляет собой стартовую точку процесса. Далее следуют этапы инициализации, включающие "Запуск отладчика" и "Запуск основной программы". Эти шаги предназначены для подготовки системы к работе, обеспечения ее корректной функциональности и готовности к выполнению команд.

После инициализации происходит "Определение режима управления", что является ключевым этапом. Если выбран режим ручного управления, система переходит к блоку "Ручное управление", который обеспечивает непосредственное управление роботом оператором. Этот режим предполагает полный контроль оператора над действиями робота. После выполнения задач система проверяет, достигнута ли последняя контрольная точка. Если контрольная точка достигнута, процесс завершается, иначе управление продолжается.

Режим ручного управления не выбран, система рассматривает возможность "Совместного управления", процесс управление роботом осуществляется совместно оператором и автоматической системой. В режиме телеуправление с поддержкой оператор управляет роботом с поддержкой автоматической системы, управление осуществляется посредством телеметрии и сенсоров, обеспечивая точное выполнение действий.

Иначе система переходит в режим автономного управления. В этом случае робот выполняет задачи самостоятельно, используя встроенные алгоритмы и сенсоры для ориентации в пространстве и принятия решений.

В каждом из режимов управления после выполнения задач система проверяет, достигнута ли последняя контрольная точка. Если контрольная точка достигнута, процесс завершается, переходя к блоку "Конец". Если нет, управление продолжается, возвращаясь к соответствующему режиму для выполнения дальнейших действий.

На данном этапе разработки реализовано лишь ручное управление платформой и производятся тесты телеуправления с поддержкой. Рисунок 34 отображает структуру алгоритма ручного управления, полностью зависит от наблюдателя, управляющие сигналы приходят лишь на исполнительный уровень и отвечают за выполнение действий актуаторами системы.

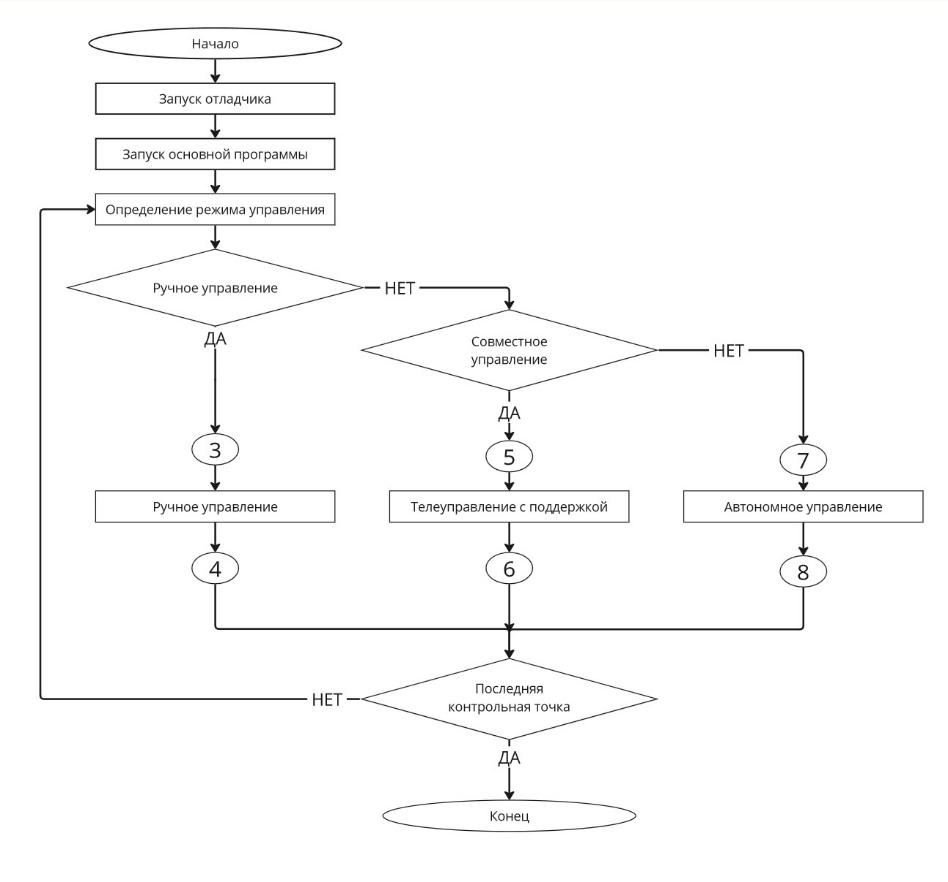


Рисунок 33–Алгоритм функционирования платформы

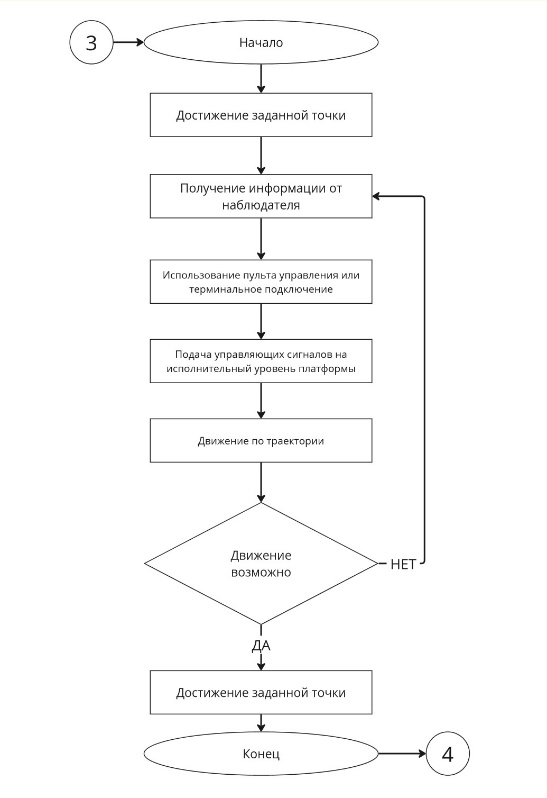


Рисунок 34 – Алгоритм платформы при ручном управлении

Алгоритм управления робототехнической платформой включает четыре уровня: интеллектуальный (планирование и контроль), стратегический (принятие решений и SLAM), тактический (детальное планирование и взаимодействие), исполнительный (выполнение команд). Использование ROS2 как основной среды разработки, ускоряет процесс и делает решение более универсальным. Общий алгоритм дает пояснение о процессе функционирования, а также дает представление о работе системы.