

**Отчет по проекту РНФ №20–79–10325 «Разработка принципов
и подходов к адаптивному управлению автономными
мобильными киберфизическими системами в условиях
изменяющегося окружения» за 2021–2022 гг.**

1. Разработка подхода к адаптивному управлению инфраструктурой киберфизической системы (КФС), сформированной на базе множества гетерогенных робототехнических средств (РС), ориентированного на обеспечение отказоустойчивости системы в изменяющейся среде, а также в случае утраты работоспособности отдельными РС

В рамках проекта при разработке подхода к адаптивному управлению инфраструктурой КФС, сформированной на базе множества гетерогенных РС, ориентированного на обеспечение отказоустойчивости системы в изменяющейся среде, а также в случае утраты работоспособности отдельными РС за основу были взяты представленные в [1, 2] подходы к построению архитектур КФС с использованием контейнеризации, способствующие упрощению процесса распределения инфраструктурных ролей и поддержанию работоспособности КФС.

Базовыми объектами инфраструктуры КФС в рамках настоящего проекта выступает множество гетерогенных РС наземного и воздушного базирования. При разработке предлагаемого подхода в качестве основных типов роботизированных агентов, входящих в состав КФС, рассматривались: маломобильные наземные РС, обладающие физическими хранилищами данных большего объема и значительными вычислительными ресурсами (далее – «мобильные роботизированные станции хранения и обработки данных», «МРСХОД»); БВС, оснащенные сетевым оборудованием с высокой пропускной способностью и характеризующиеся значительным объемом бортовых хранилищ данных (далее – «беспилотные воздушные хранилища данных», «БВХД»); наземные и воздушные РС специализированного назначения, ориентированные на непосредственную реализацию прикладных сервисов КФС (далее – «исполнительные РС»). Важно отметить, что указанная специфика роботизированных агентов носит нестрогий характер, поскольку при практической реализации КФС, как правило, преобладают РС смешанного типа, характеризующиеся более универсальным спектром возможностей.

С учетом требований к гибкости, масштабируемости, децентрализованности и отказоустойчивости КФС, для обеспечения адаптивного управления инфраструктурой системы необходимо активно

задействовать инструменты виртуализации и контейнеризации программных модулей и прикладных сервисов КФС. На рис. 1 представлена обобщенная архитектура КФС в разрезе применения средств виртуализации и контейнеризации, основанная на разработанных на предыдущем этапе реализации проекта методах автономного распределения инфраструктурных ролей между компонентами киберфизической системы в условиях изменяющегося окружения. Предложенная архитектура позволяет адаптивно осуществлять перераспределение и переназначение инфраструктурных ролей между компонентами КФС, представленными гетерогенными РС.

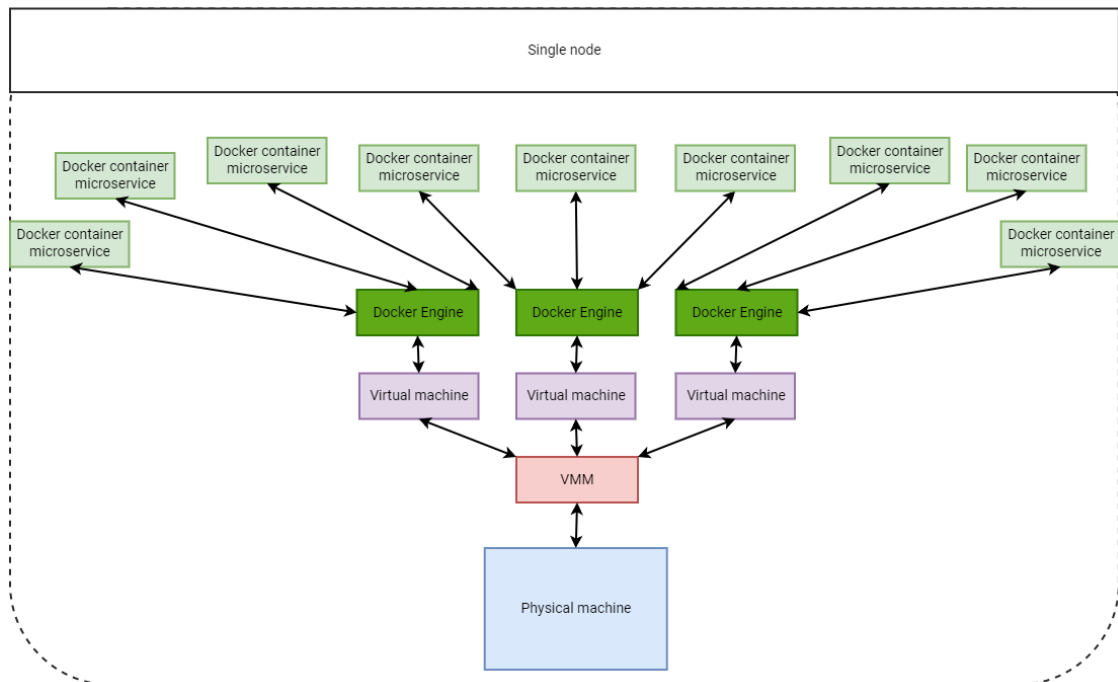


Рисунок 1 – Обобщенная архитектура КФС в разрезе применения средств виртуализации и контейнеризации

Согласно представленной архитектуре, на этапе первичной конфигурации программной составляющей каждого РС (Physical machine), являющегося компонентов КФС, выполняется установка ОС и менеджера виртуальных машин (VMM). Выбор VM-образа для каждого экземпляра виртуальной машины происходит в зависимости от инфраструктурной роли роботизированного агента, а также поставленных перед РС задач в контексте реализации прикладных сервисов КФС. Благодаря комбинации инструментов виртуализации и контейнеризации удастся добиться большей отказоустойчивости размещенных на РС микросервисов, а также избавиться от ряда возможных коллизий и конфликтов имен в случаях, когда необходимо использовать несколько реплик конкретного микросервиса в рамках одного РС [3]. На уровне виртуальной машины мониторинг может осуществляться средствами VMM.

Таким образом, в случае потери работоспособности некоторым исполнительным РС, соответствующая инфраструктурная роль может быть незамедлительно переназначена другому агенту посредством передачи ему идентичного VM-образа и развертывания данного образа на ОС такого агента. По результатам последовательной инициализации Docker-контейнеров, размещенных в VM-образе, и заложенных в них программных модулей, корректное функционирование прикладных сервисов КФС может быть оперативно восстановлено. Представленная выше архитектура отражает принятую парадигму распределения и переназначения прикладных задач между компонентами КФС, однако лишь частично характеризует процессы перераспределения ролей хранения, обработки и передачи данных в рамках КФС.

Согласно разработанному на предыдущем этапе реализации проекта методу автономного распределения инфраструктурных ролей между компонентами КФС, передача, хранение, репликация и обмен данными в рамках КФС следует реализовывать на базе открытых фреймворков и технологий больших данных вроде Hadoop и Spark. В таком случае отказоустойчивость хранилища данных при потере работоспособности узлов данных (data node) обеспечивается непосредственно на уровне технологии хранения данных, а устойчивость системы по отношению к сбою управляющего узла хранилища – узла имён (name node) обеспечивается использованием вторичного узла имён (Secondary Name Node, SNN) [4]. SNN периодически объединяет файлы журнала изменений, и не только обеспечивает возможность восстановления главного узла, но и позволяет оптимизировать использование ресурсов. При этом SNN не является репликой главного узла имён и не предназначен для выполнения его операций, поэтому для обеспечения устойчивости системы хранения роль как главного, так и вторичного узла имён дублируется несколькими компонентами КФС. Важно отметить, что при назначении инфраструктурных ролей, роли главных и вторичных узлов имен следует распределять преимущественно между БВХД. Размещение Docker-контейнеров, содержащих распределенные обработчики данных, в свою очередь, следует осуществлять преимущественно на базе МРСХОД.

Таким образом, предложенная парадигма организации подсистем хранения и обработки данных позволяет реализовать адаптивное переназначение соответствующих инфраструктурных ролей средствами непосредственно самой технологии хранения и обработки данных в том числе для случаев отказа отдельных узлов, контейнеров и/или компонентов КФС и обеспечить высокую устойчивость КФС к нарушениям целостности данных.

Минимизацию рисков потери связности между компонентами системы в процессе функционирования КФС в рамках проекта предлагается обеспечить за счет применения беспроводных сетей с ячеистой топологией на базе радиомодулей, установленных на БпЛА, для организации устойчивой системы передачи управляющих команд дальнего действия. Разработанный ранее алгоритм организации соответствующей сети передачи данных в условиях неизвестного окружения с использованием технологии LoRa представлен в работе [5]. Такая сеть не предназначена для передачи больших объемов данных и главным образом служит для доставки управляющих команд компонентам КФС, а также выступает средством реализации управления инфраструктурой КФС при организации на базе компонентов КФС локальных сетей передачи данных с высокой производительностью. Высокоскоростную передачу данных между компонентами КФС в рамках проекта предлагается обеспечивать за счет применения технологий локальных беспроводных mesh-сетей [6, 7].

Таким образом, в проекте были предложены решения по применению средств виртуализации и контейнеризации, по организации подсистем хранения и обработки данных, а также сетей передачи данных в рамках КФС, базирующиеся на использовании разработанных ранее методов автономного распределения инфраструктурных ролей между компонентами КФС. Предложенные решения позволяют архитектурно разделить прикладные задачи компонентов КФС и роли хранения, обработки и передачи данных, и, соответственно, осуществлять перераспределение и переназначение инфраструктурных ролей между компонентами КФС, представленными гетерогенными РС.

2. Разработка метода построения маршрутов движения группы РС и перемещения без коллизий членов группы по неоднородной местности

В рамках выполнения работ по проекту был разработан метод построения бесколлизийных маршрутов движения для перемещения группы РС по неоднородной местности. Были предложены решение задачи энергоэффективного планирования пути для группы наземных РС, движущихся по неровной поверхности, а также для группы воздушных РС.

Большинство существующих алгоритмов группового планирования пути для случая наземных РС не предназначены для навигации по неровной поверхности, поскольку не учитывают геометрические особенности ландшафта, что влияет как на проходимость, так и на скорость перемещения РС в среде. Для решения данной проблемы был разработан метод, основанный на комбинации разработанных членами коллектива алгоритма LRLHD-A*

(Local Roughness Local Height Difference A*) [8] и алгоритма планирования группового движения ORCA-3D (Optimal Reciprocal Collision Avoidance 3D) [9]. Исходными данными для предложенного метода группового планирования движения РС в трехмерной среде являются: карта высот, представленная в виде массива ячеек с определенным значением высот; начальные и конечные позиции РС; граничные значения параметров, характеризующих рельеф местности. Рассмотрим основные этапы работы разработанного метода (рис. 2).

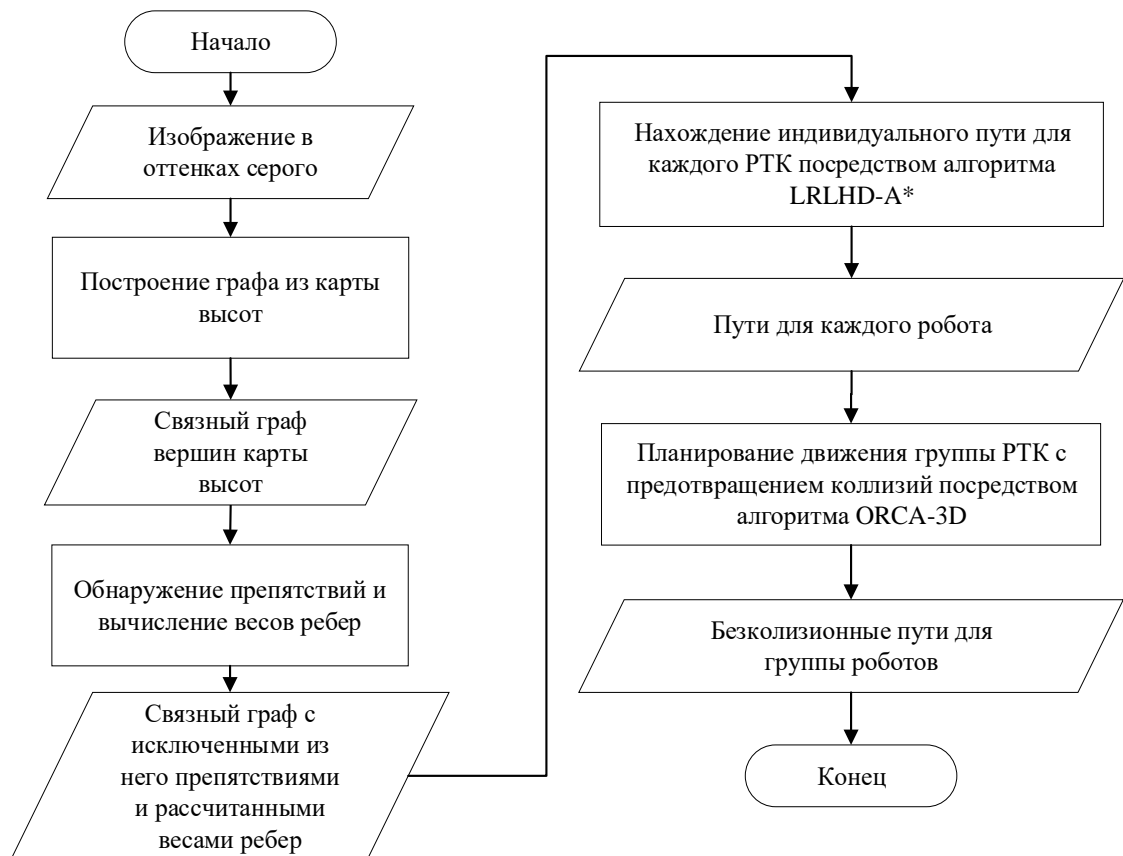


Рисунок 2 – Блок-схема алгоритма GPMUS

На начальном этапе из карты высот формируется связный граф. Карту высот необходимо представить в виде связного графа для дальнейшего планирования пути на ней. Для построения графа из карты высот, каждой ее ячейке присваивается список из восьми соседних вершин - две по горизонтали, две по вертикали и четыре по диагонали. Далее компонента связности будет использоваться при нахождении препятствий, а также в процессе построения маршрутов.

Следующим этапом алгоритма является обнаружение всех непроходимых для РС участков и вычисление весов ребер связного графа. Перед планированием пути необходимо выявить все участки местности, по которым движение РС невозможно - ямы, резкие подъемы в гору и прочие значительные неровности рельефа. Для этого необходимо вычислить

локальную неровность и максимальную локальную разницу высот для каждой вершины графа. Локальная неровность – это кривизна поверхности, вызванная перепадами высот. Неровность области $p(v)$ вокруг вершины v выражается в виде среднего угла наклона поверхности для ближайших вершин $N_{l(v)}$, находящихся на расстоянии меньшем, чем l , от вершины v . Угол между нормальными вершин рассчитывается по формуле $c_r^{(u,v)} = \arccos(n_{(u)} * n_{(v)})$, а средний угол $a_{av(v)}$ рассчитывается по формуле (1), где $N_{d(v)}$ – все непосредственные соседи вершины v .

$$a_{av(v)} = \frac{\sum_{u \in N_d(v)} c_r^{(u,v)}}{|N_{d(v)}|}, \quad (1)$$

$$p(v) = \frac{a_{av(v)} + \sum_{u \in N_{l(v)}} a_{av(u)}}{|N_{l(v)}| + 1}. \quad (2)$$

Для нахождения неровностей в рельефе (оврагов, канав и тд.) необходимо использовать оценку локальной разницы высот. Локальная разница высот $b(v)$ отражает максимальную разницу высот между вершиной v и каждой вершиной из $N_l(v)$, и рассчитывается по формуле (3), где $v(z)$ и $u(z)$ – значения z -координат вершин v и u соответственно.

$$b(v) = \max_{u \in N_l(v)} |v(z) - u(z)|, \quad (3)$$

Те вершины, у которых значение локальной неровности или локальной разности высот превышает определенный порог, помечаются как препятствия и исключаются из графа. Границы областей, в которых находятся препятствия, расширяются на расстояние, сопоставимое с размерами РС, чтобы все точки построенного пути были достаточно удалены от участков, опасных для движения РС. Алгоритм постепенной инфляции плавно снижает вес вершин, находящихся на расстоянии от r_i до r_0 от области, в которой находится препятствие. Параметр $r(v)$ отражает степень отдаленности вершины v от ближайшего препятствия и рассчитывается по формуле:

$$r(v) = c_{max} * \left(\cos \left(\frac{(d_{lethal} - r_i)}{(r_0 - r_i)} * \pi \right) + 1 \right), \quad (4)$$

где c_{max} – максимальная общая стоимость, а d_{lethal} – наименьшее расстояние до препятствия.

Для глобального планирования пути необходимо рассчитать конечные веса ребер. Конечный вес $c(u,v)$ для каждого ребра $e(u,v) \in E$ рассчитывается по формуле:

$$c^{(u,v)} = ((k_p * \max(p(u), p(v))) + 1) * \\ * \left((k_b * \max(b(u), b(v))) + 1 \right) * d^{(u,v)} + |r(u) - r(v)| \quad (5)$$

где k_p и k_b используются для определения влияния неровности p и локальной разницы высот b на изменение веса ребра, а $d(u,v)$ – евклидово расстояние между вершинами u и v .

После обнаружения препятствий и вычисления весов ребер графа, посредством алгоритма LRLHD-A* рассчитываются пути РС. Алгоритм LRLHD-A* основан на алгоритме A* и предназначен для планирования путей движения РС на трехмерной карте, представленной в виде графа. Применение данного алгоритма позволяет строить проходимые траектории движения РС по неровной поверхности, которые будут необходимы для следующего этапа работы метода группового планирования движения.

Найденные маршруты передаются в модифицированный алгоритм ORCA, где данные в последствии анализируются на наличие пересечений между ними. Если пересечения, вызывающие столкновения, обнаруживаются, алгоритм перепланирует траектории так, чтобы этих пересечений не было. Входными данными для алгоритма ORCA-3D являются: количество РС, их радиус, скорость движения, стартовые позиции, ориентация в пространстве и траектория, представленная в виде путевых точек. Алгоритм ORCA-3D корректирует длину вектора скорости исходя из перепада высот между соседними вершинами пути.

Для апробации разработанного метода решения задачи бесколлизийного энергоэффективного планирования путей для группы наземных РС выбран симулятор Gazebo, в который была импортирована карта высот размером 256*256 вершин, где расстояние между соседними вершинами равно 0,5 м. Группа РС состоит из трех Pioneer3AT (рис. 3).

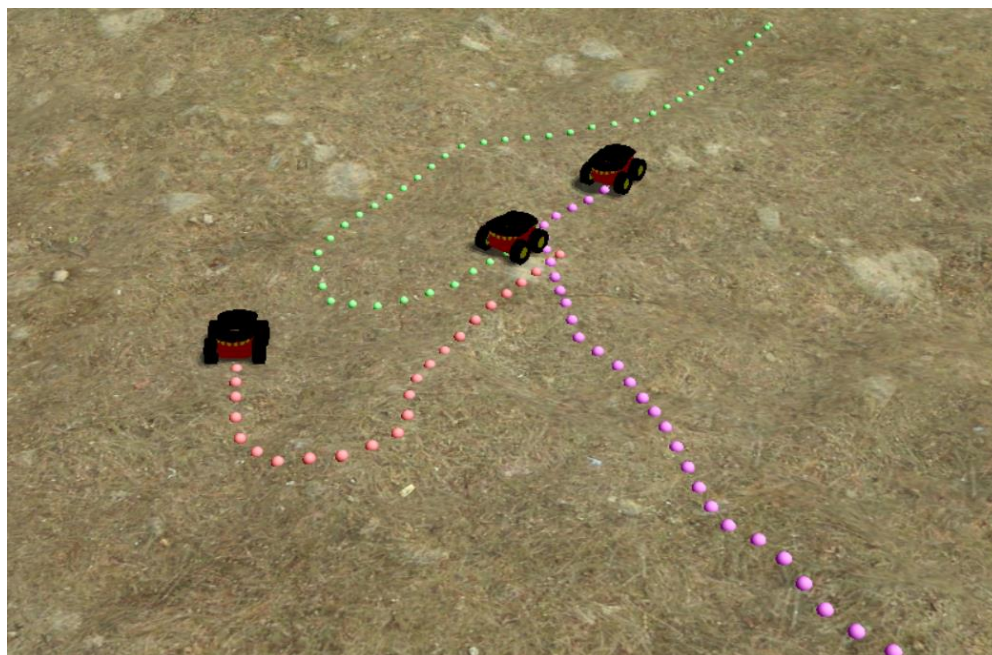


Рисунок 3 – Визуализация построенных путей в Gazebo

Как видно из рис. 4, среднее отклонение позиций роботов на каждом временном шаге от вершин путей, рассчитанных посредством метода с ORCA-3D, меньше в диапазоне от 10 до 57%, чем у путей, рассчитанных посредством метода с ORCA. Из этого следует, что ORCA-3D более точно рассчитывает позиции роботов на каждом временном шаге, чем ORCA, что обусловлено пересчетом длины вектора скорости исходя из перепада высот между соседними вершинами пути. Таким образом использование разработанного метода позволяет повысить качество планирования траекторий для группы наземных РС на неровной поверхности.

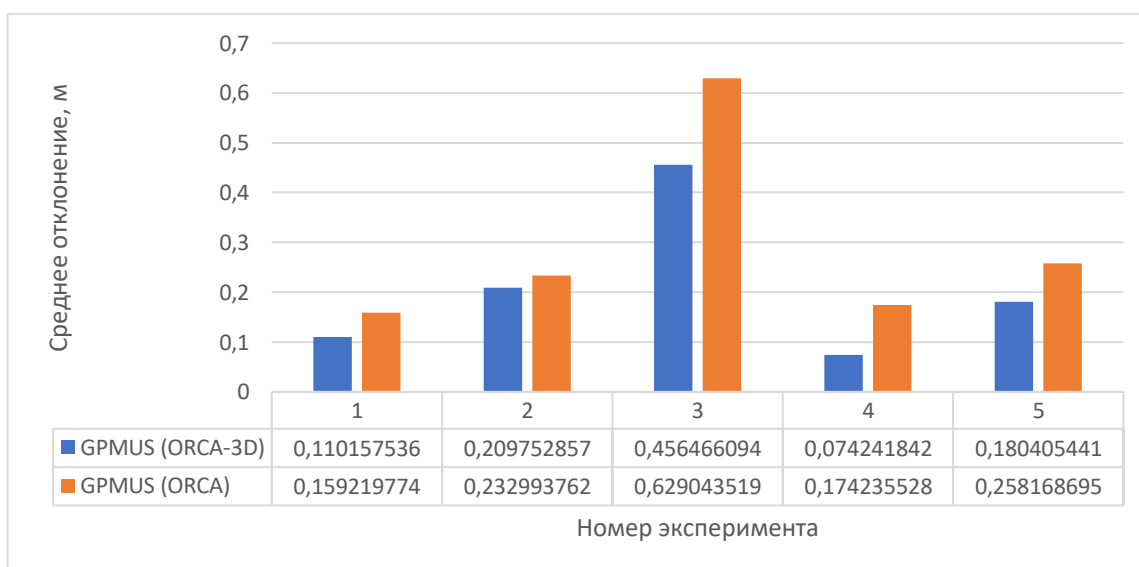


Рисунок 4 – Среднее отклонение роботов от маршрутов, рассчитанных посредством ORCA и ORCA-3D

В рамках проекта под воздушными РС понимаются малогабаритные беспилотные воздушные суда (БВС) мультироторного типа. Для решения задач проекта было принято решение выделить две группы траекторий движения воздушных РС в зависимости от размера обследуемой территории и поставленной миссии: траектория, покрывающая заданный участок (рис. 5а) и траектория движения к назначенной точке на карте (рис. 5б).

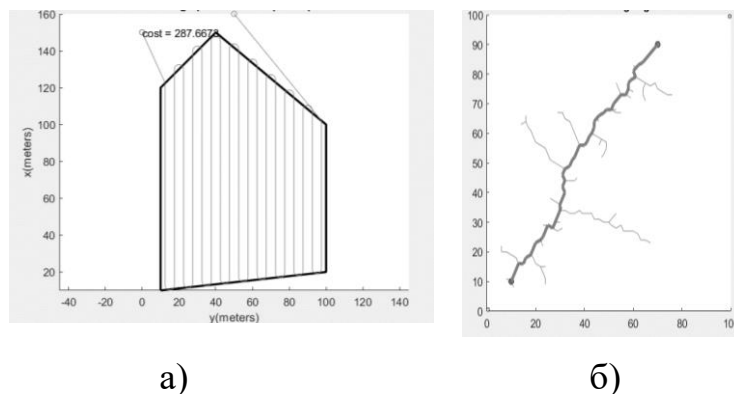


Рисунок 5 – Примеры траекторий для воздушных РС

При выполнении таких задач, как мониторинг местности, поиск объектов, построение карты для воздушных РС необходимо спланировать траекторию покрытия выделенной территории. В таком случае РС должен двигаться вдоль непрерывной, сглаженной траектории. Кроме того, РС должен избегать препятствия и участки земли, не представляющие интереса для решения поставленной задачи [10].

В рамках проекта для формирования указанных траекторий был модернизирован алгоритм, представленный в [11]. Выполненная модификация заключается в добавлении функции расчета оптимального направления развертки [12], которая сокращает количество маневров при движении РС. Также в процессе модернизации в алгоритм была интегрирована дополнительная функция сглаживания построенной траектории методом кубической интерполяции [13], что позволяет воздушному РС двигаться почти с постоянной скоростью. За счет вышеописанных нововведений увеличивается продолжительность полета БВС.

Для анализа энергоэффективности разработанной модификации алгоритма был проведен ряд экспериментов. В табл. 1 приведены результаты измерений длины траектории для двух участков экспериментов в трех случаях. Первый случай описывает построение траектории без дополнительного расчета направления оптимальной развертки. Второй случай – это построение траектории с оптимальным направлением развертки, но без сглаживания. Последний случай иллюстрирует длину итоговой траектории с рассчитанным направлением оптимальной развертки и сглаживанием.

Таблица 1. Результаты экспериментов

	Случай 1	Случай 2	Случай 3
Показатель: длина траектории			
№ участка	Длина траектории без расчета направления оптимальной развертки, м	Длина траектории без сглаживания, м	Длина траектории с оптимальным направлением развертки и сглаживанием, м
1	2720	2680	2490
2	9850	9531	9300
Показатель: время полета			
№ участка	Время полета по траектории без расчета направления оптимальной развертки, с	Время полета по траектории без сглаживания, с	Время полета по траектории с оптимальным направлением развертки и сглаживанием, с
1	7 мин 52 с	6 мин 24 с	5 мин 20 с
2	15 мин 51 с	15 мин 6 с	14 мин 12 с
Показатель: процент заряда после полета вдоль построенной траектории			
№ участка	Процент заряда аккумулятора после полета по траектории без расчета направления оптимальной развертки, %	Процент заряда аккумулятора после полета по траектории без сглаживания, %	Процент заряда аккумулятора после полета по траектории с оптимальным направлением развертки и сглаживанием, %
1	82	84	88
2	26	30	32

Для оценки результатов экспериментов было выбрано три параметра: длина траектории, время полета вдоль траектории и процент заряда аккумулятора в конце полетной миссии. Сравнивая результаты экспериментов, удалось установить, что, используя разработанный алгоритм, длина траектории на первом участке сократилась на 9%, а на втором на 6%, по сравнению с длиной траектории, построенной без расчета оптимального направления развертки и сглаживания. Также, используя улучшенный алгоритм построения траектории покрытия, время полета на первом участке сократилось на 30%, а на втором – на 10%. Экономия заряда аккумулятора после полета на двух участках составила порядка 6% от его максимального объема.

Траектория к назначенной точке используется в задачах транспортировки грузов, поиска объектов или аэрофотосъемки в заданной точке на местности. В данной формулировке задачи планирования пути для БВС цель состоит в том, чтобы найти надежный и оптимальный, относительно описанных ниже критериев путь от стартового воздушного положения до конечной точки. Критериями надежности и оптимальности пути являются: непрерывность найденного пути; отсутствие излишней извилистости пути; длина пути; время, затраченное на поиск пути.

В рамках проекта была произведена сравнительная оценка производительности наиболее часто используемых алгоритмов планирования пути в контексте двумерного планирования траектории БВС [14][15]. При проведении оценки исследуемыми параметрами являлись длина траектории и время ее вычисления. В случае БВС длина траектории преимущественно определяет и время прохождения пути. В динамичной среде время планирования пути имеет основополагающее значение. Анализ выбранных параметров важен для определения недостатков и целесообразности использования алгоритмов планирования траектории БВС.

Сравнительная оценка проводилась на рабочей области 100x100 метров. Задача планирования решалась на трёх картах: карта №1 - без препятствий, карта №2 - с двумя препятствиями, карта №3 - с пятью препятствиями. Препятствия на карте обозначены черными областями. Стартовая и целевая позиции зафиксированы во всех экспериментах. БВС мультиторного типа в экспериментах рассматривался как материальная точка, полученные результаты экспериментов сведены в табл. 2.

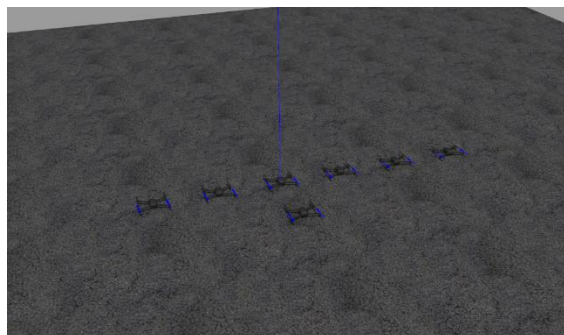
Таблица 2. Результаты экспериментов.

	Алгоритм A*		Алгоритм RRT		Алгоритм RRT*	
	Тср, с	Длина пути, м	Тср, с	Длина пути, м	Тср, с	Длина пути, м
Карта №1	1,76914	116,28280	5,28412	112,42420	6,65924	105,45960
Карта №2	1,33207	112,00000	3,79108	124,92140	7,66588	113,28050
Карта №3	1,84623	141,13400	21,37331	151,4581	21,71970	138,60260

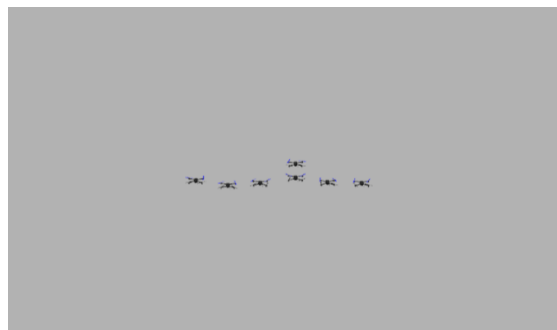
Согласно полученным данным, с помощью алгоритма A* время, затраченное на построение маршрутов, получилось меньше, чем при использовании RRT и RRT*. Оценивая длину полученного пути, можно отметить, что RRT* строил самые сглаженные пути, которые внешне похожи на реальные траектории БВС. А их длина была немного меньше длины пути построенной A*. Однако, A* исследовал только области, необходимые для строительства пути, в то время как RRT равномерно исследовал окружающую среду, что негативным образом сказывалось на скорости генерации путей. В среднем A* генерировал более короткие пути за меньшее время по сравнению с алгоритмом RRT. Разница во времени создания пути была уменьшена за счет устранения ограничений на размер шага, в то время как длина пути до сглаживания была лишь незначительно улучшена из-за зигзагообразного движения по пути с большей амплитудой.

Таким образом, высокая скорость работы алгоритма A* и низкая потребность в вычислительных ресурсах делают его ключевым кандидатом

для планирования двумерных траекторий БВС со статическими и динамическими препятствиями в режиме приближенном к реальному времени. Более того, A^* позволяет различным образом осуществлять декомпозицию рабочей среды в соответствии со специфическими требованиями в рамках различных сценариев функционирования БВС.



а)



б)

Рисунок 6 – Результат моделирования поставленной задачи в Gazebo:
а) группа БВС перед полетом; б) группа БВС в полёте до целевой точки

Симуляция группового полета по построенной алгоритмом A^* траектории проводилась в среде Gazebo (рис. 6) со стандартными моделями БВС типа квадрокоптер. Используемый БВС «iris» имеет следующие характеристики: габариты 440мм x 330мм x 100мм., вес 1.2 кг, максимальная скорость 18 м/с. Для целей моделирования скорость БВС была установлена 11 м/с. В результате серии из 50 экспериментов в среде Gazebo по построению траекторий БВС посредством алгоритма A^* удалось установить, что среднее время расчета траектории составляет 1.5 с, а построенные траектории были проходимы для РС. Это позволяет сделать вывод, что предлагаемое решение возможно использовать в режиме предполетного планирования траекторий. Включение кинематической и динамической модели БВС, ограничений и неопределенностей датчиков вместе с фиксированным и движущимся положением, скоростью и направлением препятствий позволило комплексно оценить производительность алгоритма A^* в рамках симуляционных экспериментов.

Для избегания столкновений между воздушными РС в рамках экспериментов использовался алгоритм разрешения столкновений DORCA (Directional optimal reciprocal collision avoidance), представленный в работе [16]. При сравнении результатов экспериментов, полученных для модернизированного алгоритма DORCA и исходного ORCA, с точки зрения общего количества отказов – DORCA характеризуется значительно более высокой устойчивостью и более низкой вероятностью отказов.

Таким образом, в рамках проекта был разработан и успешно апробирован метод построения бесколлизийных маршрутов для перемещения групп как наземных РС, так и групп БВС.

3. Разработка метода автономного распределения прикладных задач между множеством агентов, представленных гетерогенными РС на основе модернизации алгоритма PSO-based Dynamic Distributed Algorithm посредством использования подходов на основе клеточной декомпозиции в процессе распределения РС по кластерам

В данном разделе рассматривается решение задачи, где множество РС – R и множество задач, которые должны выполнить R – T . РС из R необходимо объединить в кластеры на основе стоимостей путей до ближайших задач из T , где каждое РС из R может быть назначено только одной задаче из T , а каждой задаче из T может быть назначено несколько РС из R . На основе данных, полученных посредством кластеризации, требуется найти такое распределение РС, которое обеспечило бы выполнение всех поставленных задач за минимально-возможное время.

Разработанный в рамках проекта метод автономного распределения прикладных задач между множеством агентов, представленных гетерогенными РС, включает в себя: алгоритм оценки стоимостей путей для каждого РС из R ; алгоритм кластеризации РС; алгоритм DDPSO для распределения РС по задачам.

В ходе выполнения работ был разработан новый алгоритм кластеризации, использующий в качестве метрики оценки стоимостей путей от текущего местоположения РС до точек выполнения задач. Алгоритм использует следующие входные данные: множество роботов R , множество задач T , множество оценок стоимостей путей $Pcosts$ для каждого робота из R к каждой задаче из T . Стоимость пути рассчитывается аналогично принципу, рассмотренному в работе [17]. Разработанный алгоритм итеративно перебирает различные комбинации наборов РС и задач, минимизируя сумму квадратов стоимостей пути от каждого РС в сформированном наборе до задачи, на которую данный набор РС будет назначен. Алгоритм стремится к созданию кластеров одинакового размера, не превышающего значения, определяемого по формуле (6). Критериями останова для итеративного процесса кластеризации являются: превышение числом итераций заданного значения I_{iter} ; отсутствие снижения функции ошибки в течение I_u итераций, где в качестве функции ошибки выступает сумма квадратов стоимостей путей РС, агрегированная по кластерам.

$$cl_{size} = \frac{|R|}{|T|} + 1, \quad (6)$$

где R – множество роботов, T – множество задач

По завершению итерационного процесса алгоритм возвращает текущий лучший список кластеров C_B . На рис. 7 представлена блок-схема разработанного алгоритма кластеризации.

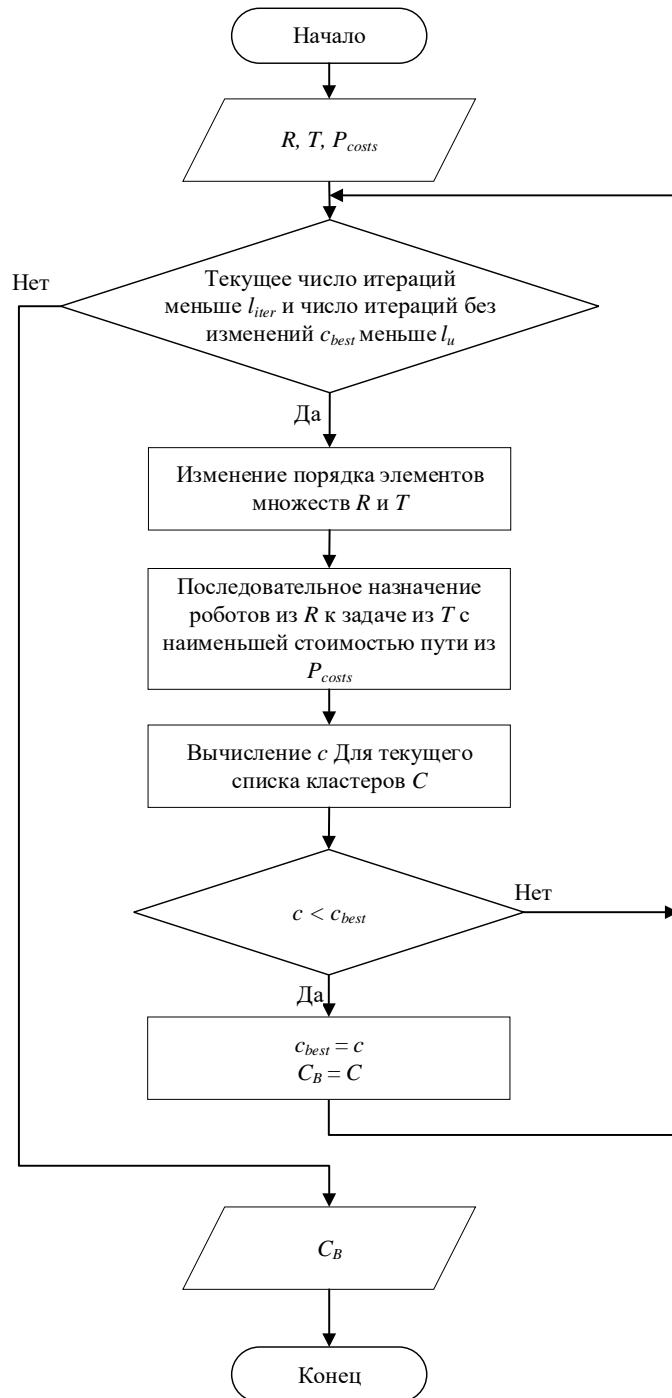


Рисунок 7 – Алгоритм кластеризации

По завершению кластеризации производится распределение прикладных задач, которое осуществляется с помощью алгоритма Dynamic Distributed Particle Swarm Optimization (DDPSO) [18]. В ходе выполнения проекта был задействован подход к использованию PSO для распределения задач между РС, представленный в [19]. Данный подход использует позицию частицы, как полноценное распределение задач. Количество измерений частицы равно $|T|$, где числовое значение в каждом измерении находится в диапазоне $[1, |R|]$. Порядковый номер измерения соответствует номеру задачи, а числовое значение координаты в данном измерении соответствует порядковому номеру робота, который может быть назначен на выполнение данной задачи. Значения координат строго целочисленные.

Алгоритм DDPSO нацелен на минимизацию стоимостей путей от РС из R до задач из набора T с целью достижения минимального времени выполнения прикладных задач, поставленных перед КФС. Распределение задач посредством DDPSO выполняется для каждого созданного на предыдущем этапе кластера.

4. Проектирование модели РС, позволяющей реализовать автоматизированную замену отдельных модулей РС посредством других роботизированных агентов

Для создания многофункциональных робототехнических платформ (МРП) актуальным является использование модульного принципа, который позволяет устанавливать различное навесное оборудование в зависимости от задач, поставленных перед РС. Для реализации автоматизированной замены модулей в ходе выполнения проекта было разработано унифицированное устройство сопряжения функциональных модулей с базовой робототехнической платформой, которое обеспечит механическую фиксацию, передачу энергии и информационный обмен. В работе [20] проведен анализ исследовательских работ в направлении решений для сопряжения модулей в модульных системах, отмечены их достоинства и недостатки. На основе проведенного анализа и собственных исследований разработано электромеханическое устройство сопряжения, конструктивно-кинематическая схема которого приведена на рисунке 8а.

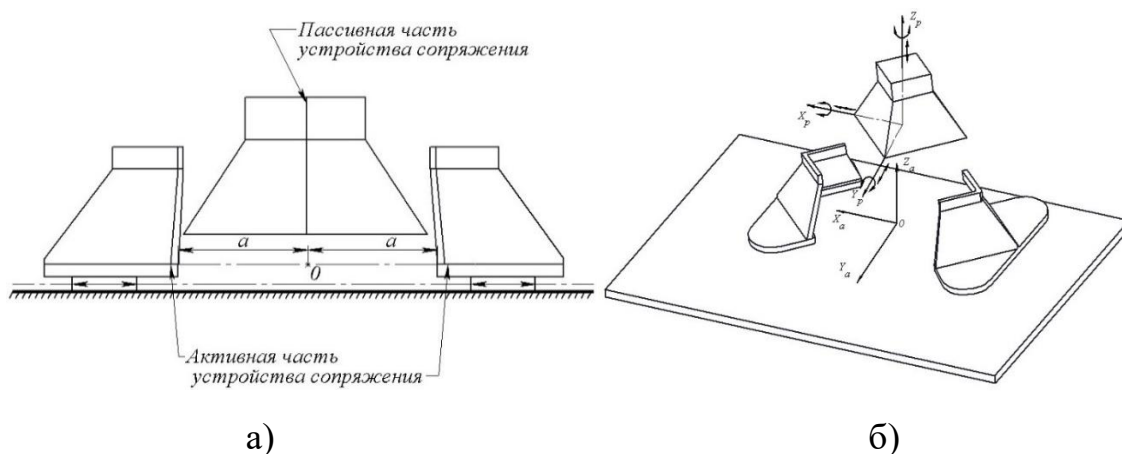


Рисунок 8 – Устройство сопряжения: а) конструктивно-кинематическая схема; б) привязка систем отсчета к пассивной и активной частям

На рисунке приведены две основные составляющие устройства: пассивная и активная части. Концепция устройства сопряжения основывается на принципе фиксации активной частью захватного механизма кинематически не связанной с ним пассивной части с возможностью контактной передачи энергии и данных. Целью сопряжения является такое физическое зацепление пассивной части, при котором система отсчета в точке Р (рисунок 8б), которая привязана к нижней грани пассивной части, совмещается с системой отсчета в точке О (рисунок 8б), привязанной к активной части. Для этого необходимо, чтобы обе составляющие активной части устройства сопряжения линейно перемещались на одинаковое расстояние «а» (рисунок 8а), что позволит зафиксировать пассивную часть с учетом заданных ограничений на начальные смещения. Данное решение разработано с учетом требований к ограниченной точности позиционирования РС.

Выбор представленных форм активной и пассивной частей обусловлен появлением дополнительных сил и моментов, которые будут направлять пассивную часть к точке О. При этом допускается, что пассивная часть относительно системы координат активной части в точке О до начала процесса сопряжения может иметь определенные линейные и угловые отклонения по положению и ориентации, т.е. пассивная часть имеет 6 степеней свободы относительно активной части. Для сопряжения необходимо определить ограничения на эти параметры.

На рисунке 9 приведены две проекции предлагаемого устройства сопряжения. На изображениях приняты следующие обозначения. Размер b_1 – размер крепежной площадки для крепления пассивной части устройства сопряжения к модулю системы. Размер c_0 – это размер ребра губки активной части устройства, размер c_1 связан с длиной ребра пассивной части устройства. Угол β соответствует углу наклона ребра пассивной части устройства и углу наклона губки активной части. Длина a_0 соответствует максимальному расстоянию, на которое могут быть разведены губки активной части

устройства сопряжения, a_1 – это длина большей диагонали основания пассивной части устройства. Угол α показывает угол между ребрами пассивной части устройства, который будет входить в контакт с губками активной части, угол между ребрами которой также равен α . Размер k_0 связан с длиной стороны угла верхней части губок, с которой пассивная часть будет контактировать при сопряжении. На рисунке также приведена система координат с центром в точке O , относительно которой происходит сопряжение.

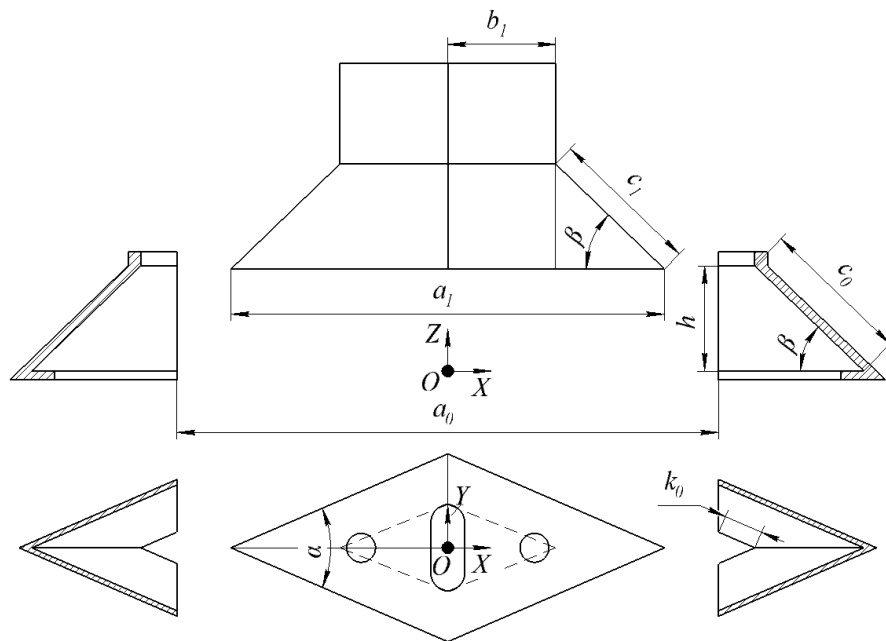


Рисунок 9 – Проекция элементов устройства сопряжения

Допустимое линейное отклонение при сопряжении d_x вдоль оси X соответствует граничному положению при z_{max} , которое определяется высотой губок. В соответствии с рисунком 9 $z_{max} = h$. Определение данного допустимого отклонения при z_{max} обусловлено тем, что в процессе сопряжения при z_{max} для успешного сопряжения d_x будет наименьшим. При этом чем меньше расстояние между активной и пассивной частью захвата по оси Z , тем больше может быть d_x . Под успешным сопряжением подразумевается возможность достижения конечного положения, соответствующее достижению точки O (см. рисунок 9) пассивной частью устройства. Тогда допустимому линейному отклонению d_x вдоль оси X , исходящей из системы координат в точке O , необходимо удовлетворять условию (7):

$$\begin{cases} -\frac{a_0 - a_1}{2} < d_x < \frac{a_0 - a_1}{2} \\ a_0 > a_1 \end{cases} \quad (7)$$

При проектировании устройства сопряжения параметр a_1 его пассивной части зависит от угла наклона губки активной части устройства β (см. рисунок 9). Тогда, изменяя данный угол, можно уменьшать или увеличивать параметр a_1 . Значение данного параметра вычисляется в соответствии с (8):

$$a_1 = 2 \cdot (c_1 \cos \beta + b_1) \quad (8)$$

Как видно, полученное значение входит в расчет отклонения в условии (7), поэтому, подставив выражение (8) в (7), можно определить диапазон допустимых отклонений вдоль оси X , варьируя параметр β . В соответствии с этим условие (7) приобретает следующий вид:

$$-\frac{a_0 - 2 \cdot (c_1 \cos \beta + b_1)}{2} < d_x < \frac{a_0 - 2 \cdot (c_1 \cos \beta + b_1)}{2} \quad (9)$$

Для успешного сопряжения необходимо, чтобы линейное отклонение вдоль оси X находилось в диапазоне допустимых отклонений в соответствии с (9).

Допустимое линейное отклонение при сопряжении d_y вдоль оси Y относительно центра O также определяется при $z_{max} = h$. Диапазон данных линейных отклонений определяется в соответствии с неравенством (10):

$$- \left| k_0 \sin \frac{\alpha}{2} \right| < d_y < \left| k_0 \sin \frac{\alpha}{2} \right| \quad (10)$$

Для успешного сопряжения необходимо, чтобы линейное отклонение d_y вдоль оси Y находилось между расчетными значениями в (10). Теоретически угол α может иметь значения в диапазоне $(0^\circ; 180^\circ)$, однако практически рекомендуется выбирать значения из диапазона $(45^\circ; 90^\circ)$. Это обусловлено тем, что при больших значениях угла может ухудшиться свойство центрирования сопряжения, что снизит точность конечного положения активной и пассивной частей, а также потребует прикладывать значительную силу к губкам активной части устройства сопряжения. Если выбирать значения угла, меньше 45° , то это, потребует большей точности позиционирования между активной и пассивной частями устройства для успешного сопряжения. Аналогичными причинами можно обосновать рекомендуемый диапазон значений угла $\beta \in (25^\circ; 70^\circ)$, теоретический диапазон которого $(0^\circ; 180^\circ)$.

Допустимое линейное отклонение d_z вдоль оси Z рассматривается с учетом отклонений d_x и d_y , при которых пассивная часть устройства сопряжения находится на такой высоте вдоль оси Z , чтобы активная часть устройства вошла в контакт с пассивной частью для дальнейшего сопряжения. Отклонение d_z зависит от угла β и длины c_1 и для успешного сопряжения необходимо, чтобы выполнялось неравенство (11).

$$0 \leq d_z < c_0 \cos \beta \quad (11)$$

Допустимый диапазон углового отклонения относительно оси Z при сопряжении определяются по неравенству (12) при z_{max} , что обусловлено

необходимостью контакта ребра пассивной части устройства сопряжения с ребрами губок активной части устройства.

$$-\left| \operatorname{atan} \left(\frac{2(k_0 \sin \frac{\alpha}{2} - d_y)}{a_1} \right) \right| < \theta_z < \left| \operatorname{atan} \left(\frac{2(k_0 \sin \frac{\alpha}{2} - d_y)}{a_1} \right) \right| \quad (12)$$

Таким образом, для успешного сопряжения необходимо, чтобы поворот пассивной и активной частей устройства сопряжения относительно друг друга вдоль оси Z находился в расчетном диапазоне (12).

Для апробации предложенной методики расчета были заданы исходные требования по допустимым отклонениям при сопряжении и необходимые размеры для будущего прототипа. Исходные данные для расчета приведены в таблице 3. Приведенные отклонения являются предельными допустимыми значениями для успешного сопряжения.

Таблица 3 – Исходные данные для расчета

Параметр	a_1 , мм	b_1 , мм	β , град	d_x , мм	d_y , мм	d_z , мм	θ_z , град
Значение	57	14	60	10	11	13	20

С учетом исходных размеров и принятых допустимых отклонений необходимо выразить линейные размеры a_0 , c_0 и k_0 устройства сопряжения. Поскольку устройство сопряжения является симметричным относительно точки O , то далее рассматриваются только неотрицательные отклонения d_x , d_y , d_z и θ_z .

Из (7) с учетом вышесказанного получается следующее неравенство:

$$a_0 > a_1 + 2d_x \quad (13)$$

Подставив в (13) исходные данные, получается, что $a_0 > 77$ мм. Далее принято $a_0 = 78$ мм.

Ограничение величины c_0 выражается из (11):

$$\begin{cases} c_0 > \frac{d_z}{\cos \beta} \\ 25^\circ < \beta < 70^\circ \end{cases} \quad (14)$$

С подстановкой исходных данных в (14) получается неравенство $c_0 > 26$ мм. Далее принимается $c_0 = 27$ мм.

В соответствии с практической рекомендацией на ограничение угла α было принято значение $\alpha = 85^\circ$. Тогда из неравенства (10) можно получить ограничение для k_0 :

$$k_0 > \frac{d_y}{\sin \frac{\alpha}{2}} \quad (15)$$

В соответствии с (15) получается неравенство $k_0 > 16$ мм, далее выбрано $k_0 = 17$ мм.

С учетом принятых в соответствии с (13) - (15) значениями и заданными параметрами в САПР были определены новые размеры и реализована модель электромеханического устройства сопряжения. Далее на модели были визуализированы рассчитанные значения допустимых отклонений при сопряжении. На рисунке 10 (а-г) представлены изображения, на которых показано взаимное расположение пассивной и активной частей устройства сопряжения при максимально допустимых отклонениях.

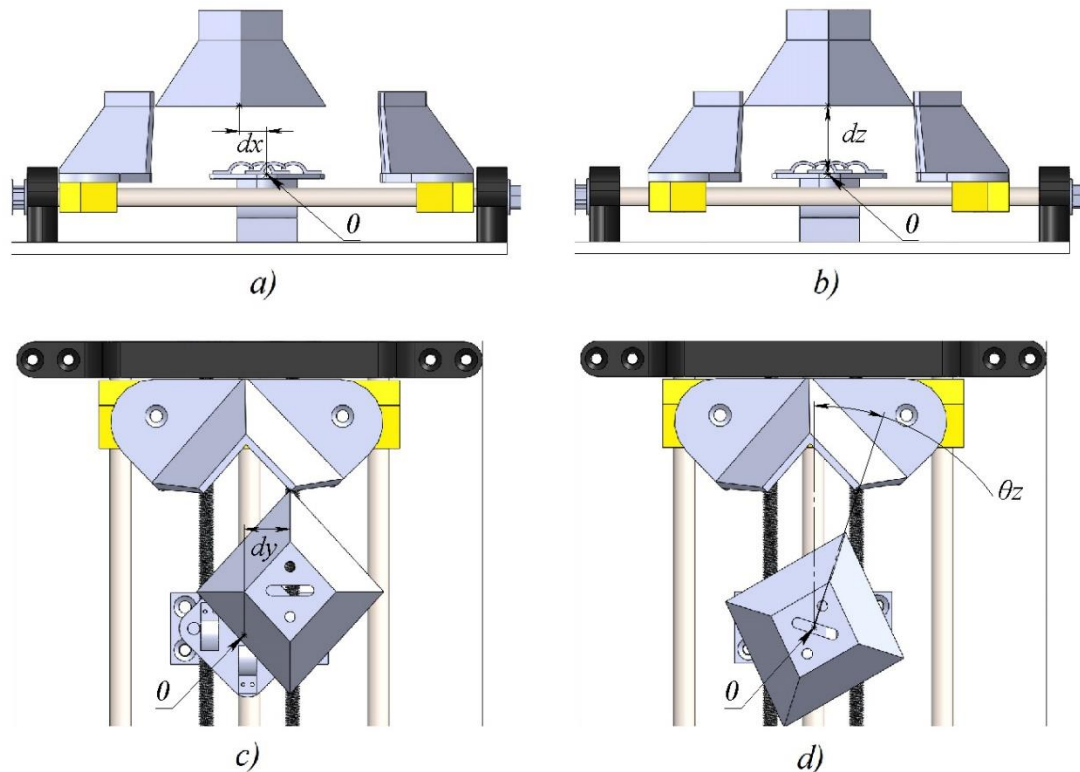


Рисунок 10 – Демонстрация рассчитанных значений допустимых отклонений на модели устройства сопряжения: а) отклонение d_x ; б) отклонение d_z ; в) отклонение d_y ; г) отклонение θ_z

Как видно из рисунка, при отклонениях, заданных при расчете, сопряжение должно проходить успешно. Использование граничных значений расчетных размеров и углов в деталях устройства сопряжения в совокупности с неточностями изготовления может привести к снижению вероятности успешного сопряжения в реальном устройстве. В соответствии с расчетными соотношениями предложенной структуры электромеханического устройства сопряжения был реализован экспериментальный прототип, модель которого приведена на рисунке 11.

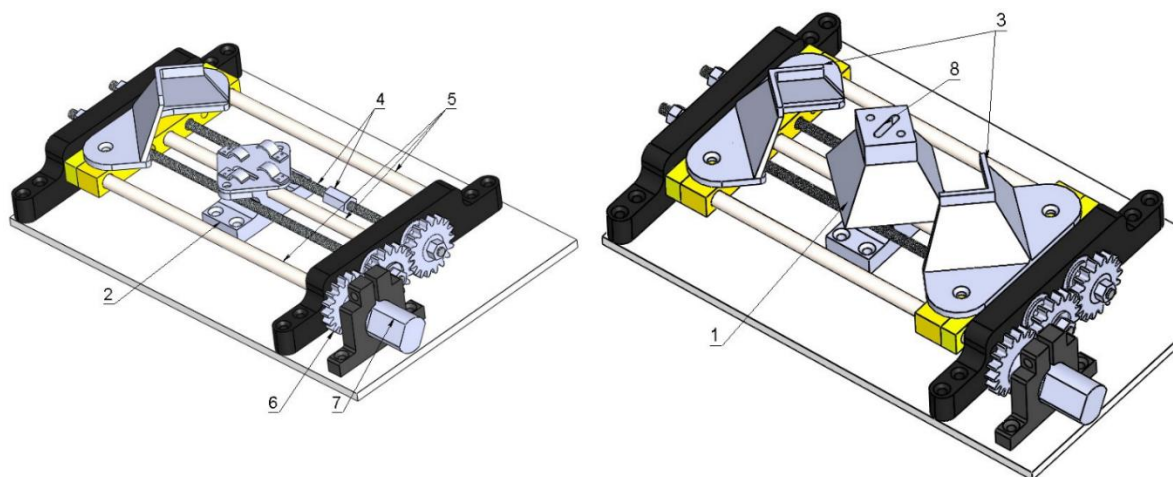


Рисунок 11 – 3D модель прототипа устройства сопряжения

Пассивная часть жестко крепится на поверхность функциональных модулей и представляет собой основание (1) с установленными на его нижнюю плоскость контактными площадками и отверстием (8) для вывода проводов. Крепление контактов на активной части осуществляется на основание (2), установленное непосредственно на корпус робототехнической платформы. Сопряжение частей устройства производится путем сведения губок захватного механизма, показанных на рисунке элементом (3), внутренняя часть которых по форме повторяет поверхность пассивной части (1). Линейное встречное перемещение захватных губок реализовано передачей винт-гайка (4) и направляющими валами (5). Передача вращения на винты, перемещающие губки, от привода (7) производится через шестерни (6), сопряженные друг с другом, за счет чего валы вращаются в разные стороны, и перемещают губки на встречу друг другу. Размеры устройства сопряжения зависят от необходимых размеров подключаемых модулей и геометрических параметров платформы. Габаритные размеры прототипа составляют $200 \times 130 \times 58$ мм. В работе [20] подробно представлен ряд экспериментов с данным прототипом устройства сопряжения. В экспериментах оценивалась успешность сопряжения в зависимости от линейных отклонений d_x , d_y и d_z , а также углового отклонения θ_z . Результаты экспериментов показали, что при использовании представленных выше расчетных соотношений вероятность успешного сопряжения снижается только при максимально допустимых значениях смещений между активной и пассивной частями. При соблюдении заданных диапазонов допустимых отклонений по осям и по углу между пассивной и активной частями устройства сопряжение проходит успешно более, чем в 98 % случаев.

В соответствии с ранее запланированными задачами по проекту разработана модель РС с возможностью автоматизированной замены отдельных модулей (рисунок 12).

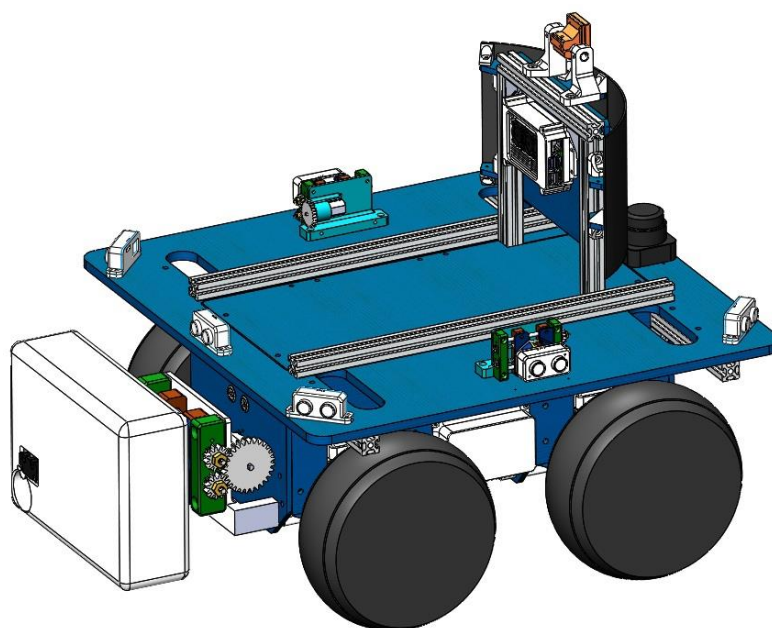


Рисунок 12 – 3D модель автономной мобильной платформы

Мобильная платформа имеет четырехколесную кинематическую схему с неповоротными колесами. Колеса попарно связаны и приводятся в движение отдельными электроприводами. Платформа стационарно оборудована восемью ультразвуковыми датчиками, лидаром, и обзорной камерой. На платформе установлено четыре устройства сопряжения (рисунок 13), разработанных на основе предложенных выше решений, для дополнительных модулей.

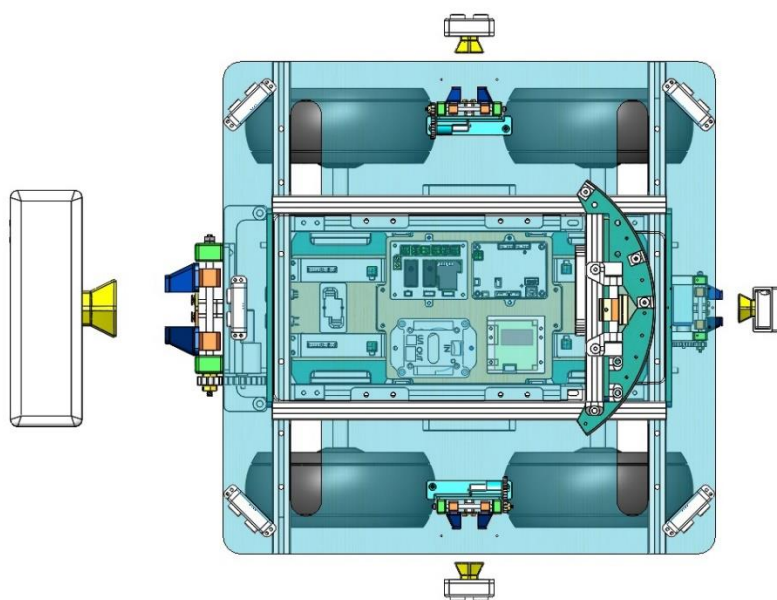


Рисунок 13 – 3D модель автономной мобильной платформы, вид сверху

Устройства сопряжения расположены в фронтальной части платформы, задней части, слева и справа. В устройстве сопряжения предусмотрены контактные группы для передачи энергии и информации (рисунок 14).

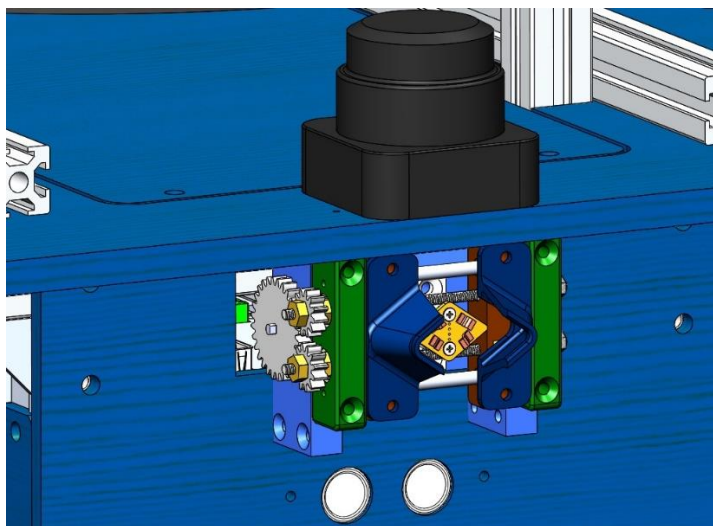


Рисунок 14 – 3D модель устройства сопряжения мобильной платформы

В каждое из устройств сопряжения возможна установка сенсорных или электромеханических исполнительных устройств. В приведенной модели в качестве примера в фронтальное устройство сопряжения установлен модуль камеры технического зрения, а в расположенные на боковой части платформы, ультразвуковые дальномеры. В устройство сопряжения, расположенное на задней части платформы, установлен модуль для расстановки автономных приемо-передатчиков по пути следования, которые позволяют организовать Mesh - сеть для передачи данных между платформой и пунктом управления. Также в устройство сопряжения, расположенное в задней части, может устанавливаться тягово-сцепное устройство, обеспечивающее присоединение к платформе колесных прицепов с дополнительным оборудованием различного назначения.

Электротехническое оснащение платформы выполнено по принципу распределенной централизованной системы управления с двух уровневой архитектурой (рисунок 15).

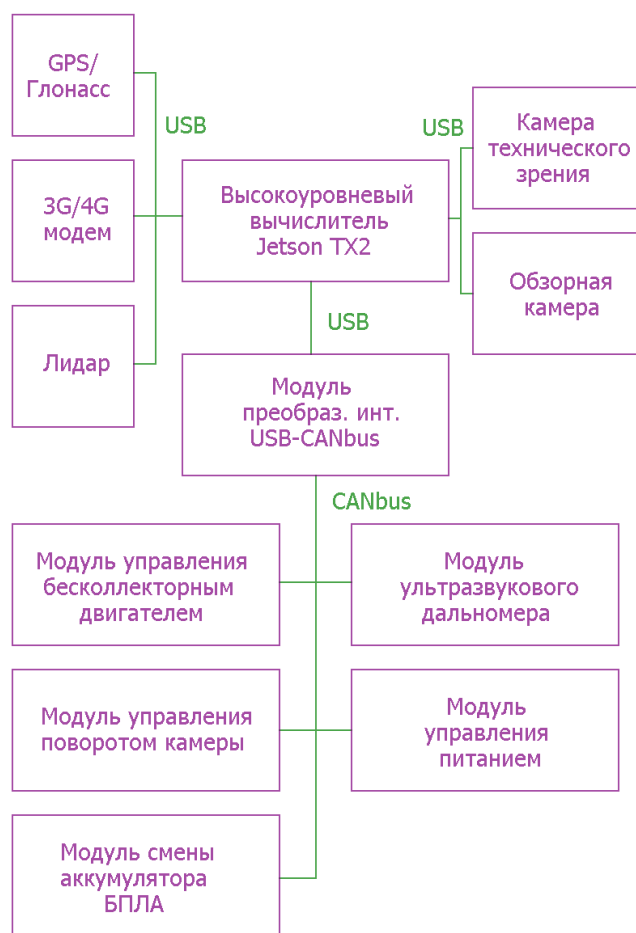


Рисунок 15 – Архитектура системы управления

Первый уровень представлен основным вычислителем. Основной вычислитель осуществляет обработку данных с лидара, изображений с обзорной камеры и камеры технического зрения, связывает мобильную платформу с облачными сервисами через беспроводной интерфейс GSM для удаленного управления и вывода изображений с камер. Основной вычислитель связан с сенсорными и функциональными модулями платформы через модуль преобразования интерфейса USB 2.0 – CANbus. Модуль преобразования интерфейса поддерживает подключение до 64 модулей любого назначения второго уровня архитектуры по шине CANbus. Каждый модуль второго уровня архитектуры имеет собственный вычислитель.

Предложенная архитектура системы обладает свойствами масштабируемости и универсальности, что позволяет создавать системы управления подвижными робототехническими комплексами различного назначения. Собственный вычислитель каждого модуля системы позволяет организовать вычисления в каждом исполнительном устройстве и сенсоре, что значительно ускоряет работу системы, снижает вероятность ошибок и увеличивает надежность и отказоустойчивость.

Применение разработанного устройства сопряжения позволит создавать мобильные многофункциональные робототехнические платформы различного

назначения, с возможностью автономной реконфигурации. Это обеспечит снижение вовлечения человека в эксплуатацию данных РС. Также предложенное решение позволит расширять функционал базовой робототехнической платформы на произвольное число модулей и размещать их в практически любых местах платформы. Кроме того, свойство масштабирования устройства сопряжения позволяет использовать модули с различными массогабаритными параметрами.

5. Разработка программного обеспечения для распределенного решения задач построения карты территории, локализации компонентов системы, навигации РС, а также обеспечения анализа геометрических характеристик внешней среды в режиме реального времени на основе применения распределенных SLAM-алгоритмов

На текущем этапе реализации проекта было разработано программное обеспечение (ПО), предназначенное для локализации и навигации на местности гетерогенной группы РС. Разработанное ПО выполняется одновременно на нескольких роботах-агентах, которые совместно осуществляют анализ характеристик внешней среды и осуществляют построение карты местности.

В решении задач по одновременной локализации компонентов КФС и построению карт местности (SLAM) и автономному функционированию, где одним из основных источников данных об окружающей среде для них является датчик LiDAR, установленный на борту РС. В качестве наземных РС выступают передвижные платформы, оснащенные вычислительными устройствами, а также бортовыми датчиками LiDAR, в качестве воздушных РС выступают БВС типа квадрокоптер.

В настоящее время SLAM-навигация получила распространение в Robotic Operating System (ROS) [21] в виде пакетов, зачастую находящихся в открытом доступе, например: «GMapping» [22], «Cartographer» [23], «Rtabmap» [24]. Проведя расширенный анализ данных SLAM-пакетов, был выбран «Cartographer». Немаловажным преимуществом данного пакета является возможность использовать его в распределенных групповых системах [25]. На базе «Cartographer» может быть реализована распределенная SLAM-навигация, основанная на использовании группы гетерогенных РС.

В рамках проекта была разработана распределенная модификация SLAM-алгоритма CORB-SLAM на основе пакета «Cartographer», реализованная в системе ROS с применением клиент-серверной архитектуры.

Согласно базовому алгоритму CORB-SLAM каждый клиентский агент осуществляет построение собственной локальной карты местности и передает соответствующую информацию на сервер, а сервер обеспечивает построение и обновление единой интегрированной глобальной карты территории [26]. Каждый раз по результатам генерации обновленной глобальной карты, сервер отправляет каждому клиентскому агенту информацию об измененной части глобальной карты местности. С целью уменьшения полосы пропускания передачи данных, авторы CORB-SLAM предложили технологию передачи облегченных данных на сервер (клиентом передается ключевой кадр и координаты агента в текущий момент времени). Поскольку построение глобальной карты местности осуществляется на основе облегченных данных, полученная таким образом карта территории не обладает достаточной степенью детализации и не может быть использована клиентскими агентами. Чтобы избежать этого недостатка в рамках проекта и была предложена распределенная модификация SLAM-алгоритма CORB-SLAM, реализованная в системе ROS с применением клиент-серверной архитектуры.

Рассмотрим предложенное решение подробнее. В системе ROS есть набор сущностей, которыми описывается общая структура ПО: узел (Node) – исполняемый файл, который использует ROS для связи с другими узлами; тема (Topic) – узлы могут публиковать сообщения в теме, а также подписываться на тему для получения сообщений в рамках системы ROS; сообщение (Message) – тип данных ROS, используемый при подписке и публикации тем.

В разработанной системе, реализующей распределенную SLAM навигацию, можно выделить три ROS узла (nodes): сенсор, клиент и сервер. На рис. 16 показана структура предлагаемой системы. Узел сенсора может быть представлен в виде камеры глубины RGB-D или лидара. Узел клиента представляет собой узел ROS, размещенный на клиентском агенте. ROS узел сенсора публикует изображение RGB и изображение глубины или же данные с лидара на клиентский узел. Клиент преобразует полученные данные в точки карты и публикует сообщения типа «ключевой кадр и точки созданной карты в данном ключевом кадре». Передача данных происходит посредством связи клиента и сервера одной темой (topic). Так как темы в ROS работают асинхронно, система SLAM на клиенте может осуществлять построение локальной карты местности вне зависимости от состояния сервера. При этом сервер по запросу от клиентского агента, согласно разработанному решению, обеспечивает передачу актуальной версии сгенерированной глобальной карты и реализует ее интеграцию с локальной картой клиентского агента. В оригинальной реализации пакета «Cartographer» предусмотрено исполнение трех асинхронных потоков на каждом клиентском агенте: отслеживание (Tracking), локальное сопоставление (Local mapping) и закрытие цикла (Loop

closing). В потоке отслеживания происходит извлечение данных с сенсора с последующим выбором ключевых кадров. В потоке локального сопоставления выполняется корректировка положений ключевых кадров и точек карты. В потоке закрытия цикла реализуется процедура оптимизации для уточнения существующей локальной карты местности. Предложенная модификация заключается в добавлении в пакет «Cartographer» отдельного потока, который передает данные карты от клиента к серверу (Streaming).

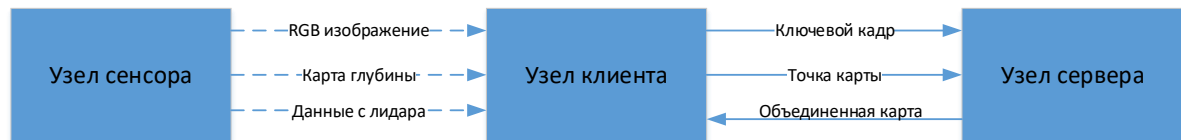


Рисунок 16 – Структура предлагаемой системы

На рис. 17 приведена обновленная схема потоков данных в рамках пакета «Cartographer» с учетом добавленного потока Streaming. Согласно предложенной модификации распределённого SLAM-алгоритма сгенерированный ключевой кадр и точки карты из потока отслеживания, а также данные из потоков локального сопоставления и закрытия цикла передаются непосредственно на сервер. После получения ключевого кадра и точек карты от потока отслеживания поток локального сопоставления, функционирующий на сервере, удаляет или корректирует информацию о ключевом кадре. Данная схема передачи данных позволяет увеличить точность генерации глобальной карты местности, а также снизить задержки в процессе связи сервера с клиентскими агентами.

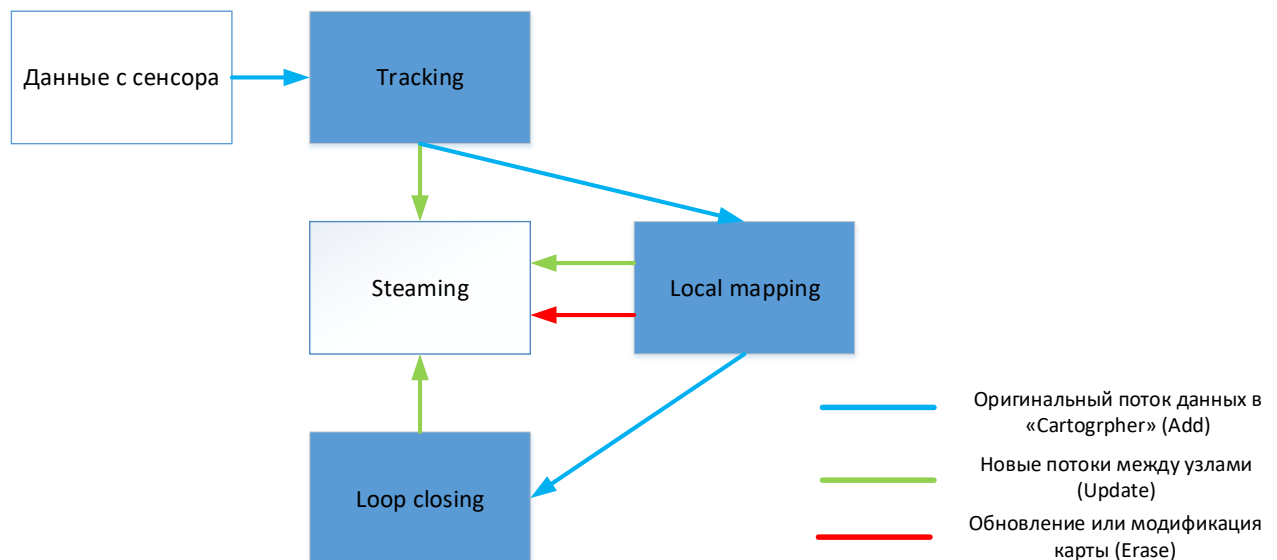
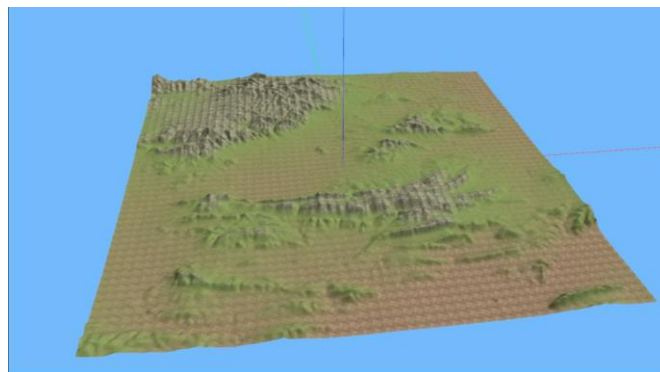


Рисунок 17 – Диаграмма потоков передачи данных клиентского агента в рамках модификации пакета «Cartographer»

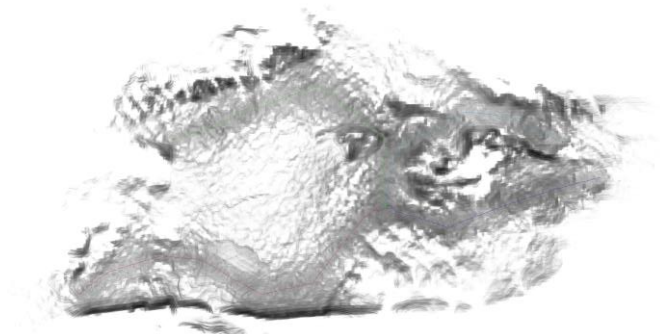
Таким образом, предложенная распределённая модификация SLAM-алгоритма CORB-SLAM отличается более высокой пропускной способностью

и повышенной устойчивостью в случае нестабильных каналов передачи данных.

В ходе выполнения проекта была дополнительно проведена экспериментальная апробация предложенного решения в симуляторе Gazebo. Для проведения эксперимента была записана последовательность кадров в формате Rosbag. Группа клиентских агентов состояла из четырех РС Pioneer3AT. Исходная модель местности представлена на рис. 18а, а результаты генерации карты местности с использованием разработанного решения – на рис. 18б.



а)



б)

Рисунок 18 – Результаты моделирования: а) исходная модель карты б) построенная карта при помощи разработанного распределенного SLAM алгоритма, развернутого на базе группы из четырех наземных РС

В рамках эксперимента каждому РС была назначена собственная целевая координата. При помощи разработанного в рамках проекта алгоритма планирования движения GMPUS для группы наземных РС, каждое РС строит проходимый маршрут с учетом рельефа и препятствий на местности. На рис. 19 представлены результаты генерации глобальной карты территории посредством разработанного распределенного SLAM алгоритма в проекции на плоскость YOZ глобальной системы отсчета.

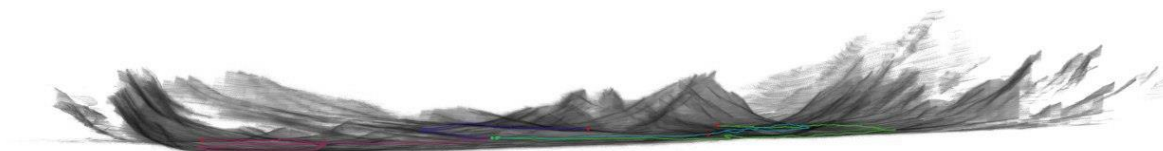


Рисунок 19 – Результаты генерации глобальной карты территории посредством разработанного распределенного SLAM алгоритма в проекции на плоскость YOZ глобальной системы отсчета

Предложенное решение устойчиво к потере соединения сервера с отдельными клиентскими агентами и способно функционировать до тех пор, пока все клиенты не утратят соединение с сервером. Дополнительно, использование модели данных, при которой ключевые кадры и точки карты разделены и имеют собственные независимые идентификационные номера, позволило уменьшить временную задержку при обработке картографической информации от клиентских агентов и реализовать построение детализированных глобальных карт местности на сервере в режиме приближенном к реальному времени.

Список использованных источников

1. Mohamed M., Engel R., Warke A., Berman S., Ludwig H. Extensible persistence as a service for containers // *Future Generation Computer Systems*. – 2019. – Т. 97. – С. 10-20.] [Ahmed B., Seghir B., Al-Osta M., Abdelouahed G. Container based resource management for data processing on IoT gateways // *Procedia Computer Science*. – 2019. – Т. 155. – С. 234-241.
2. Dehury C. K., Sahoo P. K. Design and implementation of a novel service management framework for IoT devices in cloud // *Journal of Systems and Software*. – 2016. – Т. 119. – С. 149-161.
3. Dehury C. K., Sahoo P. K. Design and implementation of a novel service management framework for IoT devices in cloud // *Journal of Systems and Software*. – 2016. – Т. 119. – С. 149-161.
4. Chaâri R., Ellouze F., Koubâa A., Qureshi B., Pereira N., Youssef H., Tovar E. Cyber-physical systems clouds: A survey // *Computer Networks*. – 2016. – Т. 108. – С. 260-278.
5. Iakovlev R., Saveliev A. Approach to implementation of local navigation of mobile robotic systems in agriculture with the aid of radio modules // *Telfor Journal*. – 2020. – Т. 12. – №. 2. – С. 92-97.
6. Höftberger, O., & Obermaisser, R. (2013, June). Ontology-based runtime reconfiguration of distributed embedded real-time systems. In *16th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2013)* (pp. 1-9). IEEE.
7. Ratasich, D., Höftberger, O., Isakovic, H., Shafique, M., & Grosu, R. (2017, May). A self-healing framework for building resilient cyber-physical systems. In *2017 IEEE 20th International Symposium on Real-Time Distributed Computing (ISORC)* (pp. 133-140). IEEE.
8. R. Yue, J. Xiao, S. L. Joseph, and S. Wang, "Modeling and path planning of the city-climber robot part II: 3D path planning using mixed integer linear programming," in *Proceedings of the IEEE International Conference on*

- Robotics and Biomimetics (ROBIO '09), vol. 6, pp. 2391–2396, Guilin, China. 2009
9. S. J. Anderson, S. C. Peters, T. E. Pilutti, and K. Iagnemma, “An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance Scenarios,” *International Journal of Vehicle Autonomous Systems*, vol. 8, no. 2–4, pp. 190–216. 2010
 10. Bai Y., Wang Y., Svinin M., Magid E., Sun R. Adaptive multi-agent coverage control with obstacle avoidance. *IEEE Control Systems Letters*. 2021;6:944-949. (In English).
 11. Севостьянова Н. Н. и др. Инновационный подход к автоматизированной фотоактивации посевных площадей посредством БПЛА с целью стимуляции роста культур //Информатика и автоматизация. – 2021. – Т. 20. – №. 6. – С. 1395-1417.
 12. Li, D.; Wang, X.; Sun, T. Energy-optimal coverage path planning on topographic map for environment survey with unmanned aerial vehicles. *Electron. Lett.* 2016, 52, 699–701.
 13. Nam, L.; Huang, L.; Li, X.; Xu, J. An Approach for Coverage Path Planning for UAVs. In *Proceedings of the 2016 IEEE 14th International Workshop on Advanced Motion Control (AMC)*, Auckland, New Zealand, 22–24 April 2016; pp. 411–416.][The adaptive vortex search algorithm of optimal path planning for forest fire rescue UAV, C Wang, P Liu, T Zhang, J Sun - 2018
 14. Zhang Z., Wu J., Dai J., He C. Optimal path planning with modified A-Star algorithm for stealth unmanned aerial vehicles in 3D network radar environment. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*. 2022;236(1):72-81.], RRT и RRT* [Wu X., Xu L., Zhen R., Wu X. Biased sampling potentially guided intelligent bidirectional RRT* algorithm for UAV path planning in 3D environment. *Mathematical Problems in Engineering*. 2019

15. Lebedev I., Lebedeva V. Analysis of «Leader–Followers» Algorithms in Problem of Trajectory Planning for a Group of Multi-rotor UAVs. Proceedings of the Computational Methods in Systems and Software. Springer, Cham. 2021:870-884.
16. Haotian Niu, Cunbao Ma, Pei Han, Directional optimal reciprocal collision avoidance, Robotics and Autonomous Systems, Volume 136, 2021, 103705, ISSN 0921-8890
17. K. Zakharov, A. Saveliev, O. Sivchenko. Energy-efficient Path Planning Algorithm on Three-dimensional Large-scale Terrain Maps for Mobile Robots. Proceedings of 5th International Conference ICR 2020. 2020.
18. A. Ayari, S. Bouamama. A new multiple robot path planning algorithm: dynamic distributed particle swarm optimization. Robot. Biomim. 4, 8. 2017. <https://doi.org/10.1186/s40638-017-0062-6>
19. A. Salman, I. Ahmad, S. Al-Madani. Particle swarm optimization for task assignment problem. Microprocessors and Microsystems 26 (2002) 363–371. 2002. [https://doi.org/10.1016/S0141-9331\(02\)00053-4](https://doi.org/10.1016/S0141-9331(02)00053-4)
20. Крестовников К.Д., Ерашов А.А., Васюнина Ю.Г., Савельев А.И. Разработка устройства сопряжения для модульной сельскохозяйственной робототехнической платформы. Сельскохозяйственные машины и технологии. 2022;16(1):78-88. <https://doi.org/10.22314/2073-7599-2022-16-1-78-88>
21. ROS [Электронный ресурс] URL: <https://www.ros.org>, свободный ресурс – Загл. с экрана. Язык англ. Дата обращения: 14.02.2022 г.
22. Gmapping [Электронный ресурс] URL: <http://wiki.ros.org/gmapping>, свободный ресурс – Загл. с экрана. Язык англ. Дата обращения: 14.02.2022 г.
23. Google Cartographer ROS [Электронный ресурс] URL: <https://google-cartographer-ros.readthedocs.io/en/latest/#>, свободный ресурс – Загл. с экрана. Язык англ. Дата обращения: 04.01.2022 г.

24. RTAB-Map, Real-Time Appearance-Based Mapping [Электронный ресурс]
URL: <http://introlab.github.io/rtabmap/>, свободный ресурс – Загл. с экрана.
Язык англ. Дата обращения: 22.01.2022 г.
25. Yagfarov, R., Ivanou, M., & Afanasyev, I. (2018). Map Comparison of Lidar-based 2D SLAM Algorithms Using Precise Ground Truth. 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV).
26. LI, Fu, et al. Corb-slam: a collaborative visual slam system for multiple robots. In: International Conference on Collaborative Computing: Networking, Applications and Worksharing. Springer, Cham, 2017. p. 480-490.