

А.И. САВЕЛЬЕВ, Д.А. АНИКИН
**МЕТОД РАСЧЕТА КАПСУЛООБРАЗНЫХ ВОЗДУШНЫХ
КОРИДОРОВ БЕЗОПАСНЫХ МАРШРУТОВ ГРУППЫ
БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ**

Савельев А.И., Аникин Д.А. Метод расчета капсулообразных воздушных коридоров безопасных маршрутов группы беспилотных летательных аппаратов.

Аннотация. В работе рассматривается задача построения безопасных маршрутов группы беспилотных летательных аппаратов в ограниченном воздушном пространстве над сельскохозяйственным участком. Актуальность исследования обусловлена растущим применением групп БПЛА в агропромышленном комплексе для мониторинга, картографирования и обработки полей, что требует обеспечения безопасности полётов в условиях высокой плотности воздушного движения, ограниченной связи, а также воздействия внешних факторов. Особую сложность представляет необходимость автономного выполнения миссий при наличии навигационных погрешностей и природных воздействий. Предлагается метод планирования маршрутов, основанный на представлении траектории каждого аппарата в виде капсульного воздушного коридора – трёхмерного объёма фиксированного радиуса, сформированного вдоль отрезков траектории. Пространственное резервирование обеспечивает безопасное разнесение траекторий на этапе планирования, исключая конфликты при последующем автономном выполнении полётов без необходимости непрерывной координации между агентами. Радиус капсулы включает запас на возможные отклонения от запланированной траектории, что обеспечивает устойчивость к навигационным ошибкам. В основе метода лежит последовательное формирование маршрутов для каждого аппарата по четырёхфазной схеме, включающей вертикальный подъём от точки старта до рабочей высоты, горизонтальный переход к входу в зону обработки, возврат от выхода из зоны к точке начала снижения и вертикальный спуск к исходной позиции. Каждый новый маршрут строится с учётом уже зарезервированных воздушных коридоров через аналитическую проверку геометрических пересечений между капсулами различных траекторий и выпуклыми многогранниками зон обработки. Для повышения вычислительной эффективности применяется иерархическая пространственная фильтрация на основе ограничивающих параллелепипедов, позволяющая на предварительном этапе быстро отсекаать заведомо непересекающиеся объекты и выполнять точную геометрическую проверку только для потенциально конфликтующих сегментов маршрутов. Численные эксперименты проводились для групп от 2 до 32 аппаратов на типовом сельскохозяйственном участке площадью один квадратный километр. Установлено нелинейное возрастание времени планирования и количества итераций с увеличением числа агентов, что обусловлено необходимостью построения каждого последующего маршрута в уже частично занятом пространстве с возрастающим числом пространственных ограничений. Длина маршрутов демонстрирует тенденцию к росту, особенно выраженную на начальных этапах масштабирования, что связано с необходимостью обхода уже зарезервированных воздушных коридоров.

Ключевые слова: БПЛА, воздушные коридоры, сельскохозяйственные задачи, Rapidly-Exploring Random Tree*, Loose Octree, автономная навигация, групповое планирование.

1. Введение. Современное сельское хозяйство активно использует беспилотные летательные аппараты (БПЛА) для

мониторинга посевов [1, 2], картографирования местности [3, 4], агрохимической обработки [5], оценки урожайности и анализа состояния полей [6]. Чем больше дронов работает одновременно и чем сложнее становится организация воздушного движения, тем острее встает вопрос слаженного взаимодействия групп БпЛА. Когда множество аппаратов выполняют различные задачи в одном воздушном пространстве, критически важно обеспечить их пространственную координацию и предотвратить столкновения на пересекающихся маршрутах.

При выполнении сельскохозяйственных операций БпЛА работают в условиях плотной загрузки выделенного воздушного пространства, жестких временных рамок для выполнения задач и ограниченной вычислительной мощности бортовых систем [7]. Кроме того, в реальной эксплуатации на выполнение полётных заданий влияют внешние природные воздействия и внутренние технические отклонения, включая погрешности телеметрии, сбои навигационных систем, временные потери связи с наземной станцией, отказы отдельных аппаратов, а также порывистый ветер и неустойчивые погодные условия, способные отклонить фактическую траекторию от запланированной. Исходя из этого, маршруты полета должны обеспечивать безопасность движения и исключение конфликтов между аппаратами в условиях прерывистой связи с центром управления, одновременно компенсируя навигационные ошибки при возможных отклонениях от запланированного курса.

Существующие методы и алгоритмы многоагентного планирования обеспечивают различный уровень оптимальности, но, как правило, не адаптированы к требованиям и ограничениям практической эксплуатации в аграрной отрасли.

Алгоритм Conflict-Based Search (CBS) [8] и его производные формируют маршруты путём централизованного разрешения конфликтов между агентами в пространственно-временной области через построение дерева конфликтов. Хотя базовый CBS обеспечивает оптимальность решения, он предполагает дискретное пространственно-временное планирование и строгое следование фиксированным траекториям, что делает его непригодным для прямого применения в реальных сценариях с БпЛА [9]. Такие алгоритмы не предусматривают отклонений от запланированного поведения – при выходе одного БпЛА из строя или отклонении его траектории возникает риск несогласованности с действиями других агентов, что может потребовать полной переработки конфигурации маршрутов для сохранения безопасности. Модификация для работы с

непрерывным временем – Continuous-time CBS (CCBS) [10], также сталкивается с вычислительными ограничениями при масштабировании и усложняется необходимостью учёта физических характеристик и динамических параметров реальных летательных аппаратов. Алгоритм Multi-Conflict-Based Search (MCBS) [11] показывает, что при росте числа агентов и плотности взаимодействий структура дерева конфликтов значительно усложняется – количество возможных ветвлений растёт экспоненциально, что ограничивает применимость метода в сценариях с высокой насыщенностью полётов, особенно при необходимости учёта трёхмерного пространства и временных ограничений.

Алгоритм Multi-Agent Rapidly-Exploring Random Tree* (MA-RRT*) [12] осуществляют поиск траекторий в объединённом конфигурационном пространстве размерности \mathbb{R}^3 , где m – число агентов. Базовый алгоритм Rapidly-Exploring Random Tree* (RRT*) [13] представляет собой стохастический подход выборочного построения деревьев, обеспечивающий асимптотическую сходимость к оптимальному решению. В многоагентной версии данная стратегия теоретически сохраняет глобальную сходимость, однако сталкивается с проклятием размерности – объём пространства поиска растёт экспоненциально с увеличением числа агентов. Кроме того, MA-RRT* предполагает точное выполнение заранее построенных траекторий и не учитывает внешние отклонения – например, сбой навигации, воздействие ветра или ошибки телеметрии. При отказе одного БПЛА или отклонении от плана нарушается согласованность всей конфигурации, что требует пересчёта маршрутов группы в целом.

Централизованные эвристические алгоритмы, такие как Cooperative A*, использующий приоритетное планирование и эвристики расстояния для координации агентов [14], и Enhanced Conflict-Based Search (ECBS), применяющий фокальный поиск и ограниченную субоптимальность для ускорения разрешения конфликтов [15], сокращают объем вычислений при планировании маршрутов. Данные подходы эффективны при фиксированном составе группы БПЛА и статичной среде, но не обладают устойчивостью к навигационным ошибкам и вынуждены инициировать глобальное перепланирование даже при незначительных отклонениях агентов от заданных траекторий.

Реактивные алгоритмы, такие как Optimal Reciprocal Collision Avoidance (ORCA) [16] и Reciprocal Velocity Obstacles (RVO) [17], обеспечивают локальное предотвращение столкновений через

корректировку векторов скорости в реальном времени без явной коммуникации между агентами. Однако данные подходы критически зависят от точности локализации и предполагают надёжное измерение скоростей окружающих агентов. Они особенно чувствительны к накапливающимся погрешностям навигации – даже небольшие ошибки в определении относительных положений и скоростей приводят к неэффективным или потенциально опасным траекториям. При временных сбоях в измерениях или задержках обновления телеметрии эти алгоритмы не имеют механизмов компенсации, что ограничивает их применение в условиях нестабильной связи и требует постоянной высокоточной локализации для безопасного функционирования.

Поведенческие и децентрализованные подходы, включая модели на основе потенциальных полей (Artificial Potential Field, APF) [18], алгоритмы имитации поведения роя (Boids, Cucker-Smale model) [19, 20], а также алгоритмы альтернативной коллективной адаптации [21] и роевого управления с элементами кластерного анализа [22] обладают высокой адаптивностью и подходят для работы в частично наблюдаемых или слабо структурированных средах. Однако такие методы, как правило, требуют тонкой эмпирической настройки параметров, их поведение сложно предсказать в новых условиях, а маршруты формируются в процессе исполнения, а не заранее. Это затрудняет проверку надёжности, воспроизводимости и безопасности движения до начала полёта.

С учётом изложенных ограничений предлагается альтернативный метод, в котором безопасность и устойчивость движения достигаются за счёт геометрически гарантированного резервирования пространства для каждого маршрута ещё до начала выполнения миссии. При этом траектория каждого БПЛА представляется в виде набора капсул, формирующих воздушный коридор – совокупность трёхмерных объёмов фиксированного радиуса, последовательно расположенных вдоль отрезков маршрута. Эти объёмы трактуются как зарезервированные безопасные зоны, исключаяющие пересечения с траекториями других агентов. Таким образом, безопасность полётов обеспечивается конструктивно, ещё на этапе планирования, без потребности в онлайн-координации или реактивном избегании конфликтов в процессе выполнения миссии.

Предложенная простая и строго геометрическая структура маршрутов группы БПЛА позволяет устранить ряд критичных для реального применения ограничений:

1. Нет необходимости в точной временной синхронизации: БПЛА могут стартовать независимо и выполнять задачи с различной продолжительностью.

2. Отсутствие зависимости от обмена данными в режиме реального времени с наземной станцией: все вычисления выполняются заранее, полёт проходит автономно.

3. Устойчивость к внешним воздействиям: выделенный безопасный объём формирует пространственный запас, позволяющий БПЛА сохранять корректность траектории даже при наличии внешних возмущений.

4. Отказоустойчивость: при потере связи или выходе из строя одного БПЛА конфликтные ситуации исключены, поскольку каждый агент движется в своём зарезервированном воздушном пространстве.

5. Простота реализации: метод не опирается на сложные динамические модели или ресурсоёмкие численные процедуры, а использует аналитическую геометрию.

Маршруты для агентов группы строятся на основе алгоритма RRT*[13] модифицированного для последовательного построения с учётом уже зарезервированных объёмов. Новый путь генерируется только в том случае, если он не пересекается с существующими коридорами. Для ускорения проверок используется структура данных «свободное октодерев» (Loose Octree) [23] – разновидность октодерева с расширенными ячейками, что позволяет эффективно индексировать пространственные объекты произвольного размера и выполнять локализованные запросы.

Построенные маршруты могут быть напрямую загружены в управляющее программное обеспечение БПЛА в виде контрольных точек. Это обеспечивает полную автономность миссии, исключает зависимость от связи в процессе выполнения и гарантирует воспроизводимость поведения.

Следует отметить, что предложенный метод не обеспечивает глобальной оптимальности маршрутов всей группы, поскольку маршруты строятся последовательно, с учётом уже зарезервированных пространственных объёмов. Для каждого агента используется алгоритм RRT*, который гарантирует асимптотическую сходимость к кратчайшему пути в рамках доступной области поиска. Поскольку эта область заранее ограничена ранее построенными маршрутами и зонами обработки, найденное решение оптимально лишь среди маршрутов, допустимых с точки зрения геометрических ограничений. Такой подход обеспечивает приемлемое качество отдельных

маршрутов при высокой вычислительной эффективности и устойчивости всей конфигурации.

Таким образом, метод объединяет пространственную независимость, отказоустойчивость, масштабируемость, автономность и простоту интеграции. Указанные характеристики обеспечивают его применимость в агропромышленной сфере, где критически важны надёжность исполнения, устойчивость к внешним воздействиям и эффективность в условиях ограниченных ресурсов на борту БПЛА.

2. Постановка задачи. Рассматривается задача построения безопасных маршрутов для группы из $m \in \mathbb{Z}$ БПЛА, выполняющих полёты к распределённым сельскохозяйственным участкам для проведения агротехнических работ. Каждый участок представлен в виде многогранника $A_j \subset \mathbb{R}^3$, $j = 1, \dots, m$, а воздушное пространство моделируется трёхмерной рабочей областью $\Omega \subset \mathbb{R}^3$.

Каждый БПЛА a_i , $i = 1, \dots, m$ стартует из точки p_1^i , выполняет миссию на выделенном участке и возвращается в точку $p_{n_i}^i$. Маршрут i -го БПЛА описывается как последовательность из $n \in \mathbb{Z}$ трёхмерных точек:

$$P_i = (p_1^i, p_2^i, \dots, p_{n_i-1}^i, p_{n_i}^i), \quad (1)$$

где n_i – количество точек маршрута i -го БПЛА.

Каждому отрезку маршрута $[p_k^i, p_{k+1}^i]$, $k = 1, \dots, n-1$ сопоставляется трехмерный объем в форме капсулы заданного радиуса R_i , объединение таких элементов определяет воздушный коридор C_i :

$$C_i = \bigcup_{k=1}^{n_i-1} \text{Capsule}(p_k^i, p_{k+1}^i, R_i), \quad (2)$$

где $\text{Capsule}(p_k^i, p_{k+1}^i, R_i)$ – объем в форме капсулы с радиусом R_i , образованный вокруг отрезка между точками p_k^i и p_{k+1}^i , моделирующий занимаемое воздушным коридором пространство; n_i – количество точек маршрута i -го БПЛА.

Сама капсула представляет собой множество точек, находящихся на расстоянии не более R_i от отрезка маршрута $[p_k^i, p_{k+1}^i]$ [24]:

$$\text{Capsule}(p_k^i, p_{k+1}^i, R_i) = \{x \in \mathbb{R}^3 : d(x, [p_k^i, p_{k+1}^i]) \leq R_i\}, \quad (3)$$

где $x \in \mathbb{R}^3$ – произвольная точка пространства; $d(x, [p_k^i, p_{k+1}^i])$ – расстояние от точки x до отрезка $[p_k^i, p_{k+1}^i]$.

Задача формируется как поиск такой конфигурации маршрутов группы БПЛА, при которой суммарная длина всех маршрутов минимальна, при этом соответствующие им воздушные коридоры не пересекаются между собой и не вторгаются в зоны активной обработки, закреплённые за другими агентами, а маршруты не выходят за пределы рабочего пространства. Математическая формализация данной задачи представляет собой оптимизацию с геометрическими ограничениями.

Целевая функция направлена на минимизацию суммарной длины всех маршрутов:

$$\min_{P_1, \dots, P_m} \sum_{i=1}^m \sum_{k=0}^{n_i-1} \|p_{k+1}^i - p_k^i\|_2, \quad (4)$$

где P_i – маршрут i -го БПЛА.

При этом на допустимые маршруты накладываются следующие пространственные ограничения:

Во-первых, воздушные коридоры различных БПЛА не должны пересекаться:

$$\forall i = 1, \dots, m, \forall j = 1, \dots, m, i \neq j \quad C_i \cap C_j = \emptyset. \quad (5)$$

Во-вторых, необходимо исключить попадание коридора одного БПЛА в активную зону обработки другого БПЛА:

$$\forall i = 1, \dots, m, \forall j = 1, \dots, m, i \neq j \quad C_i \cap Z_j = \emptyset. \quad (6)$$

В-третьих, все точки маршрутов должны находиться внутри допустимой рабочей области:

$$\forall i = 1, \dots, m, \forall k = 1, \dots, n_i \quad p_k^i \in \Omega. \quad (7)$$

3. Проверка пересечений воздушных коридоров. При построении нового маршрута требуется учитывать уже сформированные воздушные коридоры других БПЛА. Для этого на каждом шаге поиска следующей точки маршрута необходимо построить капсулу между текущей точкой и кандидатом на следующую точку, после чего выполнить проверку ее пересечения с капсулами существующих воздушных коридоров.

Пусть заданы две капсулы с несущими отрезками $[p, q]$ и $[r, s]$, а также радиусами R_1 и R_2 соответственно. Капсулы пересекаются, если выполняется условие:

$$d([p, q], [r, s]) \leq R_1 + R_2, \quad (8)$$

где $d([p, q], [r, s])$ – расстояние между отрезками.

Расстояние между отрезками определяется как наименьшее расстояние между произвольными точками, лежащими на этих отрезках.

Для аналитического вычисления расстояния удобно представить сами отрезки в параметрической форме с использованием линейной интерполяции между их конечными точками. Это позволяет выразить произвольную точку на каждом отрезке через скалярный параметр.

Точка на отрезке $[p, q]$ задаётся как:

$$x(\lambda_1) = (1 - \lambda_1)p + \lambda_1 q, \quad (9)$$

где $\lambda_1 \in [0, 1]$ определяет положение точки вдоль первого отрезка.

Точка на отрезке $[r, s]$ задается аналогично:

$$y(\lambda_2) = (1 - \lambda_2)r + \lambda_2 s, \quad (10)$$

где $\lambda_2 \in [0, 1]$ определяет положение точки вдоль второго отрезка.

Тогда поиск расстояния между отрезками сводится к минимизации евклидовой нормы между такими точками:

$$d([p, q], [r, s]) = \min_{\lambda_1, \lambda_2 \in [0, 1]} \|x(\lambda_1) - y(\lambda_2)\| = \min_{\lambda_1, \lambda_2 \in [0, 1]} \|(1 - \lambda_1)p + \lambda_1q - [(1 - \lambda_2)r + \lambda_2s]\| \quad (11)$$

Введем обозначения: $a = q - p$; $b = s - r$; $c = p - r$. Тогда точку на первом отрезке можно привести к виду:

$$x(\lambda_1) = p + \lambda_1 a, \quad (12)$$

а точку на втором отрезке к:

$$y(\lambda_2) = r + \lambda_2 b. \quad (13)$$

Для поиска оптимальных значений λ_1 и λ_2 удобно перейти к минимизации квадрата нормы, поскольку минимум нормы и минимум её квадрата достигаются при одном и том же значении параметров. Это позволяет избежать производной от квадратного корня и упростить дальнейшие вычисления. В свою очередь квадрат нормы удобно представить как скалярное произведение вектора, получаемого при вычитании векторов точек $x(\lambda_1)$ и $y(\lambda_2)$:

$$\min_{\lambda_1, \lambda_2 \in [0, 1]} \|x(\lambda_1) - y(\lambda_2)\|^2 = \min_{\lambda_1, \lambda_2 \in [0, 1]} \langle \lambda_1 a - \lambda_2 b + c, \lambda_1 a - \lambda_2 b + c \rangle \quad (14)$$

Минимум достигается в точке, где градиент функции по обоим параметрам обращается в нуль. Это условие приводит к системе линейных уравнений, полученной дифференцированием по λ_1 и λ_2 :

$$\begin{cases} \langle a, a \rangle \lambda_1 - \langle a, b \rangle \lambda_2 = -\langle a, c \rangle \\ \langle a, b \rangle \lambda_1 - \langle b, b \rangle \lambda_2 = -\langle b, c \rangle \end{cases} \quad (15)$$

Система решается по формуле Крамера [25]. Обозначим определитель:

$$\Delta = \langle a, b \rangle^2 - \langle a, a \rangle \cdot \langle b, b \rangle. \quad (16)$$

Если $\Delta > \varepsilon$, где ε – малое положительное число, решение считается устойчивым и оптимальные значения λ_1^* и λ_2^* вычисляются как:

$$\lambda_1^* = \frac{\langle b, b \rangle \cdot \langle a, c \rangle - \langle a, b \rangle \cdot \langle b, c \rangle}{\Delta}, \quad (17)$$

$$\lambda_2^* = \frac{\langle a, a \rangle \cdot \langle b, c \rangle - \langle a, b \rangle \cdot \langle a, c \rangle}{\Delta}. \quad (18)$$

Значения параметров ограничиваются отрезком $[0, 1]$, что соответствует проекции точек на соответствующие отрезки:

$$\lambda_i^{\min} = \min(\max(\lambda_i^*, 0), 1), \quad i = 1, 2. \quad (19)$$

После подстановки проецированных параметров λ_1^{\min} , λ_2^{\min} значение расстояния между отрезками вычисляется по соответствующим точкам на каждом отрезке:

$$d([p, q], [r, s]) = \left\| (1 - \lambda_1^{\min})p + \lambda_1^{\min}q - [(1 - \lambda_2^{\min})r + \lambda_2^{\min}s] \right\|. \quad (20)$$

Если $\Delta < \varepsilon$, отрезки считаются почти параллельными, и аналитическое решение может быть численно неустойчивым. Также, если хотя бы одно из значений λ_i^{\min} , $i = 1, 2$ совпадает с границами интервала $[0, 1]$, то это указывает на граничный минимум, возникающий в результате проекции оптимальной точки за пределы отрезка и последующего приведения её к ближайшему концу.

В таком случае используется перебор всех пар граничных комбинаций: расстояние вычисляется между каждой из четырёх возможных пар граничных точек, и выбирается наименьшее значение:

$$d([p, q], [r, s]) = \min \{ \|p - r\|, \|p - s\|, \|q - r\|, \|q - s\| \}. \quad (21)$$

Таким образом, на каждом шаге построения маршрута выполняется проверка пересечений между формируемой капсулой и

всеми капсулами, ранее образованными отрезками воздушных коридоров других БПЛА. Эта процедура сводится к задаче минимизации квадрата евклидова расстояния между парой отрезков с учётом граничных и вырожденных случаев, после чего полученное расстояние сравнивается с суммой радиусов соответствующих капсул.

4. Определение пересечений воздушных коридоров с зонами обработки участков. При построении траекторий БПЛА необходимо учитывать ограничения, связанные с активными зонами обработки, в пределах которых нахождение других аппаратов недопустимо. Геометрически эти зоны аппроксимируются выпуклыми многогранниками в трёхмерном пространстве.

Каждая зона обработки $Z_j \subset \mathbb{R}^3$ задаётся в виде системы линейных неравенств, определяющих пересечение полупространств:

$$Z_j = \{x \in \mathbb{R}^3 : \langle n_i^j, x \rangle \leq c_i^j, \dots, \langle n_M^j, x \rangle \leq c_M^j\}, \quad (22)$$

где $n_i^j \in \mathbb{R}^3$ – нормаль к i -й грани многогранника (направленная наружу); $c_i^j \in \mathbb{R}$ – смещение соответствующей гиперплоскости, и M – общее число граней.

Проверка осуществляется для каждого сегмента маршрута $[p, q]$, которому соответствует капсула с радиусом R_i . Положение точки на осевой линии капсулы задаётся параметрически как:

$$x(\lambda) = (1 - \lambda)p + \lambda q, \quad (23)$$

где $\lambda \in [0, 1]$ определяет положение точки вдоль отрезка.

Для каждой i -й грани многогранника рассчитывается расстояние со знаком от текущей точки до соответствующей гиперплоскости:

$$d_i(\lambda) = \langle n_i^j, x(\lambda) \rangle - c_i^j = \alpha_i \lambda + \beta_i, \quad (24)$$

где $\alpha_i = \langle n_i^j, q - p \rangle$; $\beta_i = \langle n_i^j, p \rangle - c_i^j$.

Для фиксированного значения λ отклонение точки от многогранника определяется с использованием функции максимума расстояний до всех граней:

$$d_{\text{zone}}(\lambda) = \max_{1 \leq i \leq M} d_i(\lambda). \quad (25)$$

Геометрически это значение соответствует расстоянию от точки до границы зоны в том смысле, что:

- если $d_{\text{zone}}(\lambda) < 0$, то точка гарантированно лежит внутри многогранника;
- если $d_{\text{zone}}(\lambda) = 0$, то точка принадлежит его границе;
- если $d_{\text{zone}}(\lambda) > 0$, то точка вне зоны.

Функция $d_{\text{zone}}(\lambda)$ является кусочно-линейной, непрерывной и выпуклой на отрезке $[0, 1]$, поскольку максимум конечного набора линейных функций сохраняет выпуклость, что позволяет аналитически определить точку на осевой линии капсулы, наиболее близко расположенную к границам зоны обработки.

Минимум функции $d_{\text{zone}}(\lambda)$ достигается либо на концах отрезка, либо в точках пересечения двух функций $d_i(\lambda)$ и $d_j(\lambda)$, соответствующих смене активной грани, определяющей максимум:

$$\Lambda = \{0, 1\} \cup \left\{ \lambda = \frac{\beta_j - \beta_i}{\alpha_i - \alpha_j} \mid 1 \leq i < j \leq M, \alpha_i \neq \alpha_j, \lambda \in [0, 1] \right\}. \quad (26)$$

Для каждой $\lambda \in \Lambda$ вычисляется значение $d_{\text{zone}}(\lambda)$ и затем выбирается:

$$d_{\min} = \min_{\lambda \in \Lambda} d_{\text{zone}}(\lambda). \quad (27)$$

Полученное значение d_{\min} интерпретируется как расстояние между осевой линией капсулы и поверхностью зоны обработки.

Если выполняется условие:

$$d_{\min} \leq R_i, \quad (28)$$

то это означает, что капсула проникает в зону обработки, и соответствующий сегмент маршрута считается конфликтным.

Описанный алгоритм применим в предположении выпуклости зоны. В случае, если зона имеет невыпуклую структуру, её геометрия должна быть предварительно разложена на конечное число выпуклых компонент. Для этого используются алгоритмы выпуклой

декомпозиции [26]. Тогда каждая компонентная зона проверяется независимо, а результирующее значение d_{\min} принимается как минимум по всем подзонам.

5. Использование иерархической пространственной фильтрации для ускорения проверки пересечений. Вычислительная сложность проверки пересечений между сегментами маршрутов и зонами обработки растёт квадратично с увеличением числа БПЛА и элементов среды. Для снижения вычислительных затрат применяется иерархическая структура данных «свободное октодеревцо» (Loose Octree) [23] – модификация классического октодерева с расширенными границами ячеек.

Каждому сегменту маршрута, которому соответствует капсула, сопоставляется прямоугольный параллелепипед – ограничивающий объём, минимально охватывающий капсулу (Axis-Aligned Bounding Box, AABB) [27]. Аналогично, каждой зоне обработки сопоставляется прямоугольный ограничивающий параллелепипед, построенный по всем вершинам многогранника.

Координаты ограничивающего параллелепипеда определяются компонентно. Для сегмента маршрута $[p, q]$, которому соответствует капсула радиуса R_i , границы вычисляются как:

$$x_{\min}^{(ik)} = \min(p_k^x, p_{k+1}^x) - R_i, \quad (29)$$

$$x_{\max}^{(ik)} = \max(p_k^x, p_{k+1}^x) + R_i, \quad (30)$$

$$y_{\min}^{(ik)} = \min(p_k^y, p_{k+1}^y) - R_i, \quad (31)$$

$$y_{\max}^{(ik)} = \max(p_k^y, p_{k+1}^y) + R_i, \quad (32)$$

$$z_{\min}^{(ik)} = \min(p_k^z, p_{k+1}^z) - R_i, \quad (33)$$

$$z_{\max}^{(ik)} = \max(p_k^z, p_{k+1}^z) + R_i. \quad (34)$$

Для зоны обработки Z_j , заданной в виде выпуклого многогранника, координаты ограничивающего параллелепипеда определяются как минимум и максимум по координатам всех его вершин v_1^j, \dots, v_N^j :

$$x_{\min(j)} = \min(v_1^{j,x}, \dots, v_N^{j,x}), \quad (35)$$

$$x_{\max(j)} = \max(v_1^{j,x}, \dots, v_N^{j,x}), \quad (36)$$

$$y_{\min(j)} = \min(v_1^{j,y}, \dots, v_N^{j,y}), \quad (37)$$

$$y_{\max(j)} = \max(v_1^{j,y}, \dots, v_N^{j,y}), \quad (38)$$

$$z_{\min(j)} = \min(v_1^z, \dots, v_N^z), \quad (39)$$

$$z_{\max(j)} = \max(v_1^z, \dots, v_N^z). \quad (40)$$

В данном варианте октодера ограничивающие параллелепипеды объектов индексируются таким образом, что каждая ячейка покрывает определённую часть пространства, при этом вложенные объекты могут частично выходить за её границы. Такая организация позволяет избежать избыточного дробления пространства при работе с вытянутыми объектами, сохраняя эффективность индексации. Это обеспечивает быстрое отсеечение заведомо непересекающихся элементов и выполнение детальной проверки только для потенциально конфликтующих объектов.

Процесс отсеечения выполняется следующим образом: при добавлении нового сегмента маршрута сначала определяется набор ячеек октодера, с которыми пересекается его ограничивающий параллелепипед. Затем извлекаются только те объекты, которые находятся в этих ячейках или их родительских узлах. Объекты из непересекающихся ветвей дерева автоматически исключаются из проверки. Таким образом, вместо проверки всех пар объектов, алгоритм выполняет детальную проверку только для небольшого подмножества потенциально пересекающихся элементов.

Проверка пересечения между двумя ограничивающими объёмами выполняется по координатным интервалам. Два параллелепипеда считаются пересекающимися тогда и только тогда, когда выполнены следующие условия:

$$\text{Intersect}(\text{AABB}_1, \text{AABB}_2) \Leftrightarrow \begin{cases} x_{\min}^{(1)} \leq x_{\max}^{(2)} \wedge x_{\max}^{(1)} \geq x_{\min}^{(2)} \\ y_{\min}^{(1)} \leq y_{\max}^{(2)} \wedge y_{\max}^{(1)} \geq y_{\min}^{(2)} \\ z_{\min}^{(1)} \leq z_{\max}^{(2)} \wedge z_{\max}^{(1)} \geq z_{\min}^{(2)} \end{cases} \quad (41)$$

6. Алгоритм планирования маршрутов. Планирование маршрутов группы БПЛА организовано в два уровня. На верхнем уровне для каждого аппарата формируется полный маршрут, состоящий из четырёх фаз, соответствующих ключевым этапам движения: подъёму, входу в рабочую зону, выходу из неё и снижению. На нижнем уровне выполняется построение допустимого маршрута между заданной начальной и целевой точками, с учётом активных зон обработки и ранее зарезервированных маршрутов, проверяемых на пересечение с формируемыми капсулами. Такая организация обеспечивает независимость действий агентов, исключает необходимость глобальной синхронизации и позволяет реализовать пошаговое резервирование пространства.

Для каждого БПЛА маршрут формируется в виде последовательности четырех фаз, включающей:

1. Подъём от стартовой точки до рабочей высоты.
2. Горизонтальный переход к входу в зону обработки.
3. Возврат от выхода из зоны к точке начала снижения.
4. Спуск к исходной позиции.

Каждая из фаз строится как отдельный маршрут между двумя заданными точками с учётом геометрических ограничений. Построенные сегменты представляются в виде капсул фиксированного радиуса и регистрируются в октодереве. Псевдокод верхнего уровня представлен в Листинге 1.

Алгоритм реализует построение маршрутов для группы БПЛА по четырёхфазной схеме. На вход подаются: список агентов (UAVs), каждый из которых содержит координаты стартовой, входной и выходной точек, а также радиус безопасности капсулы; высота полёта (FlightAltitude); максимальное число итераций (MaxIterations); параметр точности достижения цели (Epsilon); вероятность выбора цели при генерации случайной точки (GoalBias) и зоны обработки (Zones) в виде многогранников.

```

function PlanPathsForUAVs(UAVs, FlightAltitude, MaxIterations, Epsilon,
GoalBias, Zones)
    LooseOctree := InitializeLooseOctree()

    // Индексация всех зон обработки
    for each Zone in Zones do
        ZoneAABB := ComputeAABB(Zone)
        InsertIntoLooseOctree(LooseOctree, Zone, ZoneAABB)
    end for

    for each UAV in UAVs do
        PBase := UAV.StartPoint
        PEntry := UAV.EntryPoint
        PExit := UAV.ExitPoint
        Radius := UAV.SafeRadius

        PTakeoff := (PBase.x, PBase.y, FlightAltitude)
        PEntryH := (PEntry.x, PEntry.y, FlightAltitude)
        PExitH := (PExit.x, PExit.y, FlightAltitude)

        // Фаза 1: Подъём
        Path1 := PlanPath(PBase, PTakeoff, MaxIterations,
Epsilon, GoalBias, Radius, LooseOctree)

        // Фаза 2: До входа в зону
        Path2 := PlanPath(PTakeoff, PEntryH, MaxIterations,
Epsilon, GoalBias, Radius, LooseOctree)

        // Фаза 3: Выход из зоны
        Path3 := PlanPath(PExitH, PTakeoff, MaxIterations,
Epsilon, GoalBias, Radius, LooseOctree)

        // Фаза 4: Снижение
        Path4 := Reverse(Path1)

        PathFinal := Concatenate(Path1, Path2, Path3, Path4)

        for each Segment in PathFinal do
            Capsule := CreateCapsule(Segment, Radius)
            CapsuleAABB := ComputeAABB(Capsule)
            InsertIntoLooseOctree(LooseOctree, Capsule,
CapsuleAABB)
        end for
        UAV.Path := PathFinal
    end for
    return UAVs
end function

```

Листинг 1. Псевдокод верхнего уровня планирования

На первом этапе все зоны индексируются в октодереве для последующей фильтрации. Далее для каждого агента последовательно формируются маршруты по четырём фазам, каждая из которых строится с использованием алгоритма нижнего уровня (PlanPath, Листинг 2). После завершения всех фаз полученная траектория преобразуется в набор капсул, регистрируемых в октодереве, и используется для проверки допустимости траекторий других агентов.

```

function PlanPath(PStart, PGoal, MaxIterations, Epsilon, GoalBias, Radius,
LooseOctree)
    Tree := {PStart}

    for i = 1 to MaxIterations do
        if Random() < GoalBias then
            PRand := PGoal
        else
            PRand := SampleUniform()
        end if

        PNear := FindNearest(Tree, PRand)
        PNew := Steer(PNear, PRand)

        Capsule := CreateCapsule(PNear, PNew, Radius)

        ZoneConflict := CheckZoneIntersection(LooseOctree,
Capsule)
        CorridorConflict := CheckCapsuleIntersection
(LooseOctree, Capsule)

        if not (ZoneConflict or CorridorConflict) then
            Tree := AddToTree(Tree, PNew)

            // Перестроение дерева: проверка новых рёбер
            Neighbors := FindNearbyNodes(Tree, PNew)
            for each PNeighbor in Neighbors do
                Cost(PNew) + Distance(PNew,
PNeighbor) < Cost(PNeighbor) then
                    NewCapsule :=
CreateCapsule(PNew, PNeighbor, Radius)
                    ZConflict :=
CheckZoneIntersection(LooseOctree, NewCapsule)
                    CConflict :=
CheckCapsuleIntersection(LooseOctree, NewCapsule)

```

```

                                if not (ZConflict or
CConflict) then
                                UpdateParent(Tree,
PNeighbor, PNew)
                                end if
                                end if
                                end for
                                if ReachedGoal(PNew, PGoal, Epsilon) then
                                return ExtractPath(Tree, PGoal)
                                end if
                                end if
                                end for
                                return EmptyPath
end function

```

Листинг 2. Псевдокод нижнего уровня планирования

На нижнем уровне используется алгоритм, основанный RRT [13] со следующими модификациями:

1. Использование капсульной модели маршрута, позволяющей формировать воздушные коридоры фиксированной толщины.

2. Проверка на конфликты с уже построенными маршрутами, основанная на пересечении капсул и ускоренная с применением октодерева.

3. Аналитическая проверка на пересечение с выпуклыми зонами обработки, реализуемая через оценку расстояния от капсулы до границ многогранника.

4. Локальное перестроение дерева RRT*, при котором каждая новая вершина дополнительно проверяется на допустимость при попытке переключения соседних узлов.

5. Механизм направленного семплирования [28]: с заданной вероятностью (GoalBias) в качестве следующей точки выбирается целевая, что способствует ускорению сходимости.

В результате формируется сеть маршрутов, обеспечивающих безопасные воздушные коридоры для каждого БПЛА, примеры которых приведены на рисунках 1 и 2.

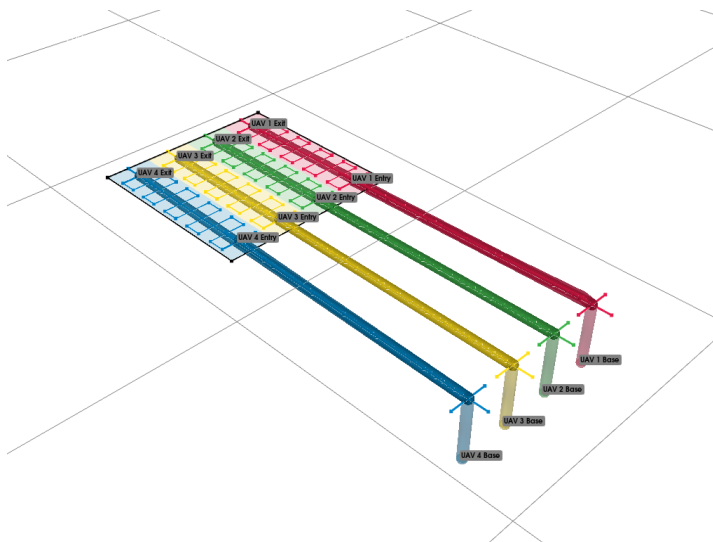


Рис. 1. Пример сети воздушных коридоров при разбиении поля на ряды для четырех БПЛА

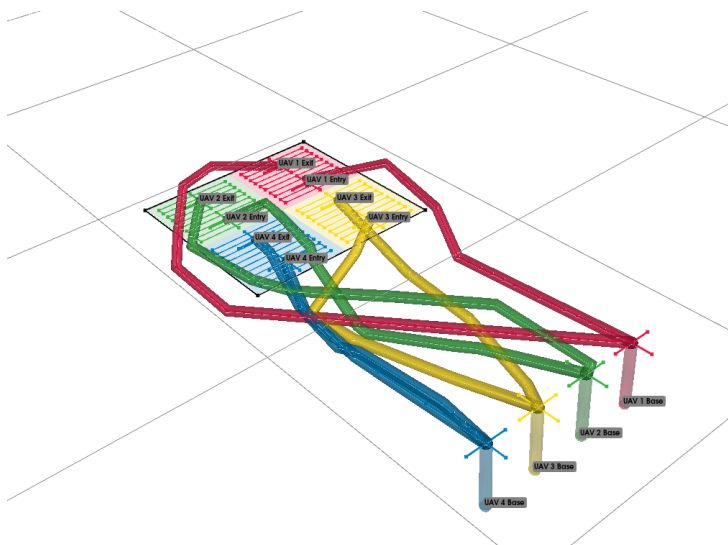


Рис. 2. Пример сети воздушных коридоров при сеточном разбиении поля для четырех БПЛА

Различие между рисунками демонстрирует влияние геометрии и параметра вероятностного выбора целевой точки на форму маршрутов: при линейном разбиении (рисунок 1) траектории близки к прямым благодаря упорядоченности расположения входов и выходов, тогда как при сеточной топологии (рисунок 2) из-за увеличения плотности и числа пересечений капсулы отклоняются от кратчайших направлений, формируя более разреженную структуру.

7. Вычислительные эксперименты. Для оценки работоспособности и вычислительной эффективности предложенного метода был реализован программный прототип на языке C++17 с использованием стандартной библиотеки и собственной реализации структуры октодеревя. Все эксперименты проводились на рабочей станции под управлением Ubuntu 22.04 с процессором Intel Core i7-12700H и 16 Гб оперативной памяти.

В качестве базовой сцены использовался прямоугольный сельскохозяйственный участок площадью 1 км², равномерно разбиваемый на прямоугольную сетку зон обработки по числу БПЛА. При этом количество строк в сетке выбирается как $\lfloor \sqrt{N} \rfloor$, а число

столбцов как $\left\lceil \frac{N}{\lfloor \sqrt{N} \rfloor} \right\rceil$, где N – общее число БПЛА. Каждому

аппарату сопоставляется отдельный подучасток, центр которого используется для задания входной и выходной точек.

Область планирования расширяется до 1.5 км по каждой горизонтальной оси, чтобы обеспечить необходимый запас пространства для формирования маршрутов с резервированием. Вертикальная граница фиксирована на уровне 25 м, что соответствует типичной высоте выполнения сельскохозяйственных задач.

Каждый маршрут строится по четырёхфазной схеме, описанной в разделе 6, а сегменты маршрута интерпретируются в виде капсул радиуса 1 м. Это значение отражает типичные габариты квадрокоптеров с добавленным зазором для безопасного пролёта.

Параметры алгоритма подобраны экспериментально для обеспечения баланса между качеством решения и вычислительной эффективностью. Экспериментальные параметры приведены в таблице 1.

Таблица 1. Экспериментальные параметры

Параметр	Значение	Обоснование
Площадь участка	1 км ²	Типичный размер сельскохозяйственного поля
Размер области планирования	1.5 × 1.5 × 0.025 км ³	Расширение для маневрирования
Радиус капсулы	1 м	Габариты БПЛА + запас безопасности
Шаг расширения RRT*	7 м	Компромисс точность/скорость
Радиус локального соединения	15 м	Охват соседних узлов без перегрузки
Максимум итераций на фазу	30000	Практический предел сходимости
Вероятность выбора цели	0.45	Баланс направленного/случайного поиска

Для обеспечения статистической достоверности результатов и оценки устойчивости алгоритма к случайным факторам инициализации каждая конфигурация тестировалась 25 раз.

В ходе численного эксперимента анализировались характеристики, отражающие поведение алгоритма в зависимости от размера группы агентов. В качестве метрик оценки эффективности выбраны:

1. Среднее время построения маршрута на одного агента, как показатель вычислительной нагрузки.
2. Среднее число итераций до завершения планирования, отражающее сложность конфигурационного пространства.
3. Длина полученного маршрута, характеризующая влияние ограничений на форму траектории.

Графики на рисунках 3 и 4 показывают, как изменяются эти показатели при увеличении числа агентов от 2 до 32.

Среднее и стандартное отклонение метрик (25 экспериментов)

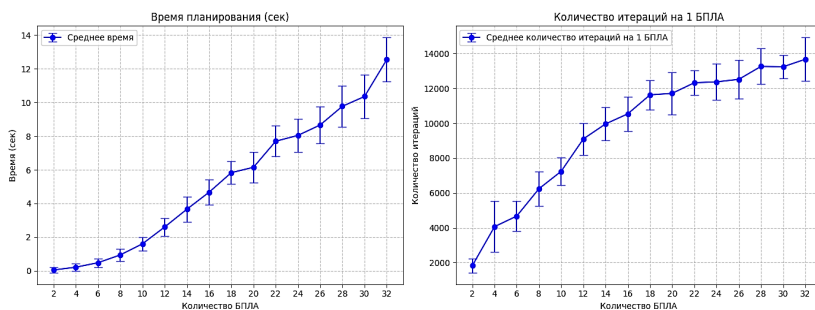


Рис. 3. Зависимости времени выполнения и количества итераций от количества БПЛА

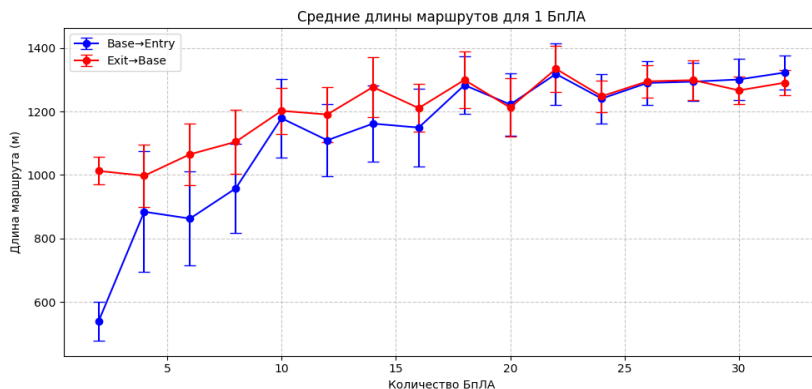


Рис. 4. Зависимость средних длин маршрутов для 1 БЛА от размера группы

Как и ожидалось, рост числа БЛА приводит к усложнению планирования: время увеличивается с 0.05 с до 12.5 с, а среднее число итераций – с 1800 до более чем 13 000. Это объясняется тем, что каждый новый маршрут строится в уже частично занятом пространстве, и каждый шаг планирования требует больше проверок на пересечения.

Длина маршрутов также постепенно увеличивается, так как новые БЛА вынуждены обходить уже построенные воздушные коридоры. Визуализации на рисунке 5 демонстрируют, как плотность маршрутов возрастает при увеличении числа агентов: свободное пространство постепенно вытесняется резервированными капсулами, оставляя всё меньше направлений для расширения дерева.

Важно отметить, что несмотря на рост вычислительной нагрузки, метод сохраняет устойчивость и успешно масштабируется до 32 агентов без существенного ухудшения качества маршрутов. При высокой плотности препятствий эффективность иерархического представления среды, основанного на октодереве, позволяет значительно сократить количество проверок на коллизии и обеспечить приемлемое вычислительное время даже в условиях высокой загруженности пространства.

Таким образом, проведённые эксперименты подтверждают применимость метода для групповой навигации БЛА в ограниченном воздушном пространстве. Результаты показывают, что предложенный подход сохраняет разумный баланс между точностью, стабильностью и вычислительной сложностью.

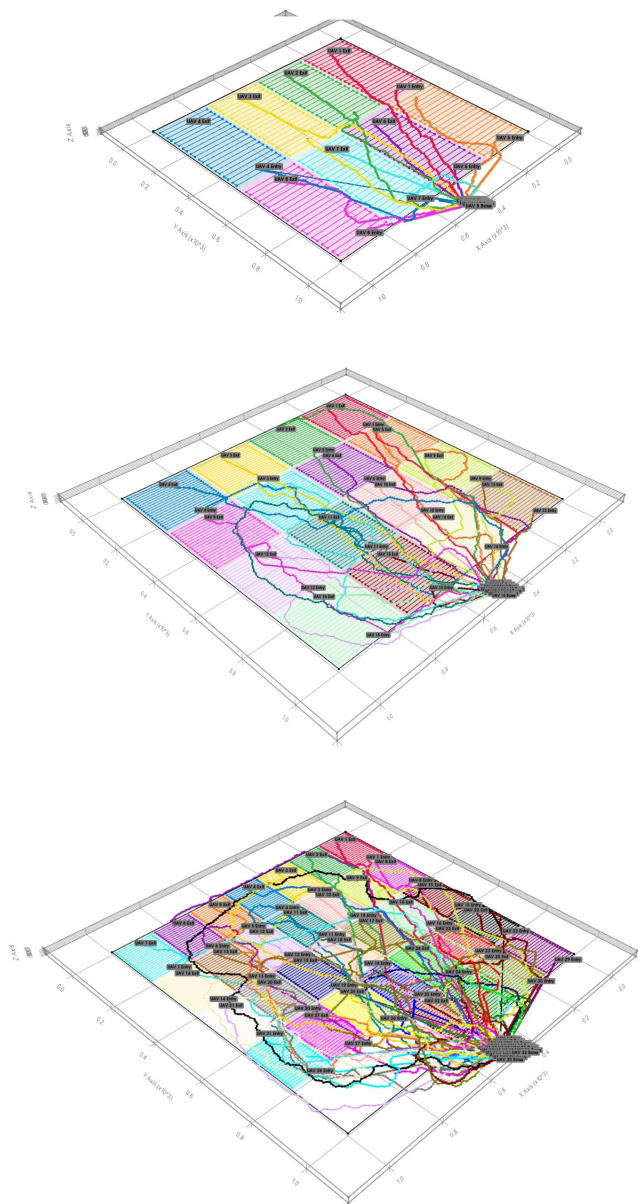


Рис. 5. Рост плотности коридорных маршрутов по мере увеличения числа БПЛА в группе на примере 8,16 и 32 БПЛА

8. Заключение. В данном исследовании представлен метод построения безопасных маршрутов группы БПЛА на основе капсульной модели воздушных коридоров, обеспечивающий пространственное резервирование траекторий в условиях плотного многоагентного движения. Предложенный метод решает ключевые проблемы групповой навигации БПЛА в агропромышленных применениях: обеспечивает автономность выполнения полётов, устойчивость к навигационным ошибкам и внешним воздействиям, независимость от централизованной координации в реальном времени.

Реализация на базе модифицированного алгоритма RRT* с интегрированными геометрическими проверками и иерархической пространственной индексацией демонстрирует вычислительную эффективность и масштабируемость.

Экспериментальные результаты подтвердили, что метод масштабируется до 32 БПЛА без значительной деградации качества маршрутов и при сохранении вычислительной эффективности. Время планирования растёт предсказуемо, без катастрофического ухудшения производительности.

Ключевым преимуществом метода является конструктивное обеспечение безопасности через геометрическое резервирование пространства, что исключает необходимость в сложных динамических моделях, онлайн-синхронизации и высокоточной навигации. Построенные маршруты могут быть напрямую загружены в управляющее программное обеспечение БПЛА в виде последовательности контрольных точек, обеспечивая полную автономность миссии.

Дальнейшее развитие метода связано с реализацией механизмов динамического освобождения воздушных коридоров после прохождения участков маршрута, что позволит повысить эффективность использования воздушного пространства при многократных полётах и увеличенных размерах групп БПЛА.

Рафаэль Мидхатович Юсупов навсегда останется в памяти как выдающийся учёный, мудрый наставник и человек высокой ответственности. Его путь – от военного инженера-исследователя до директора Санкт-Петербургского института информатики и автоматизации РАН – стал образцом преданности делу и служения стране. Более четверти века, возглавляя институт, он укреплял его научный потенциал и развивал новые направления в области информатики и теории управления. Обладая редкой глубиной мышления и организаторским даром, он создал научную школу, подготовил десятки учеников и внёс весомый вклад в развитие

информационного общества. Светлая память о Рафаэле Мидхатовиче – как о талантливом руководителе и наставнике – навсегда сохранится в сердцах коллег, учеников и всех, кому посчастливилось работать с ним.

Литература

1. Gubanov B., Lebedeva V., Lebedev I., Astapova M. Algorithms and software for evaluation of plant height in vertical farm using UAVs // *Agriculture Digitalization and Organic Production: Proceedings of the Second International Conference*. Singapore: Springer Nature Singapore, 2022. pp. 351–362. DOI: 10.1007/978-981-19-7780-0_31.
2. Шайтура Н.С. Мониторинг сельскохозяйственных земель при помощи беспилотных летательных аппаратов // *Практические аспекты применения современных беспилотных летательных аппаратов*. 2022. С. 46–57.
3. Nikolakopoulos I.A., Petropoulos G.P. Obtaining a Land Use/Cover Cartography in a Typical Mediterranean Agricultural Field Combining Unmanned Aerial Vehicle Data with Supervised Classifiers // *Land*. 2025. vol. 14. no. 3. DOI: 10.3390/land14030643.
4. Павлова А.И. Применение беспилотных и геоинформационных технологий для цифрового почвенного картографирования // *Информационные технологии, системы и приборы в АПК. Агроинфо-2021*. Т. 2021. С. 97–99.
5. Hassler S.C., Baysal-Gurel F. Unmanned aircraft system (UAS) technology and applications in agriculture // *Agronomy*. 2019. vol. 9. no. 10. DOI: 10.3390/agronomy9100618.
6. Velusamy P., Rajendran S., Mahendran R.K., Naseer S., Shafiq M., Choi J.G. Unmanned Aerial Vehicles (UAV) in precision agriculture: Applications and challenges // *Energies*. 2021. vol. 15. no. 1. DOI: 10.3390/en15010217.
7. Makam S., Komatineni B.K., Meena S.S., Meena U. Unmanned aerial vehicles (UAVs): an adoptable technology for precise and smart farming // *Discover Internet of Things*. 2024. vol. 4. DOI: 10.1007/s43926-024-00066-5.
8. Sharon G., Stern R., Felner A., Sturtevant N. R. Conflict-based search for optimal multi-agent pathfinding // *Artificial Intelligence*. 2015. vol. 219. pp. 40–66. DOI: 10.1016/j.artint.2014.11.006.
9. Wang Z., Zhang Z., Dou W., Hu G., Zhang L., Zhang M. Extending Conflict-Based Search for Optimal and Efficient Quadrotor Swarm Motion Planning // *Drones*. 2024. vol. 8. no. 12. DOI: 10.3390/drones8120719.
10. Andreychuk A., Yakovlev K., Boyarski E., Stern R. Improving continuous-time conflict based search // *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021. vol. 35. no. 13. pp. 11220–11227. DOI: 10.1609/aaai.v35i13.17338.
11. Liu X., Su Y., Wu Y., Guo Y. Multi-conflict-based optimal algorithm for multi-UAV cooperative path planning // *Drones*. 2023. vol. 7. no. 3. DOI: 10.3390/drones7030217.
12. Cap M., Novak P., Vokrinek J., Pechoucek M. Multi-agent RRT*: Sampling-based cooperative pathfinding // *arXiv preprint arXiv:1302.2828*. 2013.
13. Karaman S., Frazzoli E. Sampling based algorithms for optimal motion planning // *The International Journal of Robotics Research*. 2011. vol. 30. no. 7. pp. 846–894. DOI: 10.1177/0278364911406761.
14. Silver D. Cooperative pathfinding // *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. 2005. vol. 1. no. 1. pp. 117–122. DOI: 10.1609/aiide.v1i1.18726.
15. Barer M., Sharon G., Stern R., Felner A. Suboptimal variants of the conflict based search algorithm for the multi agent pathfinding problem // *Proceedings of the International Symposium on Combinatorial Search*. 2021. vol. 5. no. 1. pp. 19–27. DOI: 10.1609/socs.v5i1.18315.
16. Van Den Berg J., Guy S.J., Lin M., Manocha D. Reciprocal n-body collision avoidance // *Robotics Research: The 14th International Symposium ISRR*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. pp. 3–19.

17. Van den Berg J., Lin M., Manocha D. Reciprocal velocity obstacles for real-time multi-agent navigation // IEEE International Conference on Robotics and Automation. 2008. pp. 1928–1935. DOI: 10.1109/ROBOT.2008.4543489.
18. Yan X., Jiang D., Miao R., Li Y. Formation control and obstacle avoidance algorithm of a multi-USV system based on virtual structure and artificial potential field // Journal of Marine Science and Engineering. 2021. vol. 9. no. 2. DOI: 10.3390/jmse9020161.
19. Wu W., Zhang X., Miao Y. Starling behavior inspired flocking control of fixed wing unmanned aerial vehicle swarm in complex environments with dynamic obstacles // Biomimetics. 2022. vol. 7. no. 4. DOI: 10.3390/biomimetics7040214.
20. Alqudsi Y. Synchronous task allocation and trajectory optimization for autonomous drone swarm // 1st International Conference on Emerging Technologies for Dependable Internet of Things (ICETI). 2024. pp. 1–8. DOI: 10.1109/ICETI63946.2024.10777195.
21. Котов Д.В., Лебедев О.Б. Управление передвижением группы БПЛА с соблюдением геометрической структуры строя на основе альтернативной коллективной адаптации // Известия ЮФУ. Технические науки. 2024. Т. 1. С. 155–167.
22. Саенко И.Б., Лаута О.С., Митяков Е.С., Соколов А.П. Алгоритм роевого управления БПЛА с элементами кластерного анализа // Информация и космос. 2024. № 4. С. 68–75.
23. Ulrich T. Loose octrees // Game Programming Gems. 2000. vol. 1. pp. 434–442.
24. Pournin L., Weber M., Tsukahara M., Ferrez J.A., Ramaioli M., Liebling T.M. Three dimensional distinct element simulation of spherocylinder crystallization // Granular Matter. 2005. vol. 7. pp. 119–126. DOI: 10.1007/s10035-004-0188-4.
25. Bretscher O. Linear algebra with applications. NJ: Prentice Hall, 1997. 587 p.
26. Lien J.M., Amato N.M. Approximate convex decomposition of polyhedra and its applications // Computer Aided Geometric Design. 2008. vol. 25. no. 7. pp. 503–522.
27. Rendering W.P.B. Physically based rendering // Procedia IUTAM. 2015. vol. 13. pp. 3.
28. Kang G., Kim Y.B., You W.S., Lee Y.H., Oh H.S., Moon H., Choi H.R. Sampling based path planning with goal oriented sampling // IEEE International Conference on Advanced Intelligent Mechatronics (AIM). 2016. pp. 1285–1290. DOI: 10.1109/AIM.2016.7576947.

Савельев Антон Игоревич — канд. техн. наук, заведующий лабораторией, лаборатория автономных робототехнических средств, Федеральное государственное бюджетное учреждение науки «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН). Область научных интересов: математические модели управления робототехническими средствами, методы и алгоритмы для построения траекторий движения робототехнических систем. Число научных публикаций — 151. saveliev@ias.spb.su; 14-я линия В.О., 39, 199178, Санкт-Петербург, Россия; р.т.: +7(812)328-4743.

Аникин Дмитрий Андреевич — младший научный сотрудник, лаборатория автономных робототехнических средств, Федеральное государственное бюджетное учреждение науки «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН). Область научных интересов: алгоритмы группового управления наземными и воздушными роботами, алгоритмы для автоматической навигации, обнаружения препятствий и принятия решений в реальном времени. Число научных публикаций — 11. anikin.d@ias.spb.su; 14-я линия В.О., 39, 199178, Санкт-Петербург, Россия; р.т.: +7(812)328-4743.

Поддержка исследований. Работа выполнена в рамках бюджетной темы № FFZF-2025-0012.

A. SAVELIEV, D. ANIKIN
**METHOD OF CALCULATING CAPSULE-SHAPED AIR
CORRIDORS OF SAFE ROUTES FOR A GROUP OF UNMANNED
AERIAL VEHICLES**

Saveliev A., Anikin D. Method of Calculating Capsule-Shaped Air Corridors of Safe Routes for a Group of Unmanned Aerial Vehicles.

Abstract. The paper considers the problem of constructing safe routes for a group of unmanned aerial vehicles in a limited airspace over an agricultural area. The relevance of the study is due to the growing use of UAV groups in the agro-industrial complex for monitoring, mapping, and processing fields, which requires ensuring flight safety in conditions of high air traffic density, limited communication, and exposure to external factors. A particular challenge is the need for autonomous missions in the presence of navigation errors and natural impacts. A route planning method is proposed based on representing the trajectory of each device as a capsule air corridor – a three-dimensional volume of a fixed radius formed along the trajectory segments. Spatial redundancy ensures safe spacing of trajectories at the planning stage, eliminating conflicts during subsequent autonomous flight operations without the need for continuous coordination between agents. The capsule radius includes a reserve for possible deviations from the planned trajectory, which ensures resistance to navigation errors. The method is based on the sequential formation of routes for each device according to a four-phase scheme, including a vertical ascent from the starting point to the operating altitude, a horizontal transition to the entrance to the processing zone, a return from the exit from the zone to the starting point of the descent, and a vertical descent to the initial position. Each new route is built considering the already reserved air corridors through an analytical check of geometric intersections between the capsules of different trajectories and convex polyhedrons of the processing zones. To improve computational efficiency, hierarchical spatial filtering is used based on bounding parallelepipeds, which allows for the rapid cutting off of obviously non-intersecting objects at the preliminary stage and performing an accurate geometric check only for potentially conflicting route segments. Numerical experiments were carried out for groups of 2 to 32 devices on a typical agricultural plot of one square kilometer. A nonlinear increase in the planning time and the number of iterations with an increase in the number of agents was found, which is due to the need to build each subsequent route in an already partially occupied space with an increasing number of spatial constraints. The length of routes shows a tendency to increase, especially pronounced at the initial stages of scaling, which is associated with the need to bypass already reserved air corridors.

Keywords: UAVs, air corridors, agricultural tasks, Rapidly-Exploring Random Tree*, Loose Octree, autonomous navigation, group planning.

References

1. Gubanov B., Lebedeva V., Lebedev I., Astapova M. Algorithms and software for evaluation of plant height in vertical farm using UAVs. Agriculture Digitalization and Organic Production: Proceedings of the Second International Conference. Singapore: Springer Nature Singapore, 2022. pp. 351–362. DOI: 10.1007/978-981-19-7780-0_31.
2. Shaitura N.S. [Monitoring agricultural lands using unmanned aerial vehicles]. *Prakticheskie aspekty primeneniya sovremennykh bespilotnykh letatel'nykh apparatov* [Practical aspects of the use of modern unmanned aerial vehicles]. 2022. pp. 46–57. (In Russ.).

3. Nikolakopoulos I.A., Petropoulos G.P. Obtaining a Land Use/Cover Cartography in a Typical Mediterranean Agricultural Field Combining Unmanned Aerial Vehicle Data with Supervised Classifiers. *Land*. 2025. vol. 14. no. 3. DOI: 10.3390/land14030643.
4. Pavlova A.I. [Application of unmanned and geoinformation technologies for digital soil mapping]. *Informacionnye tehnologii, sistemy i pribory v APK. Agroinfo-2021 [Information technologies, systems and devices in agroindustrial complex. Agroinfo-2021]*. vol. 2021. pp. 97–99. (In Russ.).
5. Hassler S.C., Baysal-Gurel F. Unmanned aircraft system (UAS) technology and applications in agriculture. *Agronomy*. 2019. vol. 9. no. 10. DOI: 10.3390/agronomy9100618.
6. Velusamy P., Rajendran S., Mahendran R.K., Naseer S., Shafiq M., Choi J.G. Unmanned Aerial Vehicles (UAV) in precision agriculture: Applications and challenges. *Energies*. 2021. vol. 15. no. 1. DOI: 10.3390/en15010217.
7. Makam S., Komatineni B.K., Meena S.S., Meena U. Unmanned aerial vehicles (UAVs): an adoptable technology for precise and smart farming. *Discover Internet of Things*. 2024. vol. 4. DOI: 10.1007/s43926-024-00066-5.
8. Sharon G., Stern R., Felner A., Sturtevant N. R. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence*. 2015. vol. 219. pp. 40–66. DOI: 10.1016/j.artint.2014.11.006.
9. Wang Z., Zhang Z., Dou W., Hu G., Zhang L., Zhang M. Extending Conflict-Based Search for Optimal and Efficient Quadrotor Swarm Motion Planning. *Drones*. 2024. vol. 8. no. 12. DOI: 10.3390/drones8120719.
10. Andreychuk A., Yakovlev K., Boyarski E., Stern R. Improving continuous-time conflict based search. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021. vol. 35. no. 13. pp. 11220–11227. DOI: 10.1609/aaai.v35i13.17338.
11. Liu X., Su Y., Wu Y., Guo Y. Multi-conflict-based optimal algorithm for multi-UAV cooperative path planning. *Drones*. 2023. vol. 7. no. 3. DOI: 10.3390/drones7030217.
12. Cap M., Novak P., Vokrinek J., Pechoucek M. Multi-agent RRT*: Sampling-based cooperative pathfinding. *arXiv preprint arXiv:1302.2828*. 2013.
13. Karaman S., Frazzoli E. Sampling based algorithms for optimal motion planning. *The International Journal of Robotics Research*. 2011. vol. 30. no. 7. pp. 846–894. DOI: 10.1177/0278364911406761.
14. Silver D. Cooperative pathfinding. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. 2005. vol. 1. no. 1. pp. 117–122. DOI: 10.1609/aiide.v1i1.18726.
15. Barer M., Sharon G., Stern R., Felner A. Suboptimal variants of the conflict based search algorithm for the multi agent pathfinding problem. *Proceedings of the International Symposium on Combinatorial Search*. 2021. vol. 5. no. 1. pp. 19–27. DOI: 10.1609/socs.v5i1.18315.
16. Van Den Berg J., Guy S.J., Lin M., Manocha D. Reciprocal n-body collision avoidance. *Robotics Research: The 14th International Symposium ISRR*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. pp. 3–19.
17. Van den Berg J., Lin M., Manocha D. Reciprocal velocity obstacles for real-time multi-agent navigation. *IEEE International Conference on Robotics and Automation*. 2008. pp. 1928–1935. DOI: 10.1109/ROBOT.2008.4543489.
18. Yan X., Jiang D., Miao R., Li Y. Formation control and obstacle avoidance algorithm of a multi-USV system based on virtual structure and artificial potential field. *Journal of Marine Science and Engineering*. 2021. vol. 9. no. 2. DOI: 10.3390/jmse9020161.
19. Wu W., Zhang X., Miao Y. Starling behavior inspired flocking control of fixed wing unmanned aerial vehicle swarm in complex environments with dynamic obstacles. *Biomimetics*. 2022. vol. 7. no. 4. DOI: 10.3390/biomimetics7040214.

20. Alqudsi Y. Synchronous task allocation and trajectory optimization for autonomous drone swarm. 1st International Conference on Emerging Technologies for Dependable Internet of Things (ICETI). 2024. pp. 1–8. DOI: 10.1109/ICETI63946.2024.10777195.
21. Kotov D.V., Lebedev O.B. [Control of movement of a group of UAVs with observance of the geometric structure of the formation based on alternative collective adaptation]. *Izvestija JuFU. Tehnicheskie nauki – Bulletin of SFedU. Technical sciences*. 2024. vol. 1. pp. 155–167. (In Russ.).
22. Saenko I.B., Lauts O.S., Mityakov E.S., Sokolov A.P. [Algorithm for swarm control of UAVs with elements of cluster analysis]. *Informacija i kosmos – Information and Space*. 2024. no. 4. pp. 68–75. (In Russ.).
23. Ulrich T. Loose octrees. *Game Programming Gems*. 2000. vol. 1. pp. 434–442.
24. Pournin L., Weber M., Tsukahara M., Ferrez J.A., Ramaioli M., Liebling T.M. Three dimensional distinct element simulation of spherocylinder crystallization. *Granular Matter*. 2005. vol. 7. pp. 119–126. DOI: 10.1007/s10035-004-0188-4.
25. Bretscher O. *Linear algebra with applications*. NJ: Prentice Hall, 1997. 587 p.
26. Lien J.M., Amato N.M. Approximate convex decomposition of polyhedra and its applications. *Computer Aided Geometric Design*. 2008. vol. 25. no. 7. pp. 503–522.
27. Rendering W.P.B. Physically based rendering. *Procedia IUTAM*. 2015. vol. 13. pp. 3.
28. Kang G., Kim Y.B., You W.S., Lee Y.H., Oh H.S., Moon H., Choi H.R. Sampling based path planning with goal oriented sampling. *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. 2016. pp. 1285–1290. DOI: 10.1109/AIM.2016.7576947.

Saveliev Anton — Ph.D., Head of the laboratory, Laboratory of autonomous robotic means, St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS). Research interests: mathematical models of robotic means control, methods and algorithms for constructing trajectories of robotic systems. The number of publications — 151. saveliev@ias.spb.su; 39, 14th Line of V.O., 199178, St. Petersburg, Russia; office phone: +7(812)328-4743.

Anikin Dmitry — Junior researcher, Laboratory of autonomous robotic means, St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS). Research interests: algorithms for group control of ground and air robots, algorithms for automatic navigation, obstacle detection and decision-making in real time. The number of publications — 11. anikin.d@ias.spb.su; 39, 14th Line of V.O., 199178, St. Petersburg, Russia; office phone: +7(812)328-4743.

Acknowledgements. This research was carried out within the framework of budget theme No. FFZF-2025-0012.