

# Математика Математическое МОДЕЛИРОВАНИЕ

Сетевое научное издание

<http://mathmelpub.ru>

Ссылка на статью:

// Математика и математическое моделирование.  
2018. № 01. С. 15–58

DOI: [10.24108/mathm.0118.0000098](https://doi.org/10.24108/mathm.0118.0000098)

Представлена в редакцию: 06.01.2018

© НП «НЕИКОН»

УДК 517.977, 519.688

## Методы планирования пути в среде с препятствиями (обзор)

Лю В.<sup>1,\*</sup>

[\\*chenub200711@163.com](mailto:chenub200711@163.com)

<sup>1</sup>МГТУ им. Н.Э. Баумана, Москва, Россия

---

Обзор посвящен методам построения пути беспилотного аппарата в среде с препятствиями. Рассмотрены пять классов методов: методы на основе графов и клеточной декомпозиции, использование потенциальных полей, оптимизи-зационные методы, методы на основе интеллектуальных технологий. Многие алгоритмы в качестве результата дают дискретную последовательность точек, называемых опорными. Возникает задача сглаживания пути — построение гладкой траектории, проходящей через цепь опорных точек. Этой задаче также уделено внимание.

**Ключевые слова:** планирование пути, среда с препятствиями, робот, дорожная карта

---

### Введение

Планирование пути — важнейшая задача в области навигации мобильных роботов. Эта задача включает в основном три аспекта. Во-первых, спланированный путь должен пролегать от заданной начальной точки к заданной конечной точке. Во-вторых, этот путь должен обеспечивать движение робота с обходом возможных препятствий. В-третьих, путь должен среди всех возможных путей, удовлетворяющих первым двум требованиям, быть в определенном смысле оптимальным.

Методы планирования пути можно классифицировать по разным признакам. В контексте использования интеллектуальных технологий их можно разделить на традиционные методы и эвристические методы. По характеру окружающей обстановки можно разделить методы планирования на методы планирования в статической окружающей среде и в динамической среде (следует, однако, отметить, что статическая окружающая среда редко встречается на практике). Методы также можно разделить по полноте информации об окружающей среде: методы с полной информацией (в таком случае говорят о глобальном планировании пути) и методы с неполной информацией (обычно речь идет о знании обстановки в непосредственной близости от робота, в этом случае речь идет о локальном планировании пути). Отметим, что неполная информация об окружающей среде может

быть следствием меняющейся обстановки, т.е. в условиях динамической среды планирование пути, как правило, локальное.

В литературе предложено большое количество методов планирования пути, в которых используются различные эвристические приемы, вытекающие, как правило, из содержательного смысла решаемой задачи. В настоящей работе мы ограничимся рассмотрением основных подходов к решению задачи. На рис. 1 показана классификация указанных подходов.



Рис. 1. Классификация методов планирования пути

Отметим методы с использованием карты окружающей среды или описания ее с помощью графа или дерева; методы на основе клеточной декомпозиции; методы потенциальных полей; оптимизационные методы; методы, основанные на интеллектуальных технологиях, включая поведенческие методы. Во многих этих методах результатом оказывается цепь опорных точек (путевых точек), соединяющая начало и конец пути, и тогда возникает задача сглаживания полученного пути, которой также уделено определенное внимание.

### Методы на основе графов

В последние годы этот класс методов привлек заметный интерес исследователей. Основные методы этого класса показаны на рис.2 граф (или дерево) отражает состояния, в которых может находиться робот: каждый узел представляет одно состояние робота. Состоянием может быть положение, угол ориентации, скорость или ускорение робота. Переходы между состояниями характеризуются функцией затрат. Это позволяет выделить путь, который имеет минимальную общую стоимость достижения целевого состояния.

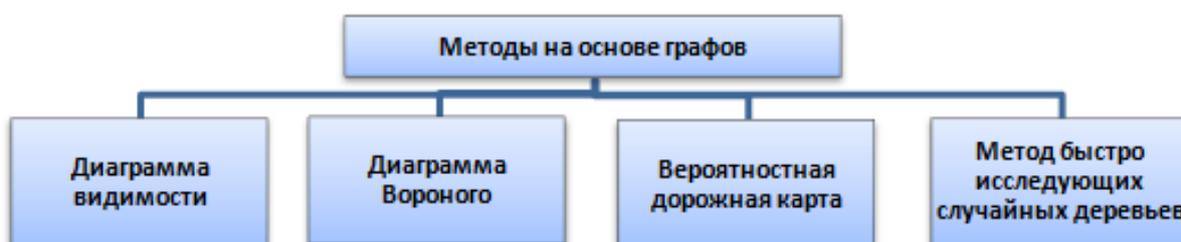


Рис. 2. Методы на основе графов

Предложен ряд методов, в основном отличающихся выбором узлов соответствующего графа. Подобные методы обычно используют в тех случаях, когда информация об окружающей среде является статической и известна полностью.

Напомним, что граф представляет собой непустое множество  $V$  вершин, определенные пары которых соединены дугами или рёбрами. На дугах задано одно из двух возможных направлений, граф с дугами называют ориентированным (орграфом), в то время как на ребрах направление не выделено, граф с ребрами называют неориентированным (рис. 3). Граф обозначают  $G = (V, E)$ .

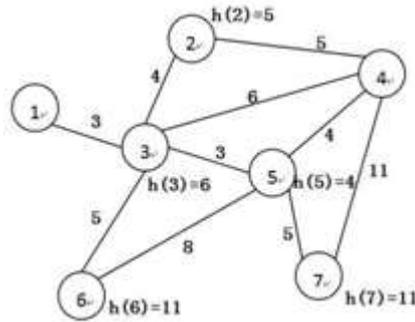


Рис.3. Взвешенный граф  $G$

Построение графа тесно связано с порядком выбора опорных точек. Типичный пример здесь — метод на основе *диаграммы видимости*. Определим диаграмму видимости как неориентированный граф  $Vg(N, L)$ , в котором  $N = V \cup \{S_0, S_f\}$  — непустое множество узлов, включающее множество вершин препятствий, начальную точку  $S_0$  и конечную точку  $S_f$  планируемого пути,  $L$  — множество рёбер, определяемых всевозможными парами точек из множества  $N$ , которые соединяются отрезком прямой, не пересекающейся с препятствиями. Для использования диаграммы видимости необходимо, чтобы препятствия имели форму многоугольника (двумерный случай) или многогранника (трехмерный случай). Поскольку путь строится по вершинам препятствий, и часть пути совпадает с краями препятствий, существует определенная опасность столкновения с препятствиями. Кроме того, при увеличении количества препятствий возрастает сложность графа, причем этот рост очень быстрый. Поэтому здесь наиболее важной задачей становятся приемы сокращения сложности графа видимости. Huang [1] предложил *метод динамической диаграммы видимости*, а Janet ввел концепцию *m-вектора* (traversability vector) [2]. Эти подходы направлены на то, чтобы определить, какие препятствия в полной окружающей среде необходимо учитывать во время движения робота, а какие не должны рассматриваться. При этом указанные подходы могут привести к тому, что полученный путь не будет являться оптимальным, особенно в трехмерном случае.

Иной подход дают *метод структурирования свободного пространства* (free space structuring method) [3] и метод на основе *диаграммы Вороного* [4]. В методе структурирования свободного пространства, реализованного в плоском случае, выделяются так называемые свободные звенья (free links) — отрезки, соединяющие вершины препятствий и не

пересекающиеся с препятствиями. Из свободных звеньев собираются выпуклые многогранники, описывающие области свободного пространства. Эти данные оформляются в виде специального графа, называемого MAKLINK, а спланированный путь в обход препятствий строится алгоритмами поиска путей в графе. Использование диаграмм Вороного также приводит к построению допустимого пути, максимально удаленного от препятствий, при этом обеспечивает дополнительные возможности с точки зрения позиционирования робота. Методы структурирования свободного пространства и диаграммы Вороного дают более безопасный путь с точки зрения столкновения с препятствиями, но общая длина пути увеличивается.

Опорные (путевые) точки можно выбирать в свободной части пространства случайным образом. Эта идея привела к зарождению нового класса методов — вероятностных. Исторически первым был предложен метод вероятностной дорожной карты (ВДК) [5]. Основная идея метода состоит в случайном выборе точек, распределенных с некоторой вероятностью в пространстве. Распределение выбирается во всем пространстве, включая препятствия, при выборе точек попавшие на препятствия отбрасываются. Случайный выбор путевых точек может привести к тому, что построенный путь не будет оптимальным [6].

Построение графа состоит не только в выборе путевых точек, но и в выборе ребер, их соединяющих. Ребром соединяются две путевые точки, если из одной в другую можно попасть по прямолинейному пути, минуя препятствия. Для построения всех ребер необходимо просмотреть все пары путевых точек. Но это трудоемкая процедура, поэтому используют упрощенный вариант — локальный, при котором для каждой точки в качестве пары просматривают только близко расположенные точки (либо ближайшие  $n$  точек, либо все точки, расположенные на расстоянии не более  $d$ ;  $n$  или  $d$  — параметр алгоритма). Построение дорожной карты превращает задачу планирования пути в задачи поиска пути в графе (один из алгоритмов поиска представлен в следующем разделе).

Метод вероятностной дорожной карты исследовался с помощью моделирования. Отметим, что, поскольку метод сильно зависит от полноты информации об окружающей среде, его трудно применять в условиях динамичной среды. На основе оригинального метода вероятностной дорожной карты разработан ряд модификаций [7], позволяющих улучшать результаты в различных ситуациях: повышать эффективность вычисления, получать более короткий путь и т.п.

К методам на основе случайного выбора относится *Метод быстро исследующих случайных деревьев* (RRT) [8]. В этом методе процесс генерации пути состоит в построении дерева опорных точек, которое последовательно расширяется от начальной точки до целевой точки. Процесс построения дерева начинается со стартового дерева  $X_{tree}$  с донной вершиной (начальной точкой) и без каких-либо ребер. На каждом шаге процесса построения выбирается новая случайная путевая точка  $X_{rand}$  (случайные точки в рассматриваемой области генерируются до тех пор, пока очередная точка не окажется вне препятствий). В

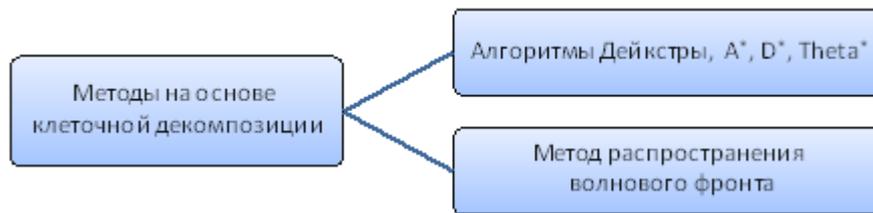
текущем дереве  $X_{tree}$  выбирается оптимальная по отношению к  $X_{rand}$  точка (обычно ближайшая), из которой достижима точка  $X_{rand}$ . Если такая точка не найдена (точка  $X_{rand}$  не достижима из точек дерева  $X_{tree}$ ), случайная точка  $X_{rand}$  отбрасывается. Если же оптимальная точка выбрана, точка  $X_{rand}$  добавляется к дереву и соединяется ребром с выбранной оптимальной точкой. Процесс построения дерева прекращается, если из последней добавленной точки достижима конечная точка или если достигнуто предельное количество вершин дерева. Если целевая точка достигнута, путь восстанавливается обратным ходом от очередной точки к той, с которой текущая точка была соединена при добавлении, т. е. с помощью отношения подчиненности, возникающего в процессе расширения графа [8].

Метод RRT позволяет прокладывать путь не только в физическом, но и в конфигурационном (фазовом) пространстве, которое может иметь высокую размерность. Кроме того, метод позволяет работать в среде с препятствиями невыпуклой формы. Метод допускает различные модификации, связанные с интерпретацией понятия достижимости, а также с правилами выбора очередной точки для включения в дерево. Так, есть варианты алгоритма, в которых при выборе очередной точки учитывается положение конечной точки [9]. Дерево может строиться по направлению от конечной точки к начальной, возможен и двунаправленный вариант метода [10].

Методы на основе графов или деревьев удобно использовать в статической окружающей среде, поскольку построение графа подразумевает использование полной информации об окружающей среде, которая должна быть известна заранее. Наличие подвижных препятствий, например, другого движущегося робота, в этих методах трудно поддается интерпретации и не позволяет строить граф.

### **Методы на основе клеточной декомпозиции**

Одна из простейших идей, возникающих в задаче планирования пути в среде с препятствиями — дискретизация окружающей среды (рис. 4). В различных реализациях этой идеи можно выделить две группы: приближенная и точная клеточные декомпозиции (cell decomposition) [11]. Приближенная клеточная декомпозиция реализуется с помощью сетки (grid map), покрывающей пространство. Метод сеток предложил Эльф [12], его идея заключается в том, чтобы разделить пространство окружающей среды на клетки одинакового размера, каждая клетка характеризуется 0 (свободна от препятствий) или 1 (занята препятствием). Сетка проста в использовании, и её легко создать и поддерживать. Однако метод сеток имеет серьезный недостаток: увеличение трудоемкости при уменьшении шагов сетки. Такое увеличение особенно заметно в окружающей среде большого объема. Предложены разные подходы, основанные на нерегулярных сетках и позволяющие снизить объем вычислений, например дерево квадрантов (quadtree) в двумерном случае или дерево кубов (octree) в трехмерном [13–15].



**Рис.4.** Методы на основе клеточной декомпозиции

При точной клеточной декомпозиции окружающую среду делят на клетки, используя грани препятствий. Здесь важную роль играет задача разложения многоугольника на выпуклые области. Хорошо известны такие подходы точной клеточной декомпозиции, как вертикальное или трапециевидное разложение [16].

В большинстве случаев при решении задачи планирования по методам на основе клеточной декомпозиции или на основе графов необходим дополнительный этап построения пути, состоящий в поиске путей на графе. Можно использовать разные стратегии поиска относительного оптимального решения. Эти стратегии делят на классические (неинформированные) и эвристические (информированные).

К неинформированным стратегиям поиска относятся: поиск в ширину, поиск по критерию стоимости, поиск в глубину, поиск с ограничением глубины, поиск в глубину с итеративным углублением и двунаправленный поиск [17].

Эвристические стратегии поиска не имеют строгого обоснования, но, тем не менее, дают приемлемое решение задачи в большинстве практически значимых случаев. Особенностью эвристической стратегии являются эвристические правила, которые формулируются для конкретной задачи. Эти правила позволяют повысить эффективность алгоритма по сравнению с соответствующей классической стратегией. Типичный пример эвристической стратегии — алгоритм  $A^*$ , который был описан в 1968 году Питером Хартом, Нильсом Нильсоном и Бертрамом Рафаэлем. По сути этот алгоритм — расширение алгоритма Дейкстры, созданного в 1959 году. Алгоритм  $A^*$  достигает более высокой производительности (по времени) с помощью эвристики. В первоначальной работе он упоминается как «алгоритм  $A$ ». Целевая функция алгоритма  $A^*$  реализуется как сумма двух других: функции стоимости достижения рассматриваемой точки из начальной и эвристической функции оценки расстояния от рассматриваемой точки до конечной точки. Эвристическая функция не должна переоценивать расстояние до целевой точки. Без второй составляющей алгоритм  $A^*$  сводится к алгоритму Дейкстры.

Одни из наиболее существенных недостатков алгоритмов на основе клеточной декомпозиции состоит в ограниченном количестве вариантов направления, вызванных структурой сетки (только восемь вариантов соединения в двумерном случае). Были разработаны различные подходы, чтобы обойти это ограничение. Очень важными примерами являются алгоритмы  $D^*$  [18] и  $\Theta^*$  [19]. Алгоритм  $D^*$  (динамический  $A^*$ ) был опубликован в 1993 году [20] и представляет собой развитие алгоритма  $A^*$  с помощью перепла-

нировки. Алгоритм  $D^*$  использует меньше клеток, чем алгоритм  $A^*$ , потому что в нем не планируется весь путь до цели. В варианте  $D^*$  [21] алгоритм улучшен за счет более эффективных расширений, сокращения количества расширенных узлов и требует меньше времени вычислений. Динамическая перепланировка также используется в алгоритмах Lifelong Planning  $A^*(LPA^*)$  [22] и  $D^*$  Lite [23]. Применение  $A^*$  и Theta\* для решения задачи планирования пути БПЛА в трехмерном пространстве было широко исследовано де Филиппом в [24–25]. Новый алгоритм (кинематический  $A^*$ ) представлен для определения допустимого пути, отвечающий требованиям динамических характеристик ЛА [26].

Еще один подход, основанный на использовании сеток — метод распространения волнового фронта (wavefront propagation method, см. рис.4), частным случаем его является метод Fast Marching Method (FMM) [27]. Особенностью подхода является отсутствие этапа поиска.

Метод распространения волнового фронта можно интуитивно понимать как расширение фронта волны: если камень брошен в пруд, то возникает волна в форме окружности, и эта волна расширяется от того места, где камень упал. В этом примере скорость распространения волны во всех направлениях одинакова. Поэтому волновой фронт оказывается круговой. Однако в неоднородной среде волна расширяется в разных направлениях с разной скоростью, так что волновой фронт уже не будет круговым [27].

В методе FMM сначала рассчитывается время  $T(X)$  как функция точки  $X$ , за которое волновой фронт, распространяясь из исходной точки, достигает заданной точки пространства. Функция  $T(X)$  является решением уравнения в частных производных

$$|\nabla T(X)|F(X) = 1,$$

называемого уравнением эйконала [27], с начальным условием  $T(X_0) = 0$ . Здесь  $X_0$  — исходная точка распространения волны;  $F(X)$  — скорость распространения волнового фронта в точке  $X$ . Сформулированная задача может быть решена приближенно с использованием подходящей разностной схемы [27–28].

После того, как получено решение уравнения, траектория из точки  $X$  может быть найдена как кривая наискорейшего спуска по поверхности  $Z = T(X)$  одним из методов спуска.

## Методы потенциальных полей

Один из широко распространенных подходов к планированию траекторий — использование потенциальных векторных полей. Общая идея методов состоит в движении вдоль векторных линий векторного поля, потенциальная функция которого отражает конфигурацию препятствий и их форму, а также цель движения. Указанный подход подходит и в двумерном, и в трехмерном случае. По типу потенциальной функции можно выделить: виртуальное силовое поле [29]; Ньютоновское потенциальное поле [30]; супербиквадратное потенциальное поле, гармоническое векторное поле [31].

Среди методов потенциальных полей самым известным является метод искусственных потенциалов (artificial potential field, APF). Его алгоритм прост, имеет низкую вычис-

лительную сложность и высокую эффективность реализации. Метод впервые предложил в 1986 году Хатиб [32]. Векторное поле разделяется на две составляющие: цель движения представляется притягивающим векторным полем, в то время как препятствия — отталкивающим векторным полем. Сложение двух векторных полей позволяет решить две задачи: движение к заданной целевой точке и обход препятствий. В свою очередь отталкивающее векторное поле есть сумма составляющих, каждое из которых описывает отдельное препятствие.

Рассматриваем робот как материальную точку. Потенциальная функция  $U(q)$  может быть записана в виде

$$U(q) = U_{\text{att}}(q) + \sum U_{\text{rep},i}(q),$$

где  $U_{\text{att}}(q)$  — притягивающая потенциальная функция, заданная для точки  $q$ ;

$\sum U_{\text{rep},i}(q)$  — слагаемые отталкивающей потенциальной функции, соответствующие отдельным препятствиям.

Притягивающая потенциальная функция имеет вид

$$U_{\text{att}}(q) = \frac{1}{2}k_p(q - q_g)^2,$$

где  $k_p$  — коэффициент притяжения, и порождает векторное поле антиградиента

$$F_{\text{att}}(q) = -\nabla U_{\text{att}}(q) = -k_p(q - q_g).$$

Для каждого  $i$ -го препятствия используем потенциальную функцию

$$U_{\text{rep},i}(q) = \begin{cases} \frac{1}{2}k_{ri} \left(\frac{1}{\rho_i} - \frac{1}{\rho_{oi}}\right)^2 (q - q_g)^n & \rho \leq \rho_0 \\ 0 & \rho > \rho_0 \end{cases}$$

где  $k_{ri}$  — коэффициент отталкивания;  $\rho_i = |q - q_i|$ ,  $q_i$  — центр препятствия;  $\rho_{oi}$  — радиус круговой области влияния препятствия. Эта функция порождает векторное поле градиента

$$F_{\text{rep},i}(q) = \begin{cases} F_{\text{rep},i}^1 \mathbf{n}_1 + F_{\text{rep},i}^2 \mathbf{n}_2 & \rho \leq \rho_0 \\ 0 & \rho > \rho_0 \end{cases}$$

Его слагаемые имеют вид

$$F_{\text{rep},i}^1 = k_r \left(\frac{1}{\rho_i} - \frac{1}{\rho_{oi}}\right) \frac{1}{\rho_i} (q - q_g)^n; \quad F_{\text{rep},i}^2 = \frac{n}{2} k_r \left(\frac{1}{\rho_i} - \frac{1}{\rho_{oi}}\right)^2 (q - q_g)^{n-1},$$

где  $\mathbf{n}_1 = \nabla(q - q_i)$  и  $\mathbf{n}_2 = -\nabla(q - q_g)$  — единичные векторы, направленные от препятствия к точке и от точки к целевой точке.

В изложенном подходе окружающая среда рассматривается как статическая, т. е. препятствия и целевая точка не двигаются. Однако в реальных условиях окружающая среда часто является динамичной, препятствия и целевая точка могут быть изменяться. Модификация метода искусственных потенциалов для динамической среды представлена в [33–34]. Для этого используются относительная скорость и относительное положение робота по отношению к целевой точке в притягивающей потенциальной функции, а также относительное положение и относительная скорость робота по отношению к препятствию в отталкивающей потенциальной функции.

Технология планирования пути по методу искусственных потенциалов проста, что упрощает контроль процесса движения в режиме реального времени. Однако метод имеет существенный недостаток: возможное существование локальных минимумов [35–36]. Предложен ряд подходов для ликвидации этого недостатка, но полностью удовлетворительного решения пока нет, а указанные подходы применялись лишь в частных ситуациях. Также отметим, что метод имеет неопределенность в построении потенциальных функций, связанную с выбором коэффициентов функции  $k_p$  и  $k_{ri}$ . Два этих фактора ограничивают широкое использование метода искусственных потенциалов в решении практических задач.

Другим недостатком метода искусственных потенциалов является проблема дрожания [37–38]. В указанных источниках предложены решения для преодоления этого недостатка. В 1989 г Боренштейн и Корен [39] предложили метод виртуального силового поля (virtual force field, VFF), сочетающий в себе метод сеток и метод потенциального поля, который может применяться для локального планирования пути.

У метода виртуального силового поля свои недостатки. Он не позволяет планировать прохождение узких областей, таких как дверь, из-за существенных изменений свойств отталкивания препятствий, возникающих при дискретизации. Метод также приводит к колебательному движению в полосообразной области (например, в коридоре). Для преодоления этих недостатков Боренштейн [40] предложил метод гистограммы векторного поля (vector field histogram, VFH).

Метод VFH является одним из самых популярных локальных методов планирования пути, используемых в режиме управления реальным временем в области мобильной робототехники. В этом методе обход препятствий осуществляется в три этапа. На первом этапе генерируется двумерная гистограмма, описывающая препятствия вокруг робота. На втором этапе по двумерной гистограмме строится одномерная полярная гистограмма. Наконец, на третьем этапе выбирается наиболее подходящий сектор с низкой плотностью препятствий, и вычисляется угол поворота в этом направлении. На основе метода VFH авторы предоставляли улучшенные методы: VFH+ [41] и VFH\* [42]. В методе VFH+ учитывается размер робота, ограничения динамики и соответствующее расширение размеров препятствий. В методе VFH\* с учетом глобальной информации окружающей среды выбирается наилучшее направление движения с использованием алгоритма  $A^*$ .

## Оптимизационные методы

Задача планирования пути в сложной окружающей среде может решаться как оптимизационная задача. Для этого движение объекта надо представить в рамках той или иной модели в виде динамической системы. Препятствия будут описываться некоторыми ограничениями, а качество допустимой траектории должно оцениваться некоторым функционалом. В результате возникает задача оптимального управления, которая не только обеспечивает траекторию объекта в обход препятствий, но и позволяет выбрать в некотором

смысле лучший вариант, например, по скорости прохождения, по энергетической эффективности и т.д.

Следует отметить, что оптимальное планирование траектории — это задача другого класса, нежели задача простого планирования. Решение первой — заметно более трудоемкое, чем решение второй. Классические методы оптимального управления основаны на нетривиальных аналитических вычислениях. Поэтому подход, основанный на оптимальном управлении, эффективен для простых, в частности, линейных систем, а для сложных нелинейных систем его реализовать труднее. Поэтому усилия исследователей постепенно сместились в сторону численных методов решения оптимизационных задач. Такие численные методы можно разделить на две группы: прямые методы и косвенные [43].

Как известно, для задачи оптимального управления существуют необходимые условия экстремума, которые формулируются в рамках принципа максимума Понтрягина. С помощью необходимых условий задача оптимального управления сводится к решению дифференциального уравнения. Такую задачу можно решать численно. Соответствующие методы называют косвенными, поскольку речь не идет о непосредственном поиске минимума функционала. Отметим, что аналитическое решение соответствующей краевой задачи, как правило, невозможно, а численные методы чувствительны по сходным данным либо обладают низкой сходимостью.

Прямые методы решения задач оптимального управления связаны с непосредственным поиском минимума целевого функционала. Однако решение такой задачи — некоторая функция, которая численным методом может быть найдена лишь в виде некоторого дискретного аналога. Необходимо исходную задачу свести к задаче с конечным набором варьируемых параметров [44], т.е. к задаче нелинейного программирования. Для решения задач нелинейного программирования разработан большой ассортимент численных методов. Таким образом, любой прямой метод включает в себя две основные подзадачи: дискретизацию, т.е. сведение задачи к задаче нелинейного программирования, и решение задачи нелинейного программирования.

В общем виде задачу оптимального управления можно сформулировать следующим образом. Требуется найти решение  $x(t)$ ,  $u(t)$  динамической системы

$$\dot{x} = f(x, u, t),$$

удовлетворяющее ограничениям:

- $l_i \leq \psi_i(x(t_0), u(t_0), t_0) \leq u_i$  (ограничения в начальной точке);
- $l_t \leq \psi_t(x(t), u(t), t) \leq u_t$  (траекторные ограничения);
- $l_f \leq \psi_f(x(t_f), u(t_f), t_f) \leq u_f$  (конечные ограничения).

При этом решение должно доставлять минимум целевому функционалу:

$$J[x(t), u(t)] = \int_{t_0}^{t_f} \phi_t(x(t), u(t), t) dt + \phi_0(x(t_0), u(t_0), t_0) + \phi_f(x(t_f), u(t_f), t_f) \rightarrow \min.$$

Сформулированная задача сводится к задаче поиска минимума функции нескольких переменных:

$$F(\rho) \rightarrow \min,$$

при наличии ограничений:

$$L \leq \begin{pmatrix} \rho \\ A\rho \\ G(\rho) \end{pmatrix} \leq U.$$

Здесь  $A$  — произвольная матрица, определяющая линейные ограничения;  $G(\rho)$  — произвольная, вообще говоря, векторнозначная функция, определяющая нелинейные ограничения.

Опишем несколько подходов к решению описанных оптимальных задач.

**Плоские модели.** Понятие дифференциальной плоскостности для нелинейных систем было предложено в 1990-е годы М. Флиссом [45]. Оно включает понятие плоского выхода, который позволяет представить динамическую систему в виде набора функциональных соотношений между входами и выходами. В случае плоской динамической системы задача оптимального управления может быть преобразована в задачу нелинейного программирования путем параметризации плоских выходов. Такой подход позволяет понизить размерность задачи, поскольку размерность пространства параметризованных плоских выходов ниже, чем размерность пространства состояний. Второе преимущество плоскостного подхода состоит в том, что он позволяет с помощью простых алгоритмов вычислить управляющие функции, реализующие траекторию.

Рассмотрим нелинейную системы со входом и выходом общего вида:

$$\begin{aligned} \dot{x} &= f(x, u), x \in R^n, u \in R^m, \\ y &= g(x), y \in R^m. \end{aligned}$$

Здесь  $u$  — вход системы, а  $y$  — выход. Если можно найти выход  $z$  такой, что

$$z = h(x, u, \dot{u}, \dots, u^{(p)})$$

(т.е. выход, который выражается через состояние  $x$ , управление  $u$  и некоторое количество производных управления по времени), то систему называют плоской, а соответствующий выход  $z$  — плоским выходом. При этом состояние и вход системы можно выразить через плоский выход и некоторое количество его производных по времени:

$$x = \varphi(z, \dot{z}, \dots, z^{(l)}), \quad u = \eta(z, \dot{z}, \dots, z^{(l)}).$$

Построение оптимальной траектории с использованием дифференциальной плоскостности может быть реализовано в три этапа.

1. *Определение плоского выхода.* Общих методов выбора плоского выхода нет. При решении практических задач плоский выход строят исходя из характера рассматриваемой системы и ее содержательного смысла.

2. *Параметризации плоских выходов.* Выбранный плоский выход необходимо задать как функцию времени. Такая функция определит во времени и сам выход, и состояние системы, и вход (управление). Для выбора функции времени можно использовать базисные функции, представляя выход как линейную комбинацию базисных функций с неизвестными коэффициентами:

$$z_i(t) = \sum_j A_{ij} \lambda_j(t).$$

Здесь  $\lambda_j(t), i = 1, \dots, N$  выбранные базисные функции, а  $A_{ij}$  — неизвестные коэффициенты, которые необходимо выбрать с учетом имеющихся ограничений и с требованием оптимальности траектории относительно заданного функционала качества. Базисные функции могут выбираться из разных соображений (например, простоты представления выходов). Возможные варианты базисных функций: полиномы, тригонометрические полиномы, сплайны.

3. *Построение траектории.* Параметризация выходов превращает задачу выбора оптимальной траектории в задачу нелинейного программирования, в которой неизвестными являются коэффициенты линейных комбинаций, задающих функции плоского выхода.

Итак, параметризация плоских выходов сводит задачу оптимального управления (планирование траектории) к задаче нелинейного программирования. Для решения такой задачи могут использоваться стандартные методы, реализованные в виде программных комплексов, например SNOPT [46], NPSOL [47].

**Смешанное целочисленное линейное программирование.** Если движение робота представлено линейной моделью  $\dot{x} = Ax + Bu$  (например, при локальном планировании траектории), то выбор оптимальной траектории может привести к задаче линейного программирования, в которой часть переменных являются дискретными (целочисленными). Такая задача представляет собой задачу смешанного целочисленного линейного программирования.

Для формирования подобной задачи динамическую систему, описывающую движение, необходимо преобразовать в систему дискретного времени. Для этого конечный промежуток времени  $T$  делится на конечное число  $N$  частичных промежутков. Здесь  $N = T/\Delta t$ , где  $\Delta t$  — время дискретизации. В результате непрерывная система управления преобразуется в дискретную систему:

$$x_{i+1} = A_c x_i + B_c u_i.$$

Параметры  $A_c, B_c$  получены из  $A$  и  $B$  с помощью соответствующих дискретизаций.

Ограничения на состояние системы могут быть заданы в виде линейных неравенств. Например, типичные ограничения на скорость робота имеют вид

$$V_{xmin} \leq V_x \leq V_{xmax}, V_{ymin} \leq V_y \leq V_{ymax}, V_{zmin} \leq V_z \leq V_{zmax}.$$

Препятствия могут быть приближенно заданы многогранниками. Ограничения на управления также могут быть линейными:  $\|u_i\| \leq U$ .

Целевую функцию оптимизационной задачи можно задать, например, в виде

$$\min_{x_i, u_i} J = \sum_{i=0}^N (w_x |x| + w_u |u|),$$

где  $w_x$  и  $w_u$  — весовые коэффициенты. Тем самым мы получим задачу смешанного целочисленного линейного программирования.

При практическом решении задачи планирования с помощью указанного подхода естественно равномерная дискретизация времени. Однако при этом следует иметь в виду

следующее. Траектория может иметь пересечения с препятствиями, попадающие между моментами дискретного времени. Для устранения таких коллизий наиболее простым подходом является более частая дискретизация времени, но это приводит к увеличению времени вычислений. Это увеличение может быть снижено с помощью итеративного метода СЦП [48]. В этом методе сначала проводится расчет с большим временем дискретизации, затем проверяется наличие столкновений. Каждый интервал  $[t_i, t_{i+1}]$ , в котором зафиксировано столкновение, дополнительно делится на меньшие интервалы и расчет повторяется. Процедура дополнительного деления может повторяться многократно.

Другой подход к упрощению возникающих задач линейного программирования — использование методов синтеза управления на основе прогнозирующих моделей (model predictive control) [49]. Прогнозирующая модель используется для локального планирования траектории, в то время как глобально траектория строится с помощью последовательности опорных точек.

**Двоично-целочисленное программирование.** Один из подходов к решению оптимизационных задач — сведение их к задачам двоично-целочисленного программирования, являющихся частным случаем задач целочисленного программирования, в которых переменные могут принимать лишь два значения 0 и 1. Методы решения подобных задач хорошо разработаны, например, могут быть использованы хорошо известные метод ветвей и границ и метод отсечения Гомори.

Задачи двоично-целочисленного программирования могут естественным образом возникать в рамках метода клеточной декомпозиции. Например, в [50] для планирования движения на плоскости в качестве клеточной декомпозиции использована триангуляция Делоне. Затем каждому элементу разбиения (треугольнику) назначена переменная, принимающая значение 0 (путь не проходит через треугольник) или 1 (путь проходит через треугольник). В качестве решения выбирается последовательность примыкающих друг к другу треугольников, начинающаяся со стартовой точки и заканчивающаяся в конечной точке. Условия примыкания треугольников формируются в виде линейных неравенств с использованием матрицы смежности треугольников. Оптимизация обеспечивается соответствующей целевой функцией. Описанный подход близок к стандартному методу клеточной декомпозиции, поскольку описание примыкающих треугольников фактически сводится к построению специального графа и поиску пути в этом графе, но поиск пути осуществляется не стандартным методом для графов, а как решение задачи целочисленного программирования. После того как цепочка примыкающих друг к другу треугольников найдена, выбирается путь, соединяющий точки пересечения медиан этих треугольников. Отметим, что такой путь гарантированно отделен от препятствий и удобен для сглаживания и использования с учетом отклонений.

Метод сведения оптимизационной задачи к задаче двоично-целочисленного программирования можно использовать и в трехмерном случае [51]. Также следует отметить, что возникающая задача двоично-целочисленного программирования может быть переформулирована как задача с ограничениями в виде линейных матричных неравенств. Для

решения подобных задач предложены эффективные алгоритмы, они оказываются более удобными, чем алгоритмы целочисленного программирования с точки зрения формирования задачи, и при этом оказываются более быстрыми [52].

### Методы на интеллектуальных алгоритмах

Задача планирования пути является одной подзадач автоматического управления роботом. Робот должен иметь способность решать задачу планирования пути в реальных условиях окружающей среды без вмешательства человека. Естественно в этой ситуации использовать алгоритмы поведения, встречающиеся в живой природе — так называемые биоинспирированные алгоритмы. Эти алгоритмы имитируют поведение или мышление человека, а также некоторых биологических сообществ.

Среди биоинспирированных алгоритмов выделяют: роевой интеллект, «муравьиный» алгоритм, метод роя частиц, «пчелиный» алгоритм, а также генетические и эволюционные алгоритмы. Как правило, эти алгоритмы формулируют как алгоритмы минимизации (или поиска), они относятся к эвристическим алгоритмам, так как не имеют строгого математического обоснования. Рассмотрим некоторые из этих алгоритмов (рис. 5).



Рис.5. Методы на интеллектуальных алгоритмах

**Муравьиный алгоритм.** Первая версия алгоритма, названная «алгоритмом оптимизации муравьиной колонии» (ant colony optimization, т. е. АСО), предложена Марко Дориго в 1992 году и была направлена на поиск оптимального пути в графе [53]. В реальном мире муравьи (первоначально) движутся в случайном порядке и, найдя пищу, возвращаются в свою колонию, помечая путь феромоном. Если другие муравьи находят такие тропы, они, вероятнее всего, пойдут по ним. Если эти муравьи в конечном итоге находят источник питания, то, возвращаясь, они повторно метят путь. Со временем феромон начинает испаряться, тем самым его привлекательная сила уменьшается. Чем больше времени требуется для прохождения пути до цели и обратно, тем сильнее испаряется феромон. На коротком пути будет более быстрым и, как следствие, сила феромона остаётся высокой. Испарение приводит к тому, что самый короткий путь от колонии до источника пищи, найденный одним из муравьев, будет более предпочтительным для других муравьев, они, скорее всего, пойдут по этому пути, а их метки в конечном итоге приведут всех муравьев к этому пути [54].

Муравьиный алгоритм ориентирован на решение трудоемких комбинаторных задач, с помощью такого алгоритма ищется близкое к оптимальному решению за приемлемое время. Подобная задача формулируется как определение в заданном конечном множестве  $\mathcal{S}$  (пространстве поиска) элемента [55], удовлетворяющего заданным ограничениям  $\Omega$  и обеспечивающего наименьшее значение заданной функции  $f$ .

Для муравьиного алгоритма задача трансформируется в задачу поиска путей в графе [55]. Каждому ребру графа приписывается объем феромона  $\tau_i^j$  (здесь  $i$  — номер вершины, из которой выходит ребро, а  $j$  — номер вершины, в которое идет ребро). На основании этих объемов рассчитываются вероятности прохода для каждого ребра. Каждый из некоторого конечного набора  $m$  муравьев строит свой путь, случайным образом выбирая очередное ребро на основе найденных вероятностей. Построенные пути затем служат основой для пересчета объемов феромонов, после чего конструирование пути повторяется. Итерационная процедура продолжается до тех пор, пока не будут выполнены условия останова. Стартовое значение объемов феромона выбирается равным некоторой постоянной.

Основная формула алгоритма — расчет вероятности перехода от  $i$ -го узла к  $j$ -му на основе имеющихся объемов феромона:

$$p_{i,j} = \tau_{i,j}^\alpha \eta_{i,j}^\beta$$

где  $\tau_{i,j}$  — количество феромона для ребра, соединяющего  $i$ -й узел с  $j$ -м;  $\alpha$  — параметр, контролирующий влияние  $\tau_{i,j}$ ;  $\eta_{i,j}$  — привлекательность ребра, соединяющего  $i$ -й узел с  $j$ -м, которую обычно берут равной величине, обратной расстоянию  $d_{i,j}$  между указанными узлами, т.е.  $\eta_{i,j} = 1/d_{i,j}$ ;  $\beta$  — параметр, контролирующий влияние  $\eta_{i,j}$ .

На каждой итерации алгоритма для каждого ребра графа объем феромона пересчитывается по формуле:

$$\tau'_{i,j} = (1 - \rho)\tau_{i,j} + \rho\Delta\tau_{i,j},$$

где  $\tau_{i,j}$  и  $\tau'_{i,j}$  — исходный и новый объем феромона для ребра  $i \rightarrow j$ ;  $\rho \in [0,1)$  — параметр, регулирующий скорость испарения феромона;  $\Delta\tau_{i,j}$  — количество отложенного феромона всех муравьев, обычно определяется формулой:

$$\Delta\tau_{i,j} = \sum_{k=1}^m \Delta\tau_{i,j}^k,$$

где

$$\Delta\tau_{i,j}^k = \begin{cases} Q/L_k, & k \text{ — й муравей использует ребро,} \\ 0, & \text{иначе.} \end{cases}$$

Здесь  $Q$  — постоянная;  $L_k$  — длина пути  $k$ -го муравья.

Варианты муравьиного алгоритма использовались для планирования пути в среде с препятствиями на основе карты или графа. Алгоритм применялся как в случае двумерной окружающей среды [55], так и в случае трехмерной [56]. В целом муравьиный алгоритм показал себя как эффективный алгоритм поиска, позволяющий найти оптимальный путь всего за несколько итераций.

**Искусственная нейронная сеть.** Искусственные нейронные сети — это группа математических моделей, имитирующих функционирование сети нервных клеток живого

организма. Такая сеть представляет собой совокупность однородных элементов (нейронов), каждый из которых имеет один или несколько входов и один или несколько выходов. Входы и выходы соединяют элементы в единую систему. Некоторые входы и выходы не задействованы и являются входами и выходами всей системы. На входы системы подается сигнал, а затем результат работы сети формируется на выходах системы. Каждый нейрон имеет настроечные параметры, влияющие на его работу. Настройка таких параметров, призванная обеспечить заданный выход сети для каждого из заданных входов, называется обучением сети. Конкретная искусственная нейронная сеть, как математическая модель, включает в себя не только внутреннюю структуру, но и алгоритм обучения. С математической точки зрения, обучение нейронных сетей — это многопараметрическая задача нелинейной оптимизации. Нейронная сеть используется в задачах адаптивного управления и как алгоритмы для робототехники [57].

Нейронные сети впервые были предложены У. Маккалоком и У. Питтсом как математические модели нервной активности [58]. В практических целях эти модели стали использовать после разработки алгоритмов обучения: в задачах прогнозирования, для распознавания образов, в задачах управления и др.

По-видимому, впервые искусственная нейронная сеть для целей планирования траекторий и обхода препятствий использовалась в 1995 г [59]. Метод быстро набрал популярность и применялся в задачах планирования пути в различных случаях. Морено и Кастро [60] предложили для планирования пути использовать расширяющуюся гибкую нейронную сеть. Нейроны сети ассоциируются с точками в конфигурационном пространстве, взаимодействие нейрона притягивающее, если соответствующая точка находится вне препятствий, а иначе отталкивающее. В процессе обучения нейронная сеть расширяется добавлением новых элементов. Нейронная сеть самоорганизуется, а линии взаимодействия нейронов представляют путь в среде с препятствиями.

К искусственным нейронным сетям близки байесовские сети. Байесовская сеть представляет собой ациклический ориентированный граф, на множестве вершин которого задано распределение вероятностей определенного типа. Байесовские сети, как и искусственные нейронные сети, могут использоваться в задачах планирования пути в среде с препятствиями [61].

**Метод роя частиц** (МРЧ) был предложен в 1995 г Кеннеди, Эберхартом и Ши [62] как попытка имитации движения стаи птиц в контексте «коллективного интеллекта» биологических популяций. МРЧ отражает способность стаи птиц, рыб и других биологических популяций адаптироваться к окружающей среде, искать пищевые ресурсы и избегать хищников путем обмена информацией. В МРЧ оптимизируют функцию, поддерживая популяцию возможных решений (каждое имитирует одну птицу), называемых частицами, и перемещая эти частицы в пространстве решений согласно простой формуле. Перемещения подчиняются принципу наилучшего найденного в этом пространстве положения, которое постоянно изменяется при нахождении частицами более выгодных положений. Таким об-

разом, возникает метод оптимизации, для использования которого не требуется знать точного градиента оптимизируемой функции.

В методе инициализируют группу частиц случайным образом. Затем ищут оптимальное решение, выполняя последовательность итераций. На каждой итерации частицы обновляют свое положение на основе двух параметров:  $p_i$  — лучшее из известных положений частицы  $i$ ;  $g$  — наилучшее известное состояние роя в целом.

Когда оптимальные значения  $p_i$  и  $g$  найдены, обновляются скорости частиц и их местоположение в соответствии со следующей формулой:

$$v_i \leftarrow \omega v_i + \varphi_p r_p (p_i - x_i) + \varphi_g r_g (g - x_i), x_i \leftarrow x_i + v_i.$$

Здесь  $\omega$  — инерционный вес;  $\varphi_p$  и  $\varphi_g$  — коэффициенты обучения, обычно равные 2;  $r_p$  и  $r_g$  — случайные числа на отрезке  $[0, 1]$ .

Метод роя частиц для задачи планирования пути летательного аппарата в пространстве использовал Чун [63]. В качестве стартового выбирался прямолинейный путь из начальной точки в конечную. Далее запускался алгоритм МРЧ, в котором текущий путь конструировался в виде сплайна. Ограничениями выступали препятствия, в качестве целевой функции использовался расход топлива.

**Реактивные методы.** Важную группу интеллектуальных алгоритмов составляют алгоритмы, основанные на имитации поведенческих реакций. Основой таких алгоритмов является концепция поведения, которую можно обозначить как типовое решение в типовой ситуации [64]. В противовес реактивному централизованное управление должно учитывать всю массу обстоятельств, связанных с окружающей средой и поставленной целью. Введение в контур управления реактивных методов приводит к децентрализации задачи управления, что в конечном счете упрощает задачу планирования, повышает восприимчивость по отношению к окружающей среде и устойчивость по отношению к внешним факторам. Каждое отдельное поведение реализуется в виде независимого модуля, который воспринимает какой-то отдельный аспект окружающей среды (например, информацию с отдельно взятого датчика).

Наиболее перспективными являются системы гибридного типа, сочетающие в себе как элементы централизованной системы, так и элементы реактивной. В такой системе особую роль играет система координации, которая стыкует различные поведения и верхний уровень системы между собой. Среди методов координации можно выделить две основные группы: арбитража и фьюжн-команды. При арбитраже в каждый момент времени выбирается только одно поведение. Арбитраж прост в реализации, однако движение объекта при таком принципе управления оказывается нестабильным [65]. Кроме того, правила арбитража не являются очевидными, выбор типа поведения — нетривиальная задача. Проблему может решить смешанная стратегия, при которой действуют все типы поведения одновременно, но с разной степенью.

Обычно рассматриваются такие варианты поведения: движение к цели, уклонение от препятствия, движение вдоль стены, выход из локального минимума. Поведение «движение к цели» может быть достигнуто при заданной информации положения цели. Поведение

ние «уклонение от препятствий» реализуется на основе информации о локальной окружающей среде, полученной по датчикам. Отметим также поведение, называемое странствованием. Речь идет о преодолении ловушек — ситуаций, когда робот не может двигаться, ориентируясь на определенное направление (например, если робот попал во внутреннюю камеру препятствия). В литературе предлагаются различные методы преодоления ловушек, основанные на одном или нескольких поведении [66–68].

В определенном смысле реактивные методы аналогичны методу искусственного потенциального поля. Однако первые имитируют человеческое поведение, в то время как метод искусственного потенциального поля строится из других соображений.

Различные типы поведения можно соединять с помощью методов нечеткой логики. Нечеткое множество характеризуется некоторой функцией (называемой функцией принадлежности), которая отражает степень принадлежности элемента множеству. Функция принадлежности может принимать значения от 1 («полностью принадлежит») до 0 («полностью не принадлежит») [69]. Нечеткие множества могут использоваться для координации связи нескольких типов поведения [67]. Нечеткая логика также может использоваться для реализации отдельного типа поведения. Например, с помощью нечеткой логики можно строить маневры летательного аппарата для ухода от столкновений на основе информации, поступающей с передней камеры [70].

## **Интеграция нескольких методов и локальные методы**

Если общая информация об окружающей среде известна и неизменна, задача планирования пути может быть решена заблаговременно (оффлайн). В этом случае не надо заботиться о времени расчета и вычислительной сложности задачи. При изменяющейся (динамической) окружающей среде метод планирования пути должен иметь способность планирования пути «на лету» (онлайн) на базе оперативно поступающей информации и динамической модели окружающей среды. Это накладывает жесткие требования на используемые вычислительные ресурсы (вычислительная мощность системы и время расчетов).

Современный алгоритм планирования пути должен в себе интегрировать преимущества различных существующих методов и быть свободен от недостатков отдельных методов. Такой результат может быть достигнут как результат соответствующего исследования. Например, важно сочетать глобальные и локальные методы. Тогда глобальный метод позволит планировать глобальный путь на основе глобальной информации об окружающей среде, а локальные методы позволят роботу избегать столкновений с препятствиями, которые не были обнаружены заранее или они являются подвижными, так что их текущее положение не может быть установлено заранее.

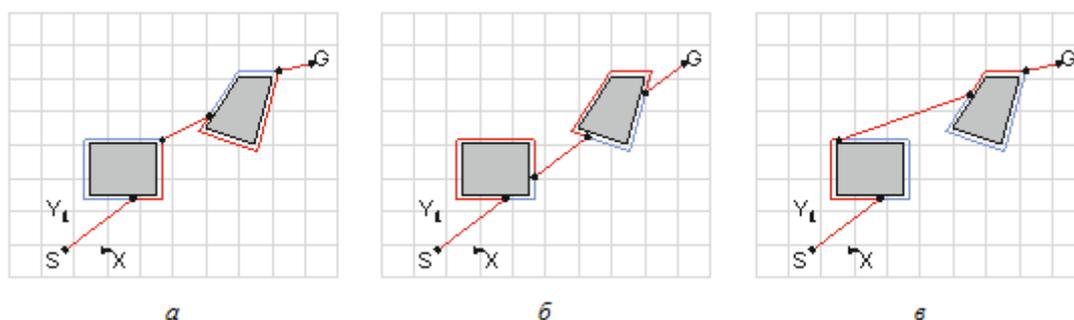
Методы планирования, описанные выше, в основном предназначены для глобального планирования. Эти алгоритмы необходимо дополнять алгоритмами локального планирования, ориентированными на обход препятствий или других движущихся объектов. К локальным методам относится описанный выше метод потенциальных полей. Другой ло-

кальный метод, так называемый Bug-алгоритм — один из самых ранних алгоритмов обхода препятствий, по которому от источника к месту назначения планируется прямой путь до тех пор, пока объект не сталкивается с препятствием.

Входной информацией Bug-алгоритма является только локальная информация об окружающей среде и информация о цели. Алгоритм реализует два режима движения: «переход к цели» и «обход препятствий». Различные версии алгоритма различаются способом обхода препятствий. Опишем некоторые варианты Bug-алгоритма.

*Алгоритм Bug 1 [71] (рис.6а)*

1. Движение в направлении цели;
2. Если встречено препятствие, происходит обгибание его до тех пор, пока робот не выйдет в точку, ближайшую к цели;
3. Продолжение движения к цели.



**Рис.6.** Bug-алгоритмы: *а* — Bug 1; *б* — Bug 2; *в* — Distbug

*Алгоритм Bug 2 [73] (рис.6б)*

1. Движение в направлении цели, фиксируется наклон прямой;
2. Если встречено препятствие, происходит обгибание его до тех пор, пока текущий наклон прямой (от текущего положения до цели) не совпадет с зафиксированным наклоном;
3. Продолжение движения к цели.

На рис.6в представлена траектория, построенная с помощью алгоритма Distbug [73]. В нем для определения момента ухода от препятствия используется структуры данных, в которых хранится некоторая дополнительная информация.

На основе алгоритмов Bug 1 и Bug 2 появилось много вариантов Bug-алгоритма [71, 72]. В [68, 73] приведены различные варианты Bug-алгоритма и проведено их сравнение. Bug-алгоритмы относятся к классу реактивных и используются для перемещения роботов в окружающей среде, которая постоянно меняется и которую невозможно полностью моделировать.

Разные алгоритмы могут соединяться в самых разных сочетаниях. Так, в [74] использован муравьиный алгоритм для поиска оптимального пути, а для подбора параметров муравьиного алгоритма использован алгоритм роя частиц. Другой тип соединения — внедрение. В [75] клеточная декомпозиция окружающей среды с несколькими разреше-

ниями использована для построения целевой функции оптимизационной задачи, в которой таким образом удастся учесть динамические и кинематические ограничения.

### Метод гладкого пути

Гладкий маршрут является необходимым условием для достижения точного отслеживания пути, как в случае наземного робота, так и в случае БПЛА.

Функция гладкая, если она в рассматриваемой области имеет производные всех порядков до некоторого значения. Параметрическая кривая гладкая, если она представлена гладкими функциями. Для описания степени гладкости кривой используют символ  $C^n$ , означающий, что функции, представляющие кривую, имеют непрерывные производные до порядка  $N$  включительно.

Путь робота к цели часто строится из нескольких кривых. Для обеспечения необходимой гладкости такого пути следует уделять большое внимание точкам сопряжения соседних кривых.

Построенный путь должен удовлетворять ограничениям, накладываемым на него логическую систему. Робот не может изменять скорость мгновенно. Колесный робот обычно не может двигаться в боковом направлении без проскальзывания, угол поворота передних колес и скорость их вращения также ограничены.

Эти требования означают, в частности, что путь должен быть гладкой кривой класса  $C^2$ . Есть и другие ограничения, например, непрерывность кривизны кривой. Такое требование равносильно гладкости кривой класса  $C^3$ . На кривизну пути могут накладываться дополнительные ограничения: ограниченный диапазон изменения  $\kappa < \kappa_{\max} = 1/R_{\min}$ , отсутствие участков нулевой кривизны и т.п.

Для учета указанных ограничений нужны специальные методы. Можно улучшать форму аппроксимирующей кривой путем изменения опорных точек. Подход, в котором изменение опорных точек основано на минимизации определенных критериев, называется фейринг (fairing). Можно выделить две группы: локальный фейринг и глобальный [76].

Первоначальный этап планирования часто заканчивается построением ломаной линии, которая упрощает планирование, но не отвечает условиям гладкости и другим ограничениям. Ломаная является исходной основой для построения точного маршрута, отвечающего всем требованиям. Этот процесс называется сглаживанием.

Основой для построения гладкого пути может быть не ломаная, а «канал» или «коридор», который выбирают так, что в нём нет препятствий. Тогда гладкий путь можно строить внутри канала и он пройдет без столкновений с препятствиями. Например, в [77] используется узкая кусочно-линейная оболочка, в которой должен располагаться В-сплайн. В [78] канал для сплайна конструируется из ячеек сетки, полученной при клеточной декомпозиции окружающей среды.

Перед процедурой сглаживания полезно применять алгоритмы редукции, направленные на сокращение количества опорных точек. Простейший алгоритм редукции основан на следующей идее [9]: в определенном порядке проверяются пары опорных точек,

пересекается ли отрезок, их соединяющий, с препятствиями. Если пересечений нет, из маршрута можно удалить все опорные точки между двумя выбранными. Алгоритм прост в применении и может значительно сократить количество звеньев маршрута, особенно если окружающая среда не слишком сложная.

Можно указать следующие подходы к задаче сглаживания пути (рис. 7).

1. *Корректировка опорных точек.* Метод состоит в том, что заданные опорные точки сдвигаются так, чтобы обеспечить требуемые ограничения и гладкость. Процедура применяется итерационно. В статье [79] корректировка опорных точек сводится к задаче квадратичного программирования, которая формулируется на основе ограничений гладкости и информации о препятствиях. В статье [80] на основе метода потенциала, строится целевой функционал, который зависит от положения опорных точек, а далее решается задача минимизации.

2. *Сглаживание ломаных в сочленениях.* При этом подходе ломаная линия корректируется только в окрестностях ее вершин. Если исходная ломаная не имеет пересечений с препятствиями, то сглаженную кривую надо проверять на пересечения только в окрестностях опорных точек. Метод сглаживания можно выбрать так, что будет легко контролировать не только пересечения с препятствиями, но и характер изменения кривизны. Наиболее распространенным является метод «отрезок – дуга» [81], в котором для сглаживания углов ломаной используется дуга окружности. Недостатком этого простого метода является то, что в точках сочленения отрезков и дуг окружностей кривизна имеет разрыв. Чтобы избежать этой проблемы, надо для сглаживания углов использовать кривые переменной кривизны, которая плавно меняется от нулевой до некоторой максимальной. Для этой цели применяют полярные полиномы [82], декартовых полиномы, клотоиды (clothoid) [83], сплайны, например кривые Безье [84], или B-сплайны [85, 86].

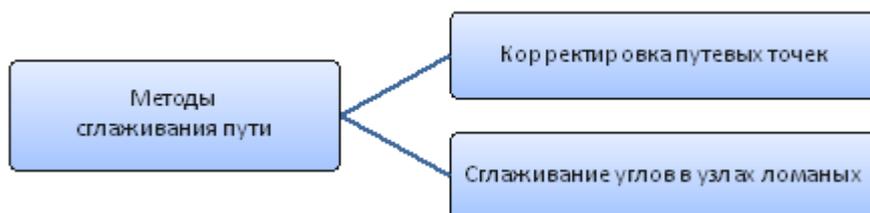
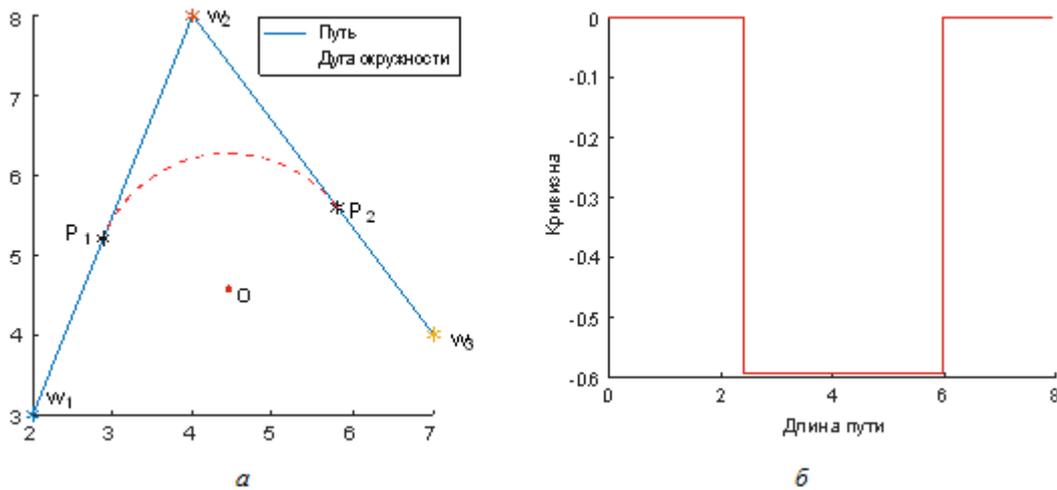


Рис.7. Методы сглаживания пути

Рассмотрим подробнее методы для сглаживания углов ломаной.

**Метод Dubins.** Один из вариантов метода «отрезок – дуга» — это метод Dubins. Этот метод сглаживания состоит в замене прилегающих к углу ломаной частей отрезков дугой окружности (рис. 8). Отметим, что, хотя путь по методу Dubins сглаживается, кривизна этого пути является разрывной, разрывы возникают в точках соединения отрезков и дуг окружностей.



**Рис.8.** Сглаживание ломаной по методу Dubins: а — ломаная и сглаженная кривая; б — изменении кривизны вдоль сглаженной кривой

Обозначим через  $w_1, w_2, w_3$  три подряд идущие вершины ломаной, причем  $w_2$  обозначает вершину, угол в которой подлежит сглаживанию. Через  $d$  обозначаем расстояние от вершины угла  $w_2$  до крайней точки дуги окружности.

Положим  $q_1 = w_1 - w_2, q_2 = w_3 - w_2$ . Это векторы, направленные по звеньям ломаной от вершины  $w_2$ . Рассмотрим их нормированные варианты:  $q_1^0 = \frac{q_1}{|q_1|}, q_2^0 = \frac{q_2}{|q_2|}$ . Положим  $\rho = \frac{1}{2}|q_1^0 - q_2^0|, \eta = \sqrt{1 - \rho^2}$ . Радиус  $R$  дуги окружности можно вычислить по формуле  $R = \frac{d \cdot \rho}{\eta}$ , угол между векторами  $q_1$  и  $q_2$  — по формуле  $\theta = \arcsin \rho$  или  $\theta = \arccos \eta$ . Для получения дуги окружности введем вспомогательный двумерный базис  $e_1, e_2$ . Вектор  $e_1$  направлен по вектору  $q_1$ , а вектор  $e_2$  расположен в плоскости векторов  $q_1, q_2$  и направлен в ту же сторону, что и вектор  $q_2$ . Для этого вычисляем

$$e_1 = q_1^0, \tilde{e}_2 = q_2^0 - (q_2^0, e_1) \cdot e_1, e_2 = \frac{\tilde{e}_2}{|\tilde{e}_2|}.$$

В базисе  $e_1, e_2$  дуга окружности задается вектор-функцией

$$\Gamma(\varphi) = \begin{pmatrix} d - R \sin \varphi \\ R - R \cos \varphi \end{pmatrix}$$

Формула дуги окружности пересчета в исходный базис получается

$$P = w_2 + \Gamma(\varphi)^T \cdot \begin{pmatrix} e_1 \\ e_2 \end{pmatrix}$$

**Метод полярных многочленов.** Для преодоления проблемы с разрывом кривизны пути можно использовать другие кривые, например, полярные многочлены, т. е. многочлены, связывающие полярные координаты:

$$r(\varphi) = a_0 + a_1 \varphi + a_2 \varphi^2 + a_3 \varphi^3 + a_4 \varphi^4 \quad (1)$$

Полярная система координат задается следующим образом: начало системы координат помещается в точку центра  $O$  сглаживающей окружности, а начальная ось совмещается с каким-либо радиусом окружности. Полярный многочлен должен соответствовать положениям начальной и конечной точек (точек касания линий и окружности с радиусом  $r$  и центральный угол  $\varphi$ ). Условия ограничения для соединения следующие:

$$\begin{cases} r = d & r' = 0 & \kappa = 0 & \varphi = 0 \\ r = d & r' = 0 & \kappa = 0 & \varphi = \Phi \end{cases} \quad (2)$$

В результате по формулам (1) и (2) получаем следующее представление полярного многочлена:

$$r(\varphi) = d \left[ 1 + \frac{\varphi^2}{2} - \frac{\varphi^3}{\Phi} + \frac{\varphi^4}{2\Phi^2} \right] \quad (3)$$

При  $\Phi > 90^\circ$  используют кусочно-полярные многочлены, состоящие из трех частей:  
 $\varphi \in [0, \beta]$  полярный многочлен;  
 $\varphi \in [\beta, \Phi - \beta]$  дуга окружности;  
 $\varphi \in [\Phi - \beta, \Phi]$  симметричный полярный многочлен.

В первой части ( $\varphi = 0 \sim \beta$ ) рассчитываем многочлен (1) с условиями

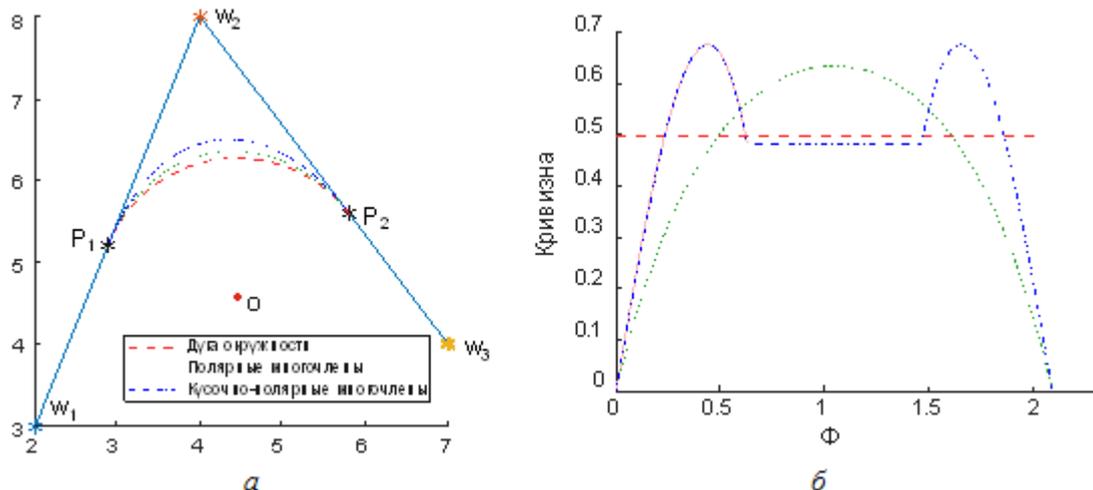
$$\begin{cases} r = d & r' = 0 & \kappa = 0 & \varphi = 0 \\ r = d & r' = 0 & \kappa = \frac{1}{R_b} & \varphi = \beta \end{cases}$$

Получаем

$$r(\varphi) = d \left[ 1 + \frac{\varphi^2}{2} - \frac{\varphi^3}{2\beta} + \frac{\varphi^5}{10\beta^3} \right] \quad (4)$$

Многочлен для третьей части можно получить, подставив  $\varphi = \Phi - \varphi$  в представление (4).

Сравнение результатов сглаживания с помощью дуги окружности, полярного многочлена и кусочно-полярного многочлена представлено на рис.9.



**Рис.9.** Сглаживание пути по методу полярных многочленов: *a* — сглаженные кривые, полученные тремя методами; *б* — сравнение кривизны трех видов сглаженной кривой

**Методы на основе сплайнов.** Рассмотрим некоторые методы сглаживания пути на основе сплайнов.

*Кубические кривые Безье.* Для построения кубической кривой Безье необходимы четыре контрольные точки ( $P_0, P_1, P_2, P_3$ ), где  $P_0$  и  $P_3$  — конечные точки кривой. Контрольные точки выбираем на звеньях ломаной (рис. 10).

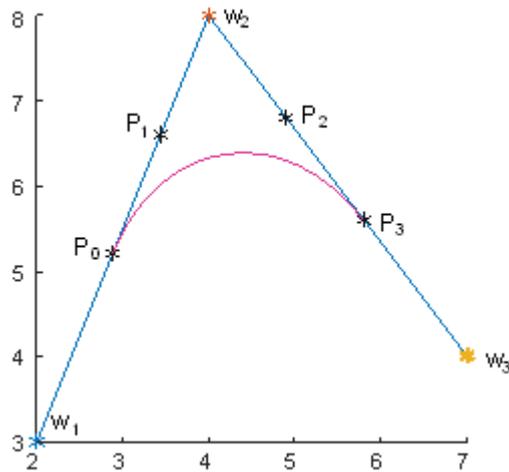


Рис.10.  $C^1$  непрерывная кубическая кривая Безье

Сначала находим контрольные точки [9]:

$$P_0 = w_2 - d_1 \cdot q_1, P_1 = w_2 - \frac{d_1 \cdot q_1}{2}, P_2 = w_2 + d_2 \cdot q_2, P_3 = w_2 + \frac{d_2 \cdot q_2}{2},$$

где  $q_1 = w_2 - w_1$ ;  $q_2 = w_3 - w_2$ ;  $d_1$  — длина от  $w_2$  до  $P_0$ ;  $d_2$  — длина от  $w_2$  до  $P_3$ .

Затем вычисляем кривую Безье по формуле

$$P(s) = \sum_{i=0}^n P_i B_{n,i}(s),$$

где  $n = 3$ ;  $B_{n,i}(s)$  — многочлены Бернштейна. Кривизна вдоль кубической кривой Безье вычисляется по формуле  $\kappa(s) = \frac{|\dot{P}(s) \times \ddot{P}(s)|}{|\dot{P}(s)|^3}$ . Отметим, что кривизна в точках стыковки имеет разрыв.

**Кривые Безье класса  $C^2$ .** Дважды непрерывно дифференцируемая кривая Безье задается восемью контрольными точками (рис. 11), нужно построить две кубические кривые Безье, чтобы получилась кривая с непрерывной кривизной [9].

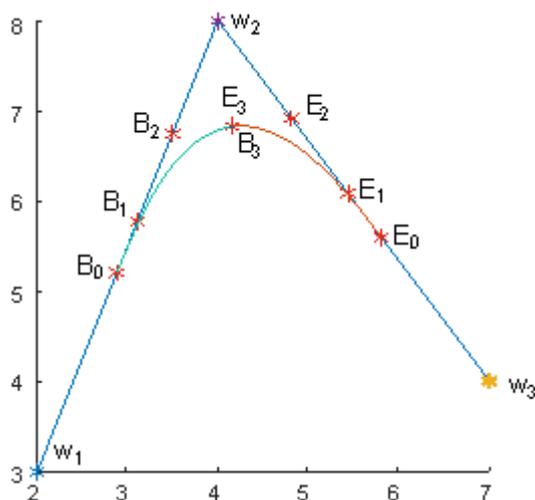


Рис.11. Кривая Безье класса  $C^2$

Первая кривая задается следующими четырьмя контрольными точками:

$$B_0 = w_2 - d_1 \cdot q_1, B_1 = B_0 + g_b \cdot q_1, \\ B_2 = B_1 + h_b \cdot q_1, B_3 = B_2 + k_b \cdot u_d.$$

Вторая кривая задается остальными четырьмя контрольными точками:

$$E_0 = w_2 + d_2 \cdot q_2, E_1 = E_0 - g_e \cdot q_2, \\ E_2 = E_1 - h_e \cdot q_1, E_3 = E_0 - k_e \cdot u_d.$$

Здесь

$$q_1 = w_2 - w_1; \quad q_2 = w_3 - w_2; \quad d_1 = d_2 = d; \quad h_b = h_e = 0.346 \cdot d; \\ g_b = g_e = 0.58 \cdot h_b; \quad k_b = k_e = 1.31 \cdot h_b \cos \beta; \quad \beta = \frac{\gamma}{2}; \quad \gamma = \pi - \theta;$$

$d_1$  – длина от  $w_2$  до  $B_0$ ;  $d_2$  – длина от  $w_2$  до  $E_0$ ;  $u_d$  — единичный вектор, направленный вдоль вектора  $\overline{B_2E_2}$ .

Для определения кривой класса  $C^2$  необходимо знать лишь параметр  $d$ . Если нужно учитывать ограничение максимальной кривизны, то

$$d = \frac{1.1228 \sin \beta}{\kappa_{max} \cdot \cos^2 \beta},$$

где  $\kappa_{max}$  – максимальная кривизна.

**Неравномерный кубический В-сплайн** [87]. Чтобы сглаживание пути с помощью сплайна обеспечивало непрерывное изменение кривизны, необходимо не менее пяти контрольных точек. Чем больше контрольных точек, тем больше степеней свободы в выборе формы кривой, но при этом число базовых функций для такой кривой увеличивается, так что возрастает вычислительная сложность. Если использовать пять контрольных точек, то в векторе  $U = \{0,0,0,0, u_4, 1,1,1,1\}$  можно регулировать только одну переменную. Если кривая сглаживания будет симметричной кривой, то  $u_4 = 0.5$  и возможностей регулировать кривую не остается. Поэтому определим  $U = \{0,0,0,0, u_4, u_5, 1,1,1,1\}$ ,  $u_4 < u_5$ ,  $u_4 + u_5 = 1$  (рис.12).

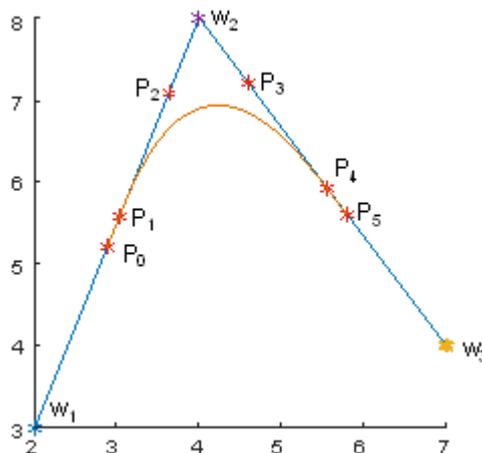


Рис.12. Неравномерный кубический В-сплайн

Для контрольных точек формулы следующие:

$$P_2 = w_2 - s \cdot q_1, P_1 = P_2 - k_1 \cdot s \cdot q_1, P_0 = P_1 - k_2 \cdot s \cdot q_1,$$

$$P_3 = w_2 + s \cdot q_2, P_4 = P_2 + k_1 \cdot s \cdot q_2, P_5 = P_1 + k_2 \cdot s \cdot q_2.$$

где  $q_1 = w_2 - w_1$ ;  $q_2 = w_3 - w_2$ . Когда известно значение  $r_{min}$  минимального радиуса и  $\theta$  угол между векторами  $-q_1$  и  $q_2$ , можно выбрать такие параметры:

$$\begin{cases} u_4 = 0.05, u_5 = 0.95 \\ k_1 = 2u_5 \\ k_2 = 2u_4 \\ s = \frac{2}{3\sqrt{2}} \cdot \frac{r_{min} \sin \theta}{u_5 (\sqrt{1 - \cos \theta})^3} \end{cases}$$

Если нет ограничения на максимальный радиус поворота, параметр  $s$  можно получить из равенства:

$$d = \|P_0 - w_2\| = \|s \cdot q_1 + k_1 \cdot s \cdot q_1 + k_2 \cdot s \cdot q_1\| = 3s.$$

На рис.13 показано, что при одинаковых точках начала поворота (параметр  $d$  одинаков) кривизна в методе Dubins наименьшая, а кривизна в методе с кубической кривой Бэзе наибольшая.

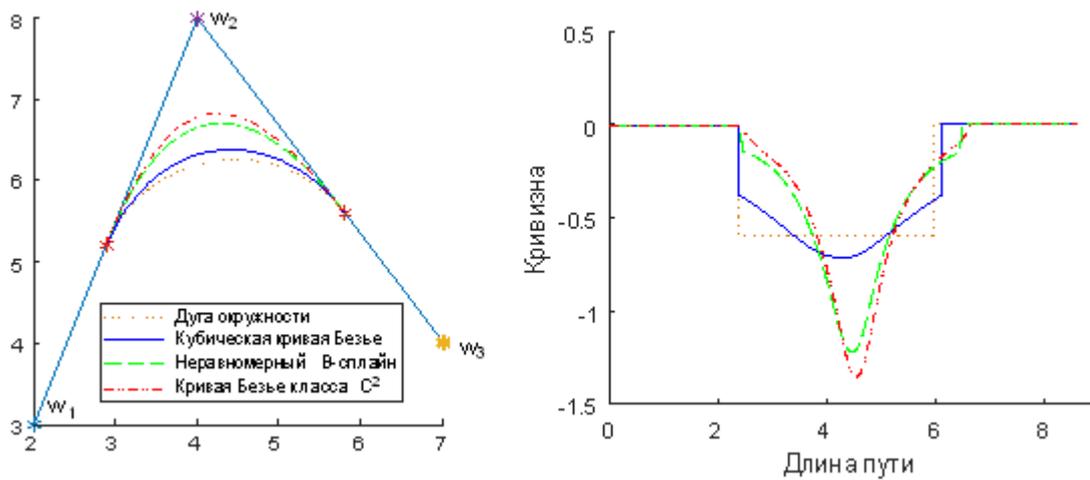


Рис.13. Сравнение методов на основе сплайна

На рис.14 представлено сравнение результатов сглаживания пути и кривизны пути при условии, что минимальный радиус поворота одинаков.

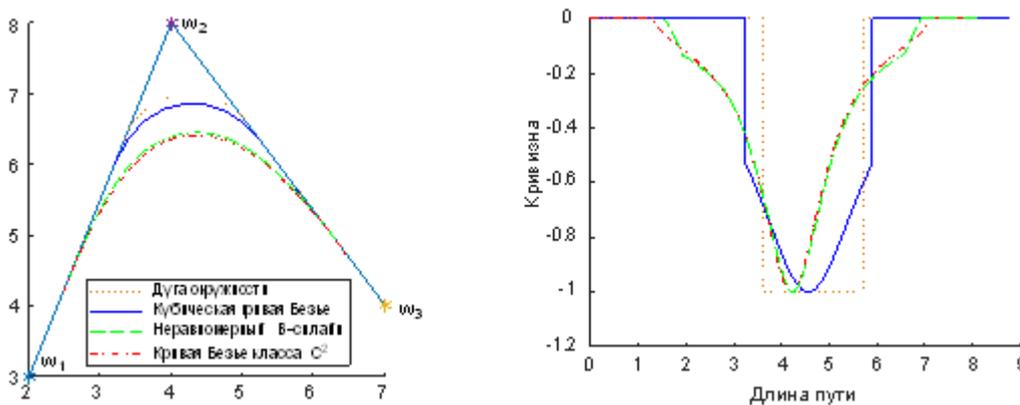


Рис.14. Сравнение методов сглаживания с помощью сплайнов

## Заключение

Методы на основе графов легко реализовать в компьютерном моделировании. Основные методы на основе графов предназначены для использования в статической окружающей среде, хотя некоторые из них могут быть преобразованы в вид, применимый для динамической среды. Поскольку граф связан с опорными точками, результат в этих методах оказывается ломаной линией. Кроме того, получаемое решение не является оптимальным: оно допустимое или относительно оптимальное.

Метод потенциальных полей позволяет построить гладкий путь. Реализация метода относительно проста и ясна. Преимуществом метода является малое время расчета. Поэтому его часто используют при локальном планировании, когда расчеты должны выполняться в режиме реального времени. Но при использовании этого метода робот может попасть в точку локального минимума и в конечном счете не достичь цели.

Оптимизационные методы имеют преимущество в том, что с помощью них можно учесть самые разные ограничения, например динамические или ограничения на управление. Можно также включать ограничения, направленные на обеспечение безопасности, надежности и других факторов. Однако с увеличением размерности пространства состояний и области окружающей среды, с увеличением степени детализации модель быстро усложняется, что приводит в результате к замедлению процесса вычисления. Реализовать оптимизационные алгоритмы в режиме реального времени не всегда представляется возможным.

Методы, основанные на интеллектуальных алгоритмах, помогают решить задачу с очень сложными ограничениями динамики. Но самый большой недостаток – цикл планирования слишком длительный, и их трудно использовать в режиме реального времени. Такие методы лучше всего применять при первоначальном глобальном планировании.

Сравнение характеристик этих методов показывается в таблице. 1.

Таблица.1. Сравнительные характеристики методов планирования пути

Методы	Реальное время	Тип окружающей среды	Характер пути	Сфера действия
На основе графа или дерева	нормально	статический	негладкий	глобальная
На основе клеточной декомпозиции	нормально	статический	негладкий	любая
Метод потенциальных полей	лучше	любой	гладкий	любая
Оптимизационные	хорошо	любой	гладкий	любая
Интеллектуальные	плохо	статический	негладкий	глобальная
Реактивные	лучше	любой	гладкий	любая

Есть несколько проблем, которые необходимо решить в процессе планирования пути в сложной трехмерной окружающей среде с препятствиями.

1. Моделирование комплексной окружающей среды. Основные подходы к моделированию окружающей среды строятся на предположении, что препятствия являются иде-

альными и регулярными. Однако это плохо согласуется с реальной ситуацией. Кроме того, важную роль играют методы трехмерной реконструкции на основе датчиков на борту, реализуемые в процессе движения. В практике планирования на данный момент используют данные радара или бортовой камеры. Но данные бортовых камер или радаров не очень информативны. Недостатки бортовых камер и радаров отсутствуют у лидаров. Однако эта технология новая, лидары громоздки и дороги, их размещение на борту относительно небольших беспилотных аппаратов невыгодно. Методы обработки данных лидара также разработаны на данный момент недостаточно, что затрудняет решение задачи планирования в реальном времени.

## 2. Планирование в режиме реального времени.

Существующие методы базируются на упрощенных моделях объектов и на использовании идеализированных предположений. Это позволяет решать задачи за приемлемое время, но вносит чрезмерную неопределенность в оценки окружающей среды, что снижает практическую значимость этих методов.

## 3. Ограничения, связанные с динамикой, и другие ограничения.

В задаче трехмерного планирования пути существуют довольно сложные и разнообразные ограничения. Необходим единый простой подход, позволяющий учитывать разнообразные ограничения.

В данной работе описаны основные методы планирования пути в среде с препятствиями. Также представлены методы сглаживания пути, сформированного в виде ломаной линии. Методы сглаживания важны потому, что негладкие траектории движения не удовлетворяют ограничениям по динамике и, кроме того, трудны для существующих методов синтеза управления. Задача планирования пути — одно из ведущих направлений в развитии современной робототехники. В работе также даны некоторые оценки состояния этого направления исследований и возможных путей дальнейшего развития.

## Список литературы

1. Meng Wang, Liu J.N.K. Fuzzy logic-based real-time robot navigation in unknown environment with dead ends // Robotics and Autonomous Systems. 2008. Vol. 56. No. 7. Pp. 625–643. DOI: [10.1016/j.robot.2007.10.002](https://doi.org/10.1016/j.robot.2007.10.002)
2. Муравьиный алгоритм. Википедия: Web-сайт. Режим доступа: [https://ru.wikipedia.org/wiki/Муравьиный\\_алгоритм](https://ru.wikipedia.org/wiki/Муравьиный_алгоритм) (дата обращения 05.12.2017).
3. Mohamad M.M., Dunnigan M.W., Taylor N.K. Ant colony robot motion planning // EUROCON 2005: Intern. conf. on computer as a tool (Belgrade, Serbia, November 21-24, 2005): Proc. N.Y.: IEEE, 2005. Vol. 1. pp. 213–216. DOI: [10.1109/EURCON.2005.1629898](https://doi.org/10.1109/EURCON.2005.1629898)
4. Искусственная нейронная сеть. Википедия: Web-сайт. Режим доступа: [https://ru.wikipedia.org/wiki/Искусственная\\_нейронная\\_сеть](https://ru.wikipedia.org/wiki/Искусственная_нейронная_сеть) (дата обращения 05.12.2017).

5. Janet J.A., Luo R.C., Kay M.G. The essential visibility graph: An approach to global motion planning for autonomous mobile robots // IEEE intern. conf. on robotics and automation (Nagoya, Japan, May 21-27, 1995): Proc. Vol. 2. N.Y.: IEEE, 1995. Pp. 1958–1963.  
DOI: [10.1109/ROBOT.1995.526023](https://doi.org/10.1109/ROBOT.1995.526023)
6. Han-Pang Huang, Shu-Yun Chung. Dynamic visibility graph for path planning // IEEE-RSJ intern. conf. on intelligent robots and systems: IROS 2004 (Sendai, Japan, Sept. 28 - Oct. 2, 2004): Proc. N.Y.: IEEE, 2004. Vol. 3. Pp. 2813–2818. DOI: [10.1109/IROS.2004.1389835](https://doi.org/10.1109/IROS.2004.1389835)
7. Habib M.K., Asama H. Efficient method to generate collision free paths for an autonomous mobile robot based on new free space structuring approach // IEEE/RSJ intern. workshop on intelligent robots and systems: IROS'91 (Osaka, Japan, November 3-5, 1991): Proc. Vol. 2. N.Y.: IEEE, 1991. Pp. 563–567. DOI: [10.1109/IROS.1991.174534](https://doi.org/10.1109/IROS.1991.174534)
8. Wallgrun J. O. Voronoi graph matching for robot localization and mapping // Transactions on computational science IX. B.: Springer, 2010. Pp. 76–108. DOI: [10.1007/978-3-642-16007-3\\_4](https://doi.org/10.1007/978-3-642-16007-3_4)
9. Amato N.M., Wu Y. A randomized roadmap method for path and manipulation planning // IEEE intern. conf. on robotics and automation (Minneapolis, USA, April 22-28, 1996): Proc. Vol. 1. N.Y.: IEEE, 1996. Pp. 113–120. DOI: [10.1109/ROBOT.1996.503582](https://doi.org/10.1109/ROBOT.1996.503582)
10. Ladd A.M., Kavraki L.E. Measure theoretic analysis of probabilistic path planning // IEEE Trans. on Robotics and Automation. 2004. Vol. 20. No. 2. Pp. 229–242.  
DOI: [10.1109/TRA.2004.824649](https://doi.org/10.1109/TRA.2004.824649)
11. Geraerts R., Overmars M.H. A comparative study of probabilistic roadmap planners // Algorithmic foundations of robotics B.: Springer, 2004. Pp. 43–57. DOI: [10.1007/978-3-540-45058-0\\_4](https://doi.org/10.1007/978-3-540-45058-0_4)
12. LaValle S.M. Planning algorithms. Camb.; N.Y.: Camb. Univ. Press, 2006. 826 p.
13. Yang K., Sukkariéh S. 3D smooth path planning for a UAV in cluttered natural environments // IEEE/RSJ intern. conf. on intelligent robots and systems: IROS 2008 (Nice, France, Sept. 22-26, 2008): Proc. N.Y.: IEEE, 2008. Pp. 794–800.  
DOI: [10.1109/IROS.2008.4650637](https://doi.org/10.1109/IROS.2008.4650637)
14. Kuffner J.J., LaValle S.M. RRT-connect: An efficient approach to to single-query path planning // IEEE intern. conf. on robotics and automation: ICRA'2000 (San Francisco, CA, USA, April 24-28, 2000): Proc. N.Y.: IEEE, 2000. Vol. 2. Pp. 995–1001.  
DOI: [10.1109/ROBOT.2000.844730](https://doi.org/10.1109/ROBOT.2000.844730)
15. Sleumer N.H., Tschichold-Gurman N. Exact cell decomposition of arrangements used for path planning in robotics. Zurich: Inst. of Theoretical Computer Science, 1999.  
DOI: [10.3929/ethz-a-006653440](https://doi.org/10.3929/ethz-a-006653440)
16. Elfes A. Using occupancy grids for mobile robot perception and navigation // Computer. 1989. Vol. 22. No. 6. Pp. 46–57. DOI: [10.1109/2.30720](https://doi.org/10.1109/2.30720)
17. Yahja A., Stentz A., Singh S., Brumitt B.L. Framed-quadtrees path planning for mobile robots operating in sparse environments // IEEE intern. conf. on robotics and automation

- (Leuven, Belgium, May 20, 1998): Proc. N.Y.: IEEE, 1998. Vol. 1. Pp. 650–655.  
DOI: [10.1109/ROBOT.1998.677046](https://doi.org/10.1109/ROBOT.1998.677046)
18. Kitamura Y., Tanaka T., Kishino F., Yachida M. 3-D path planning in a dynamic environment using an octree and an artificial potential field // IEEE-RSJ intern. conf. on intelligent robots and systems: IROS'95 (Pittsburgh, PA, USA, Aug. 5-9, 1995): Proc. N.Y.: IEEE, 1995. Vol. 2. Pp. 474–481. DOI: [10.1109/IROS.1995.526259](https://doi.org/10.1109/IROS.1995.526259)
  19. Redding J., Amin J., Boskovic J., Kang Y., Hedrick K., Howlett J., Poll S. A real-time obstacle detection and reactive path planning system for autonomous small-scale helicopters // AIAA Guidance, navigation and control conf. and exhibit (Hilton Head, USA, Aug. 20–23, 2007): Proc. Wash.: AIAA, 2007. Pp. 989–1010. DOI: [10.2514/6.2007-6413](https://doi.org/10.2514/6.2007-6413)
  20. Chazelle B., Palios L. Triangulating a nonconvex polytope // Discrete and Computational Geometry. 1990. Vol. 5. No. 5. Pp. 505–526. DOI: [10.1007/BF02187807](https://doi.org/10.1007/BF02187807)
  21. Russell S.J., Norvig P. Artificial intelligence: A modern approach. 3<sup>rd</sup> ed. Upper Saddle River: Prentice Hall, 2010. 1132 pp.
  22. Ferguson D., Stentz A. Using interpolation to improve path planning: The field D\* algorithm // J. of Field Robotics. 2006. Vol. 23. No. 2. Pp. 79–101.  
DOI: [10.1002/rob.20109](https://doi.org/10.1002/rob.20109)
  23. Daniel K., Nash A., Koenig S., Felner A. Theta\*: Any-angle path planning on grids // J. of Artificial Intelligence Research. 2010. Vol. 39. Pp. 533–579. DOI: [10.1613/jair.2994](https://doi.org/10.1613/jair.2994)
  24. Stentz A. Optimal and efficient path planning for unknown and dynamic environments. Pittsburgh: The Robotics Inst.; Carnegie Mellon Univ., 1993. 38 p.
  25. Stentz A. The focussed D\* algorithm for real-time replanning // 14<sup>th</sup> intern. joint conf. on artificial intelligence: IJCAI'95 (Montreal, Canada, Aug. 20-25, 1995): Proc. Vol. 2. San Francisco: Morgan Kaufmann Publ., 1995. Pp. 1652–1659.
  26. Koenig S., Likhachev M., Furcy D. Lifelong planning A\* // Artificial Intelligence. 2004. Vol. 155. No. 1-2. Pp. 93–146. DOI: [10.1016/j.artint.2003.12.001](https://doi.org/10.1016/j.artint.2003.12.001)
  27. Koenig S., Likhachev M. D\* lite // 18<sup>th</sup> national conf. on artificial intelligence (Edmonton, Alberta, Canada, July 28–August 1, 2002): Proc. Menlo Park: AAAI Press, 2002. Pp. 476–483.
  28. De Filippis L., Guglieri G., Quagliotti F. A minimum risk approach for path planning of UAVs // J. of Intelligent and Robotic Systems. 2011. Vol. 61. No. 1–4. Pp. 203–219.  
DOI: [10.1007/s10846-010-9493-9](https://doi.org/10.1007/s10846-010-9493-9)
  29. De Filippis L., Guglieri G., Quagliotti F. Path planning strategies for UAVs in 3D environments // J. of Intelligent and Robotic Systems. 2012. Vol. 65. No.1–4. Pp. 247–264.  
DOI: [10.1007/s10846-011-9568-2](https://doi.org/10.1007/s10846-011-9568-2)
  30. De Filippis L. Advanced path planning and collision avoidance algorithms for UAVs: Doct. diss. Torino: Ist. Politecnico di Torino, 2012. 142 p.

31. Alvarez D., Gomez J.V., Garrido S., Moreno L. 3D robot formations path planning with fast marching square // J. of Intelligent and Robotic Systems. 2015. Vol. 80. No. 3-4. Pp. 507–523. DOI: [10.1007/s10846-015-0187-1](https://doi.org/10.1007/s10846-015-0187-1)
32. Osher S., Sethian J.A. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations // J. of Computational Physics. 1988. Vol. 79. No. 1. Pp. 12–49. DOI: [10.1016/0021-9991\(88\)90002-2](https://doi.org/10.1016/0021-9991(88)90002-2)
33. Ge S.S., Cui Y.J. New potential functions for mobile robot path planning // IEEE Trans. on Robotics and Automation. 2000. Vol. 16. No. 5. Pp. 615–620. DOI: [10.1109/70.880813](https://doi.org/10.1109/70.880813)
34. Jing R., McIsaac K.A., Patel R.V. Modified Newton's method applied to potential field-based navigation for mobile robots // IEEE Trans. on Robotics. 2006. Vol. 22. No. 2. Pp. 384–391. DOI: [10.1109/TRO.2006.870668](https://doi.org/10.1109/TRO.2006.870668)
35. Ferrara A., Rubagotti M. Second-order sliding-mode control of a mobile robot based on a harmonic potential field // IET Control Theory and Applications. 2008. Vol. 2. No. 9. Pp. 807–818. DOI: [10.1049/iet-cta:20070424](https://doi.org/10.1049/iet-cta:20070424)
36. Khatib O. Real-time obstacle avoidance for manipulators and mobile robots // Intern. J. of Robotics Research. 1986. Vol. 5. No. 1. pp. 90–98. DOI: [10.1177/027836498600500106](https://doi.org/10.1177/027836498600500106)
37. Fujimura K., Samet H. A hierarchical strategy for path planning among moving obstacles (mobile robot) // IEEE Trans. on Robotics and Automation. 1989. Vol. 5. No. 1. Pp. 61–69. DOI: [10.1109/70.88018](https://doi.org/10.1109/70.88018)
38. Conn R.A., Kam M. Robot motion planning on N-dimensional star worlds among moving obstacles // IEEE Trans. on Robotics and Automation. 1998. Vol. 14. No. 2. Pp. 320–325. DOI: [10.1109/70.681250](https://doi.org/10.1109/70.681250)
39. Mabrouk M.H., McInnes C.R. Solving the potential field local minimum problem using internal agent states // Robotics and Autonomous Systems. 2008. Vol. 56. No. 12. Pp. 1050–1060. DOI: [10.1016/j.robot.2008.09.006](https://doi.org/10.1016/j.robot.2008.09.006)
40. Zou Xi-yong, Zhu Jing. Virtual local target method for avoiding local minimum in potential field based robot navigation // J. of Zhejiang Univ. - Science A. 2003. Vol. 4. No. 3. Pp. 264–269. DOI: [10.1631/jzus.2003.0264](https://doi.org/10.1631/jzus.2003.0264)
41. Masoud A.A. Solving the narrow corridor problem in potential field-guided autonomous robots // IEEE intern. conf. on robotics and automation: ICRA 2005 (Barcelona, Spain, April 18-22, 2005): Proc. N.Y.: IEEE, 2005. Pp. 2909–2914. DOI: [10.1109/ROBOT.2005.1570555](https://doi.org/10.1109/ROBOT.2005.1570555)
42. Fan Xiao-ping, Li Shuang-yan, Chen Te-fang. Dynamic obstacle-avoiding path plan for robots based on a new artificial potential field function // Control Theory and Applications. 2005. Vol. 22. No. 5. Pp. 703–707. Режим доступа: [http://en.cnki.com.cn/Article\\_en/CJFDTOTAL-KZLY200505005.htm](http://en.cnki.com.cn/Article_en/CJFDTOTAL-KZLY200505005.htm) (дата обращения 05.12.2017).

43. Borenstein J., Koren Y. Real-time obstacle avoidance for fast mobile robots // IEEE Trans. on Systems, Man, and Cybernetics. 1989. Vol. 19. No. 5. Pp. 1179–1187.  
DOI: [10.1109/21.44033](https://doi.org/10.1109/21.44033)
44. Borenstein J., Koren Y. The vector field histogram-fast obstacle avoidance for mobile robots // IEEE Trans. on Robotics and Automation. 1991. Vol. 7. No. 3. Pp. 278–288.  
DOI: [10.1109/70.88137](https://doi.org/10.1109/70.88137)
45. Ulrich I., Borenstein J. VFH+: Reliable obstacle avoidance for fast mobile robots // IEEE intern. conf. on robotics and automation (Leuven, Belgium, May 20, 1998): Proc. N.Y.: IEEE, 1998. Vol. 2. Pp. 1572–1577. DOI: [10.1109/ROBOT.1998.677362](https://doi.org/10.1109/ROBOT.1998.677362)
46. Ulrich I., Borenstein J. VFH\*: Local obstacle avoidance with look-ahead verification // IEEE intern. conf. on robotics and automation: ICRA'00 (San Francisco, CA, USA, April 24-28, 2000): Proc. N.Y.: IEEE, 2000. Vol. 3. Pp. 2505–2511.  
DOI: [10.1109/ROBOT.2000.846405](https://doi.org/10.1109/ROBOT.2000.846405)
47. Betts J.T. Survey of numerical methods for trajectory optimization // J. of Guidance, Control and Dynamics. 1998. Vol. 21. No. 2. Pp. 193–207. DOI: [10.2514/2.4231](https://doi.org/10.2514/2.4231)
48. Ross I.M., Fahroo F. A perspective on methods for trajectory optimization // AIAA/AAS Astrodynamics specialist conf. and exhibit (Monterey, CA, USA, August 5-8, 2002): Proc. Wash.: AIAA, 2002. Pp. 1–7. DOI: [10.2514/6.2002-4727](https://doi.org/10.2514/6.2002-4727)
49. Fliess M., Levine J., Martin P., Rouchon P. Flatness and defect of non-linear systems: introductory theory and examples // Intern. J. of Control. 1995. Vol. 61. No. 6. Pp. 1327–1361. DOI: [10.1080/00207179508921959](https://doi.org/10.1080/00207179508921959)
50. Gill P.E., Murray W., Saunders M.A. User's guide for SNOPT Version 7: software for large scale nonlinear programming. Stanford: Stanford Univ. Publ., 2006. 116 p.
51. Gill P.E., Murray W., Saunders M.A., Wright M.H. User's guide for NPSOL (version 4.0): a Fortran package for nonlinear programming. Stanford: Stanford Univ., 1986.
52. Culligan K., Valenti M., Kuwata Y., How J.P. Three-dimensional flight experiments using on-line mixed-integer linear programming trajectory optimization // Amer. control conf.: ACC'2007 (New York, NY, USA, July 9-13, 2007): Proc. N.Y.: IEEE, 2007. Pp. 5322–5327. DOI: [10.1109/ACC.2007.4283101](https://doi.org/10.1109/ACC.2007.4283101)
53. Schouwenaars T., De Moor B., Feron E., How J. Mixed integer programming for multi-vehicle path planning // Eur. control conf.: ECC 2001 (Porto, Portugal, Sept. 4-7, 2001): Proc. N.Y.: IEEE, 2001. Pp. 2603–2608.
54. Earl M.G., D'Andrea R. Iterative MILP methods for vehicle-control problems // IEEE Trans. on Robotics. 2005. Vol. 21. No. 6. Pp. 1158–1167. DOI: [10.1109/TRO.2005.853499](https://doi.org/10.1109/TRO.2005.853499)
55. Kuwata Y. Real-time trajectory design for unmanned aerial vehicles using receding horizon control: Doct. diss. Camb., MA: Massachusetts Inst. of Technology, 2003. 151 p.
56. Habibi G., Masehian E., Beheshti M.T.H. Binary integer programming model of point robot path planning // 33<sup>rd</sup> annual conf. of the IEEE Industrial Electronics Soc.: IECON 2007

- (Taipei, Taiwan, Nov. 5-8, 2007): Proc. N.Y.: IEEE, 2007. Pp. 2841–2845.  
DOI: [10.1109/IECON.2007.4460315](https://doi.org/10.1109/IECON.2007.4460315)
57. Masehian E., Habibi G. Robot path planning in 3D space using binary integer programming // Intern. J. of Computer, Information, Systems and Control Engineering. 2007. Vol. 1. No. 5. Pp. 1240-1245.
58. Masehian E., Habibi G. Motion planning and control of mobile robot using Linear Matrix Inequalities (LMIs) // IEEE/RSJ intern. conf. on intelligent robots and systems: IROS 2007 (San Diego, CA, USA, Oct. 29 - Nov. 2, 2007): Proc. N.Y. IEEE, 2007. Pp. 4277–4282.  
DOI: [10.1109/IROS.2007.4399641](https://doi.org/10.1109/IROS.2007.4399641)
59. Dorigo M., Birattari M., Stutzle T. Ant colony optimization // IEEE Computational Intelligence Magazine. 2006. Vol. 1. No. 4. Pp. 28–39. DOI: [10.1109/MCI.2006.329691](https://doi.org/10.1109/MCI.2006.329691)
60. Mohamad M.M., Dunnigan M.W., Taylor N.K. Ant colony robot motion planning // Intern. conf. on “Computer as a tool”: EUROCON 2005 (Belgrade, Serbia, Nov. 21-24, 2005): Proc. N.Y.: IEEE, 2005. Vol. 1. Pp. 213–216. DOI: [10.1109/EURCON.2005.1629898](https://doi.org/10.1109/EURCON.2005.1629898)
61. Гэн К.К., Тань Лиго, Чулин Н.А., Хэ Юн. Планирование маршрута для квадрокоптера в неизвестной среде на основе монокулярного компьютерного зрения // Автоматизация. Современные технологии. 2015. № 12. С. 14–19.
62. Мак-Каллок У.С., Питтс В. Логическое исчисление идей, относящихся к нервной активности // Автоматы: Сб. / Под ред. К.Э. Шеннона, Дж. МакКарти. М.: Изд-во иностр. лит., 1956. С. 363–384.
63. Glasius R., Komoda A., Stan C.A.M. Gielen. Neural network dynamics for path planning and obstacle avoidance // Neural Networks. 1995. Vol. 8. No. 1. Pp. 125–133.  
DOI: [10.1016/0893-6080\(94\)E0045-M](https://doi.org/10.1016/0893-6080(94)E0045-M)
64. Moreno J.A., Castro M. Heuristic algorithm for robot path planning based on a growing elastic net // Progress in artificial intelligence: 12<sup>th</sup> Portuguese conf. on artificial intelligence: EPIA 2005 (Covilhã, Portugal, December 5-8, 2005): Proc. B.: Springer, 2005. Pp. 447-454.  
DOI: [10.1007/11595014\\_44](https://doi.org/10.1007/11595014_44)
65. Fu X., Gao X., Chen D. A Bayesian optimization algorithm for UAV path planning // Intelligent information processing II: Intern. conf. on intelligent information processing: IIP 2004 (Beijing, China, Oct. 21-23, 2004): Proc. Boston: Springer, 2005. Pp. 227–232.  
DOI: [10.1007/0-387-23152-8\\_29](https://doi.org/10.1007/0-387-23152-8_29)
66. Eberhart R., Kennedy J. A new optimizer using particle swarm theory // 6<sup>th</sup> intern. symp. on micromachine and human science: MHS'95 (Nagoya, Japan, Oct. 4-6, 1995): Proc. N.Y.: IEEE, 1995. Pp. 39–43. DOI: [10.1109/MHS.1995.494215](https://doi.org/10.1109/MHS.1995.494215)
67. Jung L.F., Knutzon J.S., Oliver J.H., Winer E.H. Three-dimensional path planning of unmanned aerial vehicles using particle swarm optimization // 11<sup>th</sup> AIAA/ISSMO multidisciplinary analysis and optimization conf. (Portsmouth, Virginia, USA, September 6–8, 2006): Proc. Wash.: AIAA, 2006. Pp. 992–1001. DOI: [10.2514/6.2006-6995](https://doi.org/10.2514/6.2006-6995)

68. Huq R., Mann G.K.I., Gosine R.G. Mobile robot navigation using motor schema and fuzzy context dependent behavior modulation // *Applied Soft Computing*. 2008. Vol. 8. No. 1. Pp. 422–436. DOI: [10.1016/j.asoc.2007.02.006](https://doi.org/10.1016/j.asoc.2007.02.006)
69. Egerstedt M. Behavior based robotics using hybrid automata // *Hybrid systems: Computation and control: 3<sup>rd</sup> intern. workshop on hybrid systems: HSCC 2000* (Pittsburgh, PA, USA, March 23-25, 2000): Proc. B.: Springer, 2000. Pp. 103–116. DOI: [10.1007/3-540-46430-1\\_12](https://doi.org/10.1007/3-540-46430-1_12)
70. Ma J.-C., Zhang Q., Ma L.-Y., Xie W. Multi-behavior fusion-based path planning for mobile robot // *Beijing Ligong Daxue Xuebao / Trans. of Beijing Inst. of Technology*. 2014. Vol. 34. No. 6. Pp. 576–581.
71. Motlagh O.R.E., Hong T.S., Ismail N. Development of a new minimum avoidance system for a behavior-based mobile robot // *Fuzzy Sets and Systems*. 2009. Vol. 160. No. 13. Pp. 1929–1946. DOI: [10.1016/j.fss.2008.09.015](https://doi.org/10.1016/j.fss.2008.09.015)
72. Бекасов Д.Е. Применение аппарата нечеткой логики при решении задачи поиска пути в неизвестном окружении // *Молодежный науч.-техн. вестник. МГТУ им. Н.Э. Баумана: электрон. журн.* 2012. No. 5. С. 40.
73. Keke G., Wei L., Ligu T. A fuzzy controller: Using monocular computer vision to see and avoid obstacle for quadcopter // *5<sup>th</sup> intern. workshop on computer science and engineering: Information processing and control engineering: WCSE 2015-IPSE* (Moscow, Russia, April 15-17, 2015): Proc. Chenghu: Science and Engineering Inst., 2015.
74. Ng J., Braunl T. Performance comparison of bug navigation algorithms // *J. of Intelligent and Robotic Systems*. 2007. Vol. 50. No. 1. Pp. 73–84. DOI: [10.1007/s10846-007-9157-6](https://doi.org/10.1007/s10846-007-9157-6)
75. Lumelsky V., Stepanov A. Dynamic path planning for a mobile automaton with limited information on the environment // *IEEE Trans. on Automatic Control*. 1986. Vol. 31. No. 11. Pp. 1058–1063. DOI: [10.1109/TAC.1986.1104175](https://doi.org/10.1109/TAC.1986.1104175)
76. Yufka A., Parlaktuna O. Performance comparison of the BUG's algorithms for mobile robots // *Intern. symp. on INnovations in intelligent SYSTems and applications: INISTA 2009* (Trabzon, Turkey, June 29–July 1, 2009): Proc. N.Y.: IEEE, 2009. Pp. 416–421.
77. Shi C., Bu Y., Liu J. Mobile robot path planning in three-dimensional environment based on ACO-PSO hybrid algorithm // *IEEE/ASME intern. conf. on advanced intelligent mechatronics: AIM 2008* (Xian, China, July 2-5, 2008): Proc. N.Y.: IEEE, 2008. Pp. 252–256. DOI: [10.1109/AIM.2008.4601668](https://doi.org/10.1109/AIM.2008.4601668)
78. Mettler B., Toupet O. Receding horizon trajectory planning with an environment-based cost-to-go function // *44<sup>th</sup> IEEE conf. on decision and control and the European control conf.: CDC-ECC'05* (Seville, Spain, Dec. 15, 2005): Proc. N.Y.: IEEE, 2005. Pp. 4071–4076. DOI: [10.1109/CDC.2005.1582799](https://doi.org/10.1109/CDC.2005.1582799)
79. Gilimyanov R.F., Pesterev A.V., Rapoport L.B. Smoothing curvature of trajectories constructed by noisy measurements in path planning problems for wheeled robots // *J. of*

Computer and Systems Sciences International. 2008. Vol. 47. No. 5. Pp. 812–819.

DOI: [10.1134/S1064230708050158](https://doi.org/10.1134/S1064230708050158)

80. Lutterkort D., Peters J. Smooth paths in a polygonal channel // 15<sup>th</sup> annual symp. on computational geometry: SCG'99 (Miami Beach, FLA, USA, June 13-16, 1999): Proc. N.Y.: ACM Press, 1999. Pp. 316–321. DOI: [10.1145/304893.304985](https://doi.org/10.1145/304893.304985)
81. Jung D., Tsiotras P. On-line path generation for unmanned aerial vehicles using B-spline path templates // J. of Guidance, Control, and Dynamics. 2013. Vol. 36. No. 6. Pp. 1642–1653. DOI: [10.2514/1.60780](https://doi.org/10.2514/1.60780)
82. Zhao Y., Tsiotras P. A quadratic programming approach to path smoothing // Amer. control conf.: ACC 2011 (San Francisco, CA, USA, June 29 – July 1, 2011): Proc. N.Y.: IEEE, 2011. Pp. 5324–5329. DOI: [10.1109/ACC.2011.5990880](https://doi.org/10.1109/ACC.2011.5990880)
83. Гилимьянов Р.Ф., Рапопорт Л.Б. Метод деформации пути в задачах планирования движения роботов при наличии препятствий // Проблемы управления. 2012. № 1. С. 70–76.
84. Dubins L.E. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents // Amer. J. of Mathematics. 1957. Vol. 79. No. 3. Pp. 497–516. DOI: [10.2307/2372560](https://doi.org/10.2307/2372560)
85. Nelson W. Continuous-curvature paths for autonomous vehicles // IEEE intern. conf. on robotics and automation (Scottsdale, AZ, USA, May 14-19, 1989): Proc. N.Y.: IEEE, 1989. Vol. 3. Pp. 1260–1264. DOI: [10.1109/ROBOT.1989.100153](https://doi.org/10.1109/ROBOT.1989.100153)
86. Van der Molen G.M. Trajectory generation for mobile robots with clothoids // Robotic systems: Advanced techniques and applications. Dordrecht: Springer, 1992. Pp. 399–406. DOI: [10.1007/978-94-011-2526-0\\_46](https://doi.org/10.1007/978-94-011-2526-0_46)
87. Walton D.J., Meek D.S., Ali J.M. Planar G2 transition curves composed of cubic Bézier spiral segments // J. of Computational and Applied Mathematics. 2003. Vol. 157. No. 2. Pp. 453–476. DOI: [10.1016/S0377-0427\(03\)00435-7](https://doi.org/10.1016/S0377-0427(03)00435-7)
88. Komoriya K., Tanie K. Trajectory design and control of a wheel-type mobile robot using B-spline curve // IEEE/RSJ intern. workshop on intelligent robots and systems: IROS'89 (Tsukuba, Japan, Sept. 4-6, 1989): Proc. N.Y.: IEEE, 1989. Pp. 398–405. DOI: [10.1109/IROS.1989.637937](https://doi.org/10.1109/IROS.1989.637937)
89. Berglund T., Jonsson H., Soderkvist I. An obstacle-avoiding minimum variation B-spline problem // Intern. conf. on geometric modeling and graphics (London, UK, July 16–18, 2003): Proc. N.Y.: IEEE, 2003. Pp. 156–161. DOI: [10.1109/GMAG.2003.1219681](https://doi.org/10.1109/GMAG.2003.1219681)
90. Wang Z., Zhang W., Li G., Mu X. G2 path smoothing using non-uniform B-spline // Systems Engineering and Electronics. 2011. No. 7. Pp. 1539–1543. Режим доступа: [http://en.cnki.com.cn/Article\\_en/CJFDTOTAL-XTYD201107021.htm](http://en.cnki.com.cn/Article_en/CJFDTOTAL-XTYD201107021.htm) (дата обращения 05.12.2017).

## Path Planning Methods in an Environment with Obstacles (A Review)

W. Liu<sup>1,\*</sup>

[\\*cherub200711@163.com](mailto:cherub200711@163.com)

<sup>1</sup>Bauman Moscow State Technical University, Moscow, Russia

---

**Keywords:** path planning, environment with obstacles, robot, road map

---

Planning the path is the most important task in the mobile robot navigation. This task involves basically three aspects. First, the planned path must run from a given starting point to a given endpoint. Secondly, it should ensure robot's collision-free movement. Thirdly, among all the possible paths that meet the first two requirements it must be, in a certain sense, optimal.

Methods of path planning can be classified according to different characteristics. In the context of using intelligent technologies, they can be divided into traditional methods and heuristic ones. By the nature of the environment, it is possible to divide planning methods into planning methods in a static environment and in a dynamic one (it should be noted, however, that a static environment is rare). Methods can also be divided according to the completeness of information about the environment, namely methods with complete information (in this case the issue is a global path planning) and methods with incomplete information (usually, this refers to the situational awareness in the immediate vicinity of the robot, in this case it is a local path planning). Note that incomplete information about the environment can be a consequence of the changing environment, i.e. in a dynamic environment, there is, usually, a local path planning.

Literature offers a great deal of methods for path planning where various heuristic techniques are used, which, as a rule, result from the denotative meaning of the problem being solved. This review discusses the main approaches to the problem solution. Here we can distinguish five classes of basic methods: graph-based methods, methods based on cell decomposition, use of potential fields, optimization methods,  $\phi$ TV methods based on intelligent technologies.

Many methods of path planning, as a result, give a chain of reference points (waypoints) connecting the beginning and end of the path. This should be seen as an intermediate result. The problem to route the reference points along the constructed chain arises. It is called the task of smoothing the path, and the review addresses this problem as well.

## References

1. Meng Wang, Liu J.N.K. Fuzzy logic-based real-time robot navigation in unknown environment with dead ends. *Robotics and Autonomous Systems*, 2008, vol. 56, no. 7, pp. 625–643. DOI: [10.1016/j.robot.2007.10.002](https://doi.org/10.1016/j.robot.2007.10.002)
2. *Murav'inyj algoritm* [Ant algorithm]: Wikipedia. Available at: [https://ru.wikipedia.org/wiki/Муравьиный\\_алгоритм](https://ru.wikipedia.org/wiki/Муравьиный_алгоритм), accessed 05.12.2017 (in Russian).
3. Mohamad M.M., Dunnigan M.W., Taylor N.K. Ant colony robot motion planning. *EUROCON 2005: Intern. conf. on computer as a tool* (Belgrade, Serbia, November 21-24, 2005): Proc. N.Y.: IEEE, 2005. Vol. 1. pp. 213–216. DOI: [10.1109/EURCON.2005.1629898](https://doi.org/10.1109/EURCON.2005.1629898)
4. *Iskusstvennaia nejronnaia set'* [Artificial neural network]: Wikipedia. Available at: [https://ru.wikipedia.org/wiki/Искусственная\\_нейронная\\_сеть](https://ru.wikipedia.org/wiki/Искусственная_нейронная_сеть), accessed 05.12.2017 (in Russian).
5. Janet J.A., Luo R.C., Kay M.G. The essential visibility graph: An approach to global motion planning for autonomous mobile robots. *IEEE intern. conf. on robotics and automation* (Nagoya, Japan, May 21-27, 1995): Proc. Vol. 2. N.Y.: IEEE, 1995. Pp. 1958–1963. DOI: [10.1109/ROBOT.1995.526023](https://doi.org/10.1109/ROBOT.1995.526023)
6. Han-Pang Huang, Shu-Yun Chung. Dynamic visibility graph for path planning. *IEEE-RSJ intern. conf. on intelligent robots and systems: IROS 2004* (Sendai, Japan, Sept. 28 - Oct. 2, 2004): Proc. N.Y.: IEEE, 2004. Vol. 3. Pp. 2813–2818. DOI: [10.1109/IROS.2004.1389835](https://doi.org/10.1109/IROS.2004.1389835)
7. Habib M.K., Asama H. Efficient method to generate collision free paths for an autonomous mobile robot based on new free space structuring approach. *IEEE/RSJ intern. workshop on intelligent robots and systems: IROS'91* (Osaka, Japan, November 3-5, 1991): Proc. Vol. 2. N.Y.: IEEE, 1991. Pp. 563–567. DOI: [10.1109/IROS.1991.174534](https://doi.org/10.1109/IROS.1991.174534)
8. Wallgrun J. O. Voronoi graph matching for robot localization and mapping. *Transactions on computational science IX*. B.: Springer, 2010. Pp. 76–108. DOI: [10.1007/978-3-642-16007-3\\_4](https://doi.org/10.1007/978-3-642-16007-3_4)
9. Amato N.M., Wu Y. A randomized roadmap method for path and manipulation planning. *IEEE intern. conf. on robotics and automation* (Minneapolis, USA, April 22-28, 1996): Proc. Vol. 1. N.Y.: IEEE, 1996. Pp. 113–120. DOI: [10.1109/ROBOT.1996.503582](https://doi.org/10.1109/ROBOT.1996.503582)
10. Ladd A.M., Kavraki L.E. Measure theoretic analysis of probabilistic path planning. *IEEE Trans. on Robotics and Automation*, 2004, vol. 20, no. 2, pp. 229–242. DOI: [10.1109/TRA.2004.824649](https://doi.org/10.1109/TRA.2004.824649)
11. Geraerts R., Overmars M.H. A comparative study of probabilistic roadmap planners. *Algorithmic foundations of robotics*. B.: Springer, 2004. Pp. 43–57. DOI: [10.1007/978-3-540-45058-0\\_4](https://doi.org/10.1007/978-3-540-45058-0_4)
12. LaValle S.M. *Planning algorithms*. Camb.; N.Y.: Camb. Univ. Press, 2006. 826 p.

13. Yang K., Sukkarieh S. 3D smooth path planning for a UAV in cluttered natural environments. *IEEE/RSJ intern. conf. on intelligent robots and systems: IROS 2008* (Nice, France, Sept. 22-26, 2008): Proc. N.Y.: IEEE, 2008. Pp. 794–800.  
DOI: [10.1109/IROS.2008.4650637](https://doi.org/10.1109/IROS.2008.4650637)
14. Kuffner J.J., LaValle S.M. RRT-connect: An efficient approach to to single-query path planning. *IEEE intern. conf. on robotics and automation: ICRA '2000* (San Francisco, CA, USA, April 24-28, 2000): Proc. N.Y.: IEEE, 2000. Vol. 2. Pp. 995–1001.  
DOI: [10.1109/ROBOT.2000.844730](https://doi.org/10.1109/ROBOT.2000.844730)
15. Sleumer N.H., Tschichold-Gurman N. Exact cell decomposition of arrangements used for path planning in robotics. Zurich: Inst. of Theoretical Computer Science, 1999.  
DOI: [10.3929/ethz-a-006653440](https://doi.org/10.3929/ethz-a-006653440)
16. Elfes A. Using occupancy grids for mobile robot perception and navigation. *Computer*, 1989, vol. 22, no. 6, pp. 46–57. DOI: [10.1109/2.30720](https://doi.org/10.1109/2.30720)
17. Yahja A., Stentz A., Singh S., Brumitt B.L. Framed-quadtrees path planning for mobile robots operating in sparse environments. *IEEE intern. conf. on robotics and automation* (Leuven, Belgium, May 20, 1998): Proc. N.Y.: IEEE, 1998. Vol. 1. Pp. 650–655.  
DOI: [10.1109/ROBOT.1998.677046](https://doi.org/10.1109/ROBOT.1998.677046)
18. Kitamura Y., Tanaka T., Kishino F., Yachida M. 3-D path planning in a dynamic environment using an octree and an artificial potential field. *IEEE-RSJ intern. conf. on intelligent robots and systems: IROS'95* (Pittsburgh, PA, USA, Aug. 5-9, 1995): Proc. N.Y.: IEEE, 1995. Vol. 2. Pp. 474–481. DOI: [10.1109/IROS.1995.526259](https://doi.org/10.1109/IROS.1995.526259)
19. Redding J., Amin J., Boskovic J., Kang Y., Hedrick K., Howlett J., Poll S. A real-time obstacle detection and reactive path planning system for autonomous small-scale helicopters. *AIAA Guidance, navigation and control conf. and exhibit* (Hilton Head, USA, Aug. 20–23, 2007): Proc. Wash.: AIAA, 2007. Pp. 989–1010. DOI: [10.2514/6.2007-6413](https://doi.org/10.2514/6.2007-6413)
20. Chazelle B., Palios L. Triangulating a nonconvex polytope. *Discrete and Computational Geometry*, 1990, vol. 5, no. 5, pp. 505–526. DOI: [10.1007/BF02187807](https://doi.org/10.1007/BF02187807)
21. Russell S.J., Norvig P. Artificial intelligence: A modern approach. 3<sup>rd</sup> ed. Upper Saddle River: Prentice Hall, 2010. 1132 pp.
22. Ferguson D., Stentz A. Using interpolation to improve path planning: The field D\* algorithm. *J. of Field Robotics*, 2006, vol. 23, no. 2, pp. 79–101. DOI: [10.1002/rob.20109](https://doi.org/10.1002/rob.20109)
23. Daniel K., Nash A., Koenig S., Felner A. Theta\*: Any-angle path planning on grids. *J. of Artificial Intelligence Research*, 2010, vol. 39, pp. 533–579. DOI: [10.1613/jair.2994](https://doi.org/10.1613/jair.2994)
24. Stentz A. Optimal and efficient path planning for unknown and dynamic environments. Pittsburgh: The Robotics Inst.; Carnegie Mellon Univ., 1993. 38 p.
25. Stentz A. The focussed D\* algorithm for real-time replanning. *14<sup>th</sup> intern. joint conf. on artificial intelligence: IJCAI'95* (Montreal, Canada, Aug. 20-25, 1995): Proc. Vol. 2. San Francisco: Morgan Kaufmann Publ., 1995. Pp. 1652–1659.

26. Koenig S., Likhachev M., Furcy D. Lifelong planning A\*. *Artificial Intelligence*, 2004, vol. 155, no. 1-2, pp. 93–146. DOI: [10.1016/j.artint.2003.12.001](https://doi.org/10.1016/j.artint.2003.12.001)
27. Koenig S., Likhachev M. D\* lite. *18<sup>th</sup> national conf. on artificial intelligence* (Edmonton, Alberta, Canada, July 28–August 1, 2002): Proc. Menlo Park: AAAI Press, 2002. Pp. 476–483.
28. De Filippis L., Guglieri G., Quagliotti F. A minimum risk approach for path planning of UAVs. *J. of Intelligent and Robotic Systems*, 2011, vol. 61, no. 1–4, pp. 203–219. DOI: [10.1007/s10846-010-9493-9](https://doi.org/10.1007/s10846-010-9493-9)
29. De Filippis L., Guglieri G., Quagliotti F. Path planning strategies for UAVs in 3D environments. *J. of Intelligent and Robotic Systems*, 2012, vol. 65, no. 1–4, pp. 247–264. DOI: [10.1007/s10846-011-9568-2](https://doi.org/10.1007/s10846-011-9568-2)
30. De Filippis L. Advanced path planning and collision avoidance algorithms for UAVs: Doct. diss. Torino: Ist. Politecnico di Torino, 2012. 142 p.
31. Alvarez D., Gomez J.V., Garrido S., Moreno L. 3D robot formations path planning with fast marching square. *J. of Intelligent and Robotic Systems*, 2015, vol. 80, no. 3-4, pp. 507–523. DOI: [10.1007/s10846-015-0187-1](https://doi.org/10.1007/s10846-015-0187-1)
32. Osher S., Sethian J.A. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. of Computational Physics*, 1988, vol. 79, no. 1, pp. 12–49. DOI: [10.1016/0021-9991\(88\)90002-2](https://doi.org/10.1016/0021-9991(88)90002-2)
33. Ge S.S., Cui Y.J. New potential functions for mobile robot path planning. *IEEE Trans. on Robotics and Automation*, 2000, vol. 16, no. 5, pp. 615–620. DOI: [10.1109/70.880813](https://doi.org/10.1109/70.880813)
34. Jing R., McIsaac K.A., Patel R.V. Modified Newton's method applied to potential field-based navigation for mobile robots. *IEEE Trans. on Robotics*, 2006, vol. 22, no. 2, pp. 384–391. DOI: [10.1109/TRO.2006.870668](https://doi.org/10.1109/TRO.2006.870668)
35. Ferrara A., Rubagotti M. Second-order sliding-mode control of a mobile robot based on a harmonic potential field. *IET Control Theory and Applications*, 2008, vol. 2, no. 9, pp. 807–818. DOI: [10.1049/iet-cta:20070424](https://doi.org/10.1049/iet-cta:20070424)
36. Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. *Intern. J. of Robotics Research*, 1986, vol. 5, no. 1, pp. 90–98. DOI: [10.1177/027836498600500106](https://doi.org/10.1177/027836498600500106)
37. Fujimura K., Samet H. A hierarchical strategy for path planning among moving obstacles (mobile robot). *IEEE Trans. on Robotics and Automation*, 1989, vol. 5, no. 1, pp. 61–69. DOI: [10.1109/70.88018](https://doi.org/10.1109/70.88018)
38. Conn R.A., Kam M. Robot motion planning on N-dimensional star worlds among moving obstacles. *IEEE Trans. on Robotics and Automation*, 1998, vol. 14, no. 2, pp. 320–325. DOI: [10.1109/70.681250](https://doi.org/10.1109/70.681250)
39. Mabrouk M.H., McInnes C.R. Solving the potential field local minimum problem using internal agent states. *Robotics and Autonomous Systems*, 2008, vol. 56, no. 12, pp. 1050–1060. DOI: [10.1016/j.robot.2008.09.006](https://doi.org/10.1016/j.robot.2008.09.006)

40. Zou Xi-yong, Zhu Jing. Virtual local target method for avoiding local minimum in potential field based robot navigation. *J. of Zhejiang Univ. - Science A*, 2003, vol. 4, no. 3, pp. 264–269. DOI: [10.1631/jzus.2003.0264](https://doi.org/10.1631/jzus.2003.0264)
41. Masoud A.A. Solving the narrow corridor problem in potential field-guided autonomous robots. *IEEE intern. conf. on robotics and automation: ICRA 2005* (Barcelona, Spain, April 18-22, 2005): Proc. N.Y.: IEEE, 2005. Pp. 2909–2914. DOI: [10.1109/ROBOT.2005.1570555](https://doi.org/10.1109/ROBOT.2005.1570555)
42. Fan Xiao-ping, Li Shuang-yan, Chen Te-fang. Dynamic obstacle-avoiding path plan for robots based on a new artificial potential field function. *Control Theory and Applications*, 2005, vol. 22, no. 5. Pp. 703–707. Available at: [http://en.cnki.com.cn/Article\\_en/CJFDTOTAL-KZLY200505005.htm](http://en.cnki.com.cn/Article_en/CJFDTOTAL-KZLY200505005.htm), accessed 05.12.2017.
43. Borenstein J., Koren Y. Real-time obstacle avoidance for fast mobile robots. *IEEE Trans. on Systems, Man, and Cybernetics*, 1989, vol. 19, no. 5, pp. 1179–1187. DOI: [10.1109/21.44033](https://doi.org/10.1109/21.44033)
44. Borenstein J., Koren Y. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Trans. on Robotics and Automation*, 1991, vol. 7, no. 3, pp. 278–288. DOI: [10.1109/70.88137](https://doi.org/10.1109/70.88137)
45. Ulrich I., Borenstein J. VFH+: Reliable obstacle avoidance for fast mobile robots. *IEEE intern. conf. on robotics and automation* (Leuven, Belgium, May 20, 1998): Proc. N.Y.: IEEE, 1998. Vol. 2. Pp. 1572–1577. DOI: [10.1109/ROBOT.1998.677362](https://doi.org/10.1109/ROBOT.1998.677362)
46. Ulrich I., Borenstein J. VFH\*: Local obstacle avoidance with look-ahead verification. *IEEE intern. conf. on robotics and automation: ICRA '00* (San Francisco, CA, USA, April 24-28, 2000): Proc. N.Y.: IEEE, 2000. Vol. 3. Pp. 2505–2511. DOI: [10.1109/ROBOT.2000.846405](https://doi.org/10.1109/ROBOT.2000.846405)
47. Betts J.T. Survey of numerical methods for trajectory optimization. *J. of Guidance, Control and Dynamics*, 1998, vol. 21, no. 2, pp. 193–207. DOI: [10.2514/2.4231](https://doi.org/10.2514/2.4231)
48. Ross I.M., Fahroo F. A perspective on methods for trajectory optimization. *AIAA/AAS Astrodynamics specialist conf. and exhibit* (Monterey, CA, USA, August 5-8, 2002): Proc. Wash.: AIAA, 2002. Pp. 1–7. DOI: [10.2514/6.2002-4727](https://doi.org/10.2514/6.2002-4727)
49. Fliess M., Levine J., Martin P., Rouchon P. Flatness and defect of non-linear systems: introductory theory and examples. *Intern. J. of Control*, 1995, vol. 61, no. 6, pp. 1327–1361. DOI: [10.1080/00207179508921959](https://doi.org/10.1080/00207179508921959)
50. Gill P.E., Murray W., Saunders M.A. User's guide for SNOPT Version 7: software for large scale nonlinear programming. Stanford: Stanford Univ. Publ., 2006. 116 p.
51. Gill P.E., Murray W., Saunders M.A., Wright M.H. User's guide for NPSOL (version 4.0): a Fortran package for nonlinear programming. Stanford: Stanford Univ., 1986.
52. Culligan K., Valenti M., Kuwata Y., How J.P. Three-dimensional flight experiments using on-line mixed-integer linear programming trajectory optimization. *Amer. control conf.: ACC'2007* (New York, NY, USA, July 9-13, 2007): Proc. N.Y.: IEEE, 2007. Pp. 5322–5327. DOI: [10.1109/ACC.2007.4283101](https://doi.org/10.1109/ACC.2007.4283101)

53. Schouwenaars T., De Moor B., Feron E., How J. Mixed integer programming for multi-vehicle path planning. *Eur. control conf.: ECC 2001* (Porto, Portugal, Sept. 4-7, 2001): Proc. N.Y.: IEEE, 2001. Pp. 2603–2608.
54. Earl M.G., D’Andrea R. Iterative MILP methods for vehicle-control problems. *IEEE Trans. on Robotics*, 2005, vol. 21, no. 6, pp. 1158–1167. DOI: [10.1109/TRO.2005.853499](https://doi.org/10.1109/TRO.2005.853499)
55. Kuwata Y. Real-time trajectory design for unmanned aerial vehicles using receding horizon control: Doct. diss. Camb., MA: Massachusetts Inst. of Technology, 2003. 151 p.
56. Habibi G., Masehian E., Beheshti M.T.H. Binary integer programming model of point robot path planning. *33<sup>rd</sup> annual conf. of the IEEE Industrial Electronics Soc.: IECON 2007* (Taipei, Taiwan, Nov. 5-8, 2007): Proc. N.Y.: IEEE, 2007. Pp. 2841–2845. DOI: [10.1109/IECON.2007.4460315](https://doi.org/10.1109/IECON.2007.4460315)
57. Masehian E., Habibi G. Robot path planning in 3D space using binary integer programming. *Intern. J. of Computer, Information, Systems and Control Engineering*, 2007, vol. 1, no. 5, pp. 1240-1245.
58. Masehian E., Habibi G. Motion planning and control of mobile robot using Linear Matrix Inequalities (LMIs). *IEEE/RSJ intern. conf. on intelligent robots and systems: IROS 2007* (San Diego, CA, USA, Oct. 29 - Nov. 2, 2007): Proc. N.Y. IEEE, 2007. Pp. 4277–4282. DOI: [10.1109/IROS.2007.4399641](https://doi.org/10.1109/IROS.2007.4399641)
59. Dorigo M., Birattari M., Stutzle T. Ant colony optimization. *IEEE Computational Intelligence Magazine*, 2006, vol. 1, no. 4, pp. 28–39. DOI: [10.1109/MCI.2006.329691](https://doi.org/10.1109/MCI.2006.329691)
60. Mohamad M.M., Dunnigan M.W., Taylor N.K. Ant colony robot motion planning. *Intern. conf. on “Computer as a tool”: EUROCON 2005* (Belgrade, Serbia, Nov. 21-24, 2005): Proc. N.Y.: IEEE, 2005. Vol. 1. Pp. 213–216. DOI: [10.1109/EURCON.2005.1629898](https://doi.org/10.1109/EURCON.2005.1629898)
61. Gen Ke Ke, Tan Ligo, Chulin N.A., Khe Yun. Route planning for quadcopter in an unknown environment based on monocular computer vision. *Avtomatizatsiia. Sovremennye tekhnologii* [Automation. Modern Technology], 2015, no. 12, pp. 14-19 (in Russian).
62. McCulloch W.S., Pitts W. Logicheskoe ischislenie idej, otnosiashchikhsia k nervnoj aktivnosti [A logical calculus of the ideas immanent in nervous activity]. *Avtomaty* [Automata studies] / ed. by C.E. Shannon, J. McCarthy. Moscow: Foreign Literature Publ., 1956. Pp. 363-384 (in Russian).
63. Glasius R., Komoda A., Stan C.A.M. Gielen. Neural network dynamics for path planning and obstacle avoidance. *Neural Networks*, 1995, vol. 8, no. 1, pp. 125–133. DOI: [10.1016/0893-6080\(94\)E0045-M](https://doi.org/10.1016/0893-6080(94)E0045-M)
64. Moreno J.A., Castro M. Heuristic algorithm for robot path planning based on a growing elastic net. *Progress in artificial intelligence: 12<sup>th</sup> Portuguese conf. on artificial intelligence: EPIA 2005* (Covilhã, Portugal, December 5-8, 2005): Proc. B.: Springer, 2005. Pp. 447-454. DOI: [10.1007/11595014\\_44](https://doi.org/10.1007/11595014_44)
65. Fu X., Gao X., Chen D. A Bayesian optimization algorithm for UAV path planning. *Intelligent information processing II: Intern. conf. on intelligent information processing: IIP*

- 2004 (Beijing, China, Oct. 21-23, 2004): Proc. Boston: Springer, 2005. Pp. 227–232.  
DOI: [10.1007/0-387-23152-8\\_29](https://doi.org/10.1007/0-387-23152-8_29)
66. Eberhart R., Kennedy J. A new optimizer using particle swarm theory. *6<sup>th</sup> intern. symp. on micromachine and human science: MHS'95* (Nagoya, Japan, Oct. 4-6, 1995): Proc. N.Y.: IEEE, 1995. Pp. 39–43. DOI: [10.1109/MHS.1995.494215](https://doi.org/10.1109/MHS.1995.494215)
67. Jung L.F., Knutzon J.S., Oliver J.H., Winer E.H. Three-dimensional path planning of unmanned aerial vehicles using particle swarm optimization. *11<sup>th</sup> AIAA/ISSMO multidisciplinary analysis and optimization conf.* (Portsmouth, Virginia, USA, September 6–8, 2006): Proc. Wash.: AIAA, 2006. Pp. 992–1001. DOI: [10.2514/6.2006-6995](https://doi.org/10.2514/6.2006-6995)
68. Huq R., Mann G.K.I., Gosine R.G. Mobile robot navigation using motor schema and fuzzy context dependent behavior modulation. *Applied Soft Computing*, 2008, vol. 8, no. 1, pp. 422–436. DOI: [10.1016/j.asoc.2007.02.006](https://doi.org/10.1016/j.asoc.2007.02.006)
69. Egerstedt M. Behavior based robotics using hybrid automata. *Hybrid systems: Computation and control: 3<sup>rd</sup> intern. workshop on hybrid systems: HSCC 2000* (Pittsburgh, PA, USA, March 23-25, 2000): Proc. B.: Springer, 2000. Pp. 103–116. DOI: [10.1007/3-540-46430-1\\_12](https://doi.org/10.1007/3-540-46430-1_12)
70. Ma J.-C., Zhang Q., Ma L.-Y., Xie W. Multi-behavior fusion-based path planning for mobile robot. *Beijing Ligong Daxue Xuebao / Trans. of Beijing Inst. of Technology*, 2014, vol. 34, no. 6, pp. 576–581.
71. Motlagh O.R.E., Hong T.S., Ismail N. Development of a new minimum avoidance system for a behavior-based mobile robot. *Fuzzy Sets and Systems*, 2009, vol. 160, no. 13, pp. 1929–1946. DOI: [10.1016/j.fss.2008.09.015](https://doi.org/10.1016/j.fss.2008.09.015)
72. Bekasov D.E. The use of fuzzy logic in solving the problem of path finding in an unknown environment. *Molodezhnyj nauchno-tehnicheskij vestnik MGTU im. N.E. Baumana* [Youth Scientific and Technical Bulletin of the Bauman MSTU], 2012, no. 5, p. 40 (in Russian).
73. Keke G., Wei L., Ligu T. A fuzzy controller: Using monocular computer vision to see and avoid obstacle for quadcopter. *5<sup>th</sup> intern. workshop on computer science and engineering: Information processing and control engineering: WCSE 2015-IPSE* (Moscow, Russia, April 15-17, 2015): Proc. Chenghu: Science and Engineering Inst., 2015.
74. Ng J., Braunl T. Performance comparison of bug navigation algorithms. *J. of Intelligent and Robotic Systems*, 2007, vol. 50, no. 1, pp. 73–84. DOI: [10.1007/s10846-007-9157-6](https://doi.org/10.1007/s10846-007-9157-6)
75. Lumelsky V., Stepanov A. Dynamic path planning for a mobile automaton with limited information on the environment. *IEEE Trans. on Automatic Control*, 1986, vol. 31, no. 11, pp. 1058–1063. DOI: [10.1109/TAC.1986.1104175](https://doi.org/10.1109/TAC.1986.1104175)
76. Yufka A., Parlaktuna O. Performance comparison of the BUG's algorithms for mobile robots. *Intern. symp. on INnovations in intelligent SYSTems and applications: INISTA 2009* (Trabzon, Turkey, June 29–July 1, 2009): Proc. N.Y.: IEEE, 2009. Pp. 416–421.
77. Shi C., Bu Y., Liu J. Mobile robot path planning in three-dimensional environment based on ACO-PSO hybrid algorithm. *IEEE/ASME intern. conf. on advanced intelligent*

- mechatronics: AIM 2008* (Xian, China, July 2-5, 2008): Proc. N.Y.: IEEE, 2008. Pp. 252–256. DOI: [10.1109/AIM.2008.4601668](https://doi.org/10.1109/AIM.2008.4601668)
78. Mettler B., Toupet O. Receding horizon trajectory planning with an environment-based cost-to-go function. *44<sup>th</sup> IEEE conf. on decision and control and the European control conf.: CDC-ECC'05* (Seville, Spain, Dec. 15, 2005): Proc. N.Y.: IEEE, 2005. Pp. 4071–4076. DOI: [10.1109/CDC.2005.1582799](https://doi.org/10.1109/CDC.2005.1582799)
  79. Gilimyanov R.F., Pesterev A.V., Rapoport L.B. Smoothing curvature of trajectories constructed by noisy measurements in path planning problems for wheeled robots. *J. of Computer and Systems Sciences International*, 2008, vol. 47, no. 5, pp. 812–819. DOI: [10.1134/S1064230708050158](https://doi.org/10.1134/S1064230708050158)
  80. Lutterkort D., Peters J. Smooth paths in a polygonal channel. *15<sup>th</sup> annual symp. on computational geometry: SCG'99* (Miami Beach, FLA, USA, June 13-16, 1999): Proc. N.Y.: ACM Press, 1999. Pp. 316–321. DOI: [10.1145/304893.304985](https://doi.org/10.1145/304893.304985)
  81. Jung D., Tsiotras P. On-line path generation for unmanned aerial vehicles using B-spline path templates. *J. of Guidance, Control, and Dynamics*, 2013, vol. 36, no. 6, pp. 1642–1653. DOI: [10.2514/1.60780](https://doi.org/10.2514/1.60780)
  82. Zhao Y., Tsiotras P. A quadratic programming approach to path smoothing. *Amer. control conf.: ACC 2011* (San Francisco, CA, USA, June 29 – July 1, 2011): Proc. N.Y.: IEEE, 2011. Pp. 5324–5329. DOI: [10.1109/ACC.2011.5990880](https://doi.org/10.1109/ACC.2011.5990880)
  83. Gilimyanov R.F., Rapoport L.B. Path deformation method in robot motion planning problems in the presence of obstacles. *Problemy upravleniia* [Control Sciences], 2012, no. 1, pp. 70-76 (in Russian).
  84. Dubins L.E. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Amer. J. of Mathematics*, 1957, vol. 79, no. 3, pp. 497–516. DOI: [10.2307/2372560](https://doi.org/10.2307/2372560)
  85. Nelson W. Continuous-curvature paths for autonomous vehicles. *IEEE intern. conf. on robotics and automation* (Scottsdale, AZ, USA, May 14-19, 1989): Proc. N.Y.: IEEE, 1989. Vol. 3. Pp. 1260–1264. DOI: [10.1109/ROBOT.1989.100153](https://doi.org/10.1109/ROBOT.1989.100153)
  86. Van der Molen G.M. Trajectory generation for mobile robots with clothoids. *Robotic systems: Advanced techniques and applications*. Dordrecht: Springer, 1992. Pp. 399–406. DOI: [10.1007/978-94-011-2526-0\\_46](https://doi.org/10.1007/978-94-011-2526-0_46)
  87. Walton D.J., Meek D.S., Ali J.M. Planar G2 transition curves composed of cubic Bézier spiral segments. *J. of Computational and Applied Mathematics*, 2003, vol. 157, no. 2, pp. 453–476. DOI: [10.1016/S0377-0427\(03\)00435-7](https://doi.org/10.1016/S0377-0427(03)00435-7)
  88. Komoriya K., Tanie K. Trajectory design and control of a wheel-type mobile robot using B-spline curve. *IEEE/RSJ intern. workshop on intelligent robots and systems: IROS'89* (Tsukuba, Japan, Sept. 4-6, 1989): Proc. N.Y.: IEEE, 1989. Pp. 398–405. DOI: [10.1109/IROS.1989.637937](https://doi.org/10.1109/IROS.1989.637937)

89. Berglund T., Jonsson H., Soderkvist I. An obstacle-avoiding minimum variation B-spline problem. *Intern. conf. on geometric modeling and graphics* (London, UK, July 16–18, 2003): Proc. N.Y.: IEEE, 2003. Pp. 156–161. DOI: [10.1109/GMAG.2003.1219681](https://doi.org/10.1109/GMAG.2003.1219681)
90. Wang Z., Zhang W., Li G., Mu X. G2 path smoothing using non-uniform B-spline. *Systems Engineering and Electronics*, 2011, no. 7, pp. 1539–1543. Available at: [http://en.cnki.com.cn/Article\\_en/CJFDTOTAL-XTYD201107021.htm](http://en.cnki.com.cn/Article_en/CJFDTOTAL-XTYD201107021.htm), accessed 05.12.2017.