

М.А. АГРАНОВСКИЙ, К.С. ЯКОВЛЕВ
**ЭФФЕКТИВНЫЙ МЕТОД ГЕНЕРАЦИИ
КИНЕМАТИЧЕСКИ-СОГЛАСОВАННЫХ ПРИМИТИВОВ
ДВИЖЕНИЯ НА ОСНОВЕ ПАРАМЕТРИЗАЦИИ ПО КРИВИЗНЕ**

Аграновский М.А., Яковлев К.С. Эффективный метод генерации кинематически-согласованных примитивов движения на основе параметризации по кривизне.

Аннотация. Планирование движения для мобильных роботов и беспилотных автомобилей является одной из ключевых задач современной робототехники. Основная её сложность заключается в необходимости учёта кинематических ограничений, таких как гладкость траектории и ограничения на минимальный радиус поворота. Одним из наиболее часто используемых подходов к решению этой задачи является построение траектории из набора заранее сгенерированных, кинематически-согласованных фрагментов, называемых примитивами движения. Робастность и скорость генерации этих примитивов напрямую определяют эффективность всего процесса планирования. В данной работе предлагается новый метод генерации примитивов движения для систем с кинематикой велосипедного типа. Метод основан на представлении траектории в виде кривой с полиномиальной функцией кривизны, а задача генерации примитива, соединяющего два заданных состояния (положение, ориентация, кривизна), сводится к решению системы нелинейных уравнений методом Ньютона. В отличие от существующих подходов, использующих коэффициенты полинома в качестве параметров, применяется новая репараметризация, основанная на значениях кривизны в ключевых точках траектории. Такая параметризация позволяет улучшить сходимость и устойчивость численного метода поиска решения, что демонстрируют проведённые эксперименты.

Ключевые слова: планирование траекторий, решётка состояний, примитивы движения, кинематические ограничения, беспилотный автомобиль, велосипедная модель, метод Ньютона.

1. Введение. Задача планирования движения является одной из фундаментальных в современной робототехнике и имеет ключевое значение для таких приложений, как беспилотные автомобили, промышленные манипуляторы и автономные летательные аппараты. Для большинства мобильных агентов эта задача усложняется наличием нетривиальных неголономных ограничений на движение. Эти кинематические ограничения, обусловленные конструкцией агента, не позволяют ему совершать произвольные перемещения. Например, автомобиль не может мгновенно сместиться вбок или поворачивать на месте, а траектория его движения должна обладать достаточной гладкостью, чтобы быть физически реализуемой.

Игнорирование таких ограничений на начальных этапах планирования приводит к построению геометрически кратчайших, но практически невыполнимых путей. Поэтому современные алгоритмы стремятся интегрировать кинематические ограничения непосредственно

в процесс поиска траектории. Одним из наиболее успешных и широко применяемых подходов для этого является планирование на решетке состояний (англ. *state-lattice planning*) [1–3]. Основная идея подхода заключается в дискретизации пространства состояний агента (координаты, ориентация, кривизна и т.д.) и построении графа, узлы которого соответствуют дискретным состояниям. Ребра этого графа представляют собой заранее сгенерированные короткие, кинематически-согласованные фрагменты траекторий – примитивы движения. Итоговый путь от начального до целевого состояния возможно найти с помощью графовых алгоритмов, таких как A* [4] или его производных [5]. Данный по построению путь является выполнимым для робота (так как примитивы движения стыкуются в дискретных состояниях, непрерывно продолжая друг друга). Пример построенной таким образом траектории представлен на рисунке 1.

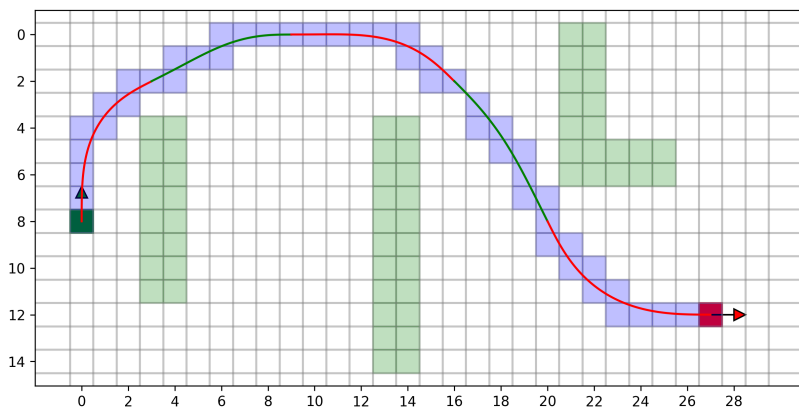


Рис. 1. Пример итоговой кинематически-согласованной траектории, собранной из примитивов движения (фрагменты чередующегося цвета). Также голубым выделены клетки пространства, через которые траектория проходит

Помимо классического поиска на решётке состояний, примитивы движения находят применение и в других парадигмах планирования. Например, они используются для расширения дерева в семплирующих методах (RRT, RRT*), что позволяет направить поиск в сторону динамически выполнимых траекторий.

Таким образом, разработка эффективных и быстрых методов генерации примитивов движения является базовой задачей, от которой зависит производительность целого класса передовых алгоритмов планирования. Центральным в генерации примитивов является решение двухточечной краевой задачи: для заданных начального и конечного состояний (включая координаты, ориентацию и кривизну) необходимо найти гладкую траекторию, соединяющую их.

В данной работе мы предлагаем метод генерации примитивов движения, который развивает идеи, представленные в [6]. Ключевой особенностью нашего подхода является введение новой параметризации траектории, которая, в отличие от стандартной полиномиальной, обладает более подходящими для нашей задачи свойствами. Как будет показано далее, это позволяет значительно улучшить устойчивость и скорость сходимости численного метода, используемого для генерации примитива.

2. Обзор литературы. Задачи поиска пути и планирования траекторий являются фундаментальными для многих областей искусственного интеллекта и робототехники. В основе многих классических методов лежит алгоритм A^* [4], обеспечивающий оптимальный поиск на графах. В простейшем случае, при работе на графах регулярной декомпозиции, таких как 8-связная решетка, были разработаны высокоэффективные специализированные алгоритмы. Ярким примером является JPS (Jump Point Search), который значительно ускоряет поиск на сетках с равномерной стоимостью за счет «прыжков» через узлы, не являющиеся точками принятия решения [7]. Этот подход был впоследствии расширен для работы на взвешенных картах [8] и адаптирован для трехмерных пространств [9].

Однако пути, ограниченные ребрами сетки, часто не являются оптимальными по длине в евклидовом пространстве. Это привело к разработке методов *any-angle* планирования, которые ищут пути, не привязанные строго к сетке. Ключевым алгоритмом в этом направлении является Theta*, который, используя структуру сетки, строит траектории с линией прямой видимости между несмежными узлами [10].

Помимо оптимизации длины пути, многие практические задачи требуют учета дополнительных факторов, что привело к развитию планирования с учетом ограничений на ресурсы. Это могут быть ограничения по времени, энергии, стоимости или другим параметрам. В этой области были предложены как быстрые точные алгоритмы для задачи поиска кратчайшего пути с ограничениями на ресурсы (RCSPP) [11], так и эффективные двунаправленные методы поиска [12].

Отдельное внимание в робототехнике уделяется планированию для роботов-манипуляторов, работающих в многомерных конфигурационных пространствах. Здесь также применяются подходы, основанные на поиске, в том числе с использованием примитивов движения для описания сложных действий [13], а также методы на основе семплирования, использующие целенаправленную выборку для повышения эффективности [14].

Еще одним важным и сложным направлением является планирование в динамических средах, содержащих движущиеся препятствия. Для решения этой задачи требуется планирование в пространстве-времени. Основополагающим методом здесь стал SIPP (Safe Interval Path Planning), который вычисляет безопасные интервалы времени для нахождения в каждом узле графа [15]. Последние исследования в этой области фокусируются на повышении эффективности поиска. Например, в рамках ограниченно-субоптимального SIPP [16, 17], а также на решении многокритериальных задач в динамических окружениях [18].

Этот краткий обзор иллюстрирует огромное разнообразие подходов к планированию, каждый из которых адаптирован для своего класса задач. Среди этого многообразия особое место занимает задача планирования движения с учётом кинематических ограничений. Её различные решения подробно рассмотрены в обзорной работе [19]. Существующие подходы можно условно разделить на две большие категории: методы на основе семплирования и методы, использующие дискретные решётки состояний.

К первой категории относятся алгоритмы семейства RRT [20–22], которые строят вероятностные деревья поиска в пространстве состояний. Для учёта кинематических ограничений эти методы всё чаще интегрируют примитивы движения для построения динамически выполнимых ветвей дерева, что позволяет генерировать траектории без постобработки [23]. Ко второй категории относятся методы поиска по заранее построенной решётке состояний, где узлы соединены примитивами, образуя граф для поиска оптимального пути. Основополагающие принципы этого подхода детально изложены в работах [1–3]. Качество и структура такого графа напрямую зависят от используемого набора примитивов, что подтверждается исследованием [24], где анализируется влияние их разнообразия и плотности на итоговое решение. Данный подход успешно применяется для широкого круга задач: от планирования движения квадрокоптеров [25] до навигации подводных аппаратов [26].

В качестве развития идеи решёток состояний, направленной на преодоление проблемы высокой степени ветвления, данная работа [27] предлагает новое пространство поиска. В методе MeshA* реализован

гибридный подход: поиск ведётся непосредственно по ячейкам сетки с одновременным вписыванием в них допустимых последовательностей примитивов движения. Такое решение позволяет сохранить теоретические гарантии полноты и оптимальности, обеспечивая при этом значительное ускорение поиска.

Способы генерации самих примитивов движения также разнообразны. Они могут быть основаны на различных параметрических кривых, например, сплайнах [28–30] или получены путём решения задачи оптимального управления, как это сделано в [31]. Для систем с кинематикой автомобильного типа особенно эффективен подход, основанный на представлении траектории через полиномиальную функцию кривизны. Ключевой работой в этом направлении, которая легла в основу нашего исследования, является [6], где для генерации примитива ищутся коэффициенты кубического полинома кривизны.

В последние годы активно развиваются методы генерации примитивов на основе машинного обучения, направленные на создание более адаптивных наборов [32]. Для высокодинамичных задач, таких как автономные гонки, применяются динамические примитивы движения, генерируемые в реальном времени [33]. Современные генеративные подходы, такие как диффузионные модели, позволяют создавать примитивы «на лету», адаптированные к каждой конкретной задаче [34]. Идея адаптивности развивается и в работах, где примитивы могут динамически изменять свою форму, подстраиваясь под геометрию окружения [35]. Методы обучения также выходят на более высокий уровень абстракции, где предлагается использовать переиспользуемые примитивы как «строительные блоки» для освоения роботом новых сложных навыков [36].

Несмотря на успехи методов обучения, повышение эффективности и робастности классических численных подходов, как в [6], остаётся актуальной задачей, ведь они используются в графовых алгоритмах, имеющих теоретические гарантии, выгодно отличающие их от всех остальных.

3. Постановка задачи. В данном разделе мы формализуем задачу генерации примитива движения. Для этого сперва определим модель движения агента и его пространство состояний, а затем введём ряд физически обоснованных допущений, позволяющих упростить задачу.

3.1. Кинематическая модель и параметризация по длине дуги. В качестве модели движения мобильного агента рассматривается стандартная кинематическая модель велосипедного типа, представленная

на рисунке 2¹. Данная модель широко используется в литературе [2, 6], поскольку она, с одной стороны, адекватно описывает ключевые нетривиальные ограничения легкового автомобиля (например, минимальный радиус поворота и невозможность бокового смещения), а с другой стороны – достаточно проста для аналитических преобразований.

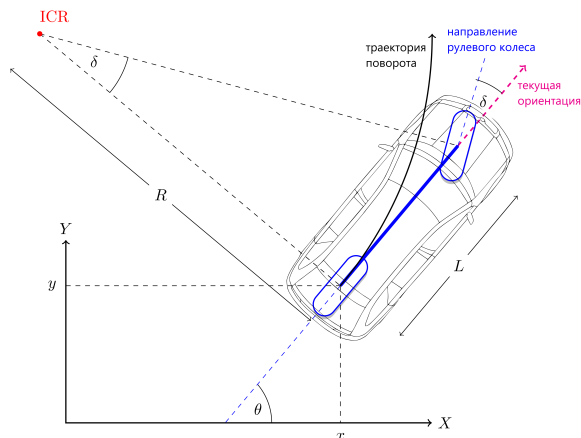


Рис. 2. Велосипедная кинематическая модель. Точка (x, y) соответствует центру задней оси в глобальной системе координат

Полное состояние (конфигурация) агента в момент времени t описывается вектором $X(t) = (x(t), y(t), \theta(t), k(t))$, где:

- (x, y) – координаты центра задней оси агента;
- θ – угол ориентации (направления) агента относительно оси Ox ;
- k – кривизна траектории в точке (x, y) . Она связана с углом поворота рулевого колеса δ и колёсной базой L соотношением $k = \frac{\tan(\delta)}{L}$.

Включение кривизны k в вектор состояния является принципиальным моментом. Оно позволяет формировать траектории непрерывные по кривизне (C^1 -гладкость пути), что необходимо для плавного сочленения примитивов движения между собой. Это, в свою очередь, обеспечивает физическую реализуемость манёвров, так как скачки кривизны соответствуют мгновенным поворотам рулевого колеса.

¹Использовано изображение из образовательного ресурса <https://thomasfermi.github.io/Algorithms-for-Automated-Driving/Control/BicycleModel.html>.

Уравнения, описывающие изменение состояния во времени, зависят от линейной скорости $v(t)$:

$$\begin{cases} \dot{x}(t) = v(t) \cos \theta(t) \\ \dot{y}(t) = v(t) \sin \theta(t) \\ \dot{\theta}(t) = v(t)k(t) \end{cases}, \quad (1)$$

где точка над переменной обозначает производную по времени t .

При генерации примитивов движения зачастую используется допущение о постоянной линейной скорости v вдоль одного фрагмента траектории. Важно отметить, что это упрощение применяется лишь локально, поскольку итоговый путь агента собирается из множества последовательных коротких фрагментов, произвольный профиль скорости на всей траектории может быть эффективно реализован посредством его кусочно-постоянной аппроксимации. Такой подход позволяет отделить геометрическую задачу построения формы траектории от планирования скоростного режима движения по ней. При этом удобнее выполнить репараметризацию траектории, перейдя от времени t к длине пройденной дуги $s = v \cdot t$. Производная по новой переменной s связана с производной по времени как $\frac{d}{ds} = \frac{dt}{ds} \frac{d}{dt} = \frac{1}{v} \frac{d}{dt}$. Применяв это правило к системе (1), получим уравнения, описывающие геометрию пути независимо от скорости:

$$\begin{cases} x'(s) = \cos \theta(s) \\ y'(s) = \sin \theta(s) \\ \theta'(s) = k(s) \end{cases}. \quad (2)$$

Здесь и далее штрих (') обозначает производную по длине дуги s .

3.2. Класс рассматриваемых траекторий и формальная постановка задачи. Система (2) не является замкнутой, поскольку в ней не определена функция кривизны $k(s)$. Для получения гладких и аналитически представимых траекторий мы, следуя подходу из [6], будем рассматривать класс кривых, у которых кривизна $k(s)$ является полиномом третьей степени по длине дуги s :

$$k(s) = a_3 s^3 + a_2 s^2 + a_1 s + a_0. \quad (3)$$

Выбор кубического полинома обеспечивает непрерывность производной кривизны, что соответствует плавному изменению угловой скорости рулевого колеса, и является хорошим компромиссом между гибкостью формы траектории и количеством определяющих её параметров.

Таким образом, **траекторией-примитивом** мы будем называть гладкую кривую $X(s) = (x(s), y(s), \theta(s), k(s))$, определенную на отрезке $[0, l]$, где функция кривизны $k(s)$ является полиномом вида (3), а остальные компоненты удовлетворяют системе уравнений (2).

Теперь задача генерации примитива может быть сформулирована следующим образом в виде *двухточечной краевой задачи*:

Дано:

1. Начальное состояние агента $X_0 = (x_0, y_0, \theta_0, k_0)$;
2. Конечное (целевое) состояние агента $X_f = (x_f, y_f, \theta_f, k_f)$;

Найти:

1. Параметры полинома кривизны $\{a_1, a_2, a_3\}$;
2. Длину примитива $l > 0$;

такие, что траектория-примитив $X(s)$, выходящая из начального состояния $X(0) = X_0$, удовлетворяет конечному условию $X(l) = X_f$.

Отметим, что коэффициент a_0 в (3) однозначно определяется из начального условия: $a_0 = k(0) = k_0$.

4. Метод генерации примитива движения. В данном разделе представлен предлагаемый метод решения поставленной двухточечной краевой задачи. Сначала мы выведем явные уравнения для траектории-примитива, а затем введем ключевую для нашего подхода репараметризацию, которая улучшает численные свойства алгоритма генерации.

4.1. Интегральная форма уравнений траектории.

Как было определено в предыдущем разделе, мы ищем траекторию, параметризованную длиной дуги s , с полиномиальной функцией кривизны. Для полноты изложения приведем здесь вывод явных уравнений траектории.

Пусть начальное состояние задано как $X_0 = (x_0, y_0, \theta_0, k_0)$. Функция кривизны $k(s)$ является полиномом третьей степени (3). Из начального условия $k(0) = k_0$ следует, что $a_0 = k_0$. Таким образом, траектория полностью определяется своей геометрией (тремя коэффициентами $\{a_1, a_2, a_3\}$ кривизны) и её общей длиной l .

Угол ориентации $\theta(s)$ и координаты $(x(s), y(s))$ получаются из системы (2) простым интегрированием с учетом начального условия $X(0) = (x_0, y_0, \theta_0, k_0)$. Таким образом, итоговые функции, описывающие траекторию, следующие:

$$\begin{cases} k(s) = a_3 s^3 + a_2 s^2 + a_1 s + a_0 \\ \theta(s) = \theta_0 + a_0 s + a_1 \frac{s^2}{2} + a_2 \frac{s^3}{3} + a_3 \frac{s^4}{4} \\ x(s) = x_0 + \int_0^s \cos \theta(z) dz \\ y(s) = y_0 + \int_0^s \sin \theta(z) dz \end{cases} \quad (4)$$

Интегралы в (4) не могут быть вычислены аналитически в замкнутой форме, поэтому их значения находят с помощью численных методов, например, квадратурных формул. Таким образом, для заданных параметров $\{a_1, a_2, a_3\}$ и длины l , конечное состояние траектории $X(l) = (x(l), y(l), \theta(l), k(l))$ может быть вычислено. Задача генерации траектории – обратная: она сводится к подбору этих четырёх параметров так, чтобы $X(l)$ совпало с целевым состоянием X_f .

4.2. Репараметризация по значениям кривизны. Поиск решения в пространстве коэффициентов полинома $\{a_1, a_2, a_3, l\}$ является стандартным подходом [6]. Однако эти параметры имеют низкую интуитивную интерпретацию и сильно взаимосвязаны, что может приводить к плохой обусловленности задачи оптимизации и, как следствие, к медленной сходимости или неустойчивости численных методов.

Мы предлагаем использовать альтернативную, более физически осмысленную параметризацию. Вместо коэффициентов полинома $\{a_1, a_2, a_3\}$ будем использовать значения кривизны в трёх ключевых точках вдоль траектории, а вместо длины l – её логарифм.

Пусть конечная длина примитива равна l . Введем три узловые точки на отрезке $[0, l]$: $s_1 = l/3$, $s_2 = 2l/3$, $s_3 = l$. Определим новые параметры следующим образом:

$$\begin{cases} k_1 := k(s_1) = k(l/3) \\ k_2 := k(s_2) = k(2l/3) \\ k_f := k(s_3) = k(l) \\ \ell := \log(l) \end{cases} \quad (5)$$

Использование логарифма длины ℓ преобразует ограниченную переменную $l > 0$ в неограниченную $\ell \in \mathbb{R}$, что часто бывает полезно для оптимизационных алгоритмов. Значение k_f – это кривизна в конечном состоянии, которая задана в постановке задачи. Таким образом, для поиска остаются только три параметра: $\{k_1, k_2, \ell\}$.

Покажем, что данная параметризация эквивалентна исходной. Переход от $\{a_1, a_2, a_3, l\}$ к $\{k_1, k_2, k_f, \ell\}$ тривиален: он заключается в вычислении значений полинома и логарифма. Для обратного перехода от $\{k_1, k_2, k_f, \ell\}$ к $\{a_1, a_2, a_3, l\}$ сперва находим $l = e^\ell$. Затем, зная значения полинома $k(s)$ в четырёх точках ($s = 0, s_1, s_2, s_3$), мы можем составить систему линейных уравнений относительно его коэффициентов $\{a_0, a_1, a_2, a_3\}$:

$$\begin{pmatrix} k_0 \\ k_1 \\ k_2 \\ k_f \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & s_1 & s_1^2 & s_1^3 \\ 1 & s_2 & s_2^2 & s_2^3 \\ 1 & s_3 & s_3^2 & s_3^3 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}. \quad (6)$$

Матрица в правой части является матрицей Вандермонда, следовательно, она всегда невырождена (поскольку $l > 0$). Это гарантирует существование и единственность решения. Выразив вектор коэффициентов через обратную матрицу, получаем:

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & l/3 & (l/3)^2 & (l/3)^3 \\ 1 & 2l/3 & (2l/3)^2 & (2l/3)^3 \\ 1 & l & l^2 & l^3 \end{pmatrix}^{-1} \begin{pmatrix} k_0 \\ k_1 \\ k_2 \\ k_f \end{pmatrix}. \quad (7)$$

Таким образом, мы показали, что предложенная параметризация $\{k_1, k_2, \ell\}$ полностью и однозначно определяет геометрию примитива.

Главное преимущество такого подхода заключается в том, что параметры k_1, k_2 имеют ясный физический смысл и напрямую влияют на форму траектории в её срединной части, что, как мы ожидаем, должно улучшить нахождение численного решения. Более того, количество подбираемых параметров сократилось с 4 до 3.

4.3. Численное решение методом Ньютона. После введения новой параметризации задача свелась к поиску такого вектора параметров $p = (k_1, k_2, \ell) \in \mathbb{R}^3$, который порождает траекторию, соединяющую начальное состояние X_0 с целевым X_f . Напомним, что четвертый параметр, кривизна k_f , уже зафиксирована целевым состоянием.

Для решения этой задачи мы можем сформулировать систему нелинейных уравнений. Введем *функцию невязки* $F : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, которая для заданного вектора параметров p вычисляет разницу между конечным состоянием порождаемой траектории и целевым состоянием X_f . Обозначим через $\mathcal{X}(p)$ конечное состояние траектории длиной $l = e^\ell$, построенной с параметрами $p = (k_1, k_2, \ell)$. Тогда функция невязки имеет вид:

$$F(p) = F(k_1, k_2, \ell) = \begin{pmatrix} \mathcal{X}(p).x - x_f \\ \mathcal{X}(p).y - y_f \\ \mathcal{X}(p).\theta - \theta_f \end{pmatrix}. \quad (8)$$

Задача генерации примитива, таким образом, эквивалентна нахождению корня этой вектор-функции, то есть решению уравнения $F(p) = 0$.

Поскольку размерность пространства параметров (3) совпадает с размерностью пространства невязок (3), для решения этой системы мы можем применить *многомерный метод Ньютона*. Это итерационный процесс, на каждом шаге которого строится новое приближение p_{n+1} на основе текущего p_n по следующей формуле:

$$p_{n+1} = p_n - \alpha \cdot [J_F(p_n)]^{-1} F(p_n), \quad (9)$$

где $J_F(p_n)$ – матрица Якоби функции F , вычисленная в точке p_n , а $\alpha \in (0, 1]$ – коэффициент скорости обучения (*learning rate*), который используется для улучшения стабильности сходимости.

Матрица Якоби J_F представляет собой матрицу частных производных компонент функции F по компонентам вектора параметров p :

$$J_F(p) = \begin{pmatrix} \frac{\partial F_x}{\partial k_1} & \frac{\partial F_x}{\partial k_2} & \frac{\partial F_x}{\partial \ell} \\ \frac{\partial F_y}{\partial k_1} & \frac{\partial F_y}{\partial k_2} & \frac{\partial F_y}{\partial \ell} \\ \frac{\partial F_\theta}{\partial k_1} & \frac{\partial F_\theta}{\partial k_2} & \frac{\partial F_\theta}{\partial \ell} \end{pmatrix}, \quad (10)$$

где $F = (F_x, F_y, F_\theta)^T$ для краткости. Аналитическое выражение для этих производных получить затруднительно из-за интегральной формы уравнений (4). Поэтому на практике элементы матрицы Якоби вычисляются численно в каждой точке p_n с помощью *метода конечных разностей*. Например, частная производная $\frac{\partial F_i}{\partial p_j}$ аппроксимируется как:

$$\frac{\partial F_i}{\partial p_j}(p_n) \approx \frac{F_i(p_n + \Delta p_j \cdot e_j) - F_i(p_n)}{\Delta p_j}, \quad (11)$$

где e_j – орт j -ой оси, а Δp_j – малое приращение (например, 10^{-6}).

Итерационный процесс продолжается до тех пор, пока норма невязки $\|F(p_n)\|$ не станет меньше заданной малой величины ϵ или пока не будет достигнуто максимальное число итераций.

Важным для сходимости метода Ньютона является выбор начального приближения $p_0 = (k_{1,0}, k_{2,0}, \ell_0)$. В качестве эвристики для p_0 можно использовать параметры траектории, которая соединяет X_0 и X_f с некоторыми упрощениями.

5. Эксперименты. Для экспериментальной проверки предложенного метода генерации примитивов и его сравнения с базовым подходом был проведён ряд вычислительных экспериментов².

5.1. Постановка экспериментов. Сравнение проводилось между двумя методами генерации примитивов, использующими многомерный метод Ньютона для поиска решения:

1. Базовый метод. Основан на прямой параметризации полинома кривизны через его коэффициенты $\{a_1, a_2, a_3\}$ и длину траектории l . В этом случае задача сводится к поиску четырех параметров, а функция невязки $F(p)$ является четырехмерной, включая в себя расхождение по конечной кривизне. Соответственно, на каждой итерации метода Ньютона вычисляется и обращается матрица Якоби размером 4×4 .

2. Предложенный метод. Использует предложенную в данной работе репараметризацию по значениям кривизны в узловых точках

²Исходный код предлагаемого алгоритма генерации примитивов движения доступен по ссылке: <https://github.com/PathPlanning/CubicPrimitivesGenerator>.

$\{k_1, k_2\}$ и логарифму длины $\ell = \log(l)$. Кривизна в конечной точке k_f определяется целевым состоянием и не является искомым параметром. Таким образом, задача сводится к поиску трех параметров, а функция невязки является трехмерной. На каждой итерации вычисляется и обращается матрица Якоби размером 3×3 .

Во всех экспериментах использовались единые настройки для метода Ньютона: начальные параметры $p_0 = (0, 0, 0)$ (что соответствует $a_1 = a_2 = a_3 = 0, l = 1$), максимальное число итераций $N_{max} = 100$, коэффициент скорости обучения $\alpha = 0.1$. Итерационный процесс считался успешно завершённым и прекращался досрочно, если евклидова норма вектора невязки становилась меньше порогового значения $\epsilon = 0.01$.

5.2. Эксперимент I: Сравнительный анализ производительности.

На первом этапе был проведен масштабный сравнительный тест для количественной оценки производительности и робастности обоих методов. Для этого было сгенерировано 2100 уникальных тестовых сценариев. В каждом сценарии требовалось найти траекторию от начального состояния $X_0 = (0, 0, 0, k_{start})$ до целевого состояния $X_f = (x_f, y_f, \theta_f, k_f)$. Параметры для генерации сценариев выбирались следующим образом:

- начальная кривизна k_{start} принимала значения из множества $\{0.0, 0.5, 1.0\}$;
- конечные координаты (x_f, y_f) соответствовали 20 точкам, равномерно распределённым на окружности единичного радиуса;
- конечная ориентация θ_f принимала 7 значений из диапазона $[-\pi, \pi]$ с шагом $\pi/3$ радиан (60°);
- конечная кривизна k_f принимала значения из множества $\{-1.0, -0.5, 0.0, 0.5, 1.0\}$.

Выбранный набор сценариев является репрезентативным благодаря инвариантности задачи относительно некоторых преобразований. Фиксация начального положения $(0, 0)$ и ориентации 0 эквивалентна рассмотрению всех возможных задач с точностью до параллельного переноса и поворота. Рассмотрение только неотрицательных начальных кривизн ($k_{start} \geq 0$) также является достаточным, поскольку любой случай с $k_{start} < 0$ сводится к случаю с $k_{start} > 0$ путем симметричного отражения всей задачи относительно оси абсцисс.

Для каждого сценария оба метода запускались для поиска соответствующей траектории. Основными измеряемыми метриками были: **успешность** (Success Rate) – доля случаев, в которых метод находил решение за отведенное число итераций, и **время выполнения** – среднее и медианное время, затраченное на один успешный поиск. Совокупность

данных показателей принята нами в качестве критерия эффективности алгоритмов: эффективным считается метод, обеспечивающий наибольшую успешность при минимальных временных затратах. Результаты эксперимента сведены в таблице 1.

Таблица 1. Сравнительные результаты производительности методов

Метрика	Базовый метод	Предложенный метод
Успешность (Success Rate)	64.76%	85.62%
Среднее время (сек.)	0.1482	0.1107
Медианное время (сек.)	0.1616	0.1061

Как видно из таблицы, предложенный метод демонстрирует значительно более высокую робастность, успешно находя решение в 85.62% случаев против 64.76% у базового подхода. Кроме того, новый метод в среднем работает быстрее. На рисунке 3 представлено попарное сравнение времени выполнения для тех сценариев, где оба метода нашли решение.

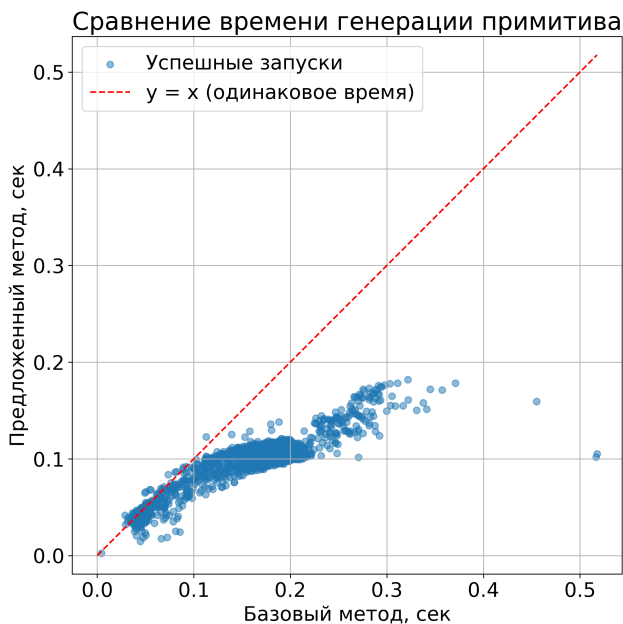


Рис. 3. Сравнение времени выполнения для успешных запусков. Точки ниже диагональной линии соответствуют случаям, где предложенный метод оказался быстрее базового

Большинство точек на графике лежит ниже диагональной линии $y = x$, что подтверждает преимущество предложенного метода в скорости на одинаковых задачах. Таким образом, первый эксперимент однозначно демонстрирует, что репараметризация по кривизне улучшает как сходимость, так и производительность численного метода.

5.3. Эксперимент 2: Анализ области достижимости. Второй эксперимент был направлен на визуальный анализ робастности и исследование «области достижимости» для каждого метода. Начальное состояние было зафиксировано в точке $X_0 = (0, 0, 0, 0)$. Целевые состояния располагались в центрах ячеек сетки 10×10 , покрывающей квадрат $[-1, 1]^2$. Для каждого центра ячейки генерировалось 12 целевых ориентаций с шагом 30 градусов (от 0° до 330°). Конечная кривизна во всех случаях была нулевой (это довольно распространённое упрощение, позволяющее не сильно потерять в выразительности итоговых траекторий, но серьёзно уменьшить размерность пространства поиска). Таким образом, было сформировано 1200 тестовых задач.

Результаты данного эксперимента представлены на рисунке 4 в виде карт достижимости. Каждая карта состоит из 100 «циферблатов», соответствующих ячейкам сетки. Каждый циферблат разделен на 12 секторов, где каждый сектор отвечает за одно из 12 целевых направлений. Зеленый цвет сектора (светлый) означает успешное нахождение траектории, а красный (темный) – неудачу. Центральная стрелка указывает начальное состояние робота.

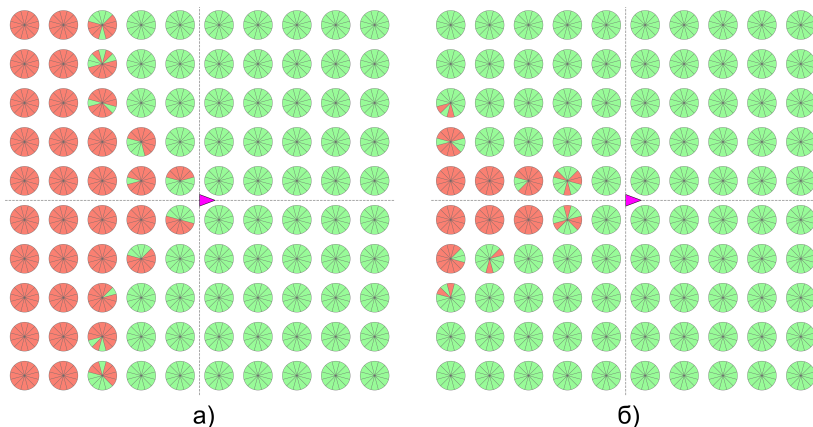


Рис. 4. Карты достижимости для: а) базового метода и б) предложенного метода.

Зеленый цвет сектора означает успех, красный – неудачу

Визуальный анализ на рисунке 4 наглядно подтверждает выводы, сделанные в первом эксперименте. На карте для базового метода (рисунок 4(а)) видны обширные «красные» области. Метод систематически не справляется с задачами, требующими сложных маневров: движения назад (отрицательные значения по оси Ox) или резких разворотов на месте. В то же время карта для предложенного метода (рисунок 4(б)) практически полностью «зеленая», что свидетельствует о его высокой робастности и способности генерировать траектории для подавляющего большинства требуемых конечных состояний. Это преимущество является критически важным для построения плотных и связных графов состояний в задачах планирования.

Стоит отметить, что карты достижимостей на рисунке 4 почти симметричны относительно оси Ox , что вполне логично, так как поставленная задача генерации примитива симметрична. Небольшие несовпадения проявляются из-за численной неустойчивости реализации.

5.4. Примеры. На рисунках 5, 6 представлено несколько иллюстраций примитивов движения, полученных предлагаемым алгоритмом (стрелка изображает целевое состояние). В частности, на рисунке 5(б) показан пример траектории в виде сложного манёвра, который не смог найти базовый подход, но который был построен предложенным.

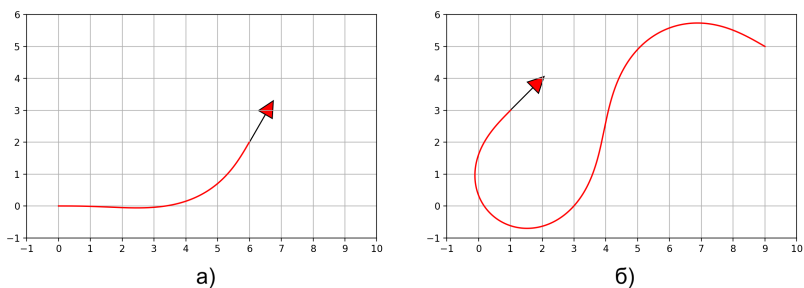


Рис. 5. Примеры траекторий:

- а) из $X_0 = (0, 0, 0, 0)$ в $X_f = (6, 2, \frac{\pi}{3}, 0)$;
 б) из $X_0 = (9, 5, \frac{5\pi}{6}, 0)$ в $X_f = (1, 3, -\frac{\pi}{4}, 0)$

Также стоит отметить интересную особенность генерации примитивов из рассматриваемого класса траекторий. Несмотря на то, что углы ориентации, отличающиеся на $2\pi k$ ($k \in \mathbb{Z}$), соответствуют одному и тому же направлению, в рамках рассматриваемой задачи

они не являются взаимозаменяемыми. Начальный θ_0 и конечный θ_f углы задают граничные условия для функции угла направления $\theta(s)$, которая должна непрерывно изменяться вдоль траектории $X(s)$ (при изменении s от 0 до l). Таким образом, выбор конкретных значений θ_0 и θ_f определяет полную угловую вариацию траектории. Это наглядно продемонстрировано на рисунке 6, где сравниваются две кривые, для которых граничные задачи различаются лишь значением конечного угла: а) $\theta_f = 11\pi/6$ и б) $\theta_f = -\pi/6$. Хотя оба угла задают идентичную конечную ориентацию, в первом случае траектория совершает полный оборот, а во втором – нет.

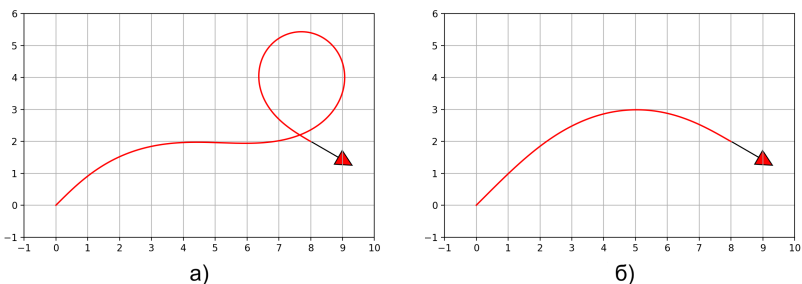


Рис. 6. Траектории для разных конечных углов:

- а) из $X_0 = (0, 0, \frac{\pi}{4}, 0)$ в $X_f = (8, 2, \frac{11\pi}{6}, 0)$;
- б) из $X_0 = (0, 0, \frac{\pi}{4}, 0)$ в $X_f = (8, 2, -\frac{\pi}{6}, 0)$

6. Заключение. В данной работе был представлен и исследован новый метод генерации кинематически-согласованных примитивов движения для систем с моделью велосипедного типа. Ключевым вкладом работы является применение репараметризации задачи поиска траектории: вместо подбора коэффициентов полинома кривизны используется оптимизация значений кривизны в узловых точках.

Проведенные вычислительные эксперименты подтвердили высокую эффективность предложенного подхода по сравнению с базовым методом. Количественный анализ на наборе из 2100 тестовых сценариев показал, что предложенная репараметризация позволяет повысить успешность нахождения решения (Success Rate) с 64.76% до 85.62%. При этом среднее время генерации одного примитива сократилось приблизительно на 25% (с 0.148 с до 0.110 с), что критически важно для использования метода в алгоритмах планирования реального времени.

Качественный анализ карт достижимости продемонстрировал, что предложенный метод значительно устойчивее при генерации траекторий, требующих существенного изменения ориентации в ограниченном пространстве, обеспечивая более плотное покрытие пространства состояний по сравнению с базовым подходом.

В качестве направлений для дальнейших исследований мы выделяем расширение сравнительной базы экспериментов, включив в анализ методы на основе сплайнов и оптимального управления. Также планируется более детально исследовать зависимость сходимости от числа итераций для разработки адаптивных стратегий выбора начального приближения (рестартов) в случаях, когда алгоритм не находит решение за отведенный бюджет вычислений.

Литература

1. Pivtoraiko M., Kelly A. Efficient constrained path planning via search in state lattices // International Symposium on Artificial Intelligence, Robotics, and Automation in Space. 2005. pp. 1–7.
2. Pivtoraiko M., Knepper R.A., Kelly A. Differentially constrained mobile robot motion planning in state lattices // Journal of Field Robotics. 2009. vol. 26. no. 3. pp. 308–333. DOI: 10.1002/rob.20285.
3. Pivtoraiko M., Kelly A. Kinodynamic motion planning with state lattice motion primitives // Proceedings of 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2011. pp. 2172–2179. DOI: 10.1109/IROS.2011.6094900.
4. Hart P., Nilsson N., Raphael B. A formal basis for the heuristic determination of minimum cost paths // IEEE Transactions on Systems Science and Cybernetics. 1968. vol. 4. no. 2. pp. 100–107. DOI: 10.1109/TSSC.1968.300136.
5. Ebdend R., Drechsler R. Weighted A* search – unifying view and application // Artificial Intelligence Journal. 2009. vol. 173. no. 14. pp. 1310–1342. DOI: 10.1016/j.artint.2009.06.004.
6. Nagy B., Kelly A. Trajectory generation for car-like robots using cubic curvature polynomials // Field and Service Robotics. 2001. vol. 11. pp. 479–490.
7. Harabor D., Grastien A. Online graph pruning for pathfinding on grid maps // Proceedings of the 25th AAAI Conference on Artificial Intelligence. 2011. vol. 25. no. 1. pp. 1114–1119. DOI: 10.1609/aaai.v25i1.7994.
8. Carlson M., Moghadam S.K., Harabor D.D., Stuckey P.J., Ebrahimi M. Optimal pathfinding on weighted grid maps // Proceedings of the 37th AAAI Conference on Artificial Intelligence. 2023. vol. 37. no. 10. pp. 12373–12380. DOI: 10.1609/aaai.v37i10.26458.
9. Nobes T.K., Harabor D., Wybrow M., Walsh S.D.C. The JPS pathfinding system in 3D // Proceedings of the 15th International Symposium on Combinatorial Search. 2022. vol. 15. no. 1. pp. 145–152. DOI: 10.1609/socs.v15i1.21762.
10. Daniel K., Nash A., Koenig S., Felner A. Theta*: Any-angle path planning on grids // Journal of Artificial Intelligence Research. 2010. vol. 39. pp. 533–579. DOI: 10.1613/jair.2994.
11. Ahmadi S., Tack G., Harabor D.D., Kilby P. A fast exact algorithm for the resource constrained shortest path problem // Proceedings of the 35th AAAI

- Conference on Artificial Intelligence. 2021. vol. 35. no. 14. pp. 12217–12224. DOI: 10.1609/aaai.v35i14.17450.
12. Ahmadi S., Tack G., Harabor D.D., Kilby P. Weight constrained path finding with bidirectional A* // Proceedings of the 15th International Symposium on Combinatorial Search. 2022. vol. 15. no. 1. pp. 2–10. DOI: 10.1609/socs.v15i1.21746.
 13. Cohen B.J., Chitta S., Likhachev M. Search-based planning for manipulation with motion primitives // 2010 IEEE International Conference on Robotics and Automation. 2010. pp. 2902–2908. DOI: 10.1109/ROBOT.2010.5509685.
 14. Kang G., Kim Y.B., Lee Y.H., Oh H.S., You W.S., Choi H.R. Sampling-based motion planning of manipulator with goal-oriented sampling // Intelligent Service Robotics. 2019. vol. 12. pp. 265–273. DOI: 10.1007/s11370-019-00281-y.
 15. Phillips M., Likhachev M. SIPP: Safe interval path planning for dynamic environments // 2011 IEEE International Conference on Robotics and Automation. 2011. pp. 5628–5635. DOI: 10.1109/ICRA.2011.5980306.
 16. Yakovlev K., Andreychuk A., Stern R. Revisiting bounded-suboptimal safe interval path planning // Proceedings of the 30th International Conference on Automated Planning and Scheduling. 2020. vol. 30. no. 1. pp. 300–304. DOI: 10.1609/icaps.v30i1.6674.
 17. Rybecky T., Kulich M., Andreychuk A., Yakovlev K. Towards narrowing the search in bounded-suboptimal safe interval path planning // Proceedings of the 12th International Symposium on Combinatorial Search, SoCS 2021. 2021. vol. 12. no. 1. pp. 136–140. DOI: 10.1609/socs.v12i1.18562.
 18. Ren Z., Rathinam S., Likhachev M., Choset H. Multi-objective safe-interval path planning with dynamic obstacles // IEEE Robotics and Automation Letters. 2022. vol. 7. no. 3. pp. 8154–8161. DOI: 10.1109/LRA.2022.3187270.
 19. Gonzalez D., Perez J., Milanes V., Nashashibi F. A review of motion planning techniques for automated vehicles // IEEE Transactions on Intelligent Transportation Systems. 2016. vol. 17. no. 4. pp. 1135–1145. DOI: 10.1109/TITS.2015.2498841.
 20. La Valle S.M., Kuffner Jr J.J. Randomized kinodynamic planning // The International Journal of Robotics Research. 2001. vol. 20. no. 5. pp. 378–400. DOI: 10.1177/02783640122067453.
 21. Karaman S., Frazzoli E. Incremental sampling-based algorithms for optimal motion planning // Robotics: Science and Systems. 2010. vol. 104. no. 2. pp. 267–274. DOI: 10.1177/0278364911406761.
 22. Otte M., Frazzoli E. RRTX: Real-time motion planning/replanning for environments with unpredictable obstacles // Algorithmic Foundations of Robotics XI. Springer Tracts in Advanced Robotics. 2015. vol. 107. pp. 461–478. DOI: 10.1007/978-3-319-16595-0_27.
 23. Paudel A. Motion primitives based path planning with rapidly-exploring random tree. arXiv preprint arXiv: 2210.15784. 2022.
 24. Головин В.А., Яковлев К.С. Прimitives движения робота в задаче планирования траектории с кинематическими ограничениями // Информатика и автоматизация. 2023. Т. 22. №6. С. 1354–1386. DOI: 10.15622/ia.22.6.4.
 25. Jingyu Y., Jianjun M. Search-based trajectory planning with motion primitives for quadrotors using pruning A* algorithm // 2022 37th Youth Academic Annual Conference of Chinese Association of Automation (YAC). 2022. pp. 996–1000. DOI: 10.1109/YAC57282.2022.10023615.
 26. Scharff J., Gonzalez D., Hernandez J.D., Pairet E., Petitot Y. Online 3-dimensional path planning with kinematic constraints in unknown environments using hybrid A* with tree pruning // Sensors. 2021. vol. 21. no. 4. 1152. p.

27. Agranovskiy M., Yakovlev K. MeshA*: Efficient path planning with motion primitives // Proceedings of the 40th AAAI Conference on Artificial Intelligence. 2026. vol. 40. no. 43. pp. 36785–36792. DOI: 10.1609/aaai.v40i43.41004.
28. Piazzì A., Bianco C.G.L., Bertozzi M., Fascioli A., Broggi A. Quintic G/sup 2/-splines for the iterative steering of vision-based autonomous vehicles // IEEE Transactions on Intelligent Transportation Systems. 2002. vol. 3. no. 1. pp. 27–36. DOI: 10.1109/6979.994793.
29. Walton D.J., Meek D.S. A controlled clothoid spline // Computers Graphics. 2005. vol. 29. no. 3. pp. 353–363. DOI: 10.1016/j.cag.2005.03.008.
30. Farouki R.T. Pythagorean-hodograph curves: algebra and geometry inseparable // Geometry and Computing. Springer. 2008. vol. 1. pp. 131–196. DOI: 10.1007/978-3-540-73398-0_8.
31. Яковлев К.С., Белинская Ю.С., Макаров Д.А., Андрейчук А.А. Безопасно-интервальное планирование и метод накрытий для управления движением мобильного робота в среде со статическими и динамическими препятствиями // Автоматика и телемеханика. 2022. №6. С. 96–117. DOI: 10.31857/S0005231022060083.
32. De Iaco R., Smith S.L., Czarnecki K. Learning a lattice planner control set for autonomous vehicles // IEEE Intelligent Vehicles Symposium (IV). 2019. pp. 549–556. DOI: 10.1109/IVS.2019.8813797.
33. Weaver C., Capobianco R., Wurman P.R., Stone P., Tomizuka M. Real-time trajectory generation via dynamic movement primitives for autonomous racing // 2024 American Control Conference (ACC). 2024. pp. 352–359. DOI: 10.23919/ACC60939.2024.10644244.
34. Franke J., Moldagalieva A., Hanfeld P., Honig W. Accelerating db-A* for kinodynamic motion planning using diffusion. arXiv preprint arXiv: 2503. 05539. 2025.
35. Kraljusic B., Ajanovic Z., Covic N., Lacevic B. Search-based robot motion planning with distance-based adaptive motion primitives // 2025 XXX International Conference on Information, Communication and Automation Technologies (ICAT). 2025. pp. 1–6. DOI: 10.1109/ICAT66432.2025.11189289.
36. Yao Y., Liu S., Song H., Qu D., Chen Q., Ding Y., Zhao B., Wang Z., Li X., Wang D. Think small, act big: primitive prompt learning for lifelong robot manipulation // Proceedings of the Computer Vision and Pattern Recognition Conference. 2025. pp. 22573–22583.

Аграновский Марат Антонович — инженер-исследователь, Санкт-Петербургский государственный университет (СПбГУ); студент, Московский государственный университет имени М.В. Ломоносова (МГУ). Область научных интересов: дискретные алгоритмы, планирование траекторий и навигация, мультиагентные системы, эвристический поиск, математические основы ИИ. Число научных публикаций — 3. agrinscience@gmail.com; Университетская набережная, 7–9, 199034, Санкт-Петербург, Россия; р.т.: +7(931)366-7576.

Яковлев Константин Сергеевич — канд. физ.-мат. наук, доцент, доцент, Санкт-Петербургский государственный университет (СПбГУ); ведущий научный сотрудник, Федеральное государственное учреждение "Федеральный исследовательский центр "Информатика и управление" Российской академии наук" (ФИЦ ИУ РАН). Область научных интересов: искусственный интеллект, робототехника, мульти-агентные системы. Число научных публикаций — 156. yakovlev@isa.ru; Университетская набережная, 7–9, 199034, Санкт-Петербург, Россия; р.т.: +7(926)270-8979.

M. AGRANOVSKIY, K. YAKOVLEV
**AN EFFICIENT METHOD FOR GENERATING KINEMATICALLY
FEASIBLE MOTION PRIMITIVES VIA CURVATURE
PARAMETERIZATION**

Agranovskiy M., Yakovlev K. An Efficient Method for Generating Kinematically Feasible Motion Primitives via Curvature Parameterization.

Abstract. Motion planning for mobile robots and autonomous vehicles is a key challenge in modern robotics. The main difficulty lies in accommodating non-trivial kinematic constraints, such as trajectory smoothness and minimum turning radius limitations. One of the most effective approaches to this problem is planning in a discretized state space, which utilizes a precomputed set of kinematically feasible trajectories known as motion primitives. The quality and generation speed of these primitives directly determine the efficiency of the entire planning process. This paper proposes a novel method for generating motion primitives for systems with bicycle-type kinematics. The method is based on representing the trajectory as a curve with a polynomial curvature function. Unlike existing approaches that use polynomial coefficients as parameters, we introduce a new reparameterization based on curvature values at key points along the trajectory. This reparameterization has a more intuitive physical meaning, which improves the convergence and stability of the numerical solution-finding method. The task of generating a primitive connecting two given states (position, orientation, curvature) is reduced to solving a system of nonlinear equations using Newton's method. The conducted experiments demonstrate the advantage of the proposed reparameterization in terms of speed and robustness compared to the baseline approach.

Keywords: trajectory planning, state lattice, motion primitives, kinematic constraints, autonomous vehicle, bicycle model, Newton's method.

References

1. Pivtoraiko M., Kelly A. Efficient constrained path planning via search in state lattices. International Symposium on Artificial Intelligence, Robotics, and Automation in Space. 2005. pp. 1–7.
2. Pivtoraiko M., Knepper R.A., Kelly A. Differentially constrained mobile robot motion planning in state lattices. Journal of Field Robotics. 2009. vol. 26. no. 3. pp. 308–333. DOI: 10.1002/rob.20285.
3. Pivtoraiko M., Kelly A. Kinodynamic motion planning with state lattice motion primitives. Proceedings of 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2011. pp. 2172–2179. DOI: 10.1109/IROS.2011.6094900.
4. Hart P.E., Nilsson N.J., Raphael B. A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics. 1968. vol. 4. no. 2. pp. 100–107. DOI: 10.1109/TSSC.1968.300136.
5. Ebendt R., Drechsler R. Weighted A* search – unifying view and application. Artificial Intelligence Journal. 2009. vol. 173. no. 14. pp. 1310–1342. DOI: 10.1016/j.artint.2009.06.004.
6. Nagy B., Kelly A. Trajectory generation for car-like robots using cubic curvature polynomials. Field and Service Robotics. 2001. vol. 11. pp. 479–490.
7. Harabor D., Grastien A. Online graph pruning for pathfinding on grid maps. Proceedings of the 25th AAAI Conference on Artificial Intelligence. 2011. vol. 25. no. 1. pp. 1114–1119. DOI: 10.1609/aaai.v25i1.7994.

8. Carlson M., Moghadam S.K., Harabor D.D., Stuckey P.J., Ebrahimi M. Optimal pathfinding on weighted grid maps. *Proceedings of the 37th AAAI Conference on Artificial Intelligence*. 2023. vol. 37. no. 10. pp. 12373–12380. DOI: 10.1609/aaai.v37i10.26458.
9. Nobes T.K., Harabor D., Wybrow M., Walsh S.D.C. The JPS pathfinding system in 3D. *Proceedings of the 15th International Symposium on Combinatorial Search*. 2022. vol. 15. no. 1. pp. 145–152. DOI: 10.1609/socs.v15i1.21762.
10. Daniel K., Nash A., Koenig S., Felner A. Theta*: Any-angle path planning on grids. *Journal of Artificial Intelligence Research*. 2010. vol. 39. pp. 533–579. DOI: 10.1613/jair.2994.
11. Ahmadi S., Tack G., Harabor D.D., Kilby P. A fast exact algorithm for the resource constrained shortest path problem. *Proceedings of the 35th AAAI Conference on Artificial Intelligence*. 2021. vol. 35. no. 14. pp. 12217–12224. DOI: 10.1609/aaai.v35i14.17450.
12. Ahmadi S., Tack G., Harabor D.D., Kilby P. Weight constrained path finding with bidirectional A*. *Proceedings of the 15th International Symposium on Combinatorial Search*. 2022. vol. 15. no. 1. pp. 2–10. DOI: 10.1609/socs.v15i1.21746.
13. Cohen B.J., Chitta S., Likhachev M. Search-based planning for manipulation with motion primitives. *2010 IEEE International Conference on Robotics and Automation*. 2010. pp. 2902–2908. DOI: 10.1109/ROBOT.2010.5509685.
14. Kang G., Kim Y.B., Lee Y.H., Oh H.S., You W.S., Choi H.R. Sampling-based motion planning of manipulator with goal-oriented sampling. *Intelligent Service Robotics*. 2019. vol. 12. pp. 265–273. DOI: 10.1007/s11370-019-00281-y.
15. Phillips M., Likhachev M. SIPP: Safe interval path planning for dynamic environments. *2011 IEEE International Conference on Robotics and Automation*. 2011. pp. 5628–5635. DOI: 10.1109/ICRA.2011.5980306.
16. Yakovlev K., Andreychuk A., Stern R. Revisiting bounded-suboptimal safe interval path planning. *Proceedings of the 30th International Conference on Automated Planning and Scheduling*. 2020. vol. 30. no. 1. pp. 300–304. DOI: 10.1609/icaps.v30i1.6674.
17. Rybecky T., Kulich M., Andreychuk A., Yakovlev K. Towards narrowing the search in bounded-suboptimal safe interval path planning. *Proceedings of the 12th International Symposium on Combinatorial Search, SoCS 2021*. 2021. vol. 12. no. 1. pp. 136–140. DOI: 10.1609/socs.v12i1.18562.
18. Ren Z., Rathinam S., Likhachev M., Choset H. Multi-objective safe-interval path planning with dynamic obstacles. *IEEE Robotics and Automation Letters*. 2022. vol. 7. no. 3. pp. 8154–8161. DOI: 10.1109/LRA.2022.3187270.
19. Gonzalez D., Perez J., Milanés V., Nashashibi F. A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*. 2016. vol. 17. no. 4. pp. 1135–1145. DOI: 10.1109/TITS.2015.2498841.
20. La Valle S.M., Kuffner Jr J.J. Randomized kinodynamic planning. *The International Journal of Robotics Research*. 2001. vol. 20. no. 5. pp. 378–400. DOI: 10.1177/02783640122067453.
21. Karaman S., Frazzoli E. Incremental sampling-based algorithms for optimal motion planning. *Robotics: Science and Systems*. 2010. vol. 104. no. 2. pp. 267–274. DOI: 10.1177/0278364911406761.
22. Otte M., Frazzoli E. RRTX: Real-time motion planning/replanning for environments with unpredictable obstacles. *Algorithmic Foundations of Robotics XI*. Springer Tracts in Advanced Robotics. 2015. vol. 107. pp. 461–478. DOI: 10.1007/978-3-319-16595-0_27.
23. Paudel A. Motion primitives based path planning with rapidly-exploring random tree. *arXiv preprint arXiv: 2210.15784*. 2022.

24. Golovin V., Yakovlev K. [Robot motion primitives in trajectory planning with kinematic constraints]. *Informatika i avtomatizatsiya – Informatics and Automation*. 2023. vol. 22. no. 6. pp. 1354–1386. DOI: 10.15622/ia.22.6.4. (In Russ.).
25. Jingyu Y., Jianjun M. Search-based trajectory planning with motion primitives for quadrotors using pruning A* algorithm. 2022 37th Youth Academic Annual Conference of Chinese Association of Automation (YAC). 2022. pp. 996–1000. DOI: 10.1109/YAC57282.2022.10023615.
26. Scharff J., Gonzalez D., Hernandez J.D., Pairet E., Petillot Y. Online 3-dimensional path planning with kinematic constraints in unknown environments using hybrid A* with tree pruning. *Sensors*. 2021. vol. 21. no. 4. 1152. p.
27. Agranovskiy M., Yakovlev K. MeshA*: Efficient path planning with motion primitives. *Proceedings of the 40th AAI Conference on Artificial Intelligence*. 2026. vol. 40. no. 43. pp. 36785–36792. DOI: 10.1609/aaai.v40i43.41004.
28. Piazza A., Bianco C.G.L., Bertozzi M., Fascioli A., Broggi A. Quintic G/sup 2/-splines for the iterative steering of vision-based autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*. 2002. vol. 3. no. 1. pp. 27–36. DOI: 10.1109/6979.994793.
29. Walton D.J., Meek D.S. A controlled clothoid spline. *Computers Graphics*. 2005. vol. 29. no. 3. pp. 53–363. DOI: 10.1016/j.cag.2005.03.008.
30. Farouki R.T. *Pythagorean-hodograph curves: algebra and geometry inseparable*. *Geometry and Computing*. Springer. 2008. vol. 1. pp. 131–196. DOI: 10.1007/978-3-540-73398-0_8.
31. Yakovlev K., Belinskaya J., Makarov D., Andreychuk A. [Safe-interval planning and coverage method for controlling the movement of a mobile robot in an environment with static and dynamic obstacles]. *Avtomatika i telemekhanika – Automation and telemechanics*. 2022. no. 6. pp. 96–117. DOI: 10.31857/S0005231022060083. (In Russ.).
32. De Iaco R., Smith S.L., Czarnecki K. Learning a lattice planner control set for autonomous vehicles. *IEEE Intelligent Vehicles Symposium (IV)*. 2019. pp. 549–556. DOI: 10.1109/IVS.2019.8813797.
33. Weaver C., Capobianco R., Wurman P.R., Stone P., Tomizuka M. Real-time trajectory generation via dynamic movement primitives for autonomous racing. 2024 American Control Conference (ACC). 2024. pp. 352–359. DOI: 10.23919/ACC60939.2024.10644244.
34. Franke J., Moldagalieva A., Hanfeld P., Honig W. Accelerating db-A* for kinodynamic motion planning using diffusion. *arXiv preprint arXiv: 2503. 05539*. 2025.
35. Kraljusic B., Ajanovic Z., Covic N., Lacevic B. Search-based robot motion planning with distance-based adaptive motion primitives. 2025 XXX International Conference on Information, Communication and Automation Technologies (ICAT). 2025. pp. 1–6. DOI: 10.1109/ICAT66432.2025.11189289.
36. Yao Y., Liu S., Song H., Qu D., Chen Q., Ding Y., Zhao B., Wang Z., Li X., Wang D. Think small, act big: primitive prompt learning for lifelong robot manipulation. *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025. pp. 22573–22583.

Agranovskiy Marat — Research engineer, Saint Petersburg State University (SPbSU); Student, Lomonosov Moscow State University (MSU). Research interests: discrete algorithms, trajectory planning and navigation, multi-agent systems, heuristic search, mathematical foundations of AI. The number of publications — 3. agrinscience@gmail.com; 7–9, University Embankment, 199034, Saint-Petersburg, Russia; office phone: +7(931)366-7576.

Yakovlev Konstantin — Ph.D., Associate professor, Saint Petersburg State University (SPbSU); Leading researcher, Federal Research Center "Computer Science and Control" of the Russian

Academy of Sciences (FRC CSC RAS). Research interests: artificial intelligence, robotics, multi-agent systems. The number of publications — 156. yakovlev@isa.ru; 7–9, University Embankment, 199034, Saint-Petersburg, Russia; office phone: +7(926)270-8979.