

В.Н. МЕЩЕРЯКОВ, С.Е. КОНДРАТЬЕВ, Н.В. КАЗЮРА
**ПЛАНИРОВАНИЕ ТРАЕКТОРИИ ДВИЖЕНИЯ АВТОНОМНЫХ
МОБИЛЬНЫХ РОБОТОВ НА ОСНОВЕ ГРАДИЕНТНОГО
ПОТОКА С ПРОПОРЦИОНАЛЬНО-ИНТЕГРАЛЬНОЙ
КОРРЕКЦИЕЙ**

Мещеряков В.Н., Кондратьев С.Е., Казюра Н.В., Планирование траектории движения автономных мобильных роботов на основе градиентного потока с пропорционально-интегральной коррекцией.

Аннотация. Планирование траектории движения является одной из ключевых задач при разработке автономных мобильных роботов, обеспечивающей их безопасную и эффективную навигацию в средах с препятствиями. Существующие методы планирования, включая алгоритмы на основе выборки, оптимизационные и геометрические подходы, испытывают трудности при балансировке между вычислительной эффективностью, оптимальностью траектории и гарантированным выполнением ограничений безопасности. В данной работе представлен новый метод планирования траектории, основанный на градиентном потоке с пропорционально-интегральной коррекцией. Задача безопасной навигации робота формулируется как задача условной оптимизации, которая преобразуется в задачу безусловной оптимизации и решается методом градиентного потока с применением аппарата функционального анализа. Для гарантированного выполнения ограничений достижения целевого состояния и обхода препятствий динамика градиентного потока дополняется пропорциональным и интегральным членами обратной связи. Барьерные функции для описания препятствий строятся на основе гладких квадратичных форм с возможностью расширения на сложную геометрию посредством функций расстояния. Сформулированы достаточные условия в форме линейных матричных неравенств, при выполнении которых строго доказана локальная экспоненциальная сходимость алгоритма к оптимальным управляющим траекториям. Эффективность метода подтверждена численным моделированием навигации колёсного робота в загромождённом пространстве с множественными препятствиями. Проведено сравнение с гибридным алгоритмом поиска на графах, кинодинамическими быстрорастущими случайными деревьями и методом прямой коллокации как в автономном, так и в онлайн-режимах. Результаты демонстрируют, что предложенный алгоритм обеспечивает наименьшее время вычислений и наибольшую точность достижения целевого состояния. Обсуждены ограничения метода, включая локальный характер сходимости и возможность застревания в локальных минимумах в средах со сложной топологией. Показано, что дискретизированная версия алгоритма сохраняет вычислительную эффективность, достаточную для адаптивного управления в реальном времени с периодическим перепланированием.

Ключевые слова: планирование движения, обход препятствий, градиентный поток, пропорционально-интегральное управление, экспоненциальная сходимость, функциональный анализ, линейные матричные неравенства.

1. Введение. Планирование движения является ключевой функцией автономных систем, обеспечивающей безопасную и эффективную навигацию в условиях динамических ограничений [1–3]. В отечественной литературе данная проблематика также активно

исследуется [4, 5]. Тем не менее достижение работы в реальном времени с гарантиями безопасности остаётся сложной задачей, особенно для неполноприводных систем с меньшим числом управляющих входов, чем степеней свободы [6].

1.1. Существующие методы и их ограничения. Подходы к планированию движения можно условно разделить на методы на основе выборки, оптимизационные методы, геометрические методы и методы на основе обучения [7]. Алгоритмы на основе выборки [8–10] вероятно исследуют конфигурационное пространство и эффективны в многомерных задачах. Однако они обычно не обеспечивают гарантий оптимальности и могут страдать от медленной сходимости в загромождённых средах, что зачастую приводит к снижению качества решения в пользу вычислительной эффективности. Оптимизационные методы [11–13] ориентированы на уточнение траектории и эффективны в генерации гладких и энергоэффективных путей. Хотя некоторые из них способны обрабатывать нелинейные ограничения, эти методы, как правило, вычислительно затратны и чувствительны к локальным минимумам, что делает их менее пригодными для приложений реального времени. Геометрические методы [14, 15] используют инструменты дифференциальной геометрии для моделирования робототехнического движения, предлагая новые перспективы для генерации траекторий. Тем не менее они обычно опираются на большие штрафные члены для обеспечения ограничений, что увеличивает жёсткость задачи и вычислительную нагрузку. Недавние гибридные подходы, такие как геометрический тепловой поток (ГТП) с тонкой настройкой [16], направлены на смягчение этих проблем путём комбинирования геометрических и оптимизационных техник, достигая лучшей вычислительной эффективности и доказуемой сходимости при определённых условиях. Методы на основе обучения, особенно использующие глубокое обучение с подкреплением [17–19], показали высокую производительность в динамических и неопределённых средах, где традиционные методы могут не справиться. Однако эти методы требуют обширных обучающих данных и не имеют строгих теоретических гарантий, особенно в отношении безопасности и устойчивости.

Несмотря на эти достижения, остаются фундаментальные проблемы. Многие существующие подходы испытывают трудности с балансировкой оптимальности, выполнения ограничений и вычислительной эффективности. Более того, лишь немногие алгоритмы предоставляют формальные гарантии устойчивости и безопасности, особенно для нелинейных неполноприводных систем. Эти ограничения подчёркивают

необходимость создания структур планирования движения, которые были бы одновременно теоретически обоснованными и практически применимыми для автономных приложений реального времени.

1.2. Вклад данной работы. Данная работа вносит следующие ключевые вклады в область планирования движения с обходом препятствий:

Проектирование алгоритма. На основе классических методов пропорционально-интегрального управления предложен новый градиентно-пропорционально-интегральный поток (ГПИ-поток) непрерывного времени, систематически интегрирующий динамику градиентного потока с механизмами пропорциональной и интегральной обратной связи с использованием инструментов функционального анализа [24–26]. Стационарные состояния данного алгоритма соответствуют допустимым управляющим траекториям, которые по своей природе удовлетворяют как ограничениям достижения целевого состояния, так и ограничениям обхода препятствий.

Теоретические гарантии. Поскольку задача оптимизации является нелинейной и невыпуклой, вводится разумное предположение относительно градиентных свойств функционала ограничений. При выполнении условия, сформулированного как проверяемое линейное матричное неравенство (ЛМН), строго доказывается, что ГПИ-поток гарантирует локальную экспоненциальную сходимость к допустимым, достигающим целевого состояния и траектории без столкновений. Данный анализ устанавливает первую формальную связь между градиентными потоками непрерывного времени с ПИ-обратной связью и гарантированным выполнением ограничений в задачах планирования движения.

Численная валидация. Эффективность алгоритма продемонстрирована посредством численного моделирования навигации колёсного робота в загромождённых средах. Примечательно, что даже после дискретизации для практической реализации структура ГПИ-потока сохраняет вычислительную эффективность, обеспечивая практическую применимость решений непрерывного времени.

Онлайн-реализация. Также продемонстрирована адаптивность алгоритма путём его встраивания в периодический цикл онлайн-выполнения во время работы транспортного средства, что позволяет достичь эффективного и робастного отслеживания при обеспечении обхода препятствий без необходимости предварительно вычисленных путей.

1.3. Структура статьи. Оставшаяся часть данной статьи организована следующим образом. Раздел 2 содержит необходимые предварительные сведения, включая обозначения и постановку задачи планирования движения для аффинных по управлению систем. Раздел 3 представляет предлагаемый ГПИ-поток с теоретическим доказательством сходимости. Раздел 4 демонстрирует результаты моделирования, показывающие эффективность ГПИ-потока как в автономном, так и в онлайн-режимах, а также сравнивает его производительность с существующими методами. Наконец, раздел 5 завершает статью, резюмируя вклады и потенциальные направления будущих исследований.

2. Предварительные сведения. В данном разделе устанавливается математический фундамент путём введения ключевых обозначений и формулировки задачи планирования движения для аффинных по управлению систем.

2.1. Обозначения. Пусть \mathbb{R} обозначает множество действительных чисел, $\mathbb{R}_{\geq 0}$ – множество неотрицательных действительных чисел, \mathbb{N} – множество неотрицательных целых чисел. Для вектора $x \in \mathbb{R}^n$ выражение $\|x\|$ представляет его евклидову норму. Также используются матричные обозначения I_n для n -мерных единичных матриц. Для симметричной матрицы $A \in \mathbb{R}^{n \times n}$ обозначение $A \succeq 0$ (соотв. \succ) означает, что A положительно полуопределена (соотв. положительно определена).

Определим следующие два функциональных пространства между множествами \mathcal{A} и \mathcal{B} :

- $L^2(\mathcal{A}, \mathcal{B})$: квадратично интегрируемые функции;
- $C^1(\mathcal{A}, \mathcal{B})$: непрерывно дифференцируемые функции.

В частности, определим $\mathcal{U} := L^2([0, T], \mathbb{R}^m)$ для некоторого $T > 0$, которое является гильбертовым пространством со скалярным произведением $\langle f, g \rangle := \int_0^T f(t)^\top g(t) dt$. Таким образом, для $u \in \mathcal{U}$, $\|u\| := \sqrt{\langle u, u \rangle}$. Оператор Якоби для $f \in C^1(\mathcal{A}, \mathcal{B})$ обозначается $\mathcal{J}f$. Когда $f : \mathcal{A} \rightarrow \mathbb{R}$, это сводится к оператору градиента ∇f .

2.2. Постановка задачи. Рассмотрим аффинную по управлению систему:

$$\dot{x}(t) = g(x(t)) + \sum_{i=1}^m h_i(x(t))u_i(t), \quad t \in [0, T], \quad (1)$$

где:

- вектор состояния $x = [x_1 \dots x_n]^\top \in \mathbb{R}^n$;
- управляющий вход $u = [u_1 \dots u_m]^\top \in \mathbb{R}^m$;

– дрейфовый вектор $g \in C^1(\mathbb{R}^n, \mathbb{R}^n)$, который предполагается глобально ограниченным;

– допустимые направления управления $h_1, \dots, h_m \in C^1(\mathbb{R}^n, \mathbb{R}^n)$, которые также предполагаются глобально ограниченными.

Возможно, что $m < n$, так что система является неполноприводной. Пусть $\varphi(\cdot; \alpha, u)$ обозначает траекторию системы (также известную как траектория, полученная интегрированием) из $x(0) = \alpha$ при управлении $u \in \mathcal{U}$.

Задача планирования движения формулируется как:

$$\min_{u \in \mathcal{U}} J_1(u) \quad \text{при } \varphi(T; \alpha, u) = \beta, \varphi(t; \alpha, u) \in \Upsilon \forall t \in [0, T]. \quad (2)$$

Здесь J_1 – целевой функционал, определённый как:

$$J_1(u) := \int_0^T u^\top(t) Q u(t) dt \quad (3)$$

для некоторого $Q \succ 0$. Множество $\Upsilon \subset \mathbb{R}^n$ – это операционное пространство, которое содержит начальное и целевое состояния $\alpha, \beta \in \Upsilon$ и кодирует ограничения препятствий с операционными пределами. Иными словами, рассматриваемая задача планирования движения по существу состоит в минимизации Q -взвешенной нормы входа u при одновременном достижении желаемого целевого состояния и обходе препятствий. Более подробное описание постановки задачи можно найти в работе [16].

3. Алгоритм планирования движения. В данном разделе представлен алгоритм решения задачи планирования движения (2). Основная идея состоит в деформации произвольной начальной входной траектории $u_{\text{init}} \in \mathcal{U}$ таким образом, чтобы она в конечном счёте стала решением задачи (2). Поскольку $u \in \mathcal{U}$, эта деформация также должна принадлежать \mathcal{U} . Следовательно, как проектирование, так и анализ алгоритма требуют инструментов функционального анализа.

3.1. Проектирование алгоритма ГПИ-потока. Аналогично методу ГТП, разработанному в [15], вводится дополнительная переменная s помимо u , так что она становится $u(t, s) \in L^2 \times C^1([0, T] \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^m)$. Заметим, что для каждого фиксированного s , $u(\cdot, s)$ по-прежнему представляет кривую в \mathcal{U} . По мере увеличения s кривая $u(\cdot, s)$ постепенно деформируется, причём её бесконечно малая деформация задаётся $\frac{\partial}{\partial s} u(\cdot, s)$, сокращённо u_s . Далее определяются

подходящие значения для u_s , чтобы гарантировать сходимость $u(\cdot, s)$ к допустимому решению задачи (2) при $s \rightarrow \infty$.

Для обработки ограничений в задаче (2) сначала определим:

$$J_{2a}(u) := \|\varphi(T; \alpha, u) - \beta\|^2, \quad (4)$$

что связано с ограничением достижения целевого состояния, и введём функционал:

$$J_{2b}(u) := \int_0^T b(\varphi(t; \alpha, u)) dt, \quad (5)$$

который обрабатывает ограничение обхода препятствий. Здесь $b \in C^1(\mathbb{R}^n, \mathbb{R}_{\geq 0})$ – барьерная функция, удовлетворяющая условиям: $b(x) = 0$, если $x \in \Upsilon$; $b(x) > 0$, если $x \notin \Upsilon$.

Для препятствия сферической формы радиуса r с центром в точке $x_0 \in \mathbb{R}^n$ возможным выбором для b является:

$$b(x) := (\max\{0, r^2 - \|x - x_0\|^2\})^2. \quad (6)$$

Данная форма барьерной функции выбрана по следующим причинам: (1) она непрерывно дифференцируема на всём \mathbb{R}^n , что обеспечивает гладкость градиента; (2) квадратичная структура обеспечивает численную устойчивость вблизи границы препятствия; (3) функция обращается в нуль вне препятствия с нулевым градиентом, что минимизирует влияние на траекторию в допустимой области.

Для препятствий более сложной геометрии (полигональных, эллипсоидальных, невыпуклых) барьерные функции могут быть построены на основе функций знакового расстояния или R-функций Рвачёва [27]. В случае нескольких препятствий общая барьерная функция формируется как сумма индивидуальных барьеров для каждого препятствия.

Дальнейшее обсуждение проектирования барьерных функций можно найти в [20].

Замечание о выборе весовых коэффициентов. Параметры c_a и c_b определяют относительную важность ограничений достижения целевого состояния и обхода препятствий соответственно. На практике рекомендуется выбирать $c_a \geq c_b$, поскольку точное достижение цели обычно более критично. При $c_a \gg c_b$ алгоритм приоритизирует

достижение цели, что может приводить к траекториям вблизи границ препятствий. При $c_a \ll c_b$ обеспечивается большой запас безопасности, но возможно увеличение отклонения от целевого состояния. В представленных экспериментах использовались значения $c_a = 2, c_b = 1$ для автономного режима и $c_a = 1, c_b = 5/7$ для онлайн-режима, обеспечивающие приемлемый компромисс. Корректность аппроксимации обеспечивается при конечных значениях $c_a, c_b > 0$, если исходная задача (2) имеет допустимое решение.

Выберем веса $c_a, c_b > 0$ и определим:

$$J_2(u) := c_a J_{2a}(u) + c_b J_{2b}(u). \quad (7)$$

Путём преобразования жёстких ограничений в мягкие ограничения, задача оптимизации с ограничениями (2) может быть аппроксимирована задачей безусловной оптимизации:

$$\min_{u \in \mathcal{U}} J_1(u) + k_P J_2(u), \quad (8)$$

где $k_P > 0$ – большой коэффициент усиления. Задача (8) может быть локально решена с использованием «градиентного» потока:

$$u_s = -\nabla J_1(u) - k_P \nabla J_2(u). \quad (9)$$

Отметим, что дифференциальное уравнение (9) является лишь концептуальным, поскольку J_1 и J_2 – это функционалы, градиенты которых ещё не определены должным образом. Выражения для градиентов J_1, J_{2a} и J_{2b} , определённых в (3), (4) и (5) соответственно, будут приведены далее в разделе 3.2.

Тем не менее существуют две проблемы с приближённым решением задачи (2) через (9): во-первых, поскольку жёсткие ограничения были преобразованы в мягкие, всегда будет существовать разрыв между $\varphi(T; \alpha, u)$ и β , и ограничения обхода препятствий могут быть нарушены; во-вторых, увеличение коэффициента k_P может уменьшить нарушения ограничений, но это существенно увеличивает вычислительную сложность, как отмечено в [16].

На основе метода интегратора из [21], разработанного для обеспечения условий консенсуса, в уравнение (9) вводится интегральный член для обеспечения выполнения ограничений:

$$u_s = -\nabla J_1(u) - k_P \nabla J_2(u) - k_I \int_0^s \nabla J_2(u(\cdot, \tau)) d\tau. \quad (10)$$

Члены $\nabla J_1(u)$, $\nabla J_2(u)$ и $\int_0^s \nabla J_2(u(\cdot, \tau)) d\tau$ представляют градиент функционала стоимости, пропорциональную обратную связь и интегральную обратную связь для достижения целевого состояния и обхода препятствий соответственно. Это мотивирует аббревиатуру ГПИ-поток (Градиентно-Пропорционально-Интегральный Поток).

Определяя $z(t, s) := \int_0^s \nabla J_2(u(t, \tau)) d\tau$, выражение (10) можно эквивалентно записать как:

$$u_s = -\nabla J_1(u) - k_P \nabla J_2(u) - k_I z, \quad (11a)$$

$$z_s = \nabla J_2(u), \quad (11b)$$

с начальными условиями:

$$u(t, 0) = u_{\text{init}}(t) \quad \forall t \in [0, T], \quad (12)$$

$$z(t, 0) = 0 \quad \forall t \in [0, T]. \quad (12)$$

В стационарном состоянии, когда $\lim_{s \rightarrow \infty} u(\cdot, s) = u^*$, имеем $u_s(t, s) = z_s(t, s) = 0$ для почти всех $t \in [0, T]$, что влечёт:

$$\nabla J_2(u^*)(t) = 0 \quad \text{для почти всех } t \in [0, T]. \quad (13)$$

Вспоминая определения J_2 , J_{2a} и J_{2b} в (7), (4) и (5) соответственно, видно, что (13) является необходимым условием для $J_{2a}(u^*) = J_{2b}(u^*) = 0$. Таким образом, можно непосредственно заключить, что стационарные решения ГПИ-потока удовлетворяют необходимым условиям допустимости в планировании движения с обходом препятствий. Локальная сходимость к допустимому решению будет доказана в разделе 3.3.

3.2. Определение градиента функционала. Здесь определяется градиент функционала в $C^1(\mathcal{U}, \mathbb{R})$ и способ вычисления градиентов вышеупомянутых J_1 , J_{2a} , J_{2b} .

Как предложено в работе [22], для любого $J \in C^1(\mathcal{U}, \mathbb{R})$ градиент $\nabla J(u)$ характеризуется таким образом, что для любых $v \in \mathcal{U}$ и $\epsilon > 0$:

$$J(u + \epsilon v) = J(u) + \epsilon \langle \nabla J(u), v \rangle + O(\epsilon^2), \quad (14)$$

где напомним, что $\langle \cdot, \cdot \rangle$ – скалярное произведение, определённое для \mathcal{U} . Следовательно, нетрудно увидеть, что:

$$\nabla J_1(u) = 2Qu. \quad (15)$$

Для вычисления градиента функционала J_{2a} применяется метод возмущений из работы [16], в результате чего получается:

$$\delta \dot{x}(t) = A_u(t)\delta x(t) + B_u(t)\epsilon v(t) + O(\epsilon^2), \quad (16)$$

где $\epsilon > 0$, $v \in \mathcal{U}$ таковы, что ϵv – аддитивное возмущение управления u , δx – результирующее возмущение состояния x , и:

$$A_u(t) := \mathcal{J}g(\varphi(t; \alpha, u)) + \sum_{i=1}^m \mathcal{J}h_i(\varphi(t; \alpha, u))u_i(t), \quad (17a)$$

$$B_u(t) := [h_1(\varphi(t; \alpha, u)) \quad \dots \quad h_m(\varphi(t; \alpha, u))]. \quad (17b)$$

Заметим, что (16) – это возмущённая линейная система с переменными параметрами с состоянием δx и управляющим входом v . Более того, поскольку $\delta x(0) = 0$, из формулы вариации постоянных следует, что:

$$\delta x(t) = \epsilon \int_0^t \Phi_u(t, \tau) B_u(\tau) v(\tau) d\tau + O(\epsilon^2), \quad (18)$$

где $\Phi_u(t, \tau)$ – матрица перехода состояний, порождённая $A_u(s)$ от момента τ до t . В частности, имеем:

$$\delta x(T) = \epsilon \int_0^T \Phi_u(T, \tau) B_u(\tau) v(\tau) d\tau + O(\epsilon^2).$$

В результате, поскольку $\delta x(T) = \varphi(T; \alpha, u + \epsilon v) - \varphi(T; \alpha, u)$, получаем:

$$\begin{aligned}
 J_{2a}(u + \epsilon v) - J_{2a}(u) &= \|\varphi(T; \alpha, u + \epsilon v) - \beta\|^2 - \|\varphi(T; \alpha, u) - \beta\|^2 \\
 &= 2\epsilon(\varphi(T; \alpha, u) - \beta)^\top \int_0^T \Phi_u(T, \tau) B_u(\tau) v(\tau) d\tau + O(\epsilon^2).
 \end{aligned}$$

Сравнивая с (14), получаем:

$$\nabla J_{2a}(u)(t) = 2B_u^\top(t) \Phi_u^\top(T, t) (\varphi(T; \alpha, u) - \beta). \quad (19)$$

Наконец, вычислим градиент J_{2b} . Используя разложение Тейлора, имеем:

$$\begin{aligned}
 &b(\varphi(t; \alpha, u + \epsilon v)) - b(\varphi(t; \alpha, u)) \\
 &= \epsilon(\nabla b)^\top(\varphi(t; \alpha, u)) \int_0^t \Phi_u(t, \tau) B_u(\tau) v(\tau) d\tau + O(\epsilon^2).
 \end{aligned}$$

Таким образом,

$$\begin{aligned}
 J_{2b}(u + \epsilon v) - J_{2b}(u) &= \int_0^T b(\varphi(t; \alpha, u + \epsilon v)) - b(\varphi(t; \alpha, u)) dt \\
 &= \epsilon \int_0^T \int_t^T (\nabla b)^\top(\varphi(\tau; \alpha, u)) \Phi_u(\tau, t) B_u(t) v(t) d\tau dt + O(\epsilon^2).
 \end{aligned}$$

Сравнивая с (14), имеем:

$$\nabla J_{2b}(u)(t) = \int_t^T B_u^\top(t) \Phi_u^\top(\tau, t) \nabla b(\varphi(\tau; \alpha, u)) d\tau. \quad (20)$$

Подводя итог, ГПИ-поток (11) начинается с начальных состояний (12) для решения задачи планирования движения с обходом препятствий (2), где ∇J_1 , $\nabla J_2 = c_a \nabla J_{2a} + c_b \nabla J_{2b}$ определены через (15), (19) и (20) с некоторыми параметрами $c_a, c_b, k_P, k_I > 0$.

3.3. Теоретическое доказательство сходимости. В данном разделе доказывается, что ГПИ-поток является локально экспоненциально устойчивым, что означает: если $\|u_{\text{init}} - u^*\|$ достаточно

мало, где $u^* \in \mathcal{U}$ – оптимальное решение задачи (2), то $\|u(\cdot, s) - u^*\|$ сходится к нулю экспоненциально быстро при $s \rightarrow \infty$.

Поскольку задача оптимизации (2) нелинейна и невыпукла, необходимо ввести разумные предположения, чтобы обеспечить возможность проведения анализа. Для этого рассмотрим случай без препятствий, т.е. $J_2 = J_{2a}$. Из (19) следует, что $\|\nabla J_2(u)\|$ должна быть «пропорциональна» $\|u - u^*\|$. Это является частным случаем неравенства:

$$\int_0^T (\nabla J_2(u))^\top(t) M_1 \nabla J_2(u)(t) dt \leq \int_0^T (u(t) - u^*(t))^\top M_2 (u(t) - u^*(t)) dt, \quad (21)$$

с конкретным выбором $kM_1 = M_2 = I_m$, где $k > 0$. Таким образом, неравенство (21) принимается как предположение для анализа сходимости.

Замечание о неравенстве (21). Данное неравенство означает, что норма градиента функционала ограничений мажорируется взвешенной нормой отклонения управления от оптимального. Это условие выполняется для широкого класса систем, включая линейные и локально линеаризуемые нелинейные системы с гладкими ограничениями. Матрицы M_1 и M_2 характеризуют локальную структуру задачи вблизи оптимума: M_1 связана с чувствительностью ограничений к изменению управления, а M_2 – с кривизной целевого функционала. На практике для проверки условия (21) можно использовать численную оценку градиентов в окрестности предполагаемого решения или аналитический анализ линеаризованной системы.

Теорема 1. Пусть $u^* \in \mathcal{U}$ – оптимальное решение задачи (2), и предположим, что существуют окрестность $\Omega := \{u \in \mathcal{U} : \|u - u^*\| \leq r\}$, $r > 0$, m -мерные матрицы $M_1, M_2 \succeq 0$ такие, что неравенство (21) выполняется для всех $u \in \Omega$. Тогда, если существуют $2m$ -мерная матрица $P \succ 0$, константы $\lambda \geq 0$, $\varepsilon > 0$ такие, что ЛМН:

$$\lambda \begin{bmatrix} -M_2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & M_1 \end{bmatrix} - \begin{bmatrix} PA + A^\top P + \varepsilon P & PB \\ B^\top P & 0 \end{bmatrix} \succ 0 \quad (22)$$

выполняется, где:

$$A := \begin{bmatrix} -2Q & -k_I I_m \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{2m \times 2m}, \quad B := \begin{bmatrix} -k_P I_m \\ I_m \end{bmatrix} \in \mathbb{R}^{2m \times m},$$

и k_I достаточно велико, то оптимальное решение u^* является локально экспоненциально устойчивым для алгоритма (11).

Доказательство. Поскольку u^* – оптимальное решение, $\nabla J_2(u^*)(t) = 0$ для почти всех $t \in [0, T]$. Более того, оптимальность влечёт существование некоторого $z^* \in \mathcal{U}$ такого, что:

$$\nabla J_1(u^*)(t) + k_I z^*(t) = 0 \quad \text{для почти всех } t \in [0, T]. \quad (23)$$

Определим $\xi^* := \begin{bmatrix} u^* \\ z^* \end{bmatrix}$. Тогда из (15) и (23) следует, что:

$$A\xi^*(t) = 0 \quad \text{для почти всех } t \in [0, T]. \quad (24)$$

Далее определим $e(t, s) := \begin{bmatrix} u(t, s) \\ z(t, s) \end{bmatrix} - \xi^*(t)$, где u, z – решение системы (11). Имеем:

$$\begin{aligned} \frac{\partial}{\partial s} e(t, s) &= \frac{\partial}{\partial s} \begin{bmatrix} u(t, s) \\ z(t, s) \end{bmatrix} = \begin{bmatrix} -2Qu(t, s) - k_P \nabla J_2(u(t, s)) - k_I z(t, s) \\ \nabla J_2(u(t, s)) \end{bmatrix} \\ &= A \begin{bmatrix} u(t, s) \\ z(t, s) \end{bmatrix} + B \nabla J_2(u(t, s)) = Ae(t, s) + B \nabla J_2(u(t, s)), \end{aligned} \quad (25)$$

для почти всех $t \in [0, T]$, где использован факт (24) для последнего равенства.

С другой стороны, пусть $\zeta(t, s) := \begin{bmatrix} e(t, s) \\ \nabla J_2(u(\cdot, s))(t) \end{bmatrix} \neq 0$. Умножая слева на $\zeta^\top(t, s)$ и справа на $\zeta(t, s)$ неравенство (22) и интегрируя по t от 0 до T , получаем:

$$\begin{aligned} & \lambda \int_0^T \zeta^\top(t, s) \begin{bmatrix} -M_2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & M_1 \end{bmatrix} \zeta(t, s) dt > \\ & > \int_0^T \zeta^\top(t, s) \begin{bmatrix} PA + A^\top P + \varepsilon P & PB \\ B^\top P & 0 \end{bmatrix} \zeta(t, s) dt. \end{aligned}$$

Иными словами, если:

$$\int_0^T \zeta^\top(t, s) \begin{bmatrix} -M_2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & M_1 \end{bmatrix} \zeta(t, s) dt \leq 0, \quad (26)$$

то:

$$\int_0^T \zeta^\top(t, s) \begin{bmatrix} PA + A^\top P + \varepsilon P & PB \\ B^\top P & 0 \end{bmatrix} \zeta(t, s) dt < 0. \quad (27)$$

Наблюдается, что (26) эквивалентно (21), тогда как (27) эквивалентно:

$$\begin{aligned} & \int_0^T \left[e^\top(t, s)(PA + A^\top P)e(t, s) + e^\top(t, s)PB \nabla J_2(u(\cdot, s))(t) + \right. \\ & \left. + (\nabla J_2(u(\cdot, s)))^\top(t)B^\top Pe(t, s) \right] dt \leq - \int_0^T \varepsilon e^\top(t, s)Pe(t, s) dt. \end{aligned} \quad (28)$$

Определим кандидата функции Ляпунова:

$$V(u, z) := \int_0^T e^\top(t)Pe(t)dt, \quad (29)$$

где $e := \begin{bmatrix} u \\ z \end{bmatrix} - \xi^*$. Очевидно,

$$\underline{\sigma} \|e\|^2 \leq V(u, z) \leq \bar{\sigma} \|e\|^2, \quad (30)$$

где $\underline{\sigma}, \bar{\sigma} > 0$ – наименьшее и наибольшее сингулярные значения матрицы P . Следовательно, когда $u(\cdot, t) \in \Omega$, неравенство (21) влечёт (28), и из (25) следует, что:

$$\begin{aligned} \frac{\partial}{\partial s} V(u(\cdot, s), z(\cdot, s)) &= \int_0^T \frac{\partial}{\partial s} e^\top(t, s) P e(t, s) + e(t, s) P \frac{\partial}{\partial s} e(t, s) dt \\ &\leq \int_0^T \left[(Ae(t, s) + B \nabla J_2(u(t, s)))^\top P e(t, s) + \right. \\ &\quad \left. + e(t, s) P (Ae(t, s) + B \nabla J_2(u(t, s))) \right] dt \\ &\leq -\varepsilon \int_0^T e^\top(t, s) P e(t, s) dt = -\varepsilon V(u(\cdot, s), z(\cdot, s)). \end{aligned}$$

Следовательно, из (30) и (23) следует, что:

$$\begin{aligned} \|u(\cdot, s) - u^*\|^2 &\leq \|e(\cdot, s)\|^2 \leq \frac{1}{\underline{\sigma}} V(u(\cdot, s), z(\cdot, s)) \\ &\leq \frac{1}{\underline{\sigma}} V(u(\cdot, 0), z(\cdot, 0)) e^{-\varepsilon s} \leq \frac{\bar{\sigma}}{\underline{\sigma}} \|e(\cdot, 0)\|^2 e^{-2\varepsilon s} \\ &= \frac{\bar{\sigma}}{\underline{\sigma}} (\|u_{\text{init}} - u^*\|^2 + \|z^*\|^2) e^{-2\varepsilon s} = \\ &= \frac{\bar{\sigma}}{\underline{\sigma}} \left(\|u_{\text{init}} - u^*\|^2 + \frac{\|\nabla J_1(u^*)\|^2}{k_I^2} \right) e^{-2\varepsilon s}. \end{aligned}$$

Следовательно, если $\|u_{\text{init}} - u^*\|$ достаточно мало и k_I достаточно велико, так что $\|u_{\text{init}} - u^*\|^2 + \frac{\|\nabla J_1(u^*)\|^2}{k_I^2} \leq \frac{\underline{\sigma}}{\bar{\sigma}} r^2$, доказывается, что не только $u(\cdot, s) \in \Omega$ для всех $s \in \mathbb{R}_{\geq 0}$, но также $\|u(\cdot, s) - u^*\| \leq r e^{-\varepsilon s}$, так что u^* локально экспоненциально устойчиво для алгоритма (11).

Замечание о подборе параметров. Матрица P и константы $\lambda, \varepsilon, k_I$ в условии (22) определяются следующим образом. Матрица P выбирается как решение ЛМН (22), которое может быть найдено стандартными численными методами (например, с помощью пакетов YALMIP, CVX, SDPT3). На практике начальное приближение для P может быть выбрано как $P = \text{diag}(Q, \mu I_m)$ с последующей итеративной корректировкой. Параметр $\lambda \geq 0$ характеризует степень выполнения условия (21) и обычно выбирается в диапазоне $[0, 1; 10]$. Параметр $\varepsilon > 0$

определяет скорость экспоненциальной сходимости – большие значения обеспечивают быструю сходимость, но сужают область притяжения. В экспериментах использовались: $P = I_{2m}$, $\lambda = 1$, $\varepsilon = 0,5$, $k_I = 5$.

4. Реализация и моделирование. В данном разделе подтверждается эффективность предложенного алгоритма посредством как автономного, так и онлайн-моделирования простой модели автомобиля, маневрирующего в загромождённых средах. Поскольку ГПИ-поток (11) работает в непрерывном времени, он сначала должен быть дискретизирован для реализации. Этот процесс дискретизации обсуждается в разделе 4.1.

4.1. Дискретизация алгоритма. Начнём с разбиения области определения u, z , $[0, T] \times \mathbb{R}_{\geq 0}$, как $\{t_0, \dots, t_k\} \times \mathbb{N}$, где $t_j = \frac{T}{N}j$, $j = 0, 1, \dots, N$. Аффинная по управлению система непрерывного времени (1) может быть аппроксимирована системой дискретного времени:

$$x(t_{j+1}) = \frac{T}{N} \left(g(x(t_j)) + \sum_{i=1}^m h_i(x(t_j))u_i(t_j) \right) + x(t_j). \quad (31)$$

Пусть $\bar{\varphi}(\cdot; \alpha, U)$ обозначает траекторию системы (31) из $x(t_0) = \alpha$ при управлении $U = [u(t_0)^\top \dots u(t_{N-1})^\top]^\top$. *Замечание об обозначениях.* Символ φ используется для обозначения траектории непрерывной системы (1), тогда как $\bar{\varphi}$ – для траектории дискретизированной системы (31). При $N \rightarrow \infty$ дискретная траектория $\bar{\varphi}$ сходится к непрерывной φ . Имеем дискретизированные версии функционалов J_1, J_{2a}, J_{2b} :

$$\begin{aligned} \bar{J}_1(U) &:= \sum_{j=0}^{N-1} u^\top(t_j)Qu(t_j), \\ \bar{J}_{2a}(U) &:= \|\bar{\varphi}(T; \alpha, U) - \beta\|^2, \\ \bar{J}_{2b}(U) &:= \sum_{j=0}^N b(\bar{\varphi}(t_j; \alpha, U)). \end{aligned}$$

Дискретизируя формулы $\nabla J_1(u)$, $\nabla J_{2a}(u)$ и $\nabla J_{2b}(u)$, получаем:

$$\nabla \bar{J}_1(U) = 2 [u(t_0)^\top Q \quad \cdots \quad u(t_{N-1})^\top Q]^\top, \quad (32)$$

$$\nabla \bar{J}_{2a}(U) = 2 \hat{M}^\top (\bar{\varphi}(t_N; \alpha, U) - \beta), \quad (33)$$

$$\nabla \bar{J}_{2b}(U) = M^\top \frac{\partial \bar{b}}{\partial X}, \quad (34)$$

где:

$$M := \begin{bmatrix} H_0 & 0 & \cdots & 0 \\ G_1 H_0 & H_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ G_{N-1} \cdots G_1 H_0 & G_{N-1} \cdots G_2 H_1 & \cdots & H_{N-1} \end{bmatrix} \in \mathbb{R}^{nN \times mN},$$

$\hat{M} \in \mathbb{R}^{n \times mN}$ – последняя блочная строка матрицы M ,

$$\frac{\partial \bar{b}}{\partial X} := \begin{bmatrix} \nabla b(\bar{\varphi}(t_1; \alpha, U)) \\ \vdots \\ \nabla b(\bar{\varphi}(t_N; \alpha, U)) \end{bmatrix} \in \mathbb{R}^{nN},$$

$$G_j := \frac{T}{N} A_u(t_j) + I_n \in \mathbb{R}^{n \times n}, \quad H_j := \frac{T}{N} B_u(t_j) \in \mathbb{R}^{n \times m},$$

и:

$$A_u(t_j) := \mathcal{J}g(\bar{\varphi}(t_j; \alpha, U)) + \sum_{i=1}^m \mathcal{J}h_i(\bar{\varphi}(t_j; \alpha, U)) u_i(t_j),$$

$$B_u(t_j) := [h_1(\bar{\varphi}(t_j; \alpha, U)) \quad \cdots \quad h_m(\bar{\varphi}(t_j; \alpha, U))].$$

Прямое вычисление градиентов также можно найти в [16]. Для каждого $k \in \mathbb{N}$ пусть:

$$U^k := \begin{bmatrix} u(t_0, k) \\ u(t_1, k) \\ \vdots \\ u(t_{N-1}, k) \end{bmatrix} \in \mathbb{R}^{Nm}, \quad Z^k := \begin{bmatrix} z(t_0, k) \\ z(t_1, k) \\ \vdots \\ z(t_{N-1}, k) \end{bmatrix} \in \mathbb{R}^{Nm},$$

которые являются дискретизированными версиями $u(\cdot, k)$, $z(\cdot, k)$ соответственно. ГПИ-поток дискретного времени для реализации получается путём дискретизации (11) с фиксированным шагом $\eta > 0$:

$$U^{k+1} = U^k - \eta (\nabla \bar{J}_1(U^k) + k_P \nabla \bar{J}_2(U^k) + k_I Z^k), \quad (35a)$$

$$Z^{k+1} = Z^k + \eta \nabla \bar{J}_2(U^k). \quad (35b)$$

Кроме того, вычисляется значение $\bar{J}_2(U^k)$ на каждом временном шаге k , и алгоритм (35) завершается, когда $\bar{J}_2(U^k) < \epsilon$ для некоторого достаточно малого ϵ . Псевдокоды для инициализации и автономного алгоритма представлены в алгоритме 1 и алгоритме 2. При последовательном запуске двух алгоритмов находится допустимая управляющая траектория для задачи планирования движения с обходом препятствий.

Замечание о выборе параметров дискретизации. Шаг η и число узлов N существенно влияют на сходимость и устойчивость дискретного алгоритма (35). Для обеспечения устойчивости шаг η должен удовлетворять условию $\eta < 2/(\|2Q\| + k_P L_J + k_I)$, где L_J – константа Липшица градиента $\nabla \bar{J}_2$, аналогично условиям устойчивости для непрерывного случая. На практике рекомендуется выбирать $\eta \in [0,5; 2]$ с последующей корректировкой при наблюдении осцилляций. Число узлов N определяет точность аппроксимации непрерывной траектории: большие значения N обеспечивают лучшую точность, но увеличивают вычислительную сложность пропорционально $O(N^2)$ из-за вычисления матрицы M . В экспериментах $N = 100$ обеспечивало приемлемый компромисс между точностью и временем вычислений.

Algorithm 1 Инициализация

Вход: Операционное пространство Υ , число точек дискретизации N .

- 1: Построить барьерную функцию b из Υ .
- 2: Вычислить ∇b .
- 3: Сгенерировать $U_{\text{init}} \in \mathbb{R}^{mN}$ случайным образом.
- 4: **вернуть** $\nabla b, U_{\text{init}} \in \mathbb{R}^{mN}$.

Algorithm 2 Автономный ГПИ-поток

Вход: Начальное и целевое состояния α, β , начальное управление U_{init} , градиент барьерной функции ∇b , дрейфовый вектор g , допустимые направления управления h_1, \dots, h_m , матрица Q , число точек дискретизации N , временной интервал T , коэффициенты усиления k_P, k_I, c_a, c_b , шаг η , допуск ϵ .

- 1: Установить $U^0 = U_{\text{init}}$.
- 2: Установить $Z^0 = 0 \in \mathbb{R}^{mN}$.
- 3: **повторять**
- 4: Вычислить $\nabla \bar{J}_1(U^k), \nabla \bar{J}_{2a}(U^k), \nabla \bar{J}_{2b}(U^k)$ по формулам (32), (33) и (34) соответственно.
- 5: Обновить U^k, Z^k согласно (35).
- 6: Вычислить $\bar{J}_{2a}(U^k), \bar{J}_{2b}(U^k)$.
- 7: Вычислить $\bar{J}_2(U^k) = c_a \bar{J}_{2a}(U^k) + c_b \bar{J}_{2b}(U^k)$.
- 8: **пока** $\bar{J}_2(U^k) < \epsilon$
- 9: **вернуть** $U^* = U^k$ и решение $\bar{\varphi}(\cdot; \alpha, U^k)$.

4.2. Автономный пример. Программно-аппаратный комплекс.

Все эксперименты выполнены на персональном компьютере с процессором AMD Ryzen 7 5800H (3,2 ГГц, 8 ядер), 16 ГБ оперативной памяти. Алгоритмы реализованы на языке Python 3.10 с использованием библиотек NumPy 1.24 и SciPy 1.11 для численных вычислений. Для решения ЛМН использовался пакет CVXPY с решателем MOSEK.

Рассмотрим простой автомобиль [1], кинематическое уравнение которого имеет вид:

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ \theta \\ \phi \end{bmatrix} = v \begin{bmatrix} \cos \phi \\ \sin \phi \\ 0 \\ \frac{1}{d} \tan \theta \end{bmatrix} + \omega \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad (36)$$

где x, y обозначают его положение, θ – угол поворота рулевого колеса, ϕ – ориентация кузова, v – линейная скорость заднего колеса, и ω – угловая скорость. Система является бездрейфовой, т.е. $g = 0$.

Рассматривается рабочая среда в плоскости xy с четырьмя препятствиями с центрами в точках $(-3, 1)$, $(0, 4)$, $(3, -2)$ и $(6, 2)$. Требуется спланировать траекторию без столкновений для простого автомобиля так, чтобы его опорная точка переместилась из $(-5, 5)$ в $(5, -5)$, ориентация ϕ изменилась с 0 до $-\frac{\pi}{2}$, а угол поворота θ остался равным 0 . Это достигается решением автономного ГПИ-потока, т.е. последовательным запуском алгоритма 1 и алгоритма 2 со следующими параметрами: $Q = I_2$, $N = 100$, $T = 10$ с, $k_P = 5$, $k_I = 5$, $c_a = 2$, $c_b = 1$, $\eta = 1$, $\epsilon = 0,0001$.

На рисунке 1 показана спланированная траектория, сгенерированная предложенным алгоритмом. Примечательно, что для достижения целевого состояния с минимальным значением целевого функционала и избежания столкновений с препятствиями траектория включает короткий манёвр движения задним ходом в точке, обозначенной как позиция 3 на рисунке 1, перед достижением цели. Эта особенность обычно не наблюдается в других алгоритмах планирования движения.

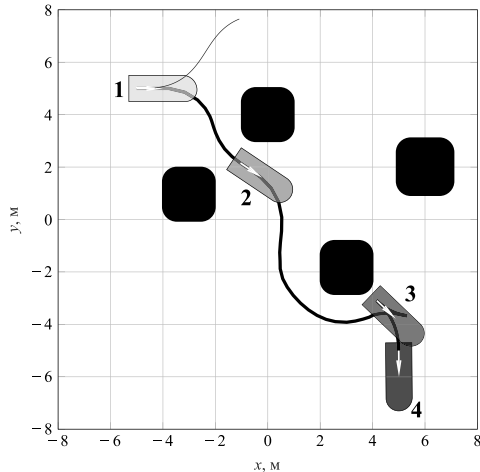


Рис. 1. Результаты автономного ГПИ-потока. Тонкая кривая представляет начальную траекторию, полученную интегрированием, полужирная кривая показывает финальную траекторию, суперэллипсы обозначают препятствия, а закругленные прямоугольники – снимки положения транспортного средства в различные моменты времени

Кроме того, проводится сравнение предложенного алгоритма с другими из литературы. Для обеспечения корректности сравнения все алгоритмы использовали одинаковую модель транспортного средства (36), идентичные начальное и целевое состояния, одинаковую конфигурацию препятствий. Параметры дискретизации унифицированы: $N = 100$ точек, $T = 10$ с. Критерий остановки для итеративных алгоритмов – достижение отклонения от цели менее 0,01 м или превышение 10000 итераций. Алгоритмы-конкуренты реализованы на основе открытых библиотек: Гибридный A^* и К-БСД* – OMPL 1.6, МПК – CasADi 3.6. В таблице 1 сравниваются время выполнения алгоритма, отклонение от целевого состояния $\Delta g := \|\bar{\varphi}(T; \alpha, U^*) - \beta\|$, длина интегрированного пути в плоскости xy и значение целевого функционала J_1 для шести различных алгоритмов: Гибридный A^* [23], Кинодинамические быстрорастущие случайные деревья (К-БСД*) [8], Метод прямой коллокации (МПК) [13], ГТП [15], ГТП с тонкой настройкой [16] и ГПИ-поток.

Результаты показывают, что предложенный ГПИ-поток достигает наименьшего времени выполнения и генерирует траекторию, полученную интегрированием, с наименьшим отклонением от целевого состояния. Хотя длина интегрированного пути и значение целевого функционала, связанное с движением простого автомобиля по этому пути, не являются оптимальными, эти значения остаются лишь незначительно выше минимума по сравнению с другими алгоритмами.

Таблица 1. Сравнение производительности алгоритмов

	Гибридный A^*	К-БСД*	МПК	ГТП	ГТП с тонкой настройкой	ГПИ-поток
Время, с	1,63	1,81	38,31	15,52	7,16	1,31
Δg , м	0,37	0,25	0,1	0,17	0,11	0,02
Длина, м	23,62	21,38	17,84	18,53	20,43	18,29
Функционал	1542,56	1565,83	1312,94	1375,79	1475,63	1428,52

4.3. Онлайн-пример. Как было продемонстрировано в предыдущем подразделе, включение члена интегральной обратной связи повышает скорость сходимости алгоритма, что делает его пригодным для приложений онлайн-отслеживания в реальном времени.

Для реализации ГПИ-потока в онлайн-режиме алгоритм выполняется периодически в течение K итераций. На каждой итерации требуется получить N_p управляющих выборок, которые могут быть применены к системе в реальном времени. В частности, во время j -й итерации, $j = 0, \dots, K - 1$, начальное состояние α для алгоритма берётся как текущее состояние системы. По мере уменьшения

временного интервала число точек дискретизации N корректируется пропорционально согласно:

$$N = (K - j)N_p. \quad (37)$$

Затем выполняется алгоритм 2 для вычисления оптимальной управляющей последовательности U^* , которая разбивается как $U^* = \begin{bmatrix} U_1^* \\ U_2^* \end{bmatrix}$, где $U_1^* \in \mathbb{R}^{mN_p}$ и $U_2^* \in \mathbb{R}^{m(N-N_p)}$. Управление U_1^* применяется к системе на временном интервале от $\frac{j}{K}T$ до $\frac{(j+1)N_p-1}{KN_p}T$, тогда как U_2^* используется как начальное управление для следующей итерации ГПИ-потока. Этот эвристический подход обеспечивает близость начального управления к оптимальному решению, тем самым ускоряя сходимость. Псевдокод для онлайн-алгоритма ГПИ-потока представлен в алгоритме 3.

Рассматривается та же задача, что и в разделе 4.2, с добавлением аддитивного белого шума к динамике, описываемой уравнением (36), где амплитуда шума составляет $\delta = 0,01$. Для дальнейшего повышения вычислительной эффективности допускается равномерное уменьшение ϵ от 1 до 0,001 в течение 10 итераций алгоритма 2. Эта корректировка отражает компромисс между точностью и скоростью, поскольку точное достижение цели менее критично на начальных этапах. Используемые параметры: $N_p = 10$, $k_P = 7$, $k_I = 4$, $c_a = 1$, $c_b = \frac{5}{7}$, тогда как остальные параметры соответствуют предыдущему примеру.

Фактическая траектория простого автомобиля, показанная на рисунке 2, демонстрирует отклонения от идеальной спланированной траектории из-за возмущений. Однако благодаря периодическому решению ГПИ-поток спланированные пути динамически корректируются в онлайн-режиме, позволяя транспортному средству достичь целевого состояния, избегая препятствий. Примечательно, что во время четвёртой итерации алгоритма 2 предложенный алгоритм планирования движения определяет обход одного из препятствий с противоположной стороны – особенность, которая не может быть достигнута другими алгоритмами планирования движения с топологическими ограничениями. Более того, общее время выполнения, которое относится к полной продолжительности выполнения алгоритмом полной задачи, составляющее 4,13 секунды, значительно меньше предписанного временного интервала $T = 10$ секунд, что подчёркивает возможность реализации алгоритма в приложениях реального времени.

Algorithm 3 Онлайн ГПИ-поток

Вход: Число точек дискретизации в интервалах N_p , число интервалов K , операционное пространство Υ , целевое состояние β , векторные поля g, h_1, \dots, h_m , матрица Q , параметры $T, k_P, k_I, \eta, \epsilon$.

- 1: Выполнить алгоритм 1 для получения ∇b и U_{init} .
- 2: для $j = 0, \dots, K - 1$ **выполнить**
- 3: Установить N согласно (37).
- 4: Установить α равным текущему состоянию транспортного средства.

- 5: Выполнить алгоритм 2 для получения U^* .
- 6: Применить U_1^* к транспортному средству на временном интервале от $\frac{j}{K}T$ до $\frac{(j+1)N_p - 1}{KN_p}T$.
- 7: Установить $U_{\text{init}} = U_2^*$.
- 8: **конец для**

Для полноты анализа в таблице 2 представлено сравнение производительности алгоритмов в онлайн-режиме при наличии возмущений. Все алгоритмы выполнялись с периодом перепланирования 1 с при тех же условиях, что и в автономном эксперименте, с добавлением аддитивного белого шума амплитудой $\delta = 0,01$.

Таблица 2. Сравнение производительности алгоритмов в онлайн-режиме

	Гибридный A^*	К-БСД*	МПК	ГПИ-поток
Общее время, с	16,9	18,3	>100	4,13
Δg , м	0,42	0,31	0,15	0,03
Столкновения, кол-во	0	1	0	0

Результаты показывают, что ГПИ-поток существенно превосходит конкурентов по времени выполнения в онлайн-режиме, обеспечивая при этом наименьшее отклонение от цели и отсутствие столкновений. МПК не смог завершить задачу за отведённое время из-за высокой вычислительной сложности.

5. Заключение. В данной работе представлен градиентно-пропорционально-интегральный поток (ГПИ-поток) непрерывного времени для планирования движения с обходом препятствий. ГПИ-поток интегрирует градиентный поток с пропорциональной и интегральной обратной связью, обеспечивая экспоненциальную сходимость к траекториям без столкновений при разумных предположениях. Результаты моделирования демонстрируют его эффективность и точность

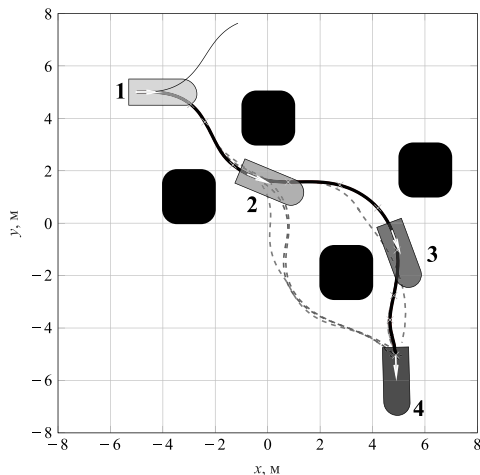


Рис. 2. Результаты онлайн ГПИ-потока. Следуя тем же обозначениям, что и на рисунке 1, кресты на полужирной кривой отмечают позиции на каждой итерации алгоритма 2, а пунктирные кривые иллюстрируют идеальные траектории к цели, сгенерированные на этих итерациях

как в автономном, так и в онлайн-режимах, что делает его пригодным для автономных систем.

Ограничения метода. Предложенный метод имеет ряд ограничений, которые следует учитывать при практическом применении. Во-первых, доказательство сходимости носит локальный характер и требует достаточной близости начального приближения к оптимальному решению. Во-вторых, в средах с локальными минимумами, вызванными тупиковыми конфигурациями или U-образными препятствиями, алгоритм может сходиться к локальному оптимуму, не достигая глобального решения – в таких случаях рекомендуется использование мультистартовой стратегии или комбинирование с глобальными методами поиска. В-третьих, для препятствий сложной невыпуклой геометрии построение гладких барьерных функций может представлять вычислительную сложность. В-четвёртых, в отличие от методов A^* и RRT, предложенный алгоритм не гарантирует полноты – то есть нахождения пути при его существовании – для произвольных конфигураций среды.

Сравнение с альтернативными подходами. По сравнению с алгоритмами A^* и RRT*, предложенный метод обеспечивает существенно меньшее время вычислений и более гладкие траектории, однако уступает им в способности находить пути в сильно загромождённых

средах с узкими проходами. Относительно методов прямой коллокации ГПИ-поток демонстрирует преимущество в вычислительной эффективности при сопоставимом качестве траекторий, но может проигрывать в точности при сложных динамических ограничениях.

Будущие исследования могут быть сосредоточены на нескольких ключевых направлениях: ослабление текущих предположений для обработки многомерных и более сложных сценариев; разработка методов выхода из локальных минимумов; проведение строгого анализа онлайн-производительности алгоритма; интеграция методов на основе обучения для повышения адаптивности в динамических средах; и расширение ГПИ-потока на многоагентные системы для задач совместного планирования движения.

Литература

1. Orthey A., Chamzas C., Kavraki L.E. Sampling-Based Motion Planning: A Comparative Review // Annual Review of Control, Robotics, and Autonomous Systems. 2024. vol. 7. pp. 285–310. DOI: 10.1146/annurev-control-061623-094742.
2. Zhang S., Jian Z., Deng X., Chen S., Nan Z., Zheng N. Hierarchical Motion Planning for Autonomous Driving in Large-Scale Complex Scenarios // IEEE Transactions on Intelligent Transportation Systems. 2022. vol. 23. no. 8. pp. 13291–13305. DOI: 10.1109/TITS.2021.3123327.
3. Zhang L., Cai K., Sun Z., Bing Z., Wang C., Figueredo L., Haddadin S., Knoll A. Motion planning for robotics: A review for sampling-based planners // Biomimetic Intelligence and Robotics. 2025. vol. 5. no. 1. 100207 p. DOI: 10.1016/j.birob.2024.100207.
4. Пшихопов В.Х., Медведев М.Ю., Костюков В.А., Хуссейн Ф., Кадим А. Алгоритмы планирования траекторий в двумерной среде с препятствиями // Информатика и автоматизация. 2022. Т. 21. №3. С. 459–492. DOI: 10.15622/ia.21.3.1.
5. Юхимец Д.А., Губанков А.С., Зуев А.В. Метод формирования пространственных траекторий мобильного робота в неизвестной обстановке // Робототехника и техническая кибернетика. 2018. №2(19). С. 46–51.
6. Cohn T., Petersen M., Simchowit M., Tedrake R. Non-Euclidean motion planning with graphs of geodesically convex sets // International Journal of Robotics Research. 2024. vol. 44. no. 10-11. pp. 1840–1862. DOI: 10.1177/02783649241302419.
7. Hu B., Cao Z., Zhou M. An Efficient RRT-Based Framework for Planning Short and Smooth Wheeled Robot Motion under Kinodynamic Constraints // IEEE Transactions on Industrial Electronics. 2021. vol. 68. no. 4. pp. 3292–3302. DOI: 10.1109/TIE.2020.2978701.
8. Huang J., Tang Y., Yan Z., Samuel A. Homotopic Path Set Planning for Robot Manipulation and Navigation // arXiv preprint arXiv:2406.02885v2. 2024.
9. Sahin A., Bhattacharya S. Topo-Geometrically Distinct Path Computation using Neighborhood-augmented Graph, and its Application to Path Planning for A Tethered Robot in 3D // IEEE Transactions on Robotics. 2024. vol. 41. pp. 20–41. DOI: 10.1109/TRO.2024.3492386.
10. Orthey A., Akbar S., Toussaint M. Multilevel motion planning: A fiber bundle formulation // International Journal of Robotics Research. 2023. vol. 43. no. 1. pp. 3–33. DOI: 10.1177/02783649231209337.
11. Jallet W., Bambade A., Arlaud E., El-Kazdadi S., Mansard N., Carpentier J. PROXDDP: Proximal Constrained Trajectory Optimization // IEEE Transactions on Robotics. 2025. vol. 41. pp. 2605–2624. DOI: 10.1109/TRO.2025.3554437.

12. Marcucci T., Petersen M., Von Wrangel D., Tedrake R. Motion planning around obstacles with convex optimization // *Science Robotics*. 2023. vol. 8. no. 84. pp. 1–39. DOI: 10.1126/scirobotics.adf7843.
13. Li Y., Han H., Kang S., Ma J., Yang H. On the Surprising Robustness of Sequential Convex Optimization for Contact-Implicit Motion Planning // *arXiv preprint arXiv:2502.01055v2*. 2025.
14. Bhattacharya S., Ghrist R. Path homotopy invariants and their application to optimal trajectory planning // *Annals of Mathematics and Artificial Intelligence*. 2018. vol. 84. pp. 139–160. DOI: 10.1007/s10472-018-9596-8.
15. Farber M. Topological Complexity of Motion Planning // *Discrete Computational Geometry*. 2003. vol. 29. pp. 211–221. DOI: 10.1007/s00454-002-0760-9.
16. Kim S., Park H. EL-AGHF: Extended Lagrangian Affine Geometric Heat Flow // *arXiv preprint arXiv:2505.24751*. 2025.
17. Li Z., Jin G., Yu R., Chen Z., Li N., Han W., Xiong L., Leng B., Hu J., Kolmanovsky I., Filev D. A Survey of Reinforcement Learning-Based Motion Planning for Autonomous Driving: Lessons Learned from a Driving Task Perspective // *arXiv preprint arXiv:2503.23650*. 2025.
18. Everett M., Chen Y.F., How J.P. Collision Avoidance in Pedestrian-Rich Environments With Deep Reinforcement Learning // *IEEE Access*. 2021. vol. 9. pp. 10357–10377. DOI: 10.1109/ACCESS.2021.3050338.
19. Kaufmann E., Bauersfeld L., Loquercio A., Muller M., Koltun V., Scaramuzza D. Champion-level drone racing using deep reinforcement learning // *Nature*. 2023. vol. 620. pp. 982–987. DOI: 10.1038/s41586-023-06419-4.
20. Ames A.D., Coogan S., Egerstedt M., Notomista G., Sreenath K., Tabuada P. Control Barrier Functions: Theory and Applications // *Proceedings of the 18th European Control Conference (ECC)*. 2019. pp. 3420–3431. DOI: 10.23919/ECC.2019.8796030.
21. Nedic A., Olshevsky A., Shi W. Achieving Geometric Convergence for Distributed Optimization Over Time-Varying Graphs // *SIAM Journal on Optimization*. 2017. vol. 27. no. 4. pp. 2597–2633. DOI: 10.1137/16M1084316.
22. Bolte J., Pauwels E. Conservative set valued fields, automatic differentiation, stochastic gradient methods and deep learning // *Mathematical Programming*. 2021. vol. 188. pp. 19–51. DOI: 10.1007/s10107-020-01501-5.
23. Pang J., Zhang S., Fu J., Liu J., Zheng N. Curvature continuous path planning with reverse searching for efficient and precise autonomous parking // *Proceedings of the IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. 2022. pp. 2798–2805. DOI: 10.1109/ITSC55140.2022.9922212.
24. Колмогоров А.Н., Фомин С.В. Элементы теории функций и функционального анализа // М.: Наука. 1976. 543 с.
25. Mordukhovich B.S. *Variational Analysis and Generalized Differentiation* // Springer. 2006. DOI: 10.1007/3-540-31247-1.
26. Поляк Б.Т. Введение в оптимизацию // М.: Наука. 1987. 384 с.
27. Рвачёв В.Л. Теория R-функций и некоторые её приложения // Киев: Наукова думка. 1982. 552 с.

Мещеряков Виктор Николаевич — д-р техн. наук, профессор, заведующий кафедрой, кафедра автоматизированного электропривода и робототехники, Федеральное государственное бюджетное образовательное учреждение высшего образования «Липецкий государственный технический университет» (ФГБОУ ВО «ЛГТУ»). Область научных интересов: автоматизированный электропривод промышленных комплексов, мехатроника

и робототехника. Число научных публикаций — 650. mesherek@yandex.ru; улица Московская, 30, 398055, Липецк, Россия; р.т.: +7(474)232-8180.

Кондратьев Сергей Евгеньевич — аспирант, ассистент, кафедры автоматизированного электропривода и робототехники, Федеральное государственное бюджетное образовательное учреждение высшего образования «Липецкий государственный технический университет» (ФГБОУ ВО «ЛГТУ»). Область научных интересов: робототехника, автоматизация, компьютерное зрение, машинное обучение. Число научных публикаций — 38. razthepsycho@yandex.ru; улица Московская, 30, 398055, Липецк, Россия; р.т.: +7(474)232-8180.

Казюра Николай Вадимович — магистр, кафедры автоматизированного электропривода и робототехники, Федеральное государственное бюджетное образовательное учреждение высшего образования «Липецкий государственный технический университет» (ФГБОУ ВО «ЛГТУ»). Область научных интересов: робототехника, автоматизация, компьютерное зрение, машинное обучение. Число научных публикаций — 12. nikolai.cazyura@yandex.ru; улица Московская, 30, 398055, Липецк, Россия; р.т.: +7(474)232-8180.

V.N. MESHCHERYAKOV, S.E. KONDRATYEV, N.V. KAZYURA
**TRAJECTORY PLANNING FOR AUTONOMOUS MOBILE ROBOTS
BASED ON GRADIENT FLOW WITH PROPORTIONAL-INTEGRAL
CORRECTION**

Meshcheryakov V.N., Kondratyev S.E., Kazyura N.V. Trajectory Planning for Autonomous Mobile Robots Based on Gradient Flow with Proportional-Integral Correction.

Abstract. Trajectory planning is one of the key tasks in the development of autonomous mobile robots, ensuring their safe and efficient navigation in environments with obstacles. Existing planning methods, including sampling algorithms, optimization, and geometric approaches, have difficulty balancing computational efficiency, optimal trajectory, and guaranteed compliance with safety constraints. This paper presents a new trajectory planning method based on gradient flow with proportional-integral correction. The problem of safe navigation of the robot is formulated as a constrained optimization problem, which is transformed into an unconstrained optimization problem and solved by the gradient flow method using the apparatus of functional analysis. To ensure that the constraints of reaching the target state and avoiding obstacles are met, the dynamics of the gradient flow is complemented by proportional and integral feedback terms. Barrier functions for describing obstacles are based on smooth quadratic shapes with the possibility of expanding to complex geometries using distance functions. Sufficient conditions are formulated in the form of linear matrix inequalities, under which the local exponential convergence of the algorithm to optimal control trajectories is rigorously proved. The effectiveness of the method is confirmed by numerical simulation of the navigation of a wheeled robot in a cluttered space with multiple obstacles. A comparison is made with a hybrid graph search algorithm, kinodynamic rapidly-exploring random trees, and the direct collocation method in both offline and online modes. The results demonstrate that the proposed algorithm provides the shortest calculation time and the highest accuracy in achieving the target state. The limitations of the method are discussed, including the local nature of convergence and the possibility of getting stuck in local minima in environments with complex topology. It is shown that the discretized version of the algorithm retains computational efficiency sufficient for adaptive control in real time with periodic rescheduling.

Keywords: motion planning, obstacle avoidance, gradient flow, proportional-integral control, exponential convergence, functional analysis, linear matrix inequalities.

References

1. Orthey A., Chamzas C., Kavraki L.E. Sampling-Based Motion Planning: A Comparative Review. *Annual Review of Control, Robotics, and Autonomous Systems*. 2024. vol. 7. pp. 285–310. DOI: 10.1146/annurev-control-061623-094742.
2. Zhang S., Jian Z., Deng X., Chen S., Nan Z., Zheng N. Hierarchical Motion Planning for Autonomous Driving in Large-Scale Complex Scenarios. *IEEE Transactions on Intelligent Transportation Systems*. 2022. vol. 23. no. 8. pp. 13291–13305. DOI: 10.1109/TITS.2021.3123327.
3. Zhang L., Cai K., Sun Z., Bing Z., Wang C., Figueredo L., Haddadin S., Knoll A. Motion planning for robotics: A review for sampling-based planners. *Biomimetic Intelligence and Robotics*. 2025. vol. 5. no. 1. 100207 p. DOI: 10.1016/j.birob.2024.100207.
4. Pshikhopov V., Medvedev M., Kostyukov V., Hussain F., Kadim A. [Trajectory Planning Algorithms in Two-Dimensional Environment with Obstacles]. *Informatika*

- i avtomatizatsiya – Informatics and Automation. 2022. vol. 21. no. 3. pp. 459–492. DOI: 10.15622/ia.21.3.1. (In Russ.).
5. Yukhimets D.A., Gubankov A.S., Zuev A.V. [Method of forming spatial trajectories of a mobile robot in an unknown environment]. *Robototekhnika i tekhnicheskaya kibernetika – Robotics and Technical Cybernetics*. 2018. no. 2(19). pp. 46–51. (In Russ.).
 6. Cohn T., Petersen M., Simchowit M., Tedrake R. Non-Euclidean motion planning with graphs of geodesically convex sets. *International Journal of Robotics Research*. 2024. vol. 44. no. 10-11. pp. 1840–1862. DOI: 10.1177/02783649241302419.
 7. Hu B., Cao Z., Zhou M. An Efficient RRT-Based Framework for Planning Short and Smooth Wheeled Robot Motion under Kinodynamic Constraints. *IEEE Transactions on Industrial Electronics*. 2021. vol. 68. no. 4. pp. 3292–3302. DOI: 10.1109/TIE.2020.2978701.
 8. Huang J., Tang Y., Yan Z., Samuel A. Homotopic Path Set Planning for Robot Manipulation and Navigation. *arXiv preprint arXiv:2406.02885v2*. 2024.
 9. Sahin A., Bhattacharya S. Topo-Geometrically Distinct Path Computation using Neighborhood-augmented Graph, and its Application to Path Planning for A Tethered Robot in 3D. *IEEE Transactions on Robotics*. 2024. vol. 41. pp. 20–41. DOI: 10.1109/TRO.2024.3492386.
 10. Orthey A., Akbar S., Toussaint M. Multilevel motion planning: A fiber bundle formulation. *International Journal of Robotics Research*. 2023. vol. 43. no. 1. pp. 3–33. DOI: 10.1177/02783649231209337.
 11. Jallet W., Bambade A., Arlaud E., El-Kazdadi S., Mansard N., Carpentier J. PROXDDP: Proximal Constrained Trajectory Optimization. *IEEE Transactions on Robotics*. 2025. vol. 41. pp. 2605–2624. DOI: 10.1109/TRO.2025.3554437.
 12. Marcucci T., Petersen M., Von Wrangel D., Tedrake R. Motion planning around obstacles with convex optimization. *Science Robotics*. 2023. vol. 8. no. 84. pp. 1–39. DOI: 10.1126/scirobotics.adf7843.
 13. Li Y., Han H., Kang S., Ma J., Yang H. On the Surprising Robustness of Sequential Convex Optimization for Contact-Implicit Motion Planning. *arXiv preprint arXiv:2502.01055v2*. 2025.
 14. Bhattacharya S., Ghrist R. Path homotopy invariants and their application to optimal trajectory planning. *Annals of Mathematics and Artificial Intelligence*. 2018. vol. 84. pp. 139–160. DOI: 10.1007/s10472-018-9596-8.
 15. Farber M. Topological Complexity of Motion Planning. *Discrete Computational Geometry*. 2003. vol. 29. pp. 211–221. DOI: 10.1007/s00454-002-0760-9.
 16. Kim S., Park H. EL-AGHF: Extended Lagrangian Affine Geometric Heat Flow. *arXiv preprint arXiv:2505.24751*. 2025.
 17. Li Z., Jin G., Yu R., Chen Z., Li N., Han W., Xiong L., Leng B., Hu J., Kolmanovsky I., Filev D. A Survey of Reinforcement Learning-Based Motion Planning for Autonomous Driving: Lessons Learned from a Driving Task Perspective. *arXiv preprint arXiv:2503.23650*. 2025.
 18. Everett M., Chen Y.F., How J.P. Collision Avoidance in Pedestrian-Rich Environments With Deep Reinforcement Learning. *IEEE Access*. 2021. vol. 9. pp. 10357–10377. DOI: 10.1109/ACCESS.2021.3050338.
 19. Kaufmann E., Bauersfeld L., Loquercio A., Muller M., Koltun V., Scaramuzza D. Champion-level drone racing using deep reinforcement learning. *Nature*. 2023. vol. 620. pp. 982–987. DOI: 10.1038/s41586-023-06419-4.
 20. Ames A.D., Coogan S., Egerstedt M., Notomista G., Sreenath K., Tabuada P. Control Barrier Functions: Theory and Applications. *Proceedings of the 18th European Control Conference (ECC)*. 2019. pp. 3420–3431. DOI: 10.23919/ECC.2019.8796030.

21. Nedic A., Olshevsky A., Shi W. Achieving Geometric Convergence for Distributed Optimization Over Time-Varying Graphs. *SIAM Journal on Optimization*. 2017. vol. 27. no. 4. pp. 2597–2633. DOI: 10.1137/16M1084316.
22. Bolte J., Pauwels E. Conservative set valued fields, automatic differentiation, stochastic gradient methods and deep learning. *Mathematical Programming*. 2021. vol. 188. pp. 19–51. DOI: 10.1007/s10107-020-01501-5.
23. Pang J., Zhang S., Fu J., Liu J., Zheng N. Curvature continuous path planning with reverse searching for efficient and precise autonomous parking. *Proceedings of the IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. 2022. pp. 2798–2805. DOI: 10.1109/ITSC55140.2022.9922212.
24. Kolmogorov A.N., Fomin S.V. *Elementy teorii funktsiy i funktsional'nogo analiza* [Elements of the theory of functions and functional analysis]. Moscow: Nauka., 1976. 543 p. (In Russ.).
25. Mordukhovich B.S. *Variational Analysis and Generalized Differentiation*. Springer. 2006. DOI: 10.1007/3-540-31247-1.
26. Polyak B.T. *Vvedenie v optimizatsiyu* [Introduction to Optimization]. Moscow: Nauka., 1987. 384 p. (In Russ.).
27. Rvachev V.L. *Teoriya R-funktsiy i nekotoryye yeye prilozheniya* [Theory of R-functions and some of its applications]. Kiev: Naukova dumka., 1982. 552 p. (In Russ.).

Meshcheryakov Victor — Ph.D., Dr. Sci., Professor, Head of the Department, Department of Automated Electric Drive and Robotics, Federal State Budgetary Educational Institution of Higher Education Lipetsk State Technical University (FSBEI HE LSTU). Research interests: automated electric drive for industrial complexes, mechatronics and robotics. The number of publications — 650. mesherek@yandex.ru; 30, Moskovskaya St., 398055, Lipetsk, Russia; office phone: +7(474)232-8180.

Kondratyev Sergey — Postgraduate student, assistant, Department of Automated Electric Drive and Robotics, Federal State Budgetary Educational Institution of Higher Education Lipetsk State Technical University (FSBEI HE LSTU). Research interests: robotics, automation, computer vision, machine learning. The number of publications — 38. razthepsycho@yandex.ru; 30, Moskovskaya St., 398055, Lipetsk, Russia; office phone: +7(474)232-8180.

Kazyura Nikolay — Master's student, Department of Automated Electric Drive and Robotics, Federal State Budgetary Educational Institution of Higher Education Lipetsk State Technical University (FSBEI HE LSTU). Research interests: robotics, automation, computer vision, machine learning. The number of publications — 12. nikolai.cazyura@yandex.ru; 30, Moskovskaya St., 398055, Lipetsk, Russia; office phone: +7(474)232-8180.