АННОТАЦИЯ

Данная работа посвящена разработке и исследованию метода управления на основе Model Predictive Path Integral (MPPI), обеспечивающего точное выполнение линейных по ускорениям ограничений (Пфаффовых ограничений) вида $A(q, v, t)\,\dot{v} = b(q, v, t)$, где $q \in \mathbb{R}^n$ — обобщённые координаты, v — обобщённые скорости, $A \in \mathbb{R}^{m \times n}$ — матрица ограничений, $b \in \mathbb{R}^m$ — вектор правой части.

Предлагаемый подход (Gauss-MPPI) основан на интеграции принципа наименьшего принуждения Гаусса в процедуру семплирования траекторий MPPI, что позволяет гарантировать выполнение ограничений на каждом шаге интегрирования динамики системы. Для проекции свободного ускорения на множество допустимых ускорений используется аналитическое решение — уравнение Удвадия–Калабы.

Работа включает обзор существующих методов обработки ограничений в MPPI, теоретическое обоснование предложенного подхода, программную реализацию алгоритма с поддержкой параллельных вычислений, а также экспериментальную проверку на задачах робототехники: неголономный мобильный робот, робот-манипулятор с ограничениями в операционном пространстве и система с контактными взаимодействиями.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Задача управления механическими системами в условиях ограничений является одной из центральных в современной робототехнике. Многие практически важные системы — мобильные роботы, манипуляторы, шагающие роботы — подчиняются кинематическим и динамическим ограничениям, которые должны выполняться точно на каждом шаге управления.

Model Predictive Path Integral (MPPI) — метод модельно-предиктивного управления, основанный на стохастической выборке траекторий. MPPI не требует вычисления градиентов функции стоимости, работает с невыпуклыми и негладкими целевыми функциями и эффективно параллелизуется на GPU. Однако исходная формулировка MPPI не предусматривает явной обработки ограничений.

Существующие подходы к обработке ограничений в MPPI — штрафные методы, барьерные функции, вероятностные фильтры — ориентированы преимущественно на ограничения типа неравенств (безопасность, границы управления). Ограничения равенств на уровне ускорений (Пфаффовы ограничения) в контексте MPPI практически не исследованы.

Целью данной работы является разработка и исследование метода управления Gauss-MPPI, обеспечивающего точное выполнение линейных по ускорениям ограничений вида:

$$A(q, v, t)\,\dot{v} = b(q, v, t). \tag{1}$$

Предлагаемый подход основан на интеграции принципа наименьшего принуждения Гаусса в процедуру семплирования траекторий MPPI, что позволяет гарантировать выполнение ограничений на каждом шаге интегрирования динамики.

Для достижения поставленной цели необходимо решить следующие **задачи**:

1. Провести анализ существующих методов обработки ограничений в MPPI, выявить их преимущества и недостатки.

2. Разработать теоретические основы метода Gauss-MPPI: сформулировать задачу проекции свободного ускорения на множество допустимых ускорений как задачу квадратичного программирования.

3. Исследовать применимость принципа наименьшего принуждения Гаусса для стохастических дифференциальных уравнений.
4. Разработать алгоритм Gauss-MPPI и его программную реализацию с поддержкой параллельных вычислений на GPU.
5. Провести экспериментальную проверку на модельных задачах робототехники.
6. Выполнить сравнительный анализ с классическим MPPI, CBF-MPPI и штрафными методами.

1. ЛИТЕРАТУРНЫЙ ОБЗОР

1.1. Теоретические основы Path Integral Control

Теория управления на основе интегралов по траекториям берёт начало в работах Каппена [1], который показал связь между стохастическим оптимальным управлением и статистической механикой. Для определённого класса задач (линейная зависимость динамики от управления, квадратичная стоимость управления) уравнение Гамильтона–Якоби–Беллмана преобразуется в линейное уравнение, решение которого выражается через интеграл по траекториям:

$$u^*(x,t) = \mathbb{E}\left[u(x,t) \cdot \exp\left(-\frac{1}{\lambda}S(\tau)\right)\right], \tag{1.1}$$

где $S(\tau)$ — стоимость траектории τ, λ — параметр температуры.

Теодору и др. [2] обобщили этот подход на более широкий класс систем и предложили алгоритм PI^2 (Policy Improvement with Path Integrals) для обучения с подкреплением.

1.2. Model Predictive Path Integral Control

Метод MPPI был предложен Уильямсом и др. [3] как подход к модельно-предиктивному управлению, основанный на стохастической выборке траекторий. В отличие от градиентных методов, MPPI генерирует множество случайных траекторий и вычисляет оптимальное управление путём взвешенного усреднения:

$$u^* = \sum_{k=1}^{K} w_k \cdot u_k, \quad w_k = \frac{\exp(-\frac{1}{\lambda}S_k)}{\sum_{j=1}^{K}\exp(-\frac{1}{\lambda}S_j)}, \tag{1.2}$$

где K — число траекторий, S_k — стоимость k-й траектории.

Ключевые преимущества MPPI:

- Не требует вычисления градиентов функции стоимости;
- Работает с невыпуклыми и негладкими функциями стоимости;
- Эффективно параллелизуется на GPU [4].

MPPI успешно применяется в автономном вождении [4], управлении квадрокоптерами и манипуляторами.

1.2.1. Постановка задачи и алгоритм MPPI

MPPI решает задачу стохастического оптимального управления для систем вида:

$$\begin{aligned} dx &= f(x,u)dt + B(x)dw \\ x(t_0) &= x_0 \end{aligned} \tag{1.3}$$

где $x \in \mathbb{R}^n$ — вектор состояния, $u \in \mathbb{R}^m$ — вектор управления, f — функция динамики, B — матрица диффузии, dw — Винеровский процесс.

Цель — минимизация функционала стоимости на горизонте T:

$$J(u) = \phi(x_T) + \int_{t_0}^{t_0+T} \mathcal{L}(x_t, u_t)\, dt, \tag{1.4}$$

где $\phi(x_T)$ — терминальная стоимость, $\mathcal{L}(x,u)$ — мгновенная стоимость.

Параметр $\lambda > 0$ регулирует «температуру» распределения: при $\lambda \to 0$ алгоритм выбирает траекторию с минимальной стоимостью, при $\lambda \to \infty$ — равномерное усреднение.

1.3. Обработка ограничений в MPPI

Исходная формулировка MPPI не предусматривает явной обработки ограничений. Существующие подходы классифицируются следующим образом.

1. Штрафные методы. Наиболее простой подход — добавление штрафов за нарушение ограничений в функцию стоимости [3]. Недостатки: не гарантируют выполнение ограничений, требуют настройки весовых коэффициентов.

2. Методы на основе барьерных функций. Shield-MPPI [5] использует Control Barrier Functions (CBF) для фильтрации небезопасных управлений. GS-MPPI [6] применяет композитные CBF для систем с множественными ограничениями. Эти методы гарантируют выполнение ограничений типа неравенств, но увеличивают вычислительную сложность.

3. Вероятностные методы. SC-MPPI [7] встраивает фильтр безопасности в процесс выборки траекторий. BC-MPPI [8] присваивает вероятность допустимости каждой траектории и корректирует веса соответственно.

4. Проекционные методы. π-MPPI [9] использует проекцию для обеспечения ограничений на величину и производные управления.

1.4. Принцип наименьшего принуждения Гаусса

Принцип наименьшего принуждения Гаусса утверждает: из всех кинематически возможных движений системы реализуется то, для которого величина принуждения минимальна. Математически принцип формулируется как задача квадратичного программирования:

$$\begin{aligned} \min_{\dot{v}} \quad & [\dot{v} - a]^T M [\dot{v} - a] \\ \text{s.t.} \quad & A\dot{v} = b, \\ & a = M^{-1}Q, \end{aligned} \tag{1.5}$$

где M — матрица масс, Q — вектор обобщённых сил, $a = M^{-1}Q$ — ускорение системы без ограничений (свободное ускорение).

1.5. Уравнение Удвадия–Калабы

Удвадия и Калаба [10, 11] разработали универсальный метод получения уравнений движения для систем с ограничениями вида $A(q, v, t)\dot{v} = b(q, v, t)$. Решение задачи минимизации принуждения имеет аналитический вид:

$$\dot{v} = \dot{v}_{\text{free}} + M^{-1/2}(AM^{-1/2})^{+}(b - A\dot{v}_{\text{free}}), \tag{1.6}$$

где $(\cdot)^+$ — псевдообратная матрица Мура–Пенроуза.

Уравнение применимо к:

- Голономным ограничениям: $\phi(q, t) = 0$;
- Неголономным ограничениям: $\phi(q, v, t) = 0$;
- Ограничениям в операционном пространстве: $J(q)\dot{v} + \dot{J}v = \ddot{x}_d$.

1.6. Выводы из обзора

Анализ литературы позволяет сделать следующие выводы:

1. Существующие методы обработки ограничений в MPPI ориентированы преимущественно на ограничения типа неравенств (безопасность, границы управления).

2. Ограничения равенств на уровне ускорений (Пфаффовы ограничения) в контексте MPPI практически не исследованы.

3. Принцип наименьшего принуждения Гаусса и уравнение Удвадия–Калабы предоставляют аналитическое решение для проекции на множество допустимых ускорений.

4. Связь между MPPI и принципом Гаусса не исследована.

2. МЕТОДОЛОГИЯ

2.1. Каноническое уравнение манипулятора

Динамика робота-манипулятора с n степенями свободы описывается каноническим уравнением:

$$M(q)\dot{v} + c(q, v) = B\tau, \tag{2.1}$$

где $q \in \mathbb{R}^n$ — вектор обобщённых координат, $v = \dot{q} \in \mathbb{R}^n$ — вектор обобщённых скоростей, $M(q) \in \mathbb{R}^{n\times n}$ — матрица масс (симметричная, положительно определённая), $c(q, v) \in \mathbb{R}^n$ — вектор кориолисовых, центробежных и гравитационных сил, $B \in \mathbb{R}^{n\times m}$ — матрица распределения управляющих воздействий, $\tau \in \mathbb{R}^m$ — вектор управляющих моментов.

Для полностью актуированной системы ($m = n$, $B = I$):

$$M(q)\dot{v} = \tau - c(q, v). \tag{2.2}$$

Свободное ускорение — ускорение системы при заданном управлении без учёта ограничений:

$$\dot{v}_{\text{free}} = M(q)^{-1}(\tau - c(q, v)). \tag{2.3}$$

Состояние системы определяется парой $x = (q, v) \in \mathbb{R}^{2n}$. Эволюция состояния при заданном ускорении $\dot{v}$ описывается полуявной (симплектической) схемой Эйлера:

$$\begin{aligned} v_{k+1} &= v_k + \dot{v}_k \cdot \Delta t, \\ q_{k+1} &= q_k + v_{k+1} \cdot \Delta t, \end{aligned} \tag{2.4}$$

которая обеспечивает лучшее сохранение энергии по сравнению с явной схемой.

2.2. Динамическая модель в алгоритме MPPI

В алгоритме MPPI динамическая модель (2.1) используется для прямого моделирования (rollout) траекторий системы. На каждой итерации генерируется K возмущённых последовательностей управления:

$$\tau_t^{(k)} = \tau_t^{\mathrm{nom}} + \epsilon_t^{(k)}, \quad \epsilon_t^{(k)} \sim \mathcal{N}(0, \Sigma), \tag{2.5}$$

где τ_t^{nom} — номинальное управление на шаге t, $\Sigma \in \mathbb{R}^{m\times m}$ — ковариационная матрица шума.

Для каждой возмущённой последовательности выполняется прямое моделирование на горизонте T шагов: вычисляется свободное ускорение $\dot{v}_{\text{free}} = M(q)^{-1}(\tau_t^{(k)} - c(q, v))$, интегрируется состояние по схеме (2.4) и накапливается стоимость. Полная стоимость k-й траектории:

$$S_k = \phi(x_T^{(k)}) + \sum_{t=0}^{T-1} \mathcal{L}(x_t^{(k)}, \tau_t^{(k)}), \tag{2.6}$$

где ϕ — терминальная стоимость, $\mathcal{L}$ — мгновенная стоимость.

Номинальное управление обновляется взвешенным усреднением возмущений:

$$\tau_t^{\mathrm{nom}} \leftarrow \tau_t^{\mathrm{nom}} + \sum_{k=1}^{K} w_k\, \epsilon_t^{(k)}, \quad w_k = \frac{\exp(-(S_k - \beta)/\lambda)}{\sum_{j=1}^{K} \exp(-(S_j - \beta)/\lambda)}, \tag{2.7}$$

где $\lambda > 0$ — параметр температуры, $\beta = \min_k S_k$ — нормировка для численной устойчивости.

2.3. Интеграция Пфаффовых ограничений через принцип Гаусса

Рассмотрим систему (2.1), подчинённую Пфаффовым ограничениям, линейным по ускорениям:

$$A(q, v, t)\dot{v} = b(q, v, t), \tag{2.8}$$

где $A \in \mathbb{R}^{p\times n}$ — матрица ограничений, $b \in \mathbb{R}^p$ — вектор правой части, p — число ограничений. Такие ограничения естественно возникают в робототехнике:

- Неголономные системы: ограничение бокового проскальзывания колёс $\dot{x}\sin\theta - \dot{y}\cos\theta = 0$;

- Ограничения в операционном пространстве: $J(q)\dot{v} = \ddot{x}_d - \dot{J}v$;
- Контактные ограничения: нормальная компонента ускорения в точке контакта равна нулю.

Ограничения (2.8) могут быть *виртуальными* — не воплощёнными в физике среды, а задаваемыми проектировщиком (например, движение конечного эффектора строго по прямой). В таком случае среда не обеспечивает их выполнение, и контроллер должен выдавать управляющий сигнал, который сам по себе порождает допустимое движение.

Согласно принципу наименьшего принуждения Гаусса, ускорение системы при наличии ограничений (2.8) определяется как решение задачи квадратичного программирования:

$$\begin{aligned} \min_{\dot{v}} \quad & \frac{1}{2}[\dot{v} - \dot{v}_{\text{free}}]^T M(q)[\dot{v} - \dot{v}_{\text{free}}] \\ \text{s.t.} \quad & A(q, v, t)\dot{v} = b(q, v, t), \end{aligned} \tag{2.9}$$

где $\dot{v}_{\text{free}}$ — свободное ускорение (2.3). Физический смысл: из всех ускорений, удовлетворяющих ограничениям, выбирается ближайшее к свободному в метрике, определяемой матрицей масс.

Аналитическое решение задачи (2.9) даётся уравнением Удвадия–Калабы:

$$\dot{v} = \dot{v}_{\text{free}} + M^{-1/2}(AM^{-1/2})^{+}(b - A\dot{v}_{\text{free}}), \tag{2.10}$$

где $(\cdot)^{+}$ — псевдообратная матрица Мура–Пенроуза. Решение допускает декомпозицию:

$$\dot{v} = \dot{v}_{\text{free}} + M^{-1}\lambda_c, \quad \lambda_c = M^{1/2}(AM^{-1/2})^{+}(b - A\dot{v}_{\text{free}}), \tag{2.11}$$

где λ_c — обобщённая сила связи, минимальная в смысле принуждения Гаусса.

Проекция Гаусса используется в алгоритме в двух ролях:

1. **Во время rollout**: проекция ускорений обеспечивает моделирование физически корректных (удовлетворяющих ограничениям) траекторий, что необходимо для правильной оценки стоимости и вычисления весов.

2. **На выходе алгоритма**: проекция номинального управления обеспечивает формирование выходного сигнала, содержащего силу связи. Без этого

шага возвращаемое управление не будет связано с наложенными ограничениями.

2.4. Алгоритм Gauss-MPPI

Полный алгоритм Gauss-MPPI представлен в виде псевдокода (Алгоритм 1).

Алгоритм 1 Gauss-MPPI

Вход: $x_0 = (q_0, v_0)$, $U = \{\tau_0, \ldots, \tau_{T-1}\}$, K, λ, Σ, Δt

Выход: τ_{out}

— *Rollout с проекцией Гаусса* —

1: **for** $k = 1, \ldots, K$ **do параллельно**

2: $\quad (q, v) \leftarrow (q_0, v_0), \quad S_k \leftarrow 0$

3: $\quad$ **for** $t = 0, \ldots, T-1$ **do**

4: $\quad\quad \epsilon_t^{(k)} \sim \mathcal{N}(0, \Sigma)$

5: $\quad\quad \hat{\tau} \leftarrow \tau_t + \epsilon_t^{(k)}$

6: $\quad\quad \dot{v}_{\text{free}} \leftarrow M(q)^{-1}(\hat{\tau} - c(q, v))$

7: $\quad\quad \dot{v} \leftarrow \dot{v}_{\text{free}} + M^{-1/2}(AM^{-1/2})^{+}(b - A\dot{v}_{\text{free}})$ $\triangleright$ проекция

8: $\quad\quad v \leftarrow v + \dot{v} \cdot \Delta t, \quad q \leftarrow q + v \cdot \Delta t$

9: $\quad\quad S_k \leftarrow S_k + \mathcal{L}((q, v),\, \hat{\tau})$

10: $\quad$ **end for**

11: $\quad S_k \leftarrow S_k + \phi(q, v)$

12: **end for**

— *Обновление номинального управления* —

13: $\beta \leftarrow \min_k S_k$

14: **for** $k = 1, \ldots, K$ **do**

15: $\quad w_k \leftarrow \exp\big(-(S_k - \beta)/\lambda\big)$

16: **end for**

17: $\eta \leftarrow \sum_{k=1}^{K} w_k$

18: **for** $t = 0, \ldots, T-1$ **do**

19: $\quad \tau_t \leftarrow \tau_t + \frac{1}{\eta} \sum_{k=1}^{K} w_k\, \epsilon_t^{(k)}$

20: **end for**

— *Проекция выходного управления* —

21: $\dot{v}_{\text{free}} \leftarrow M(q_0)^{-1}(\tau_0 - c(q_0, v_0))$

22: $\lambda_c \leftarrow M(q_0)^{1/2}\big(A_0 M(q_0)^{-1/2}\big)^{+}\big(b_0 - A_0 \dot{v}_{\text{free}}\big)$

23: $\tau_{\text{out}} \leftarrow \tau_0 + \lambda_c$

24: **return** τ_{out}

Алгоритм состоит из трёх этапов. Первый этап (строки 1–12) — rollout с проекцией Гаусса: каждая семплированная траектория моделируется с учётом ограничений, что обеспечивает корректную оценку стоимости. Второй этап (строки 13–19) — стандартное обновление MPPI: номинальная последовательность U обновляется взвешенным усреднением возмущений. Третий этап (строки 20–23) — проекция выходного управления: к номинальному моменту τ_0 добавляется обобщённая сила связи λ_c, вычисленная в текущем состоянии (q_0, v_0).

Номинальная последовательность U хранит *свободные* (без силы связи) управления. Это принципиально: сила связи λ_c зависит от состояния системы, которое на горизонте планирования неизвестно. При rollout проекция вычисляется в каждой точке траектории, а при выдаче управления — в текущем известном состоянии.

2.5. Обеспечение ограничений в Gauss-MPPI

2.5.1. Декомпозиция выходного управления

Выходной сигнал τ_{out} допускает декомпозицию:

$$\tau_{\text{out}} = \underbrace{\tau_0^{\text{nom}}}_{\text{MPPI-оптимальное}} + \underbrace{\lambda_c(q_0, v_0, \tau_0^{\text{nom}})}_{\text{сила связи}}, \tag{2.12}$$

где τ_0^{nom} — номинальное управление, найденное MPPI-оптимизацией, а λ_c — минимальная (в смысле принуждения Гаусса) сила, необходимая для выполнения ограничений.

При подаче τ_{out} на неограниченную систему $M\dot{v} + c = \tau_{\text{out}}$ результирующее ускорение:

$$\dot{v} = M^{-1}(\tau_{\text{out}} - c) = M^{-1}(\tau_0^{\text{nom}} + \lambda_c - c) = \dot{v}_{\text{free}} + M^{-1}\lambda_c, \tag{2.13}$$

что в точности совпадает с ограниченным ускорением (2.10).

2.5.2. Доказательство выполнения ограничений

Утверждение. Ускорение (2.13), порождаемое выходным управлением τ_{out}, удовлетворяет ограничению $A\dot{v} = b$.

Доказательство. Подставим (2.13) в $A\dot{v}$:

$$A\dot{v} = A\dot{v}_{\text{free}} + AM^{-1/2}(AM^{-1/2})^{+}(b - A\dot{v}_{\text{free}}). \tag{2.14}$$

Обозначим $\Phi = AM^{-1/2}$. Тогда:

$$A\dot{v} = A\dot{v}_{\text{free}} + \Phi\Phi^{+}(b - A\dot{v}_{\text{free}}). \tag{2.15}$$

Если A имеет полный строчный ранг, то $\Phi\Phi^{+} = I_p$ и:

$$A\dot{v} = A\dot{v}_{\text{free}} + b - A\dot{v}_{\text{free}} = b. \tag{2.16}$$

Равенство выполняется *для любого* $\tau_0^{\text{nom}} \in \mathbb{R}^m$. Следовательно, независимо от результата взвешенного усреднения MPPI, выходной сигнал τ_{out} порождает допустимое ускорение.

2.5.3. Роль проекции во время rollout

Проекция Гаусса во время rollout не влияет на выходное управление непосредственно, но обеспечивает *корректную оценку стоимости*. Без проекции:

- Семплированные траектории нарушали бы ограничения;
- Стоимости S_k не отражали бы реальное поведение системы под ограничениями;
- Веса w_k присваивались бы на основе нефизичных траекторий;
- MPPI-оптимизация находила бы управление, не учитывающее влияние ограничений на динамику.

Таким образом, проекция при rollout обеспечивает качество (оптимальность) управления, а проекция на выходе — выполнение ограничений.

2.6. Оптимальность Gauss-MPPI

Покажем, что Gauss-MPPI сходится к оптимальному управлению для задачи с ограничениями.

2.6.1. Ограниченная динамика как модель MPPI

Задача ограниченного оптимального управления — минимизация функционала (2.6) при динамике (2.2) и ограничениях (2.8):

$$\begin{aligned} \min_U \quad & J(U) = \phi(x_T) + \sum_{t=0}^{T-1} \mathcal{L}(x_t, \tau_t) \\ \text{s.t.} \quad & M\dot{v}_t + c = \tau_t, \quad A\dot{v}_t = b, \quad t = 0, \ldots, T-1. \end{aligned} \tag{2.17}$$

Подставив уравнение Удвадия–Калабы (2.10), ограничения на ускорения элиминируются, и задача (2.17) эквивалентна задаче без явных ограничений с модифицированной динамикой:

$$\min_U J(U), \quad \text{s.t.} \quad \dot{v}_t = f_c(x_t, \tau_t), \tag{2.18}$$

где f_c — ограниченная динамика, задаваемая формулой (2.10). Задача (2.18) не содержит явных ограничений: выполнение $A\dot{v} = b$ гарантируется структурой f_c.

Gauss-MPPI при rollout использует именно f_c в качестве модели динамики. Следовательно, с точки зрения MPPI, задача (2.18) есть стандартная задача оптимального управления с динамикой f_c.

2.6.2. Аффинность ограниченной динамики по управлению

Утверждение. Ограниченная динамика f_c аффинна по τ.

Доказательство. Подставим $\dot{v}_{\text{free}} = M^{-1}(\tau - c)$ в (2.10), обозначив $\Phi = AM^{-1/2}$:

$$\begin{aligned} \dot{v} &= M^{-1}(\tau - c) + M^{-1/2}\Phi^+\big(b - \Phi M^{-1/2}(\tau - c)\big) \\ &= \big(I - M^{-1/2}\Phi^+\Phi M^{-1/2}\big)M^{-1}(\tau - c) + M^{-1/2}\Phi^+ b. \end{aligned} \tag{2.19}$$

Обозначим:

$$N(q, v, t) = \big(I - M^{-1/2}\Phi^+\Phi M^{-1/2}\big)M^{-1}, \quad h(q, v, t) = -Nc + M^{-1/2}\Phi^+ b. \tag{2.20}$$

Тогда:

$$f_c(x, \tau) = N(x, t)\,\tau + h(x, t), \tag{2.21}$$

где N и h зависят от (q, v, t), но не от τ.

Матрица N имеет ясную интерпретацию: $M^{-1/2}\Phi^{+}\Phi M^{-1/2}$ — проектор (в метрике M) на подпространство, связанное ограничениями, а $(I - M^{-1/2}\Phi^{+}\Phi M^{-1/2})$ — проектор на ортогональное дополнение. Таким образом, N пропускает только ту компоненту управления, которая не противоречит ограничениям.

2.6.3. Сходимость к оптимальному ограниченному управлению

Теория path integral control [1,2] устанавливает следующий результат. Для системы с аффинной по управлению динамикой $\dot{v} = g(x) + G(x)\tau$, стоимостью вида $J = \phi(x_T) + \sum_t [\ell(x_t) + \frac{1}{2}\tau_t^T R\, \tau_t]$ и условием согласованности шума $\Sigma = \lambda R^{-1}$, оптимальное управление выражается через интеграл по траекториям:

$$\tau_t^* = \lim_{K\to\infty} \frac{\sum_{k=1}^{K} \tau_t^{(k)} \exp(-S_k/\lambda)}{\sum_{k=1}^{K} \exp(-S_k/\lambda)}, \tag{2.22}$$

что является в точности оценкой MPPI.

Поскольку ограниченная динамика f_c аффинна по τ (формула (2.21)), а шум управления $\epsilon \sim \mathcal{N}(0, \Sigma)$ входит через тот же канал, что и управление ($f_c(x, \tau + \epsilon) = N(\tau + \epsilon) + h$), условие согласованности шума выполняется. Следовательно, при $K \to \infty$ оценка MPPI сходится к оптимальному управлению задачи (2.18).

Для произвольных (не квадратичных по τ) функций стоимости MPPI минимизирует свободную энергию:

$$F_\lambda(U) = -\lambda \log \mathbb{E}_{E\sim\mathcal{N}(0,\Sigma)} \left[\exp\left(-\frac{J(U+E)}{\lambda}\right)\right], \tag{2.23}$$

являющуюся верхней оценкой на $\min_E J(U+E)$ (по неравенству Йенсена). При $\lambda \to 0$ свободная энергия сходится к истинному минимуму: $F_\lambda(U) \to \min_E J(U+E)$.

Итог. Gauss-MPPI сходится к оптимальному управлению задачи (2.17) при $K \to \infty$ (точность оценки) и $\lambda \to 0$ (точность аппроксимации). Проекция на выходе (строки 20–23 Алгоритма 1) не изменяет оптимальность: она детерминированно преобразует τ_0^{nom} в $\tau_{\text{out}} = \tau_0^{\text{nom}} + \lambda_c$, порождающий ту же ограниченную траекторию при подаче на неограниченную систему.

2.7. Стабилизация ограничений

При численном интегрировании ограничения на уровне ускорений (2.8) гарантируют $A\dot{v} = b$ на каждом шаге, но не предотвращают накопление ошибок на уровне координат и скоростей. Если ограничение (2.8) получено дифференцированием ограничений более высокого уровня, возникает дрейф.

2.7.1. Дрейф голономных ограничений

Рассмотрим голономное ограничение $\varphi(q) = 0$. Дифференцирование по времени даёт:

$$\begin{aligned} \dot{\varphi} &= \frac{\partial \varphi}{\partial q} v = 0, \\ \ddot{\varphi} &= \frac{\partial \varphi}{\partial q} \dot{v} + \frac{d}{dt}\left(\frac{\partial \varphi}{\partial q}\right) v = 0. \end{aligned} \tag{2.24}$$

Ограничение на уровне ускорений $A\dot{v} = b$ с $A = \partial\varphi/\partial q$ и $b = -\dot{A}v$ эквивалентно $\ddot{\varphi} = 0$. Численное интегрирование гарантирует лишь приближённое выполнение $\ddot{\varphi} \approx 0$, при этом ошибки в φ и $\dot{\varphi}$ могут накапливаться.

2.7.2. Стабилизация по Баумгарте

Метод Баумгарте [12] заменяет условие $\ddot{\varphi} = 0$ стабилизированным уравнением:

$$\ddot{\varphi} + 2\alpha\dot{\varphi} + \beta^2\varphi = 0, \tag{2.25}$$

где $\alpha, \beta > 0$ — параметры стабилизации. Уравнение (2.25) представляет собой уравнение демпфированного осциллятора для ошибки ограничения: при $\alpha = \beta$ (критическое демпфирование) отклонение φ экспоненциально убывает.

Модифицированная правая часть ограничения на уровне ускорений:

$$A\dot{v} = b - 2\alpha A v - \beta^2\varphi(q). \tag{2.26}$$

Для ограничений на уровне скоростей $\Phi(q, v) = 0$ (неголономные ограничения) стабилизация принимает вид:

$$A\dot{v} = b - \alpha\,\Phi(q, v), \tag{2.27}$$

где A и b получены дифференцированием Φ по времени.

2.7.3. Применение в Gauss-MPPI

В алгоритме Gauss-MPPI стабилизация по Баумгарте интегрируется путём замены правой части b на стабилизированную b_{stab} — как при rollout, так и при проекции выходного управления:

$$\dot{v} = \dot{v}_{\text{free}} + M^{-1/2}(AM^{-1/2})^{+}(b_{\text{stab}} - A\dot{v}_{\text{free}}). \tag{2.28}$$

При использовании стабилизации выходное управление не обеспечивает точное выполнение исходного ограничения $\varphi(q) = 0$ мгновенно, но *асимптотически стремится* к нему: динамика ошибки $\ddot{\varphi} + 2\alpha\dot{\varphi} + \beta^2\varphi = 0$ гарантирует экспоненциальное затухание отклонения с характерным временем $\sim 1/\alpha$.

Выбор параметров α и β определяет скорость коррекции дрейфа. На практике используются значения порядка $\alpha = \beta \sim 1/\Delta t$, что обеспечивает коррекцию ошибки за один–два шага интегрирования.

3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ И ЭКСПЕРИМЕНТЫ

3.1. Программный стек

Реализация алгоритма Gauss-MPPI выполнена на языке Python с использованием следующих библиотек:

- **JAX** — фреймворк для дифференцируемых вычислений с JIT-компиляцией и автоматическим распараллеливанием (`jax.vmap`) на GPU. Используется для реализации MPPI: параллельная выборка K траекторий, вычисление стоимостей и весов;
- **MuJoCo / MJX** — физический движок MuJoCo и его JAX-совместимая обёртка MJX. MJX позволяет выполнять шаги симуляции на GPU, что критично для параллельного rollout в MPPI;
- **Pinocchio** [13] — библиотека для вычисления прямой кинематики, якобианов и их производных по времени. Используется для задач, требующих аналитического доступа к кинематическим величинам (эксперимент с Dual-arm YuMi).

3.2. Реализация алгоритма Gauss-MPPI

3.2.1. Контроллер MPPI

Реализован класс `MPPI`, принимающий на вход конфигурацию (число траекторий K, горизонт T, температура λ, ковариация шума Σ) и функцию rollout. На каждом вызове `command()`:

1. Генерируются K возмущений $\epsilon^{(k)} \sim \mathcal{N}(0, \Sigma)$;
2. Возмущённые управления подаются на функцию rollout, возвращающую стоимости;
3. Вычисляются веса w_k и обновляется номинальная последовательность;
4. Возвращается первый элемент обновлённой последовательности.

Контроллер реализован полностью в JAX и JIT-компилируется для работы на GPU.

3.2.2. Ограниченная динамика

Проекция Гаусса реализована в классе `MjxWrapper`. После вычисления свободного ускорения $\dot{v}_{\text{free}}$ средствами MJX, выполняется проекция по уравнению Удвадия–Калабы:

1. Из каждого ограничения вычисляются блоки A_i, b_i;
2. Блоки объединяются: $A = [A_1; \ldots ; A_r]$, $b = [b_1; \ldots ; b_r]$;
3. Вычисляются $M^{1/2}$, $M^{-1/2}$ через собственное разложение матрицы масс;
4. Сила связи $\lambda_c = M^{1/2}(AM^{-1/2})^+(b - A\dot{v}_{\text{free}})$ добавляется к обобщённым силам.

Каждое ограничение реализуется как подкласс абстрактного класса `Constrain`
предоставляющий метод `compute(q, v)` $\to (A, b)$. Это позволяет комбинировать несколько ограничений в одном эксперименте.

3.2.3. Проекция выходного управления

Согласно Алгоритму 1, после обновления номинальной последовательности MPPI выходное управление проецируется в текущем состоянии:

$$\tau_{\text{out}} = \tau_0^{\text{nom}} + \lambda_c(q_0, v_0, \tau_0^{\text{nom}}). \tag{3.1}$$

Данная проекция вычисляется тем же механизмом, что и проекция при rollout, но однократно для текущего состояния (q_0, v_0).

3.3. Эксперимент 1: Колёсный робот без проскальзывания

3.3.1. Описание системы

Рассматривается дифференциально-приводной мобильный робот Boxer — платформа с двумя ведущими колёсами радиуса $r = 0.08$ м. Робот моделируется как свободное тело ($n_v = 6 + 2 = 8$: 6 DOF базы + 2 вращения колёс) с двумя группами виртуальных ограничений.

1. Ограничение горизонтальной плоскости. Робот удерживается на высоте $z = z_{\text{ref}}$ с фиксированной ориентацией вертикальной оси. Ограничение задаётся тремя скалярными условиями:

$$c_z = z_{\text{anchor}} - z_{\text{ref}} = 0, \quad c_{\text{orient}} = (\text{rot}(\text{anchor}) \cdot \hat{r})_{xy} = 0, \tag{3.2}$$

где $\hat{r}$ — начальное направление вертикальной оси. Стабилизация по Баумгарте:

$$A_1 \dot{v} = -\dot{A}_1 v - K_p^{(1)} c - K_d^{(1)} \dot{c}. \tag{3.3}$$

2. Ограничение отсутствия проскальзывания. Для каждого колеса i скорость точки контакта с поверхностью в проекции на плоскость равна нулю:

$$v_{\text{lin},i}^{xy} - \dot{\theta}_i \cdot r \cdot (a_i \times e_z)^{xy} = 0, \tag{3.4}$$

где $v_{\text{lin},i}$ — линейная скорость центра колеса в системе координат родительского тела, $\dot{\theta}_i$ — угловая скорость вращения колеса, a_i — ось вращения. Условие (3.4) — ограничение на уровне скоростей; при дифференцировании по времени получается ограничение вида $A_2 \dot{v} = b_2$ со стабилизацией:

$$A_2 \dot{v} = -\dot{A}_2 v - K_d^{(2)} c_w. \tag{3.5}$$

Суммарно система имеет $p = 3 + 4 = 7$ ограничений на $n_v = 8$ степеней свободы, оставляя одну управляемую степень свободы для каждого колеса (вращение).

3.3.2. Задача управления

Робот должен достичь заданной целевой позиции (x_d, y_d) на плоскости. Функция стоимости:

$$\mathcal{L}(x, \tau) = w_{\text{pos}} \|p_{xy} - p_d\|^2 + w_\tau \|\tau\|^2, \tag{3.6}$$

где p_{xy} — текущая позиция робота на плоскости, p_d — целевая позиция.

Управление $\tau \in \mathbb{R}^{n_v}$ — обобщённые силы. Проекция Гаусса гарантирует, что результирующее ускорение удовлетворяет ограничениям отсутствия проскальзывания и горизонтальной плоскости, независимо от конкретного значения τ.

3.3.3. Параметры эксперимента

Параметр	Значение
Число траекторий K	512
Горизонт T	30
Температура λ	1.0
Шаг интегрирования Δt	0.05 с
K_p (плоскость)	100
K_d (плоскость)	$2\sqrt{100} \approx 20$
K_d (проскальзывание)	$2\sqrt{100} \approx 20$
Радиус колёс r	0.08 м

Таблица 3.1. Параметры эксперимента с колёсным роботом

3.3.4. Результаты

3.4. Эксперимент 2: Dual-arm YuMi с жёстким объектом

3.4.1. Описание системы

Рассматривается двурукий манипулятор ABB IRB 14000 (YuMi), состоящий из двух 7-DOF рук. Каждая рука имеет 7 вращательных и 2 призматических (пальцы) сочленения; в эксперименте пальцы фиксированы, итого $n_v = 14$ управляемых степеней свободы.

Два манипулятора удерживают жёсткий объект (стержень), образуя кинематически замкнутую цепь. Объект фиксирован в схватах обоих манипуляторов, что создаёт голономное ограничение на 6 степеней свободы (3 позиционных + 3 ориентационных).

3.4.2. Формулировка ограничения

Пусть $\mathcal{E}_1$, $\mathcal{E}_2$ — системы координат конечных эффекторов левой и правой руки, $T_{\mathcal{W}}^{\mathcal{E}_i} \in SE(3)$ — их позы в мировой системе координат, T_A^B — постоянное преобразование между точками крепления на жёстком объекте. Голономное ограничение:

$$\varphi(q) = \mathbf{e}_{SE(3)}\Big(T_{\mathcal{W}}^{\mathcal{E}_1} \cdot T_A^B,\ T_{\mathcal{W}}^{\mathcal{E}_2}\Big) = 0, \tag{3.7}$$

где функция ошибки $\mathbf{e}_{SE(3)} : SE(3) \times SE(3) \to \mathbb{R}^6$ определена как:

$$\mathbf{e}_{SE(3)}(T_1, T_2) = \begin{bmatrix} \mathbf{p}_2 - \mathbf{p}_1 \\ \log(R_1^T R_2) \end{bmatrix}, \tag{3.8}$$

$\log : SO(3) \to \mathbb{R}^3$ — матричный логарифм.

Для приведения к форме Пфаффа $A\dot{v} = b$ используется стабилизированное кинематическое дифференциальное уравнение:

$$\begin{bmatrix} \dot{v}_{E_2} - \dot{v}_B \\ \dot{\omega}_{E_2}^{\mathcal{E}_2} - \dot{\omega}_B^{\mathcal{E}_2} \end{bmatrix} + K_d \begin{bmatrix} v_{E_2} - v_B \\ \omega_{E_2}^{\mathcal{E}_2} - \omega_B^{\mathcal{E}_2} \end{bmatrix} + K_p \cdot \mathbf{e}_{SE(3)}(\ldots) = 0, \tag{3.9}$$

где v_B, ω_B — линейная и угловая скорости точки B жёсткого объекта, вычисляемые через кинематику твёрдого тела:

$$v_B = v_{E_1} + \omega_{E_1} \times r_{AB}, \quad \omega_B = \omega_{E_1}. \tag{3.10}$$

Матрица ограничений и правая часть выражаются через якобианы конечных эффекторов:

$$\begin{aligned} A &= \begin{bmatrix} \tilde{J}_{d,v} \\ \tilde{J}_{d,\omega} \end{bmatrix}, \\ b &= -\begin{bmatrix} \dot{\tilde{J}}_{d,v} \\ \dot{\tilde{J}}_{d,\omega} \end{bmatrix} v - K_d \begin{bmatrix} v_d \\ \omega_d^{\mathcal{E}_2} \end{bmatrix} - K_p \cdot \mathbf{e}_{SE(3)}(\ldots), \end{aligned} \tag{3.11}$$

где $\tilde{J}_{d,v} = J_{E_2,v} - J_{E_1,v} + [\hat{r}_{AB}]J_{E_1,\omega}$, $\tilde{J}_{d,\omega} = R_{E_2}^T(J_{E_2,\omega} - J_{E_1,\omega})$, $[\hat{r}_{AB}]$ — оператор векторного произведения.

Устойчивость стабилизации (3.9) по позиционной компоненте следует из линейной теории ($K_p = \omega^2$, $K_d = 2\omega$ — критическое демпфирование). Устойчивость по ориентационной компоненте доказывается через функцию Ляпунова $V = \frac{1}{2}\tilde{\omega}^T\tilde{\omega} + \frac{K_p}{2}\mathbf{e}^T\mathbf{e}$ с использованием свойства $\mathbf{e}^T J_r^{-1}(\mathbf{e}) = \mathbf{e}^T$ матричного логарифма [14].

3.4.3. Задача управления

Один из конечных эффекторов (левая рука, фрейм $\mathcal{E}_1$) должен следовать заданной траектории $T_{\mathcal{W}}^{\mathcal{D}}(t)$ при сохранении жёсткой связи между руками. Управление задаётся через ограничения: ограничение жёсткой связи обеспечивает замкнутость кинематической цепи, а ограничение следования траектории задаёт желаемое движение.

В контексте Gauss-MPPI ограничение жёсткой связи (3.11) включается в проекцию Гаусса, а MPPI оптимизирует функцию стоимости, учитывающую отклонение от желаемой траектории и управляющие моменты.

3.4.4. Параметры эксперимента

Параметр	Значение
DOF системы	14 (7 + 7, пальцы фиксированы)
Размерность ограничения	6 (3 позиция + 3 ориентация)
Фрейм $\mathcal{E}_1$	`yumi_link_7_l`
Фрейм $\mathcal{E}_2$	`yumi_link_7_r`
K_p (ограничение)	1000
K_d (ограничение)	$2\sqrt{1000} \approx 63$

Таблица 3.2. Параметры эксперимента с Dual-arm YuMi

3.4.5. Результаты

4. РЕЗУЛЬТАТЫ И СРАВНИТЕЛЬНЫЙ АНАЛИЗ

4.1. Метрики оценки

Для сравнительного анализа предлагаемого метода используются следующие метрики:

- Точность выполнения ограничений (максимальная и средняя невязка);
- Качество управления (значение функции стоимости);
- Время вычислений на один шаг управления.

4.2. Сравнение с классическим MPPI

4.3. Сравнение с альтернативными методами

4.4. Анализ чувствительности к параметрам

4.5. Обсуждение результатов

ЗАКЛЮЧЕНИЕ

В данной работе рассмотрена задача управления механическими системами с линейными по ускорениям ограничениями методом MPPI. Предложен метод Gauss-MPPI, интегрирующий принцип наименьшего принуждения Гаусса в процедуру семплирования траекторий.

Основные результаты работы:

1. Проведён анализ существующих методов обработки ограничений в MPPI, выявлен пробел в обработке ограничений равенств на уровне ускорений.

2. Предложен метод Gauss-MPPI, использующий уравнение Удвадия–Калабы для проекции свободного ускорения на множество допустимых ускорений.

3. Выполнена программная реализация классического MPPI в среде MuJoCo.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Kappen H. J. Path integrals and symmetry breaking for optimal control theory // Journal of Statistical Mechanics: Theory and Experiment. 2005. Vol. 2005. No. 11. P. P11011.
2. Theodorou E., Buchli J., Schaal S. A generalized path integral control approach to reinforcement learning // Journal of Machine Learning Research. Vol. 11. 2010. P. 3137–3181.
3. Information theoretic MPC for model-based reinforcement learning / G. Williams, N. Wagener, B. Goldfain et al. // IEEE International Conference on Robotics and Automation (ICRA). 2017. P. 1714–1721.
4. Aggressive driving with model predictive path integral control / G. Williams, P. Drews, B. Goldfain et al. // IEEE International Conference on Robotics and Automation (ICRA). 2017. P. 1433–1440.
5. Ji J., Chen Z., Li Y. Shield-MPPI: Robust and safe model predictive path integral control via control barrier functions // IEEE Robotics and Automation Letters. 2024. To appear.
6. Yin S. et al. GS-MPPI: Generalized smooth MPPI with composite control barrier functions // arXiv preprint. 2024.
7. Zhu W. et al. SC-MPPI: Safe and constrained model predictive path integral control // arXiv preprint. 2023.
8. Nakamura M. et al. BC-MPPI: Barrier-certified model predictive path integral control // arXiv preprint. 2024.
9. Kim C. et al. π-MPPI: Projection-based model predictive path integral control // arXiv preprint. 2024.
10. Udwadia F. E., Kalaba R. E. A new perspective on constrained motion // Proceedings of the Royal Society of London. Series A. 1992. Vol. 439. No. 1906. P. 407–410.
11. Udwadia F. E., Kalaba R. E. Analytical Dynamics: A New Approach. Cambridge University Press. 2002. ISBN: 978-0521048330.
12. Baumgarte J. Stabilization of constraints and integrals of motion in dynamical systems // Computer Methods in Applied Mechanics and Engineering. 1972. Vol. 1. No. 1. P. 1–16.
13. The Pinocchio C++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives / J. Carpentier, G. Saurel,

G. Buondonno et al. // IEEE International Symposium on System Integration (SII). 2019. P. 614–619.
14. Somov I. Coupling robotic systems constrained by a shared object using the Udwadia-Kalaba approach // Bachelor's thesis, Innopolis University. 2025.

ПРИЛОЖЕНИЕ А