

Тема:

# Управление методом Model Predictive Path Integral для систем с линейными по ускорениям ограничениями

*Model Predictive Path Integral Control for Systems with Acceleration-Level Linear Constraints*

## 1. Цель работы

Целью данной работы является разработка и исследование метода управления на основе Model Predictive Path Integral (MPPI), обеспечивающего **точное выполнение линейных по ускорениям ограничений** вида (Пфаффовы ограничения):

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

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

## 2. Задачи исследования

Для достижения поставленной цели необходимо решить следующие задачи:

### 1. Анализ существующих методов

- Провести обзор методов обработки ограничений в MPPI (барьерные функции, штрафные методы, вероятностные подходы).
- Выявить преимущества и недостатки существующих подходов.

### 2. Разработка теоретических основ метода Gauss-MPPI

- Сформулировать задачу проекции свободного ускорения на множество допустимых ускорений как задачу квадратичного программирования (QP).
- Исследовать применимость метода наименьшего принуждения Гаусса для стохастических дифференциальных уравнений.
- Исследовать свойства сходимости модифицированного алгоритма MPPI.

### 3. Разработка алгоритма и его программная реализация

- Разработать алгоритм Gauss-MPPI с интегрированным QP-решателем.
- Реализовать параллельную версию алгоритма для GPU (Pytorch/CUDA).
- Оценить вычислительную сложность и возможность работы в реальном времени.

#### **4. Экспериментальная проверка на модельных задачах**

- Неголономный мобильный робот (ограничение отсутствия проскальзывания).
- Робот-манипулятор с ограничениями в операционном пространстве.
- Система с контактными взаимодействиями.

#### **5. Сравнительный анализ**

- Сравнить предложенный метод с классическим MPPI, CBF-MPPI и штрафными методами.
- Оценить точность выполнения ограничений, качество управления и вычислительные затраты.

### **3. Ожидаемые результаты**

1. Теоретическое обоснование метода Gauss-MPPI для систем с Пфаффовыми ограничениями.
2. Программная реализация алгоритма с поддержкой параллельных вычислений.
3. Экспериментальное подтверждение эффективности метода на задачах робототехники.

### **4. Литературный обзор**

#### **4.1. 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:

$$u^*(x, t) = \mathbb{E} \left[ u(x, t) \cdot \exp \left( -\frac{1}{\lambda} S(\tau) \right) \right], \quad (2)$$

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

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

## 4.2. 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:

$$u^* = \sum_{k=1}^K w_k \cdot u_k, \quad w_k = \frac{\exp(-\frac{1}{\lambda} S_k)}{\sum_{j=1}^K \exp(-\frac{1}{\lambda} S_j)}, \quad (3)$$

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 автономномождении [4], управлении квадрокоптерами и манипуляторами.

## 4.3. Постановка задачи и алгоритм MPPI

MPPI решает задачу стохастического оптимального управления для систем вида:

$$\begin{aligned} dx &= f(x, u)dt + B(x)dw \\ x(t_0) &= x_0 \end{aligned} \quad (4)$$

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  $T$ :

$$J(u) = \phi(x_T) + \int_{t_0}^{t_0+T} \mathcal{L}(x_t, u_t) dt, \quad (5)$$

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

---

**Алгоритм 1** Model Predictive Path Integral (MPPI)

**Вход:**  $x_0$  — начальное состояние,  $\bar{U} = \{\bar{u}_0, \dots, \bar{u}_{N-1}\}$  — номинальное управление

**Выход:**  $u^*$  — оптимальное управление

```
1: loop
2:   for  $k = 1, \dots, K$  do                                ▷ Параллельно на GPU
3:      $x_0^{(k)} \leftarrow x_0$ 
4:     for  $t = 0, \dots, N - 1$  do
5:        $\epsilon_t^{(k)} \sim \mathcal{N}(0, \Sigma)$                       ▷ Выборка возмущений
6:        $u_t^{(k)} \leftarrow \bar{u}_t + \epsilon_t^{(k)}$ 
7:        $x_{t+1}^{(k)} \leftarrow x_t^{(k)} + f(x_t^{(k)}, u_t^{(k)})\Delta t$     ▷ Интегрирование
8:     end for
9:      $S^{(k)} \leftarrow \phi(x_N^{(k)}) + \sum_{t=0}^{N-1} \mathcal{L}(x_t^{(k)}, u_t^{(k)})\Delta t$     ▷ Стоимость
10:    end for
11:    for  $k = 1, \dots, K$  do
12:       $w^{(k)} \leftarrow \exp\left(-\frac{1}{\lambda}S^{(k)}\right) / \sum_{j=1}^K \exp\left(-\frac{1}{\lambda}S^{(j)}\right)$     ▷ Веса
13:    end for
14:    for  $t = 0, \dots, N - 1$  do
15:       $\bar{u}_t \leftarrow \bar{u}_t + \sum_{k=1}^K w^{(k)} \epsilon_t^{(k)}$                                 ▷ Обновление
16:    end for
17:    Применить  $\bar{u}_0$  к системе
18:    Сдвинуть горизонт:  $\bar{U} \leftarrow \{\bar{u}_1, \dots, \bar{u}_{N-1}, \bar{u}_{N-1}\}$ 
19: end loop
```

---

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

#### 4.4. Обработка ограничений в MPPI

Исходная формулировка MPPI не предусматривает явной обработки ограничений. Существующие подходы можно классифицировать 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.**  $\pi$ -MPPI [9] использует проекцию для обеспечения ограничений на величину и производные управления.

#### 4.5. 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:

$$\begin{aligned} \min_{\dot{v}} \quad & [\dot{v} - a]^T M [\dot{v} - a] \\ \text{s.t.} \quad & A\dot{v} = b, \\ & a = M^{-1}Q, \end{aligned} \tag{6}$$

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

#### 4.6. 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:

$$\dot{v} = \dot{v}_{\text{free}} + M^{-1/2}(AM^{-1/2})^+(b - A\dot{v}_{\text{free}}), \tag{7}$$

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:  $J(q)\dot{v} + \dot{J}v = \ddot{x}_d$ .

#### 4.7. 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 представляют **аналитическое решение** для проекции на множество допустимых ускорений.

4. Связь между MPPI и принципом Гаусса не исследована.

## 5. Численное интегрирование стохастических дифференциальных уравнений

Ключевым этапом алгоритма MPPI является моделирование траекторий системы, описываемой стохастическим дифференциальным уравнением (СДУ):

$$dx = f(x, u) dt + B(x) dW, \quad (8)$$

где  $dW$  — приращение Винеровского процесса.

Качество численных методов для СДУ характеризуется двумя типами сходимости:

- **Сильная сходимость** (порядок  $\gamma$ ) — сходимость по траекториям:

$$\mathbb{E} [|x_N - x(T)|^2]^{1/2} \leq C \cdot \Delta t^\gamma. \quad (9)$$

Важна, когда требуется точное воспроизведение отдельных реализаций процесса.

- **Слабая сходимость** (порядок  $\beta$ ) — сходимость по распределениям:

$$|\mathbb{E}[g(x_N)] - \mathbb{E}[g(x(T))]| \leq C \cdot \Delta t^\beta \quad (10)$$

для гладких функций  $g$ . Важна для вычисления математических ожиданий (как в MPPI).

### 5.1. Метод Эйлера–Маруямы

Простейшая схема численного интегрирования СДУ:

$$x_{k+1} = x_k + f(x_k, u_k) \Delta t + B(x_k) \Delta W_k, \quad (11)$$

где  $\Delta W_k \sim \mathcal{N}(0, \Delta t \cdot I)$  — дискретное приращение Винеровского процесса. Метод имеет сильный порядок сходимости 0.5 и слабый порядок 1.0.

### 5.2. Метод Мильштейна

Схема более высокого порядка, учитывающая поправку Ито:

$$x_{k+1} = x_k + f(x_k, u_k) \Delta t + B(x_k) \Delta W_k + \frac{1}{2} B(x_k) \frac{\partial B}{\partial x}(x_k) [(\Delta W_k)^2 - \Delta t]. \quad (12)$$

Метод имеет сильный порядок сходимости 1.0, но требует вычисления производных матрицы диффузии.

### 5.3. Детерминированное приближение

В практических реализациях MPPI часто используется детерминированная модель с аддитивным шумом в управлении:

$$x_{k+1} = x_k + f(x_k, u_k + \epsilon_k) \Delta t, \quad \epsilon_k \sim \mathcal{N}(0, \Sigma), \quad (13)$$

что позволяет применять стандартные методы интегрирования ОДУ (Эйлера, Рунге–Кутты) и упрощает реализацию на GPU.

### 5.4. Симплектические интеграторы

Механические системы естественно описываются в гамильтоновой форме:

$$\dot{q} = \frac{\partial H}{\partial p}, \quad \dot{p} = -\frac{\partial H}{\partial q}, \quad (14)$$

где  $q$  — обобщённые координаты,  $p$  — обобщённые импульсы,  $H(q, p)$  — гамильтониан (полная энергия системы). Такие системы обладают важным свойством: фазовый поток сохраняет симплектическую структуру (объём в фазовом пространстве).

Стандартные методы (Эйлер, Рунге–Кутта) не сохраняют это свойство, что приводит к накоплению ошибки в энергии на длительных интервалах времени. Симплектические интеграторы (Штёрмера–Верле, RATTLE) сохраняют геометрическую структуру фазового потока, обеспечивая:

- Ограниченнность ошибки энергии на бесконечном интервале;
- Качественно верное поведение траекторий;
- Долговременную устойчивость численного решения.

## 6. Сделано на данный момент

### 6.1. Реализация классического MPPI

Выполнена программная реализация классического алгоритма MPPI в среде симуляции MuJoCo. Реализация включает:

- Параллельную выборку траекторий на CPU (медленно);
- Интеграцию с физическим движком MuJoCo для моделирования динамики; Пока проблемно внедрить Винеровский процесс в схему интеграции
- Тестирование на стандартных задачах управления (CartPole, Pendulum и др.) пока без визуализации.

## 6.2. Алгоритм Gauss-MPPI

Приблизительный алгоритм интеграции принципа наименьшего принуждения Гаусса в MPPI для систем с Пфаффовыми ограничениями:

---

### Алгоритм 2 Gauss-MPPI (предварительная версия)

---

**Вход:**  $x_0, \bar{U}$ , ограничения  $A(q, v, t)\dot{v} = b(q, v, t)$

```
1: for  $k = 1, \dots, K$  do
2:    $x_0^{(k)} \leftarrow x_0$ 
3:   for  $t = 0, \dots, N - 1$  do
4:      $\epsilon_t^{(k)} \sim \mathcal{N}(0, \Sigma)$ 
5:      $u_t^{(k)} \leftarrow \bar{u}_t + \epsilon_t^{(k)}$ 
6:      $\dot{v}_{\text{free}} \leftarrow M^{-1}(Q + Bu_t^{(k)})$            ▷ Свободное ускорение
7:      $\dot{v}^{(k)} \leftarrow \text{GAUSSPROJECT}(\dot{v}_{\text{free}}, A, b, M)$       ▷ Проекция
8:      $x_{t+1}^{(k)} \leftarrow \text{INTEGRATE}(x_t^{(k)}, \dot{v}^{(k)}, \Delta t)$ 
9:   end for
10:  Вычислить  $S^{(k)}, w^{(k)}$ 
11: end for
12: Обновить  $\bar{U}$ 
```

---

## 6.3. Интегрирование СДУ

Проведены эксперименты по численному интегрированию физических систем, представленных в виде стохастических дифференциальных уравнений:

- Реализованы методы Эйлера–Маруямы и детерминированное приближение;
- Исследовано влияние шага интегрирования на точность и устойчивость;
- Выявлены особенности работы с механическими системами (проверка сохранения энергии).

## 6.4. Применимость принципа Гаусса к СДУ

Исследуется вопрос корректности применения принципа наименьшего принуждения Гаусса для стохастических систем:

- Классический принцип Гаусса сформулирован для детерминированных систем;
- Открытый вопрос: как корректно определить «свободное ускорение» при наличии стохастического члена  $B(x) dW$ ;
- Рассматриваются подходы: проекция на каждом шаге интегрирования, модификация функции принуждения с учётом шума.

## 7. Дальнейшие шаги

### 7.1. Теоретическое обоснование

1. Формализовать применение принципа Гаусса для стохастических систем: определить условия корректности и границы применимости.
2. Исследовать свойства сходимости алгоритма Gauss-MPPI: показать, что модификация не нарушает сходимость к оптимальному управлению.
3. Проанализировать вычислительную сложность: оценить накладные расходы на решение QP-задачи на каждом шаге.

### 7.2. Программная реализация

1. Реализовать эффективный QP-решатель для проекции по Гауссу, совместимый с GPU (батчевое решение для всех траекторий).
2. Интегрировать Gauss-MPPI с симуляторами MuJoCo или Gazebo.
3. Оптимизировать производительность для работы в реальном времени.

### 7.3. Экспериментальная валидация

1. **Неголономный робот:** колёсный робот с ограничением отсутствия проскальзывания — проверить точность выполнения ограничения  $\dot{x} \sin \theta - \dot{y} \cos \theta = 0$ .
2. **Манипулятор:** робот-манипулятор с ограничениями в операционном пространстве — задача следования по траектории с ограничениями на конечный эффектор.
3. **Контактные задачи:** система с контактными взаимодействиями — проверить работу метода при наличии ограничений трения.

### 7.4. Сравнительный анализ

1. Сравнить Gauss-MPPI с классическим MPPI по метрикам: точность выполнения ограничений, качество управления (стоимость), время вычислений.
2. Сравнить с альтернативными методами: CBF-MPPI, штрафные методы,  $\pi$ -MPPI.
3. Исследовать чувствительность к параметрам: число траекторий  $K$ , температура  $\lambda$ , шаг интегрирования  $\Delta t$ .

## 7.5. Структурирование работы в единый репозиторий

1. Выложить текущие разработки (python notebooks) в github
2. Собрать все теоретические наработки в структурированный документ
3. Начать оформлять работу согласно шаблону магистерской работы

## Список литературы

- [1] Hilbert J. Kappen. Path integrals and symmetry breaking for optimal control theory. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(11):P11011, 2005.
- [2] Evangelos Theodorou, Jonas Buchli, and Stefan Schaal. A generalized path integral control approach to reinforcement learning. In *Journal of Machine Learning Research*, volume 11, pages 3137–3181, 2010.
- [3] Grady Williams, Andrew Aldrich, and Evangelos A. Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 40(4):344–357, 2017.
- [4] Grady Williams, Paul Drews, Brian Goldfain, James M. Rehg, and Evangelos A. Theodorou. Aggressive driving with model predictive path integral control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1433–1440, 2017.
- [5] Manan Gandhi et al. Shield model predictive path integral: A computationally efficient robust mpc method using control barrier functions. *arXiv preprint arXiv:2302.11719*, 2024.
- [6] Jun Liu et al. Guaranteed-safe mppi through composite control barrier functions for efficient sampling in multi-constrained robotic systems. *arXiv preprint arXiv:2410.02154*, 2024.
- [7] Ji Yin, Zhuo Zhang, Evangelos Theodorou, and Panagiotis Tsotras. Safe importance sampling in model predictive path integral control. *arXiv preprint arXiv:2303.03441*, 2023.
- [8] Fernando Castañeda et al. Bc-mppi: A probabilistic constraint layer for safe model-predictive path-integral control. *arXiv preprint arXiv:2510.00272*, 2024.
- [9] Angel Romero, Yunlong Sun, and Davide Scaramuzza.  $\pi$ -mppi: A projection-based model predictive path integral scheme for smooth optimal control of fixed-wing aerial vehicles. *arXiv preprint arXiv:2504.10962*, 2024.
- [10] Firdaus E. Udwadia and Robert E. Kalaba. A new perspective on constrained motion. *Proceedings of the Royal Society A*, 439:407–410, 1992.

- [11] Firdaus E. Udwadia and Robert E. Kalaba. On the foundations of analytical dynamics. *International Journal of Non-Linear Mechanics*, 37:1079–1090, 2002.