

# [F25] Основы управления роботами

Хассан Махмуд

19 декабря 2025 г.

## Аннотация

В данном отчёте представлен сравнительный анализ стратегий управления манипуляторами роботов, работающих в условиях различных неопределённостей. Мы реализуем и оцениваем два подхода к управлению: обратную динамику и скользящее управление. Исследование сосредоточено на манипуляторе UR5e, подверженном неопределённостям, включая изменяющуюся массу полезной нагрузки, трение в сочленениях, демпфирующие эффекты и внешние возмущения. Контроллер обратной динамики демонстрирует отличную производительность при точном знании модели, но значительно ухудшается при наличии неопределённостей. В противоположность этому, скользящий контроллер сохраняет рабочесть производительности, несмотря на неточности модели и возмущения. Мы также анализируем явление «дребезга» (chattering), присущее скользящему управлению, и реализуем модификации граничного слоя для достижения более плавных управляющих сигналов при сохранении рабочести. Экспериментальные результаты подтверждают превосходство скользящего управления для практических робототехнических применений, где неопределённости модели неизбежны.

## Содержание

<b>1 Введение</b>	<b>3</b>
<b>2 Теоретические основы</b>	<b>3</b>
2.1 Динамика робота с неопределённостями . . . . .	3
2.2 Управление обратной динамикой . . . . .	4
2.3 Скользящее управление . . . . .	4
2.3.1 Эквивалентное управление . . . . .	4
2.3.2 Разрывное управление . . . . .	5
2.4 Явление дребезга . . . . .	5
2.5 Модификация граничного слоя . . . . .	5
<b>3 Реализация</b>	<b>5</b>
3.1 Настройка системы . . . . .	5
3.2 Моделирование неопределённостей . . . . .	6
3.3 Параметры контроллеров . . . . .	6
3.4 Структура кода . . . . .	6
3.4.1 InverseDynamicsController . . . . .	6
3.4.2 SlidingModeController . . . . .	7
3.5 Процедура моделирования . . . . .	7

<b>4 Результаты и анализ</b>	<b>7</b>
4.1 Метрики производительности . . . . .	7
4.2 Сравнение качества слежения . . . . .	8
4.3 Статистический анализ . . . . .	8
<b>5 Обсуждение</b>	<b>8</b>
5.1 Ограничения обратной динамики . . . . .	8
5.2 Робастность скользящего управления . . . . .	8
5.3 Явление дребезга . . . . .	9
5.3.1 Причины дребезга . . . . .	9
5.3.2 Практические последствия . . . . .	9
5.4 Компромиссы граничного слоя . . . . .	9
5.5 Рекомендации по выбору параметров . . . . .	10
<b>6 Заключение</b>	<b>10</b>
6.1 Перспективы развития . . . . .	11
<b>A Приложение: Детали реализации</b>	<b>11</b>
A.1 Репозиторий кода . . . . .	11
A.2 Структура файлов . . . . .	12
A.3 Зависимости . . . . .	12
A.4 Запуск кода . . . . .	12

# 1 Введение

Современные манипуляторы роботов всё чаще работают в динамических средах, где они сталкиваются с различными неопределённостями, включая изменяющуюся массу полезной нагрузки, трение и демпфирование в сочленениях, неточности параметров модели и внешние возмущения. Хотя традиционные контроллеры на основе модели, такие как обратная динамика, хорошо работают при точных моделях, их производительность значительно ухудшается в реальных условиях, где точное значение модели недоступно.

Скользящее управление предлагает робастную альтернативу, способную справляться с ограниченными неопределённостями и возмущениями без требования точного знания модели. Ключевой принцип заключается в переводе состояния системы на заранее определённую скользящую поверхность и удержании её там с помощью разрывного закона управления. Этот подход обеспечивает внутреннюю робастность к согласованным неопределённостям, но вводит высокочастотные колебания, известные как «дребезг».

В данном отчёте представлена комплексная реализация и сравнение:

1. Управления обратной динамикой с PD-обратной связью
2. Скользящего управления с модификациями граничного слоя
3. Анализа явления дребезга и эффектов граничного слоя
4. Оценки производительности при различных неопределённостях

Реализация использует модель манипулятора UR5e в среде моделирования MuJoCo, с вычислениями динамики, выполняемыми с использованием библиотеки Pinocchio.

## 2 Теоретические основы

### 2.1 Динамика робота с неопределённостями

Динамика манипулятора робота с  $n$  степенями свободы может быть выражена как:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + D\dot{q} + F_c(\dot{q}) = \tau \quad (1)$$

где:

- $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$  – обобщённые координаты, скорости и ускорения
- $M(q) \in \mathbb{R}^{n \times n}$  – матрица инерции (неопределенная)
- $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$  – матрица кориолисовых и центробежных сил (неопределенная)
- $g(q) \in \mathbb{R}^n$  – вектор гравитационных сил (неопределенный)
- $D \in \mathbb{R}^{n \times n}$  – неизвестная диагональная матрица вязкого трения
- $F_c(\dot{q}) \in \mathbb{R}^n$  – неизвестный вектор кулоновского трения
- $\tau \in \mathbb{R}^n$  – управляющее воздействие

Неопределённости могут быть разложены как:

$$\begin{aligned} M(q) &= M_0(q) + \Delta M(q) \\ C(q, \dot{q}) &= C_0(q, \dot{q}) + \Delta C(q, \dot{q}) \\ g(q) &= g_0(q) + \Delta g(q) \end{aligned} \quad (2)$$

где индекс 0 обозначает номинальные значения, а  $\Delta$  представляет неопределённости.

## 2.2 Управление обратной динамикой

Управление обратной динамикой (также называемое управлением по вычисляемому моменту) вычисляет управляющее воздействие как:

$$\tau = M_0(q)\ddot{q}_r + C_0(q, \dot{q})\dot{q} + g_0(q) \quad (3)$$

где  $\ddot{q}_r$  – эталонное ускорение, заданное как:

$$\ddot{q}_r = \ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q) \quad (4)$$

При точном знании модели ( $M_0 = M$ ,  $C_0 = C$ ,  $g_0 = g$ ) динамика ошибки в замкнутом контуре становится линейной:

$$\ddot{e} + K_d\dot{e} + K_p e = 0 \quad (5)$$

где  $e = q_d - q$ . Однако при наличии неопределённостей модели динамика ошибки становится:

$$\ddot{e} + K_d\dot{e} + K_p e = M_0^{-1}[\Delta M\ddot{q} + \Delta C\dot{q} + \Delta g] \quad (6)$$

## 2.3 Скользящее управление

Скользящее управление определяет скользящую поверхность  $s \in \mathbb{R}^n$ :

$$s = \dot{e} + \Lambda e \quad (7)$$

где  $\Lambda > 0$  – диагональная матрица. Цель управления – привести  $s \rightarrow 0$ .

Закон управления состоит из двух компонентов:

$$\tau = \tau_{eq} + \tau_{dis} \quad (8)$$

где  $\tau_{eq}$  – эквивалентное управление, а  $\tau_{dis}$  – разрывное управление.

### 2.3.1 Эквивалентное управление

Эквивалентное управление поддерживает систему на скользящей поверхности:

$$\tau_{eq} = M_0(q)\ddot{q}_r + C_0(q, \dot{q})\dot{q}_r + g_0(q) \quad (9)$$

где  $\dot{q}_r = \dot{q}_d + \Lambda e$  и  $\ddot{q}_r = \ddot{q}_d + \Lambda \dot{e}$ .

### 2.3.2 Разрывное управление

Разрывное управление компенсирует неопределённости:

$$\tau_{dis} = K \cdot \text{sgn}(s) \quad (10)$$

где  $K$  – коэффициент разрывного управления, удовлетворяющий  $K > \rho$ , причём  $\rho$  – граница неопределённостей.

## 2.4 Явление дребезга

Функция знака в скользящем управлении вызывает высокочастотное переключение, известное как дребезг, которое может:

- Возбуждать непромоделированные высокочастотные динамики
- Вызывать чрезмерный износ механических компонентов
- Требовать высокой полосы пропускания исполнительных механизмов
- Приводить к проблемам устойчивости в цифровых реализациях

## 2.5 Модификация граничного слоя

Для уменьшения дребезга функция знака заменяется функцией насыщения:

$$\text{sat}(s) = \begin{cases} \text{sgn}(s) & \text{если } |s| > \phi \\ \frac{s}{\phi} & \text{если } |s| \leq \phi \end{cases} \quad (11)$$

где  $\phi > 0$  – толщина граничного слоя. Эта модификация создаёт непрерывный закон управления внутри граничного слоя, устраняя дребезг ценой небольшого снижения точности слежения.

# 3 Реализация

## 3.1 Настройка системы

Реализация использует следующий программный стек:

- **MuJoCo 2.3.3+:** Движок физического моделирования
- **Pinocchio:** Эффективная библиотека динамики твёрдого тела
- **Python 3.10:** Язык программирования
- **Робот UR5e:** 6-степенной манипулятор

## 3.2 Моделирование неопределённостей

Были реализованы следующие неопределённости:

Таблица 1: Неопределённости, реализованные в моделировании

Тип неопределённости	Параметр	Значение
Дополнительная масса нагрузки	Масса схвата	4.0 кг
Коэффициенты демпфирования сочленений	$[d_1, d_2, d_3, d_4, d_5, d_6]$	$[0.5, 0.5, 0.5, 0.1, 0.1, 0.1]$ Нм·с/рад
Кулоновское трение	$[f_1, f_2, f_3, f_4, f_5, f_6]$	$[1.5, 0.5, 0.5, 0.1, 0.1, 0.1]$ Нм
Внешние возмущения	Амплитуда	3.0 Нм
	Частота	2.0 Гц

## 3.3 Параметры контроллеров

Таблица 2: Параметры контроллеров

Параметр	Обратная динамика	Скользящее управление
Пропорциональный коэффициент ( $K_p$ )	100	-
Дифференциальный коэффициент ( $K_d$ )	20	-
Параметр скольжения ( $\Lambda$ )	-	10
Коэффициент разрывного управления ( $K$ )	-	50
Границный слой ( $\phi$ )	-	0.1

## 3.4 Структура кода

Реализация состоит из трёх основных классов:

### 3.4.1 InverseDynamicsController

```
class InverseDynamicsController:
    def __init__(self, urdf_path: str):
        #
        self.kp = 100 * np.ones(6)
        self.kd = 20 * np.ones(6)

    def compute_torque(self, q: np.ndarray, dq: np.ndarray):
        #
        PD -
        e = self.q_des - q
        de = self.dq_des - dq
        tau_pd = self.kp * e + self.kd * de

        #
        tau_ff = self.data.nle # C*dq + g

    return tau_ff + tau_pd
```

### 3.4.2 SlidingModeController

```
class SlidingModeController:  
    def compute_torque(self, q: np.ndarray, dq: np.ndarray):  
        #  
        e = self.q_des - q  
        de = self.dq_des - dq  
        s = de + self.lambda_val * e  
  
        #  
        qr_dot = self.dq_des + self.lambda_val * e  
        qr_ddot = self.ddq_des + self.lambda_val * de  
        tau_eq = M @ qr_ddot + nle  
  
        #  
        tau_robust = self.K * self.sat(s)  
  
        return tau_eq + tau_robust
```

## 3.5 Процедура моделирования

Моделирование следует следующим шагам:

1. Инициализация модели робота UR5e с неопределённостями
2. Установка желаемой конфигурации сочленений:  $q_d = [-1.4, -1.3, 1.0, 0, 0, 0]$  рад
3. Запуск контроллера на 10 секунд с частотой 500 Гц
4. Запись положений, скоростей, моментов и ошибок слежения сочленений
5. Анализ метрик производительности

## 4 Результаты и анализ

### 4.1 Метрики производительности

Мы оцениваем производительность контроллера с помощью:

- **СКО ошибки:**  $\sqrt{\frac{1}{T} \int_0^T \|e(t)\|^2 dt}$
- **Максимальная абсолютная ошибка:**  $\max_t |e(t)|$
- **Управляющее воздействие:**  $\frac{1}{T} \int_0^T \|\tau(t)\| dt$
- **Индекс дребезга:**  $\frac{1}{T} \int_0^T \|\dot{\tau}(t)\| dt$

## 4.2 Сравнение качества слежения

Таблица 3: Сравнение производительности контроллеров

Метрика	Обратная динамика	СУ (с ГС)	СУ (без ГС)
Среднее СКО (рад)	0.0862	0.0241	0.0238
Максимальная ошибка (рад)	0.215	0.062	0.061
Средний момент (Нм)	18.42	22.15	23.87
Дисперсия момента ( $\text{Нм}^2$ )	4.32	2.15	15.84
Индекс дребезга	1.24	0.85	8.96

## 4.3 Статистический анализ

Статистическое сравнение показывает:

- СУ с граничным слоем уменьшает СКО на 72% по сравнению с ОД
- Максимальная ошибка уменьшена на 71%
- Дребезг уменьшен на 91% с граничным слоем
- Управляющее воздействие увеличивается на 20% для СУ

## 5 Обсуждение

### 5.1 Ограничения обратной динамики

Контроллер обратной динамики хорошо работает в номинальных условиях, но показывает значительное ухудшение при наличии неопределённостей. Дополнительная масса схвата 4 кг представляет собой существенное изменение параметров (приблизительно 400% увеличение от номинальной массы), что напрямую влияет на матрицу инерции  $M(q)$ . Поскольку ОД зависит от точного знания модели, эта неопределённость приводит к значительным ошибкам слежения.

### 5.2 Робастность скользящего управления

СУ демонстрирует замечательную робастность к реализованным неопределённостям. Разрывный управляющий член  $\tau_{dis}$  эффективно компенсирует:

- Неопределённости матрицы инерции ( $\Delta M$ )
- Неопределённости кориолисовых и центробежных сил ( $\Delta C$ )
- Неопределённости гравитационных сил ( $\Delta g$ )
- Эффекты трения ( $D, F_c$ )
- Внешние возмущения

Ключевое преимущество заключается в том, что СУ требует знания только границ неопределённостей, а не их точных значений.

## 5.3 Явление дребезга

Дребезг, наблюдаемый в СУ без граничного слоя (Рисунок ??), имеет несколько практических последствий:

### 5.3.1 Причины дребезга

1. **Разрывный закон управления:** Функция знака вызывает бесконечную частоту переключения в теории
2. **Задержки переключения:** Цифровая реализация с конечной частотой дискретизации
3. **Непромоделированные динамики:** Высокочастотные структурные моды
4. **Ограничения исполнительных механизмов:** Конечная полоса пропускания и насыщение

### 5.3.2 Практические последствия

- **Износ исполнительных механизмов:** Высокочастотное переключение ускоряет механический износ
- **Потребление энергии:** Неэффективное использование мощности
- **Генерация шума:** Слышимый и электрический шум
- **Проблемы устойчивости:** Может возбуждать непромоделированные динамики

## 5.4 Компромиссы граничного слоя

Модификация граничного слоя вводит непрерывную аппроксимацию вблизи скользящей поверхности, создавая компромисс:

- **Преимущества:**
  - Устраняет дребезг
  - Уменьшает износ исполнительных механизмов
  - Меньшее управляющее воздействие
  - Более простая цифровая реализация
- **Недостатки:**
  - Небольшое снижение точности слежения
  - Потеря сходимости за конечное время
  - Робастность гарантирована только вне граничного слоя
  - Требует тщательной настройки  $\phi$

## 5.5 Рекомендации по выбору параметров

На основе нашего анализа мы рекомендуем:

1. **Параметр скольжения  $\Lambda$ :** Выбирать на основе желаемой скорости сходимости ошибки. Более высокие значения дают более быстрый отклик, но могут требовать большего управляющего воздействия.
2. **Коэффициент разрывного управления  $K$ :** Должен удовлетворять  $K > \rho$ , где  $\rho$  – граница неопределённостей. Консервативные оценки обеспечивают робастность, но увеличивают дребезг.
3. **Границный слой  $\phi$ :** Выбирать на основе требований приложения:
  - $\phi = 0.01 - 0.05$ : Приложения высокой точности
  - $\phi = 0.1$ : Робототехника общего назначения (рекомендуется)
  - $\phi = 0.2 - 0.5$ : Приложения, prioritizing плавность над точностью

## 6 Заключение

В данном отчёте представлена комплексная реализация и анализ робастного скользящего управления манипуляторами роботов в условиях неопределённостей. Основные выводы:

1. **Управление обратной динамикой** обеспечивает отличную производительность слежения при точных моделях, но значительно ухудшается при наличии реальных неопределённостей. С реализованными неопределённостями (4 кг нагрузки, трение в сочленениях, демпфирование) ОД показало на 72% большее СКО по сравнению с СУ.
2. **Скользящее управление** демонстрирует превосходную робастность, сохранив хорошую производительность, несмотря на существенные неопределённости модели. Разрывный управляющий член эффективно компенсирует ограниченные неопределённости без требования точного знания модели.
3. **Дребезг** является значительной практической проблемой в СУ, вызывая высокочастотные колебания момента, которые могут повреждать исполнительные механизмы и возбуждать непромоделированные динамики.
4. **Модификация граничного слоя** эффективно устраняет дребезг ценой небольшого снижения точности слежения. Оптимальная толщина граничного слоя  $\phi = 0.1$  обеспечивает хороший баланс между робастностью и плавностью.
5. **Компромисс производительности:** Граничный слой вводит фундаментальный компромисс между точностью слежения (робастностью) и плавностью управления. Приложения должны выбирать  $\phi$  на основе своих конкретных требований.

## 6.1 Перспективы развития

Будущие расширения этой работы могут включать:

1. **Адаптивное СУ:** Онлайн-оценка границ неопределённостей
2. **СУ высшего порядка:** Уменьшение дребезга при сохранении точного слежения
3. **Компенсация нейронными сетями:** Обучение остаточным неопределённостям
4. **Экспериментальная проверка:** Тестирование на физическом оборудовании робота
5. **Многоцелевая оптимизация:** Выбор граничного слоя по Парето-оптимальности

## Благодарности

Автор выражает благодарность преподавателям курса за их руководство и разработчикам MuJoCo и Pinocchio за предоставление отличных инструментов моделирования.

## Список литературы

### Список литературы

- [1] Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2006). *Robot Modeling and Control*. John Wiley & Sons.
- [2] Slotine, J. J. E., & Li, W. (1991). *Applied Nonlinear Control*. Prentice Hall.
- [3] Utkin, V., Guldner, J., & Shi, J. (2013). *Sliding Mode Control in Electro-Mechanical Systems*. CRC Press.
- [4] Young, K. D., Utkin, V. I., & Ozguner, U. (1999). A control engineer's guide to sliding mode control. *IEEE Transactions on Control Systems Technology*, 7(3), 328-342.
- [5] Edwards, C., & Spurgeon, S. K. (2002). *Sliding Mode Control: Theory and Applications*. CRC Press.
- [6] Khalil, H. K. (2002). *Nonlinear Systems*. Prentice Hall.

## А Приложение: Детали реализации

### A.1 Репозиторий кода

Полная реализация доступна по адресу: <https://github.com/selysecr332/>

## A.2 Структура файлов

```
robust_smc_project/
src/
    sliding_mode_control.py      # Основная реализация
    inverse_dynamics.py         # Контроллер обратной динамики
    utils.py                     # Вспомогательные функции
configs/
    controller_params.yaml     # Конфигурации параметров
logs/
    videos/                      # Записи моделирования
    plots/                       # Сгенерированные графики
    data/                        # Собранные данные
docs/
    report.pdf                  # Этот отчёт
README.md                      # Документация проекта
```

## A.3 Зависимости

```
numpy>=1.21.0
pinocchio>=2.6.0
 mujoco>=2.3.3
matplotlib>=3.5.0
scipy>=1.8.0
```

## A.4 Запуск кода

```
# Установка зависимостей
pip install -r requirements.txt

# Запуск основного анализа
python src/sliding_mode_control.py

# Генерация конкретных графиков
python src/plot_results.py --controller smc --phi 0.1
```