

Робот для работы с динамическими объектами. Игра в пинг-понг.

Студент

Алиусеф Алмасалмах Фади

Руководитель

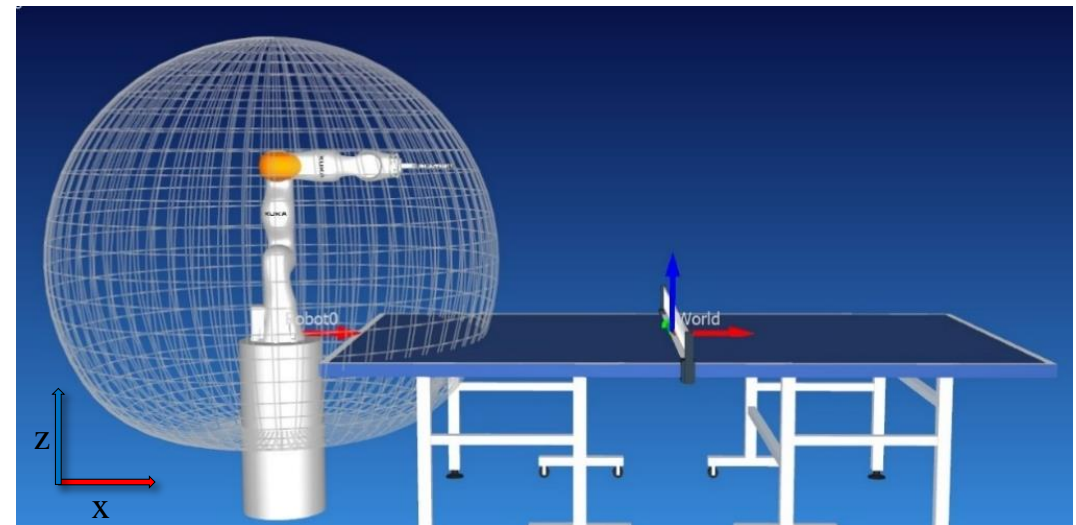
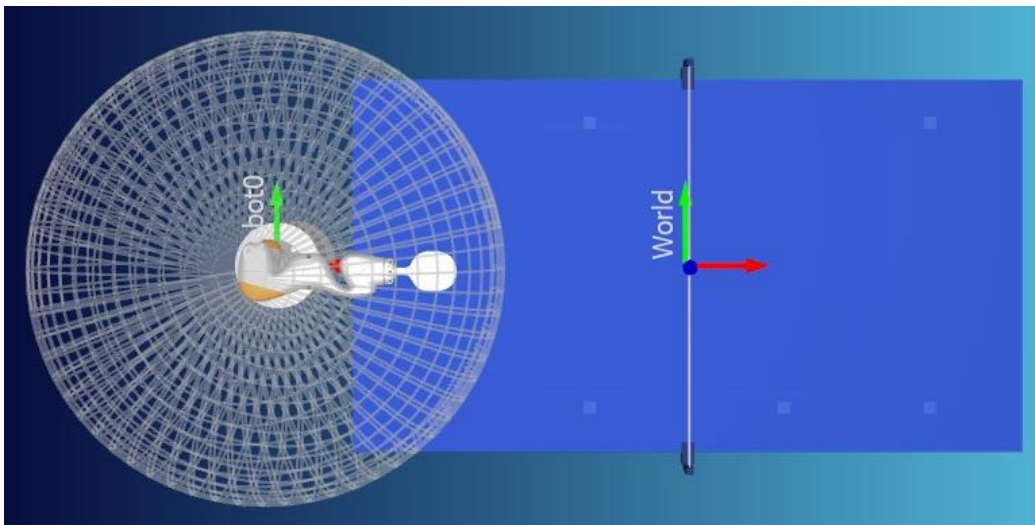
Назарова Анаид Вартановна

Постановка задачи

- **Цель** – построение модели робота, играющего в пинг-понг.
- **Частичные задачи**
 - Построение **модели мяча**.
 - Построение кинематической и динамической **модели робота**.
 - Построение **системы управления** роботом.
 - Разработка **алгоритма** принятия решений.
- Новизна работы заключается в способе выбора параметров удара и проверки безопасности робота в режиме реального времени.

Определение и введение

- Робот KUKA IIWA7 с 7-ю степенями подвижности.
- В настольном-теннисе нужны 5 степеней свободы.
- Робот имеет 2 избыточных степени свободы, мы их используем для оптимизации решения.
- Система должна работать в режиме реального времени.



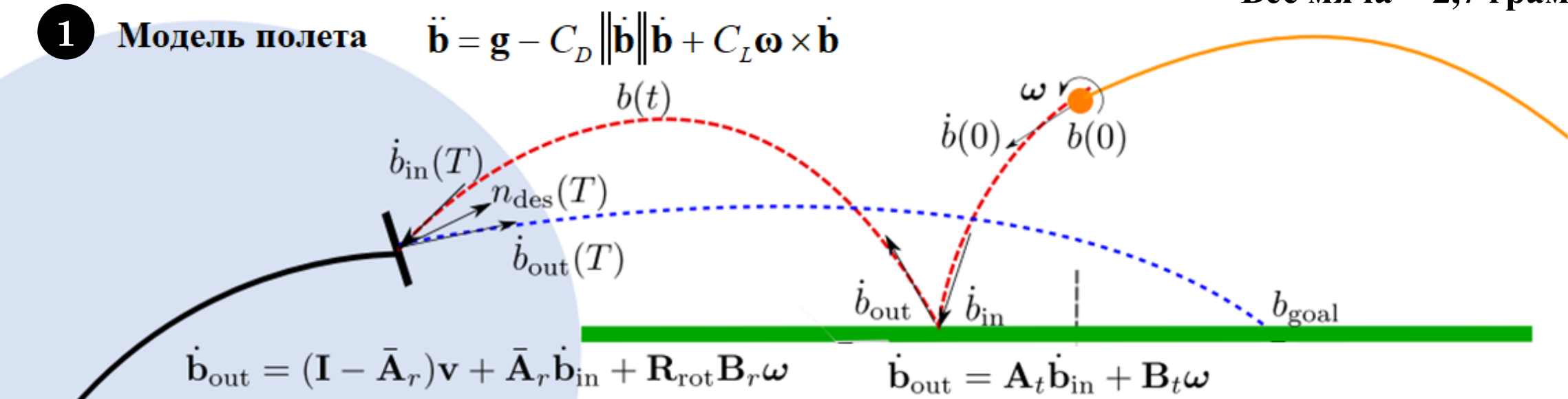
Рабочая зона робота, максимальная достигаемость – 920 мм

Размеры стола – 2750 x 1550 мм

Модель мяча

- Радиус мяча = 2 см
- Вес мяча = 2,7 грамм

1 Модель полета $\ddot{\mathbf{b}} = \mathbf{g} - C_D \|\dot{\mathbf{b}}\| \dot{\mathbf{b}} + C_L \boldsymbol{\omega} \times \dot{\mathbf{b}}$

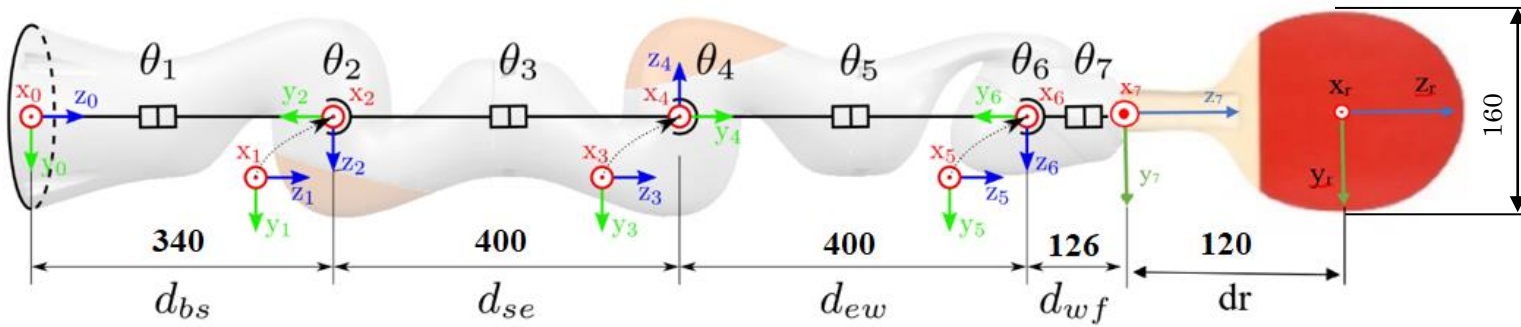


3 Модель удара с ракеткой

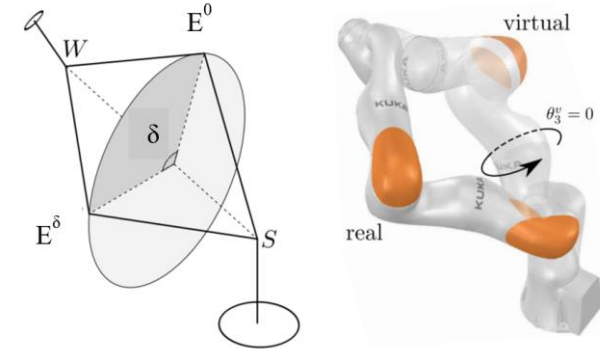
2 Модель столкновения со столом

- Если $\mathbf{b}, \dot{\mathbf{b}}, \boldsymbol{\omega}$ известны, то можно предсказать траекторию мяча.

Обратная кинематическая задача



Угол δ



$$\theta_{\text{limits}} = [\pm 170^\circ, \pm 120^\circ, \pm 170^\circ, \pm 120^\circ, \pm 170^\circ, \pm 120^\circ, \pm 170^\circ]$$

$$\theta_{1\dots 7} = IK \left(\underbrace{\begin{bmatrix} x, y, z \end{bmatrix}}_{\text{Центр ракетки}}, \underbrace{\begin{bmatrix} \alpha, \beta, \gamma \end{bmatrix}}_{\text{Углы Эйлера}}, \delta, \underbrace{\begin{bmatrix} \text{рука, локоть, кисть} \end{bmatrix}}_{\text{Конфигурация}} \right)$$

Время решения:

- В MATLAB $\approx 4.5 \times 10^{-5} \text{ sec}$
- В C++ $\approx 6.1 \times 10^{-6} \text{ sec}$

Динамическая модель

- Уравнение движения:

$$M(\theta) \cdot \ddot{\theta} + \underbrace{V(\theta, \dot{\theta})}_{\text{Вектор моментов Кориолиса и центробежных сил}} + G(\theta) = \tau$$

Матрица
массы

Вектор моментов
Кориолиса и
центробежных
сил

Вектор
моментов
тяжести

Система управления роботом

- Метод «Вычисленные крутящие моменты» - «Computed torque»

$$\tau = M(\theta)\tau' + \beta$$

$$\text{Где } \tau' = \ddot{\theta}_d + K_v \dot{E} + K_p E$$

$$\beta = V(\theta, \dot{\theta}) + G(\theta)$$

- Уравнение ошибки

$$\Rightarrow \ddot{E} + K_v \dot{E} + K_p E = 0$$

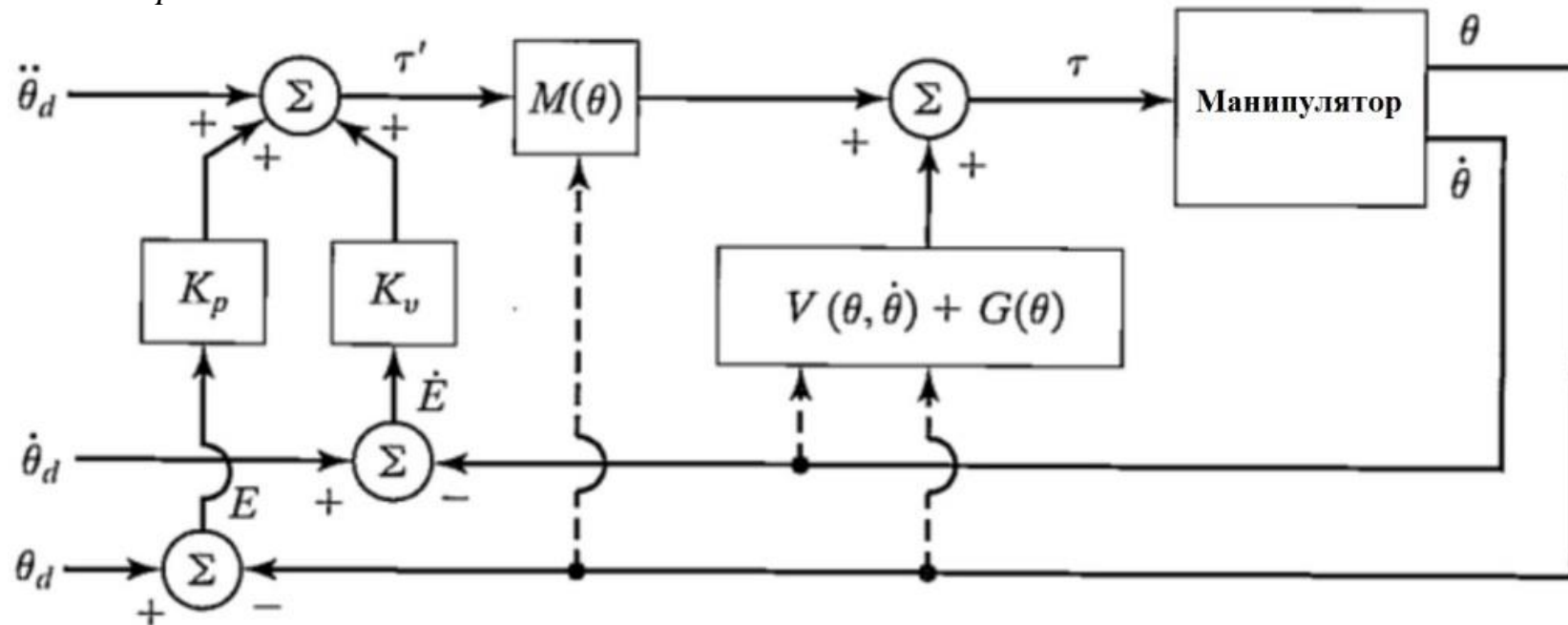
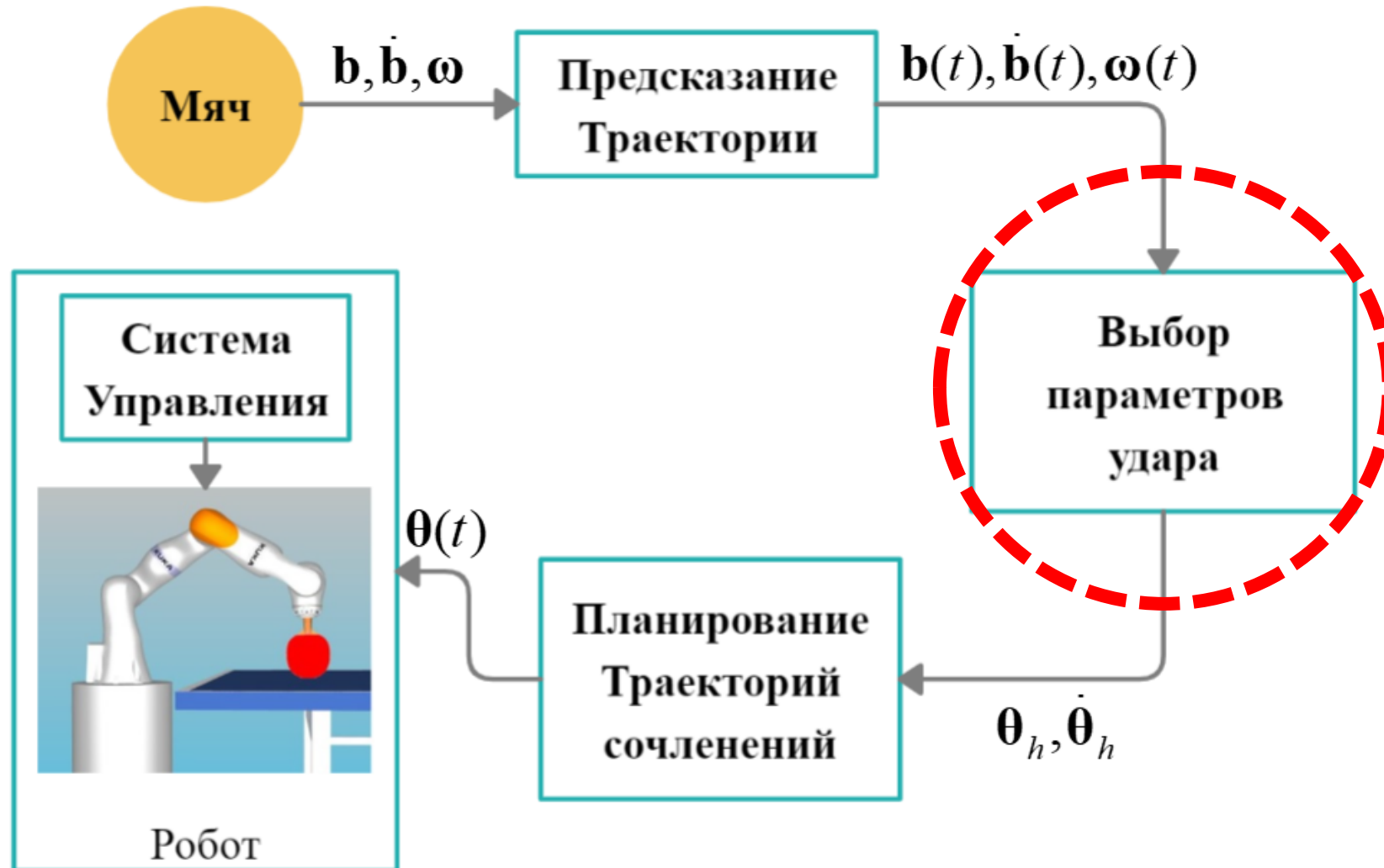
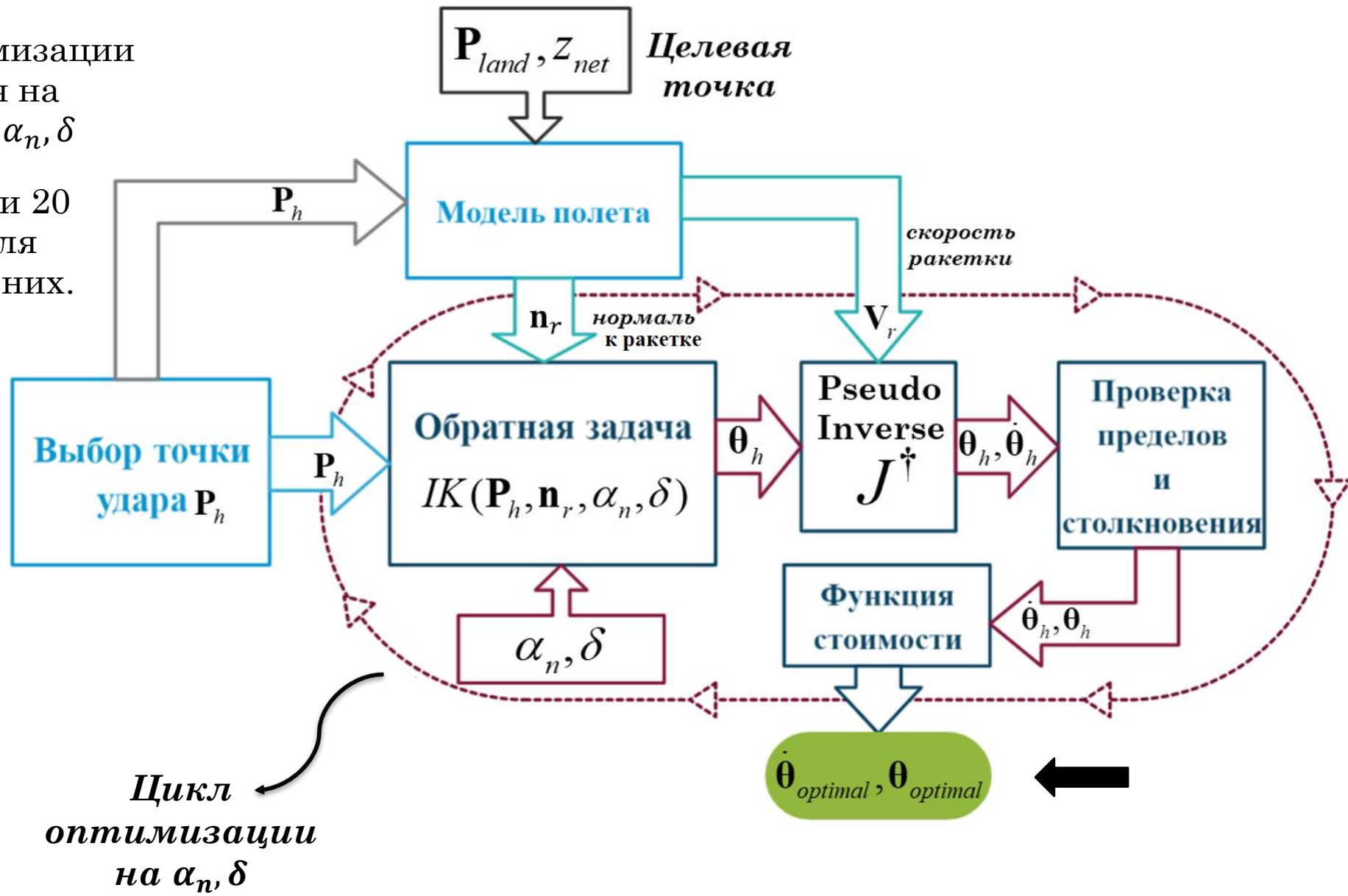


Схема алгоритма удара



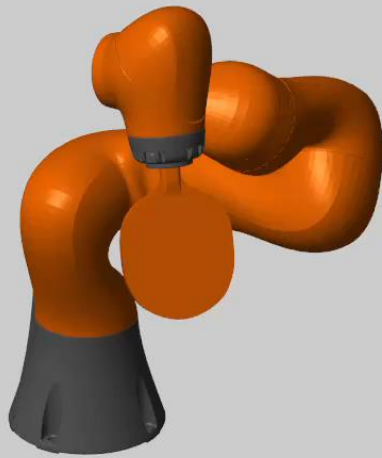
Оптимизация параметров удара

- Цикл оптимизации повторяется на количество α_n, δ
- Мы выбрали 20 значений для каждого из них.

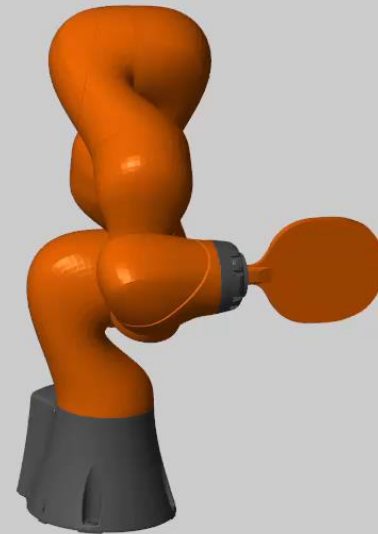


Углы α_n и δ

Угол δ

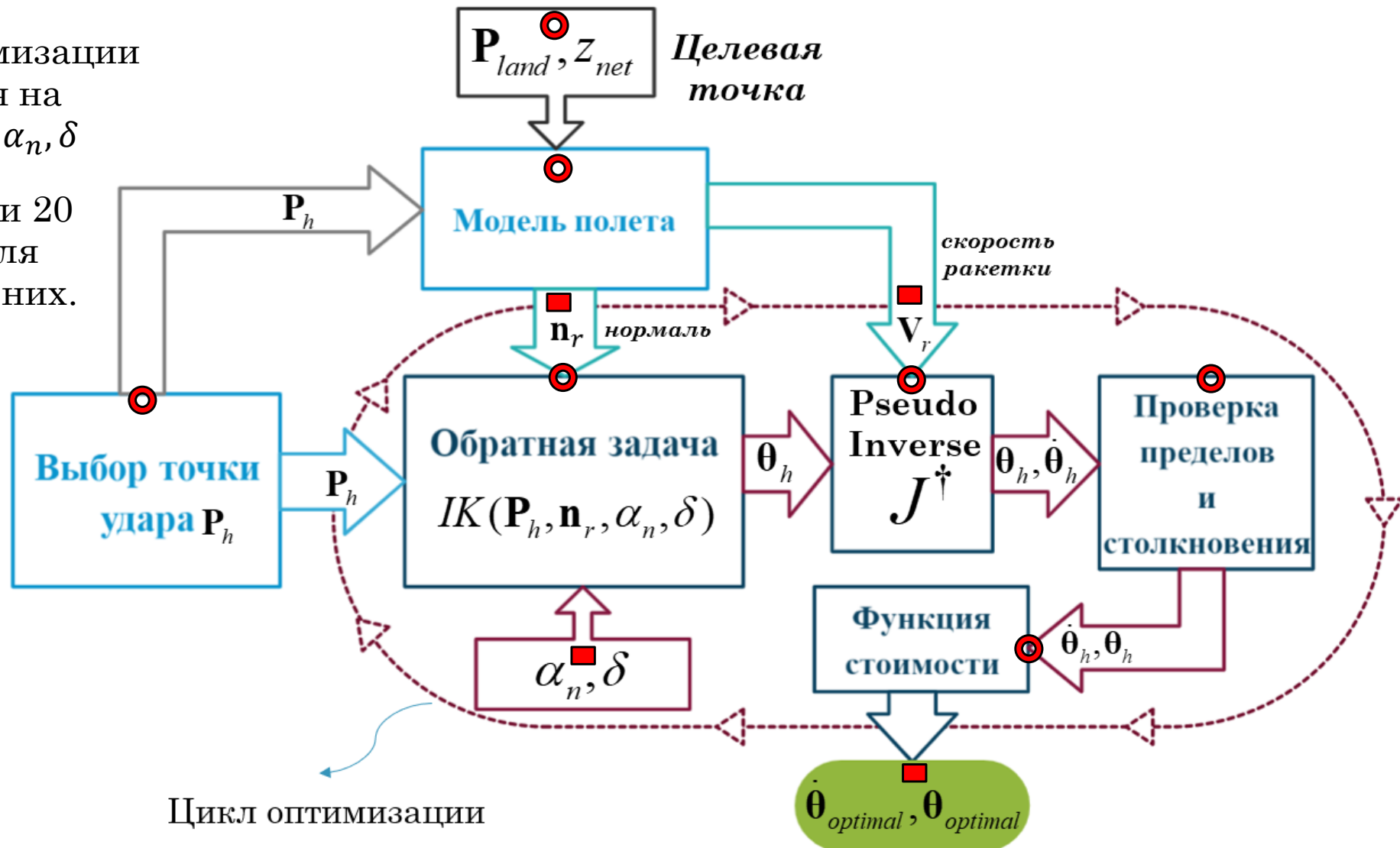


Угол α_n



Оптимизация параметров удара

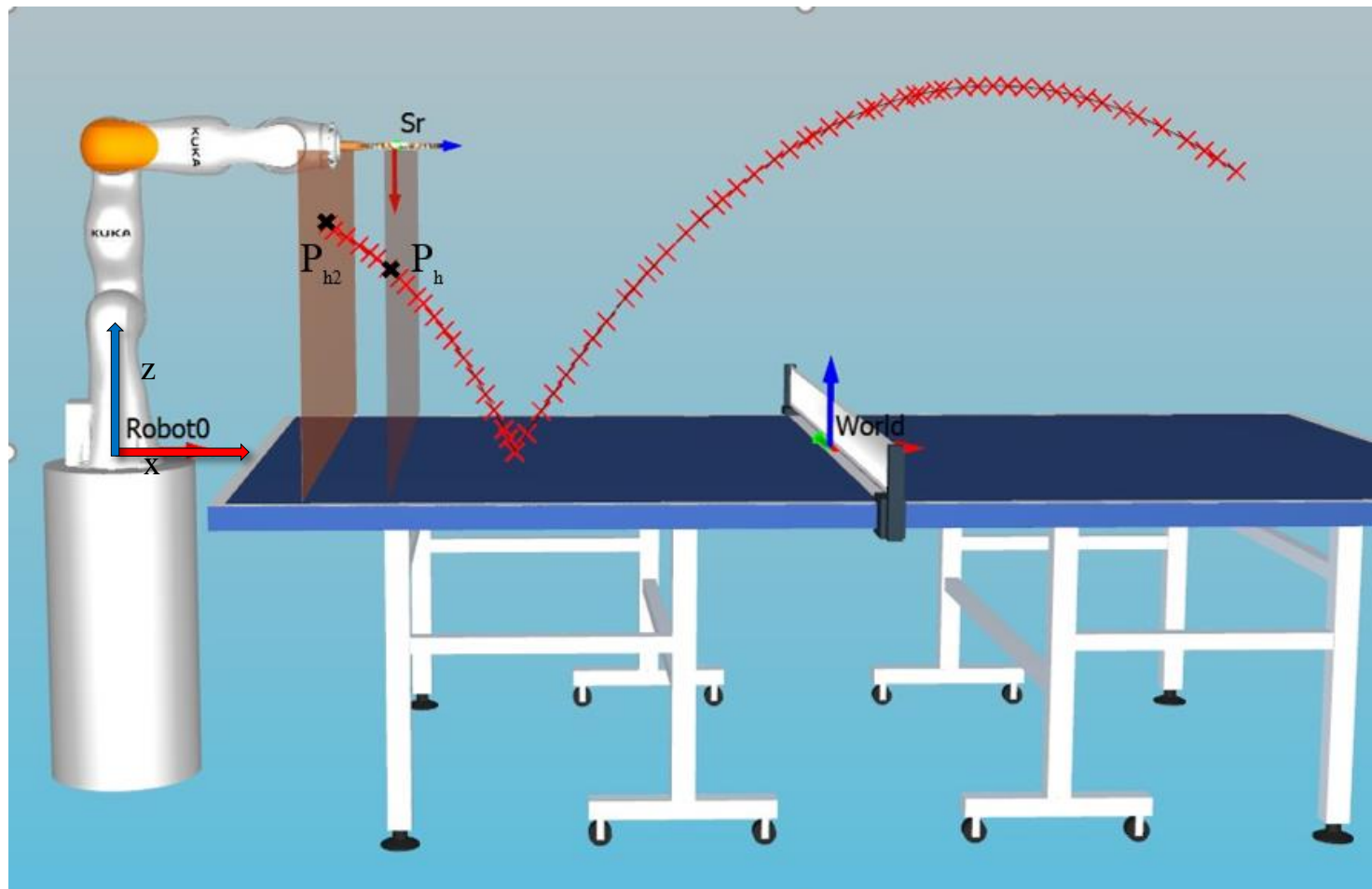
- Цикл оптимизации повторяется на количество α_n, δ
- Мы выбрали 20 значений для каждого из них.



Выбор точки удара

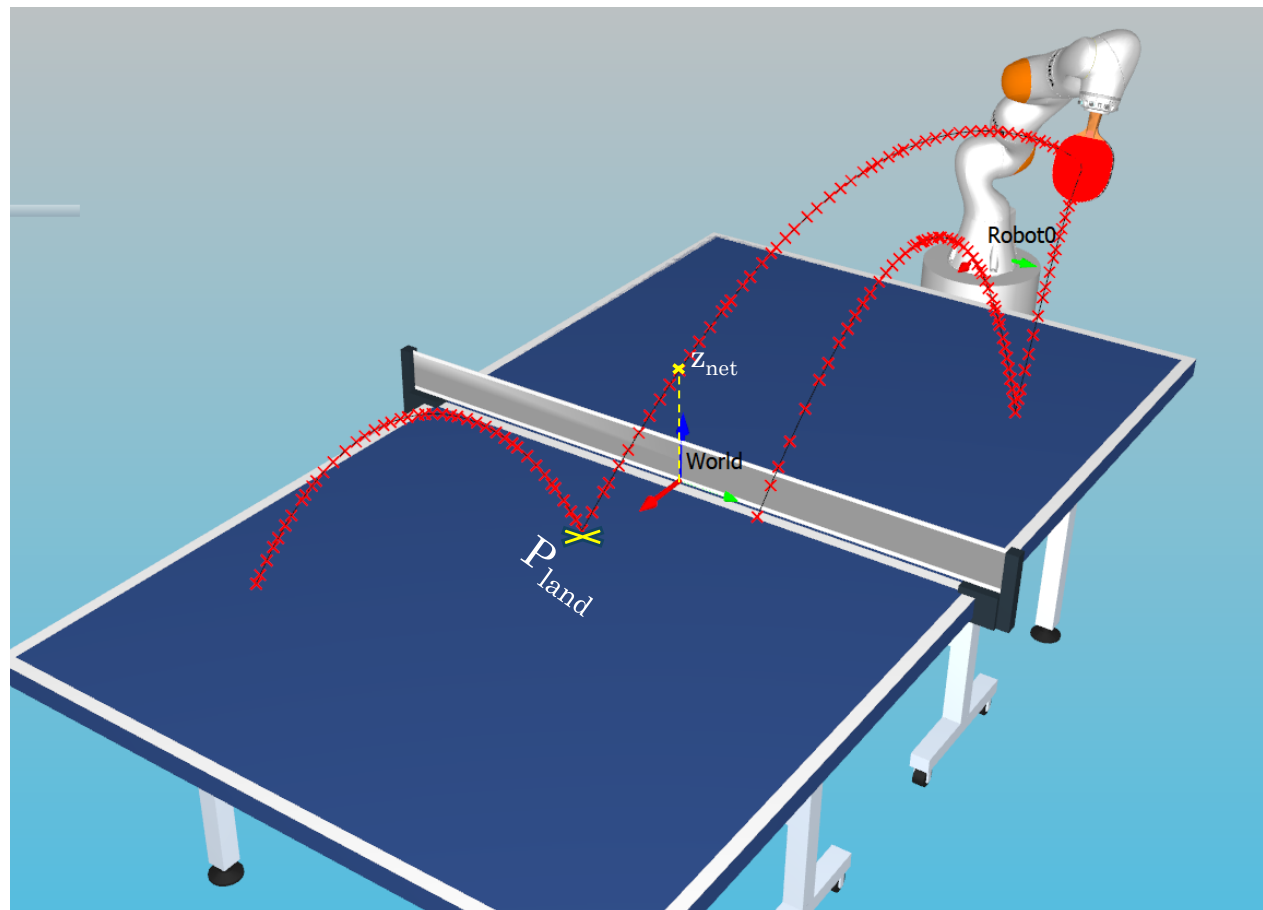
- Выбираем точку пересечения предсказуемой траектории с фиксированной плоскостью:

$$x = \text{const}$$



Выбор целевой точки

- Целевая точка — это фиксированная точка на половине соперника.
- Мяч также должен пролететь над сеткой на определенной высоте z_{net} .



Проверка пределов

- Поскольку у нас траектории 3-ого порядка можно проверять пределы по формулам:

$$\theta_i(t) = at^3 + bt^2 + ct + d$$

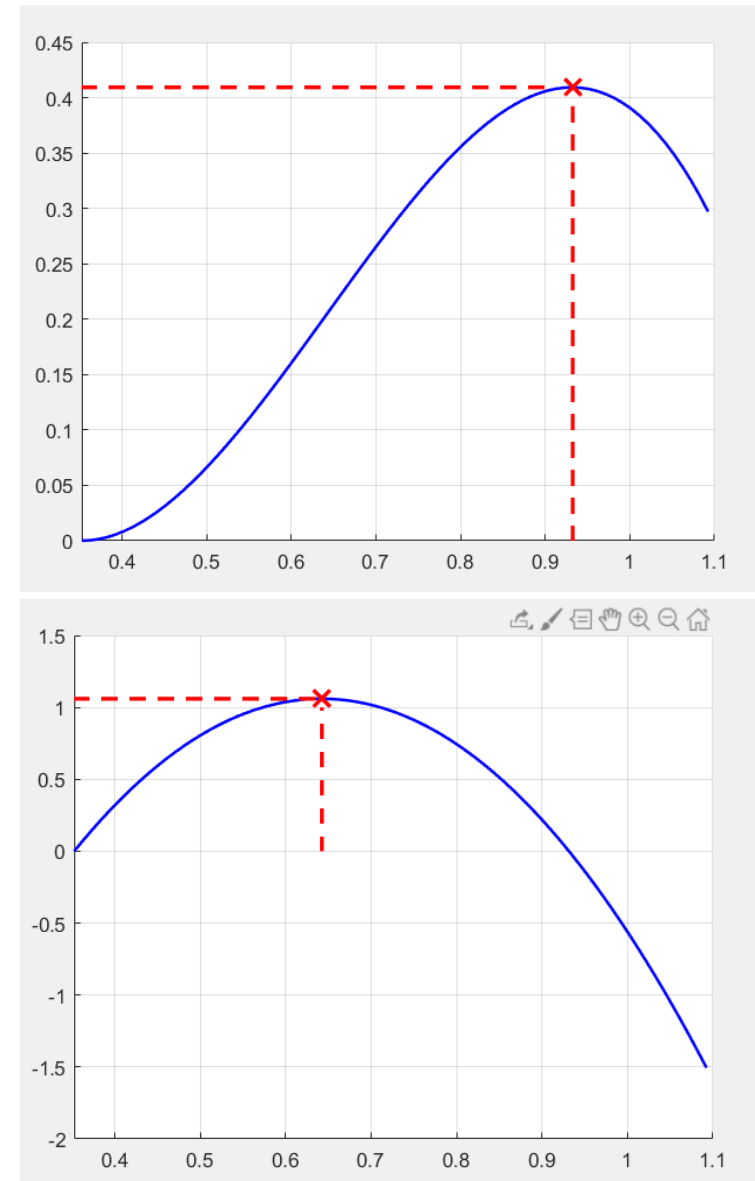
- Проверка пределов углов сочленений**

$$\theta_{i,\max 1} = \left| \frac{27a^2d + 2\sqrt{b^2 - 3ac} + 2b^3 - 9abc}{27a^2} \right| \quad t_{\theta_i \max 1} = \frac{-b - \sqrt{b^2 - 3ac}}{3a}$$
$$\theta_{i,\max 2} = \left| \frac{27a^2d - 2\sqrt{b^2 - 3ac} + 2b^3 - 9abc}{27a^2} \right| \quad t_{\theta_i \max 2} = \frac{-b + \sqrt{b^2 - 3ac}}{3a}$$

- Проверка ограничений скорости**

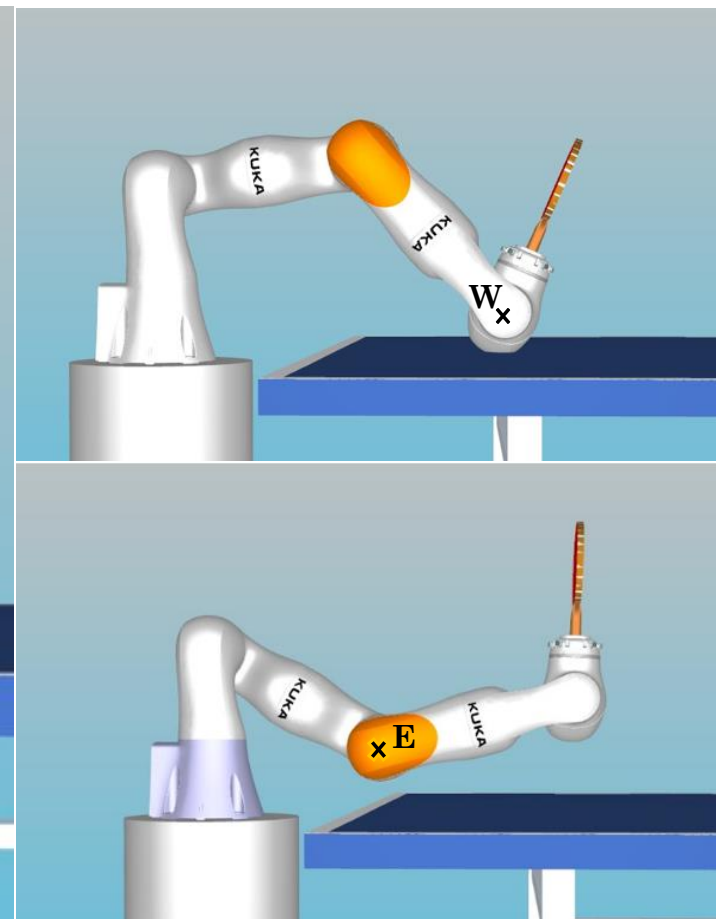
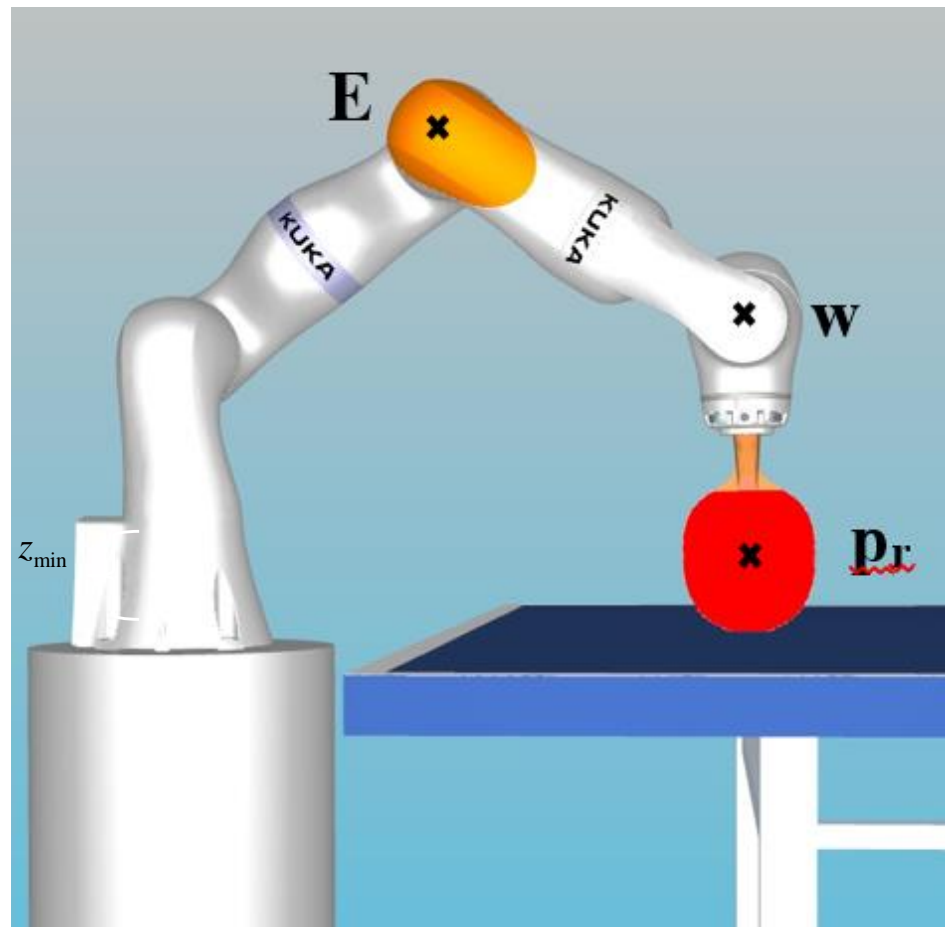
$$\dot{\theta}_{i,\max} = \frac{-b^2}{3a} + c \quad t_{\dot{\theta}_i \max} = \frac{-b}{3a}$$

- Время выполнения в MATLAB $\approx 7 \times 10^{-6} \text{ sec}$



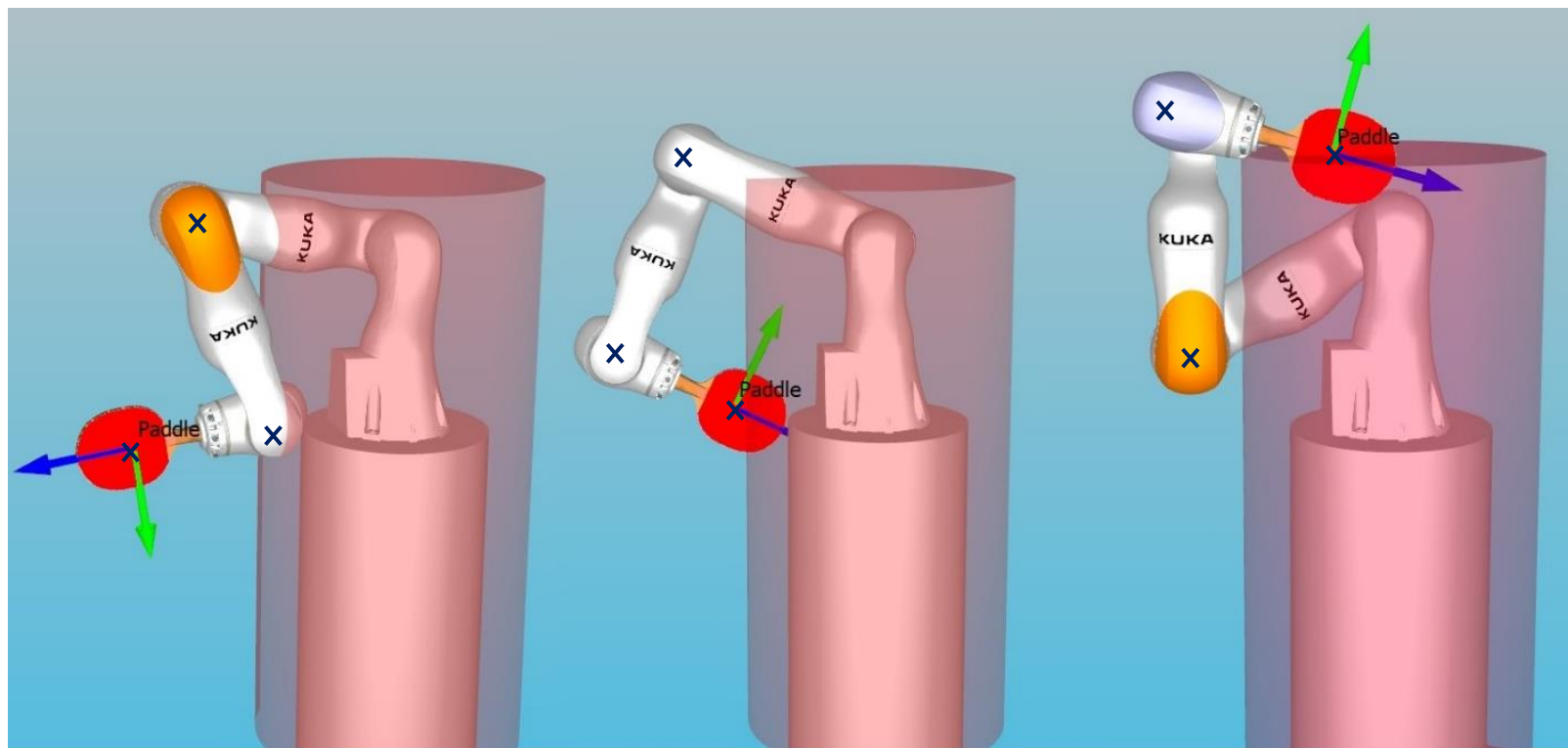
Проверка столкновения с окружающей средой

- Достаточно проверять что 3 точки (E,W,Pr) на расстоянии d_{min} от стола.



Проверка столкновения робота с самим собой

- Достаточно проверять что 3 точки (E,W,Pr) не находятся в воображаемом цилиндре (рис).
- Время проверки столкновения на одну конференцию в MATLAB:
 $\approx 2 \times 10^{-6} \text{ sec}$



Функция стоимости

$$F_{cost}(\boldsymbol{\theta}) = \alpha_1 \cdot C_{\max Qd}(\boldsymbol{\theta}) + \alpha_2 \cdot C_{dist}(\boldsymbol{\theta}) + \alpha_3 \cdot C_{limits}(\boldsymbol{\theta}) + \alpha_4 \cdot C_{manip}(\boldsymbol{\theta})$$

- Минимизировать:
 - Максимальную скорость
 - Расстояние решения от текущей конфигурации
- Максимизировать:
 - *Индекс управляемости (manipulability index)*

$$w = \sqrt{\det(J(\boldsymbol{\theta})J^T(\boldsymbol{\theta}))}$$

- Расстояние от пределов

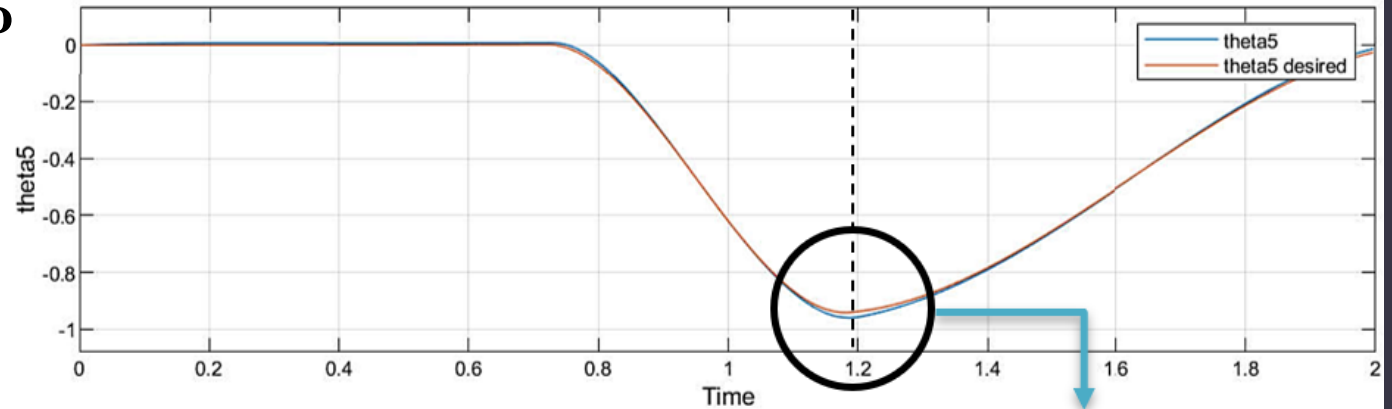
Эксперименты и результаты

- Результаты системы управления роботом, пример из реального удара:
- Траектория 5-го сочленения
- Коэффициенты усиления:

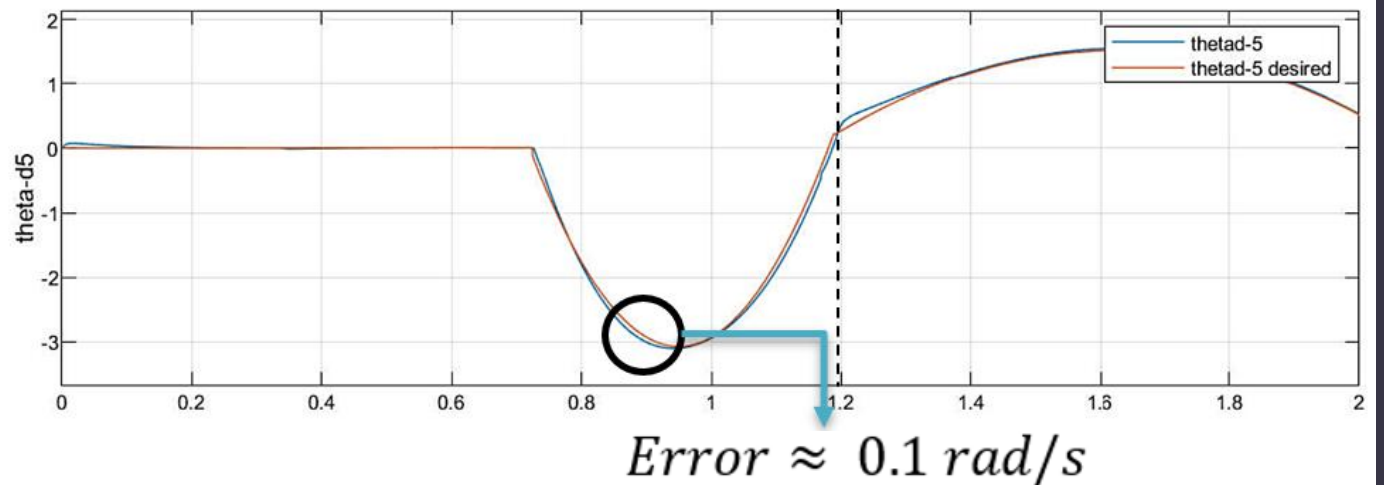
$$k_p = 2500$$

$$k_v = 100$$

Угол θ_5



Скорость $\dot{\theta}_5$



Статистический эксперимент

- Бросали мяч в сторону робота с начальным положением и скоростью:

$$\mathbf{b}_0 \sim N(\mu_p, \sigma_p)$$

$$\dot{\mathbf{b}}_0 \sim N(\mu_v, \sigma_v)$$

$$\mu_p = [2.8, 0, 0.5] \text{ m}$$

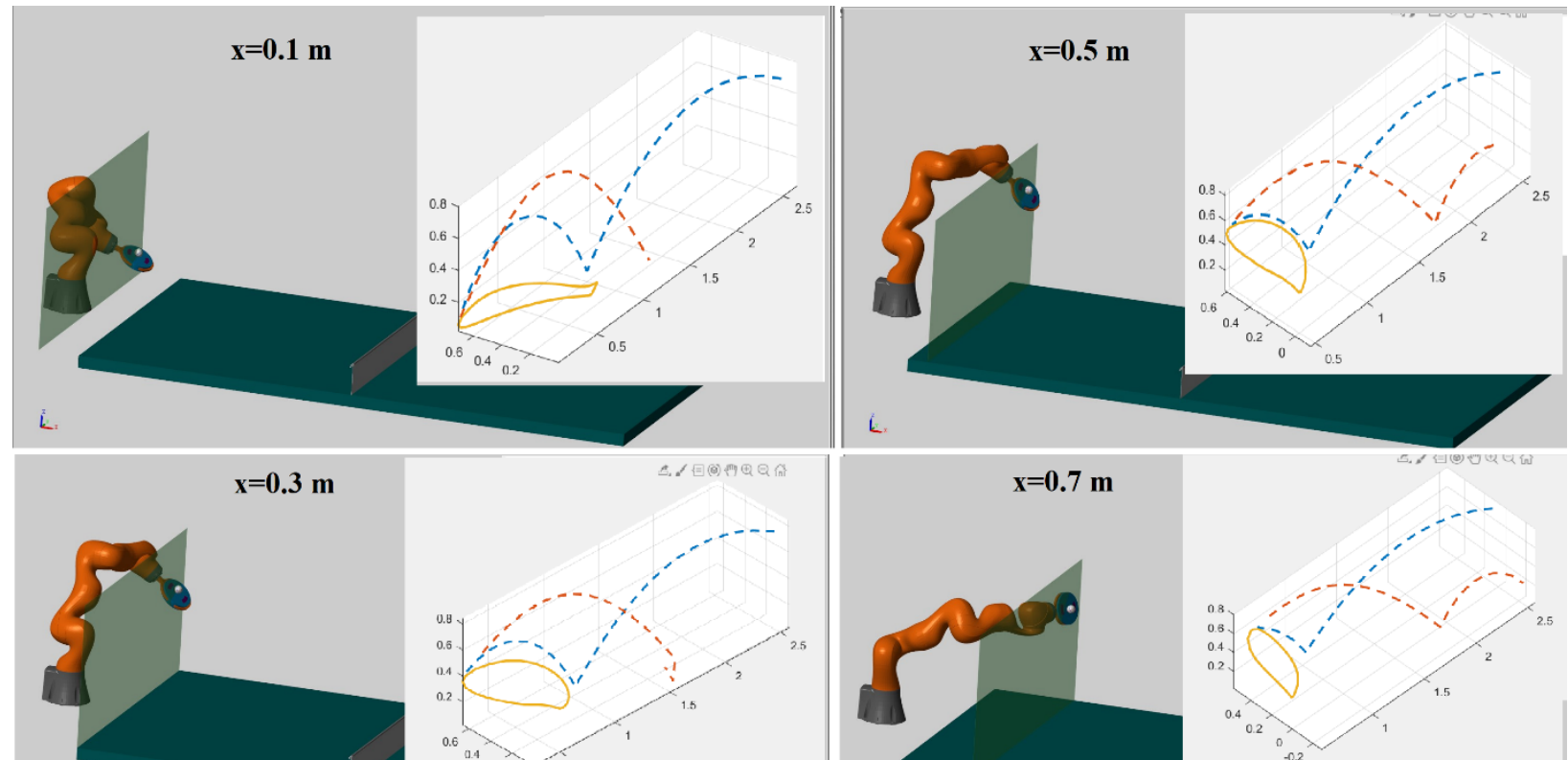
$$\sigma_p = [0.1, 0.25, 0.1] \text{ m}$$

$$\mu_v = [-3, 0, 2.3] \text{ m/s}$$

$$\sigma_v = [0.3, 0.4, 0.3] \text{ m/s}$$

- 500 раундов
- Система выбирает из 7 плоскостей для удара.

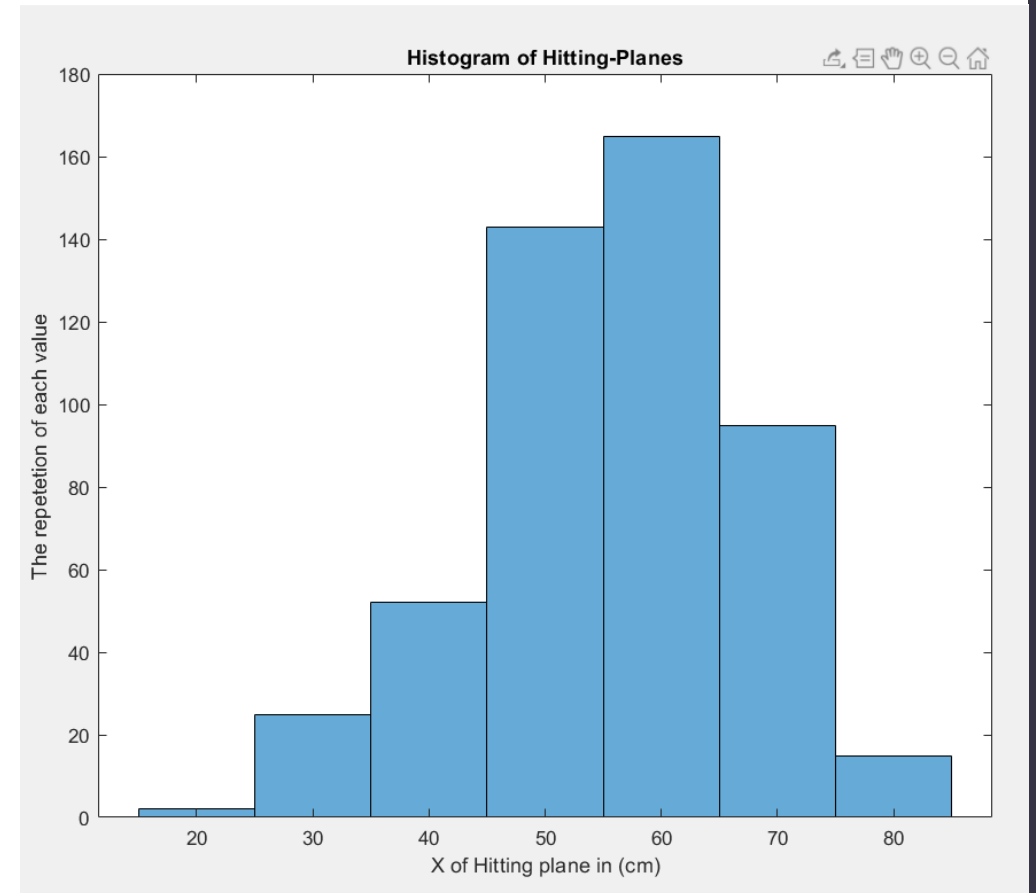
$$x = \text{const}: \text{const} \in \{0.2, \dots, 0.8\} \text{ m}$$



Результаты эксперимента

Из 500 раундов:

<i>Плоскость удара x: (m)</i>	$x \in \{0.2 \dots 0.8\}$
<i>Ударил мяч</i>	97.4%
<i>Успешно вернул мяч</i>	91.5%
<i>Error x_{land} (cm)</i>	18.4
<i>Error y_{land} (cm)</i>	4.1



Обзор литературы и новизна

1. Биомиметический подход к роботу настольного тенниса (2011)

- *Обратная задача* кинематики решена *численными методами*.
- минимизация расстояния до комфортной позиции в пространстве обобщенных координат

2. оптимальная генерация траектории для робота настольного тенниса (2018)

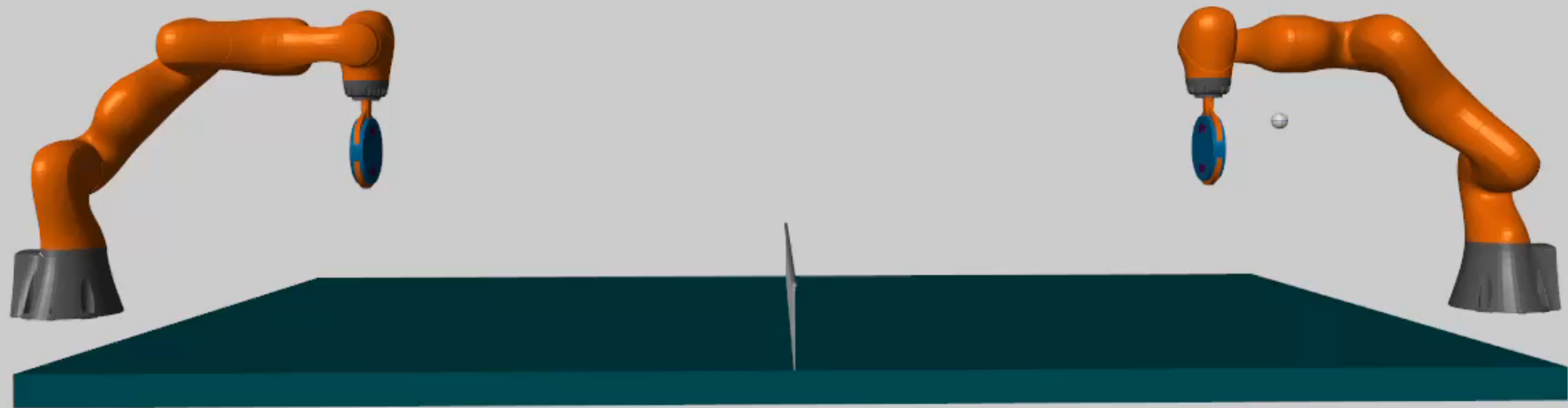
- Задача *оптимизации квадратной суммы ускорений*.
- *Не* может быть выполнена в режиме *реального времени*, поэтому они использовали справочную таблицу.

• Наша работа:

- Разработан метод для оптимизации скоростей.
- Обратная задача решена алгебро-геометрическим методом.
- Наш метод имеет фиксированное время выполнения каждый раз.
- Мы использовали *Индекс управляемости (manipulability index)*.
- Предложили метод проверки пределов и столкновения в реальном времени.
- Все наши модели учитывают вращение.

Выводы

- Построена **модель движения мяча**, и также модели столкновения.
- Построена **Кинематическая и динамическая модели робота**, и решена обратная задача.
- Построена **Система управления** роботом с методом «Вычисленные моменты».
- Система смоделирована в **MATLAB**.
- Разработан **алгоритм** для выбора **оптимальных** углов и скоростей в момент удара.
- Разработан **алгоритм** для **проверки пределов** углов и скоростей сочленений, и также проверки **столкновения** в режиме реального времени.
- В **статистическом эксперименте** с 500 раундов система смогла **ударить по мячу в 97,4% случаев**, и она **успешно вернула 91,5%** мячей на сторону соперника.



Спасибо за внимание

Список статей

1 - Построение 2D-модели робота, играющего в пинг-понг

«Политехнический молодежный журнал» МГТУ им. Н.Э. Баумана

DOI: 10.18698/2541-8009-2020-3-589

<http://ptsj.ru/catalog/menms/robots/589.html>

2 - Построение системы нелинейного динамического управления манипулятором с 7-степенями подвижности

- Ещё не опубликована, сдано в апреле 2020 на конференцию «Моделирование в инженерном деле»