

Разработка программного комплекса для решения задачи планирования траектории при управлении движением манипуляционного механизма

М. О. Чайкин

Аннотация — Планирование траектории движения выходных звеньев манипуляционных механизмов является одной из важнейших задач при практическом применении роботов в различных областях техники. В данной работе описывается разработка программного комплекса для автоматизации процесса задания траектории с условием обеспечения точности её выполнения. Разработанный комплекс позволяет формировать траекторию по различным исходным данным, решать прямую и обратную задачу кинематики, а также выполнять процедуру интерполяции с целью получения законов изменения обобщённых координат манипуляционного механизма. Интерфейс для взаимодействия с человеком-оператором представлен WEB-клиентом, написанным на Vue JS. Взаимодействие с базой данных PostgreSQL для хранения информации осуществляется посредством REST API, реализованного на языке программирования Go. Для обеспечения прохождения по заданной траектории был рассмотрен алгоритм управления, основанный на обратных задачах динамики. Расчёт закона управления осуществлялся с использованием микроконтроллеров семейства STM32 для двигателей постоянного тока с датчиком Холла для обратной связи по положению и скорости. Передача данных между WEB-клиентом и манипуляционным механизмом осуществляется при помощи микроконтроллера ESP32, на котором реализован websocket, а также REST API для задания команд управления. Результаты работы комплекса планирования траектории демонстрируются на примере совместного относительного манипулирования механизма с пятью степенями свободы.

Ключевые слова — планирование траектории, управление манипуляционным механизмом, REST API, Golang, PostgreSQL, Vue JS, ESP32, STM32

I. ВВЕДЕНИЕ

Практическое применение промышленных роботов невозможно без решения задачи формирования траектории для точного позиционирования выходного звена манипуляционного механизма при выполнении различных операций. Основными критериями, которые накладываются на планируемую траекторию

исполнительного механизма, являются обеспечение её плавности и непрерывности, а также соответствие допустимым кинематическим и динамическим характеристикам робота [1]. При этом прохождение схвата манипулятора по заданной траектории осуществляется при помощи синтеза алгоритма управления, направленного на достижение точного положения, оптимизацию времени [2, 3], энергопотребления [4] или нивелирования риска столкновения схвата робота с объектами окружающей среды [5, 6]. Планирование траектории при управлении манипуляционным механизмом затрагивает множество аспектов проектирования роботов и является комплексной задачей, требующей соответствующей автоматизации.

В рамках работы предлагается разработка программного комплекса, направленного на упрощение и автоматизацию процесса задания траектории в зависимости от выбранного типа манипулятора. Такое решение позволяет эффективно осуществлять планирование траектории с точки зрения человека-оператора, задача которого указывать только конечные цели выходного звена манипулятора, а не заниматься непосредственным программированием робота. Для фактического прохождения по заданной траектории рассматривается метод управления, основанный на решении обратной задачи динамики для достижения минимальной энергии ускорения [7]. Структура статьи построена следующим образом: в разделе II рассматривается процедура планирования траектории; в разделе III определяются основные функциональные требования, накладываемые на разрабатываемый комплекс; в разделе IV описывается предлагаемая архитектура комплекса; в разделе V приводится синтез алгоритма управления манипуляционным механизмом; в разделе VI демонстрируются результаты работы комплекса.

II. ПРОЦЕДУРА ПЛАНИРОВАНИЯ ТРАЕКТОРИИ

Задача планирования траектории подразделяется на две подзадачи: формирование траектории в декартовом пространстве для описания положения и ориентации выходного звена относительно базовой системы координат манипулятора, а также формирование траектории в пространстве обобщённых координат с

Статья получена 5 декабря 2023.

Чайкин Максим Олегович, МГТУ им. Н.Э. Баумана, аспирант кафедры РК2 «Теория машин и механизмов» (e-mail: чайкин.maksim@yandex.ru)

целью определения относительного положения кинематических пар [8]. Решение первой подзадачи даёт полную визуализацию рассматриваемой траектории в рабочем пространстве манипулятора, а второй взаимное расположение звеньев манипулятора в каждой точке траектории. Для получения наборов декартовых и обобщённых координат при генерации траектории необходимо решение прямой и обратной задач кинематики в зависимости от вводимых данных.

Примем, что формируемая траектория движения выходного звена манипуляционного механизма зависит от вида траектории, времени прохождения T и количества опорных точек N . Определим три способа задания траектории:

- В параметрической форме: путём ввода законов изменения декартовых координат и углов ориентации выходного звена механизма как функции от времени. В случае ввода траектории в параметрической форме происходит определение значений декартовых координат для $n, i = 1 \dots N$. Затем решается обратная задача кинематики.
- По декартовым координатам: пользователь вводит последовательность из N декартовых координат, которые служат входными значениями для решения обратной задачи кинематики с целью получения обобщённых координат;
- По обобщённым координатам: пользователь вводит последовательность из N обобщённых координат, по которым находятся декартовые координаты посредством решения прямой задачи кинематики.

После задания входных значений и решения задач кинематики необходимо провести валидацию полученных значений на её допустимость, чтобы

удостовериться, что планируемая траектория лежит в рабочем пространстве манипулятора и не приводит к положениям звеньев манипулятора, при которых может нарушиться его целостность.

Плавное перемещение выходного звена из начального положения в конечное возможно только с учётом обхода множества промежуточных положений, в которых необходимо восстановить параметры траектории (положение, скорость, ускорение и др.) [9]. В робототехнике для решения данных задач применяют различные методы интерполяции или аппроксимации. В случае использования методов интерполяции, получаемая кривая траектории проходит через опорные точки в заданный момент времени. При аппроксимации траектории кривая не проходит точно через опорные точки, а только приближённо соответствуют вводимым данным. Остановимся на рассмотрении методов интерполяции.

Главная цель применения интерполяции заключается в формировании закона изменения обобщённых координат от времени, который является входным значением для регуляторов приводов кинематических пар механизма. В зависимости от типа многочлена, используемого для приближения траектории, выделяют линейную и полиномиальную интерполяцию [10], а также интерполяцию сплайнами [11]. Первые два метода являются наиболее простыми с точки зрения вычисления, однако не позволяют определять скорость и ускорение в краевых точках отрезка посредством дифференцирования. Интерполяция сплайнами же обеспечивает большую гибкость задания траектории, так как обеспечивает непрерывность первых и вторых производных, что позволяет накладывать на траекторию дополнительные условия на скорость, ускорение и т.д.

Функциональные этапы процедуры планирования траектории показаны на рисунке 1.

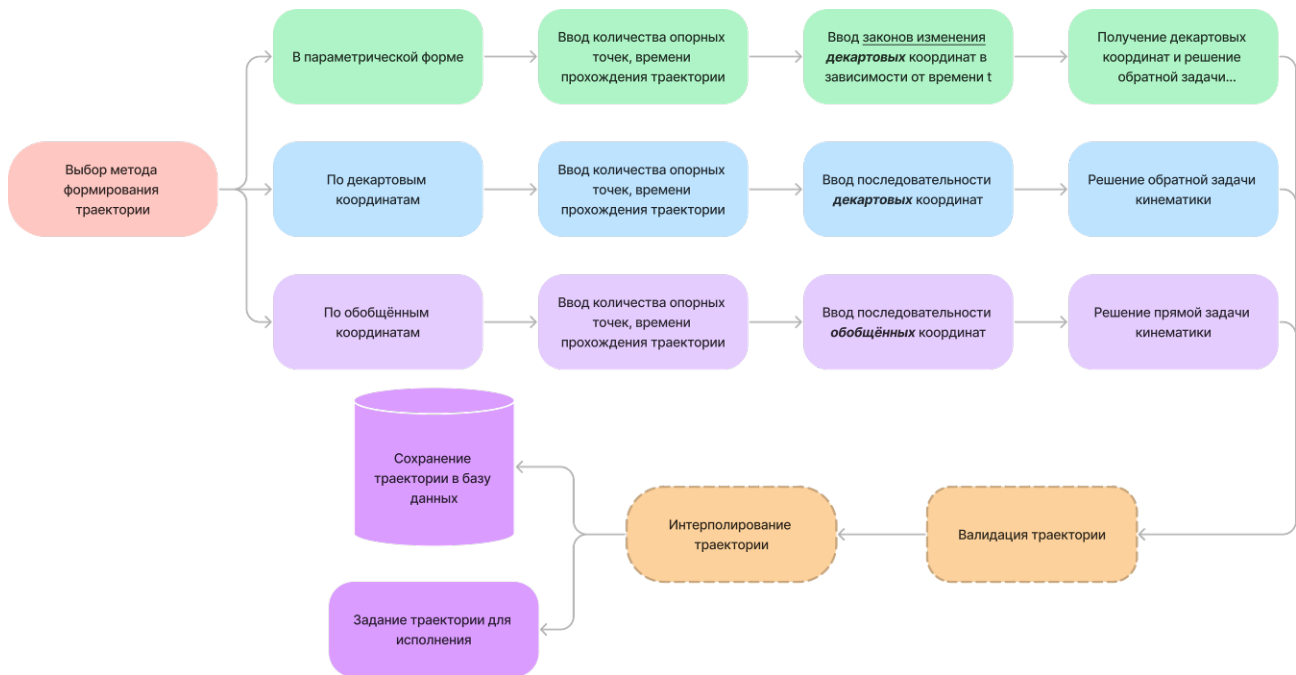


Рис. 1 – Процедура формирования траектории

III. ОПРЕДЕЛЕНИЕ ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ

Разрабатываемый комплекс планирования траектории предназначен для формирования траектории по заданным значениям с учётом решения прямой и обратной задач кинематики с последующей интерполяцией для получения законов изменения обобщённых координат. Следовательно, были выделены следующие основные функциональные требования:

1. Выбор манипуляционного механизма для проведения моделирования и задания траектории;
2. Возможность просмотра кинематических конфигураций механизмов;
3. Задание, редактирование и установление актуальной кинематической конфигурации манипуляционных механизмов;
4. Задание траектории в параметрической форме как зависимость декартовых координат от времени, а также

по наборам декартовых или обобщённых координат;

5. Решение прямой и обратной задачи кинематики;
6. Осуществление интерполяции введённой траектории;
7. Сохранение промоделированной траектории в базу данных;
8. Возможность повторной визуализации промоделированной ранее траектории;
9. Возможность задания промоделированной траектории для исполнения манипуляционным механизмом;
10. Сохранение и просмотр истории наблюдений за работой механизма.

Диаграмма вариантов использования при выбранном манипуляционном механизме приведена на рисунке 2.

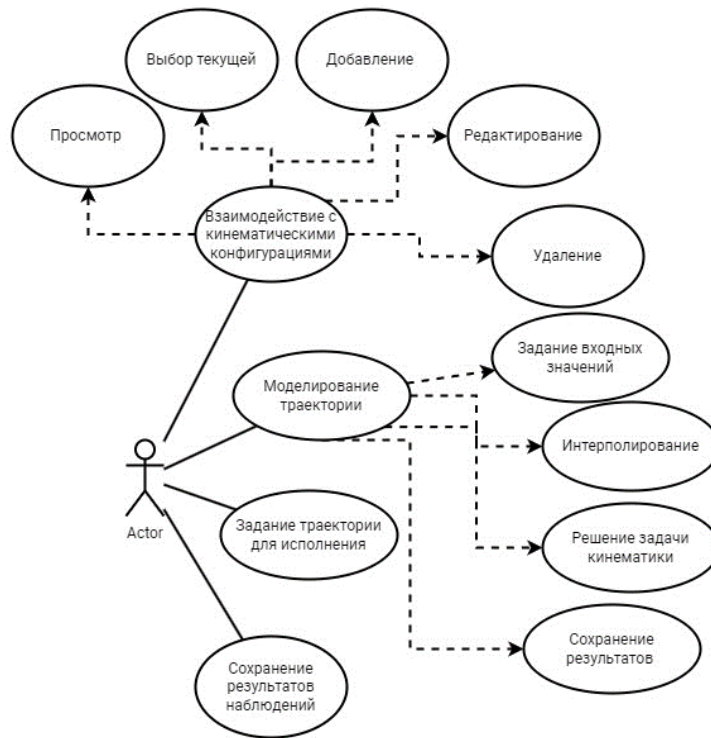


Рис. 2 – Диаграмма вариантов использования

IV. РЕАЛИЗАЦИЯ КОМПЛЕКСА ПЛАНИРОВАНИЯ ТРАЕКТОРИИ

А. Предлагаемая архитектура

Предлагаемый комплекс планирования траектории включает в себя WEB приложение для взаимодействия с человеком-оператором и программу управления исполнительным механизмом, который объединяет аппаратную часть, представленную микроконтроллерами, датчиками обратной связи, приводами и драйверами для реализации заданного алгоритма управления, а также механическую часть для непосредственного осуществления перемещения выходного звена (Рис. 3).

WEB-клиент написан с использованием фреймворка Vue JS, библиотеки компонентов Vuetify и менеджера состояний Vuex. Выбор данного решения для frontend части приложения обеспечило возможность быстрого прототипирования UI с использованием преимуществ реактивности для отображения и обновления данных. При этом по сравнению с другими JavaScript фреймворками Vue обладает большей скоростью

рендеринга компонентов и имеет более простой синтаксис, что ускоряет процесс написания кода [12]. Взаимодействие с базой данных осуществляется посредством REST API, написанного на языке программирования Go с использованием фреймворка Gin [13].

Передача информации между WEB-клиентом и уровнем управления манипуляционным механизмом осуществляется посредством websocket для передачи параметров работы в режиме реального времени, а также REST API для задания управляющих команд. Данный уровень можно реализовать с использованием любого микроконтроллера, имеющего Wi-Fi модуль. В рамках данной работы был использован микроконтроллер ESP32, поскольку он имеет встроенный Wi-Fi модуль, несколько UART и SPI интерфейсов, широкий выбор сред разработки, а также относительно низкую стоимость по сравнению с аналогами [14].

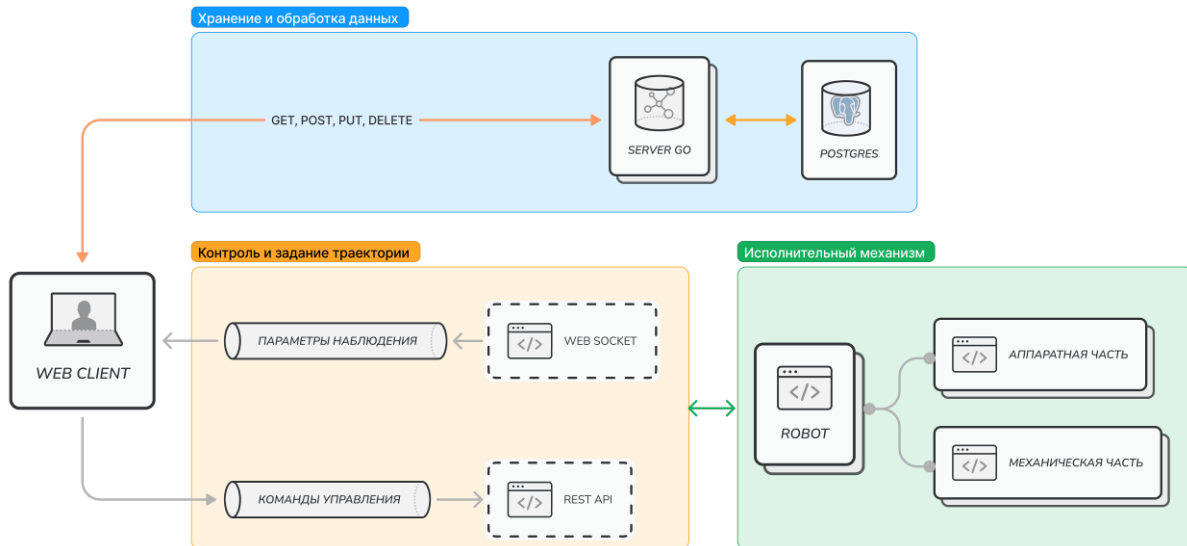


Рис. 3 – Архитектура комплекса планирования траектории

В. Структура базы данных

В качестве базы данных для хранения сведений о траектории используется СУБД PostgreSQL. Данное решение является бесплатным, обеспечивает полную поддержку SQL, а также имеет возможность хранения данных формата JSON и предоставляет механизм транзакций, что обеспечивает целостность хранимых данных [15]. ER-диаграмма базы данных представлена на рисунке 4.

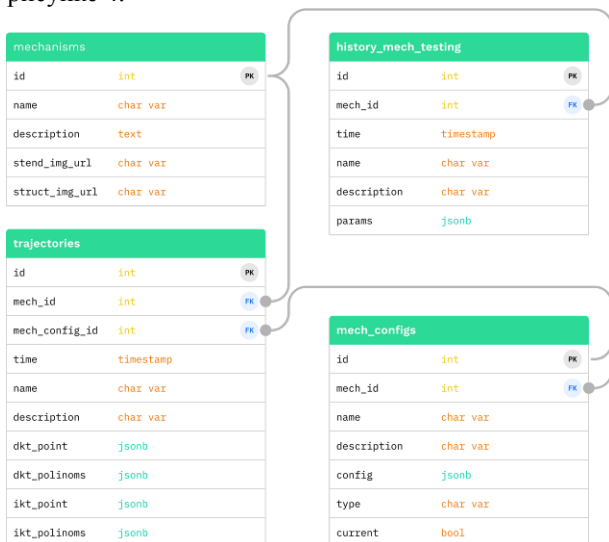


Рис. 4 – ER-диаграмма базы данных

Таблица *mechanisms* необходима для хранения информации о механизмах, доступных для исследования. Данная таблица связана отношением «многие ко многим» с таблицами *mech_configs* для описания кинематической конфигурации механизма, *trajectories* для записи промоделированных траекторий, *history_mech_testing* для хранения сеансов испытаний.

С. Описание API

Задачей REST API для взаимодействия с базой данных является обеспечение функционала для решения прямой и обратной задач кинематики, а также интерполирования при формировании траектории. При

этом API должно предусматривать возможность взаимодействия с конфигурациями механизма, траекториями и историями наблюдений. Для выполнения данного функционала были реализованы следующие endpoint API:

- `/mechanisms/` (GET): получить названия, описания, изображения, текущие конфигурации, приводы, датчики обратной связи и параметры для контроля для всех доступных манипуляционных механизмов;
- `/mechanisms/{mech_id}/configs` (GET): получить все кинематические конфигурации механизма по его идентификатору;
- `/mechanisms/{mech_id}/configs/current` (GET): получить активную (текущую) кинематическую конфигурацию исследуемого механизма;
- `/mechanisms/{mech_id}/kinematics/dkt_solver` (POST): решить прямую задачу кинематики;
- `/mechanisms/{mech_id}/kinematics/ikt_solver` (POST): решить обратную задачу кинематики;
- `/mechanisms/{mech_id}/kinematics/interpolation` (POST): найти интерполяционные многочлены для заданного набора точек;
- `/mechanisms/{mech_id}/trajectories` (GET): получить доступные траектории для заданного механизма;
- `/mechanisms/{mech_id}/trajectories/{traj_id}` (GET): получить траекторию по её идентификатору для исследуемого механизма;
- `/mechanisms/{mech_id}/trajectories/save` (POST): сохранить траекторию;
- `/mechanisms/{mech_id}/history/` (GET): получить истории наблюдений за параметрами манипуляционного механизма по его идентификатору;

- `/histories/add` (POST): добавить новую историю наблюдений за параметрами работы манипуляционного механизма;
- `/histories/{id}/edit` (POST): редактировать название и описание истории наблюдений за параметрами работы манипуляционного механизма;
- `/histories/{id}/del` (DEL): удалить историю наблюдений за параметрами манипуляционного механизма.

Структура API для управления манипуляционным механизмом включает три запроса:

- `/start` (GET): начало движения по заданной траектории;
- `/stop` (GET): завершение движения по заданной траектории;
- `/setPoint?motor_id={motor_id}` (POST): задание требуемых значений обобщённой координаты в зависимости от метода планирования траектории.

V. ОПИСАНИЕ ЭКСПЕРИМЕНТАЛЬНОГО СТЕНДА И АЛГОРИТМА УПРАВЛЕНИЯ

A. Экспериментальный стенд

Эксперименты проводились на базе стенда совместного относительного манипулирования, который включает в себя параллельный шестизвенный механизм и поступательно-поворотный механизм (Рис. 5). Представленный механизм имеет пять степеней свободы, что позволяет осуществлять позиционирование выходным звеном по трём декартовым координатам и двум углам ориентации относительно базовой системы координат, размещённой на рабочем столе поступательно-поворотного механизма.



Рис. 5 – Экспериментальный стенд

Движение кинематических пар механизма осуществляется при помощи двигателей постоянного тока с червячным редуктором с датчиком Холла для реализации обратной связи и драйвером L298N для задания ШИМ сигнала. Расчёт алгоритма управления осуществляется на микроконтроллере семейства STM32, который предназначен для получения и передачи информации по UART с микроконтроллером ESP32.

Для управления приводами были использованы возможности многопоточного управления для распараллеливания процессов расчёта регуляторов для каждого привода кинематической пары. Реализованные интерфейсы управления позволяют задавать либо

конкретное положение вала двигателя постоянного тока, либо коэффициенты интерполяционных полиномов для формирования гладкой функции изменения обобщённой координаты в зависимости от времени. Принципиальная схема работы микроконтроллера представлена на рисунке 6.

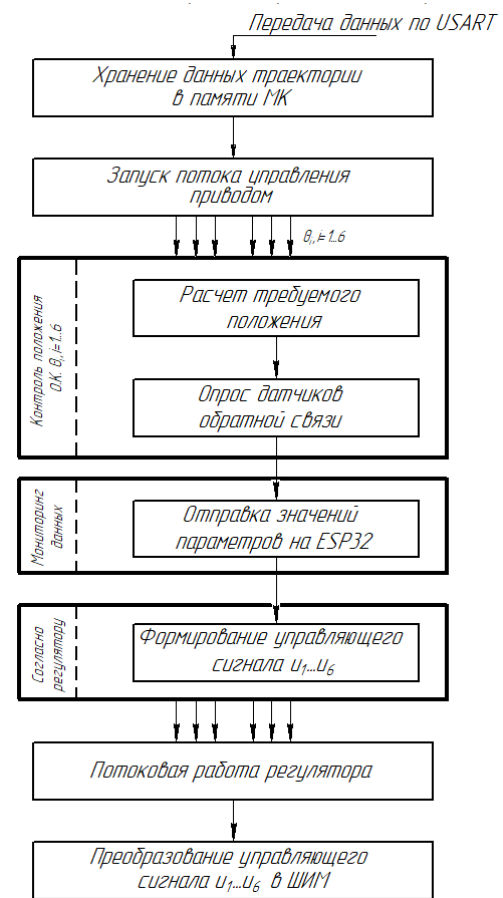


Рис. 6 – Принципиальная схема работы микроконтроллера

B. Задача управления

Задачей управления является синтез такого алгоритма, при котором рассматриваемый объект переходит из произвольного начального состояния $x_0 = x(0)$ в заданное конечное $x(\infty) = x^{уст} = const$, при этом управляемый процесс с требуемой точностью должен следовать за эталонным процессом, который задан моделью вида:

$$\ddot{x}_s(t) + \alpha_1 \dot{x}_s(t) + \alpha_0 x_s(t) = \alpha_0 x^{уст}(t), \quad (1)$$

где числовые значения параметров α_0 и α_1 подбираются из условия обеспечения заданного процесса.

Эталонный процесс выбирается исходя из требований, накладываемых на синтезируемую систему. В частности, одним из таких требований может быть обеспечение заданного быстродействия переходного процесса. Исходя из этого зададим эталонный процесс с учётом времени управления $t_y = 2.45$ с:

$$W_9(s) = \frac{3.9}{s^2 + 4s + 3.9}. \quad (2)$$

С. Алгоритм управления

Рассмотрим синтез системы управления по критерию минимума энергии ускорения. В этом случае структура алгоритма и его параметры определяются из условия минимизации функционала [7]:

$$J(u) = \frac{1}{2} [\ddot{x}_9(t) - \ddot{x}(t, u)]^2. \quad (3)$$

Для определения закона управления воспользуемся градиентной схемой первого порядка:

$$\frac{d}{dt}u(t) = \lambda \frac{d}{du}J(u), \quad \lambda - const. \quad (4)$$

Запишем передаточную функцию двигателя постоянного тока в виде дифференциального уравнения:

$$\ddot{x}(t) + a_1\dot{x}(t) + a_0x(t) = b_0u(t), \quad (5)$$

$$\ddot{x}(t, u) = b_0u(t) - a_0x(t) - a_1\dot{x}(t). \quad (6)$$

Коэффициенты b_0, a_0, a_1 в (5) и (6) определяются из математической модели ДПТ [16] как $b_0 = \frac{k_m}{J \cdot L}$; $a_0 = \frac{Rk_{em} + k_e k_m}{J \cdot L}$; $a_1 = \frac{JR + Lk_{em}}{JL}$, где k_e – электромагнитный коэффициент противоЭДС; k_m – коэффициент вязкого трения; R – сопротивление якорной цепи, [Ом]; L – индуктивность якорной цепи, [Гн]; J – момент инерции якоря ДПТ, [кг·м²].

Найдем производную функционала (3) по управлению u с учётом уравнения (6):

$$\begin{aligned} \frac{d}{du}J(u) &= \frac{d}{du} \left\{ \frac{1}{2} [\ddot{x}_9(t) - \ddot{x}(t, u)]^2 \right\} = \\ &= \frac{d}{du} \left\{ \frac{1}{2} [\ddot{x}_9(t) - b_0u(t) - a_0x(t) - a_1\dot{x}(t)]^2 \right\} = \\ &= -b_0 [\ddot{x}_9(t) - b_0u(t) - a_0x(t) - a_1\dot{x}(t)] = -b_0 [\ddot{x}_9(t) - \ddot{x}(t)]. \end{aligned}$$

Следовательно уравнение (4) примет вид:

$$\frac{d}{dt}u(t) = -\lambda b_0 [\ddot{x}_9(t) - \ddot{x}(t)] = K [\ddot{x}_9(t) - \ddot{x}(t)], \quad (7)$$

где $K = -\lambda b_0$.

Значение $x_9(t)$ находится из уравнения (5):

$$\ddot{x}_9(t) = \alpha_0 [x_{ex}(t) - x_9(t)] - \alpha_1 \dot{x}_9(t), \quad (8)$$

где $x_{ex}(t)$ – входной сигнал.

Заменим в первой части выражения (8) $x_9(t)$ на $x(t)$ и подставим его в уравнение (7). В результате получим дифференциальный закон управления:

$$\frac{d}{du}u(t) = K \{ \alpha_0 [x_{ex}(t) - x(t)] - \alpha_1 \dot{x}(t) - \ddot{x}(t) \}. \quad (9)$$

Однократно проинтегрировав с учётом нулевых начальных условий, окончательно найдем:

$$u(t) = K \left\{ \alpha_0 \int_0^t [x_{ex}(t) - x(t)] dt - \alpha_1 x(t) - \dot{x}(t) \right\} \quad (10)$$

На рисунке 7 представлена структурная схема полученной замкнутой системы управления.

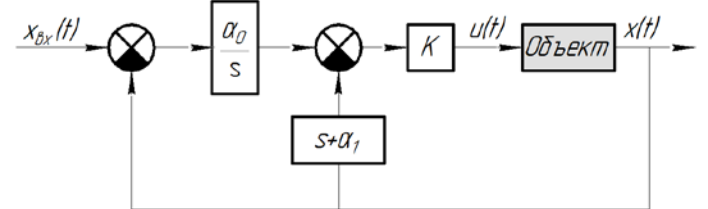


Рис. 7 – Структурная схема замкнутой системы управления по критерию минимума энергии ускорения с учётом дифференцирующего звена для вычисления выходной координаты

Для практической реализации алгоритма управления необходимо провести дискретизацию непрерывного закона управления (10). Запишем (10) в форме преобразования Лапласа:

$$\begin{aligned} U(s) &= K \left(\frac{\alpha_0}{s} (X_{ex}(s) - X(s)) - (s + \alpha_1) X(s) \right) = \\ &= \frac{K\alpha_0}{s} X_{ex}(s) - K \left((s + \alpha_1) + \frac{\alpha_0}{s} \right) X(s) \end{aligned}$$

Или после приведения к общему знаменателю:

$$sU(s) = K\alpha_0 X_{ex}(s) - K(s^2 + \alpha_1 s + \alpha_0) X(s). \quad (11)$$

После выполнения дискретизации методом Тустина или Эйлера уравнение сигнала (11) сводится к виду:

$$(z^2 + a_1 z + a_0)U(z) = (b_2 z^2 + b_1 z + b_0)X_{ex}(z) + (c_2 z^2 + c_1 z + c_0)X(z). \quad (12)$$

Вынесем z^2 в уравнении (12) и раскроем скобки:

$$\begin{aligned} U(z) + a_1 z^{-1}U(z) + a_0 z^{-2}U(z) &= b_2 X_{ex}(z) + b_1 z^{-1}X_{ex}(z) + \\ &+ b_0 z^{-2}X_{ex}(z) + c_2 X(z) + c_1 z^{-1}X(z) + c_0 z^{-2}X(z) \end{aligned} \quad (13)$$

Исходя из (13) разностное уравнение регулятора имеет вид:

$$\begin{aligned} u[k] &= b_2 x_{ex}[k] + b_1 x_{ex}[k-1] + b_0 x_{ex}[k-2] + c_2 x[k] + \\ &+ c_1 x[k-1] + c_0 x[k-2] - a_1 u[k-1] - a_0 u[k-2] \end{aligned} \quad (14)$$

В таблице 1 приведены значения коэффициентов в уравнении (14), полученные в результате дискретизации обратным методом Эйлера и методом Тустина.

Таблица 1 – Коэффициенты разностного уравнения управления по критерию минимума ускорения

Обратный метод Эйлера	Метод Тустина
$a_0 = 0$; $a_1 = -1$;	$a_0 = -1$; $a_1 = 0$;

$b_0 = 0;$ $b_1 = 0;$ $b_2 = K\alpha_0 T_\delta;$	$b_0 = \frac{1}{2}\alpha_0 T_\delta K;$ $b_1 = \alpha_0 T_\delta K; b_2 = \frac{1}{2}\alpha_0 T_\delta K;$
$c_0 = -\frac{K}{T_\delta};$ $c_1 = \frac{-K(-T_\delta\alpha_1 - 2)}{T_\delta};$ $c_2 = \frac{-K(T_\delta^2\alpha_0 + T_\delta\alpha_1 + 1)}{T_\delta}.$	$c_0 = -\frac{(T_\delta^2\alpha_0 - 2T_\delta\alpha_1 + 4)K}{2T_\delta};$ $c_1 = -\frac{(2T_\delta^2\alpha_0 - 8)K}{2T_\delta};$ $c_2 = -\frac{(T_\delta^2\alpha_0 + 2T_\delta\alpha_1 + 4)K}{2T_\delta}.$

Итоговые параметры регулятора $\alpha_0 = 3.9$, $\alpha_1 = 4.0$, $K = 44.1$. Период дискретизации T_δ равен 0.04 с. и был определён исходя из анализа запаса устойчивости разомкнутой системы управления.

Алгоритм работы микроконтроллера представлен на рисунке 8.

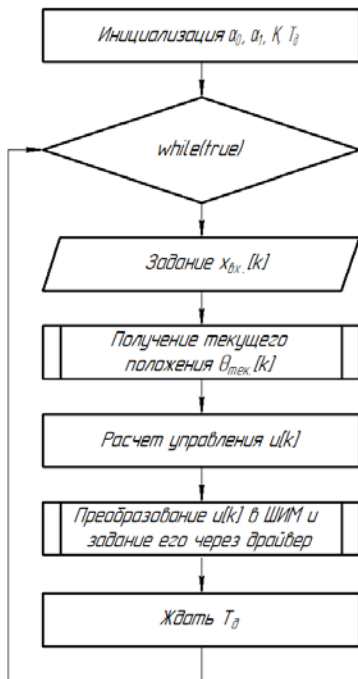


Рис. 8 – Алгоритм расчета дискретного управления на основе критерия минимума энергии ускорения на микроконтроллере

Результаты эксперимента показывают, что задача управления полностью достигнута, поскольку реальный процесс следует за эталонным, выполняя заданное требование на время управления (Рис. 9).

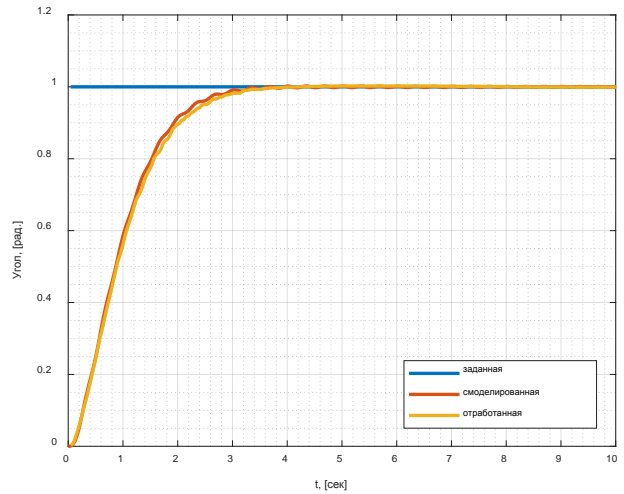


Рис. 9 – Сравнение теоретических и практических результатов работы регулятора на основе критерия минимума энергии ускорения

VI. ДЕМОНСТРАЦИЯ РАЗРАБОТАННОГО КОМПЛЕКСА

A. Редактор кинематических конфигураций

Редактор кинематических конфигураций позволяет создавать, редактировать и просматривать параметры кинематической схемы манипуляционного механизма, которые используются при расчёте прямой и обратной задач кинематики при формировании траектории (Рис. 10). Кинематическая конфигурация подразделяется на два типа: стандартная – используется в качестве шаблона при создании новой конфигурации, а также пользовательская – добавляется пользователем. Конфигурация, используемая в данный момент при формировании траектории, имеет статус «текущая». Остальные конфигурации помечаются статусом «архивная».

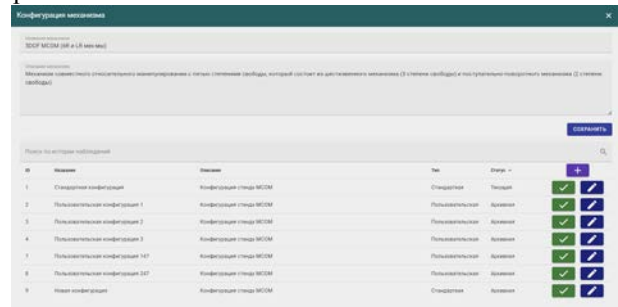


Рис. 10 – Конфигурация механизмов

Окно редактирования конфигурации показано на рисунке 11. Для каждой конфигурации пользователь может указать название, описание, а также значение каждого из кинематических параметров.

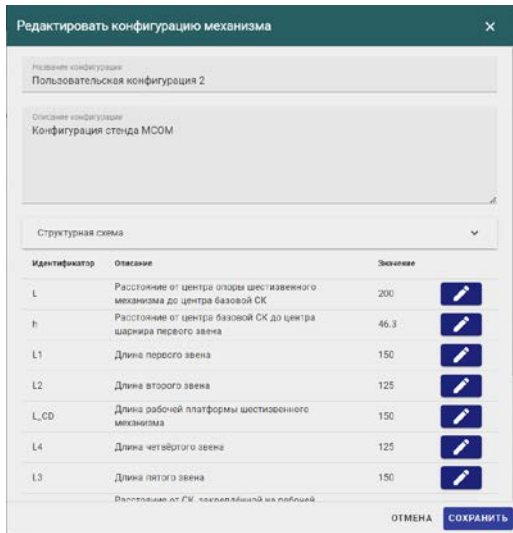


Рис.11 – Редактирование конфигурации механизма

В. Формирование траектории

Окно выбора механизма и способа для формирования траектории располагается на странице «Новая траектория» (Рис. 12).

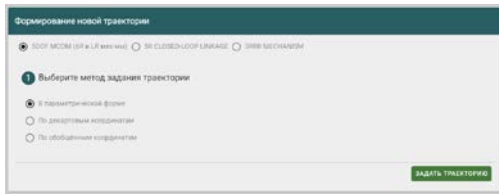


Рис. 12 – Выбор метод задания траектории

После выбора способа формирования траектории открывается окно с вводом входных значений (Рис. 13).

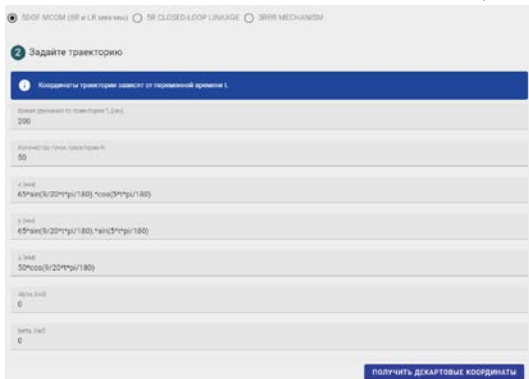


Рис. 13 – Ввод законов изменения декартовых координат

На рисунке 14 показано отображение полученных наборов декартовых и обобщённых координат в результате ввода параметрической траектории.

ДЕКАРТОВЫЕ КООРДИНАТЫ		ОБЩЕНЫЕ КООРДИНАТЫ			
№	1	x, мм	y, мм	α, deg	β, deg
0	0	0	0	0	0
1	4.081632633061225	1.952385329959332	0.4146066472412876	0	0
2	8.16326530612245	3.1517790410562982	0.14785915107181752	-0.04191519026479934	-0.90077154732371532
3	12.24489759182675	3.0043238943668155	-0.23289696399203802	-0.03136270813969957	0.0001646803014317567
4	16.3265306122449	1.306507973462113	-0.643923780183318	-0.04934621183217762	0.00112787150217049
5	20.40918265306122	-2.164162637962605	-0.9302769794238529	-0.0353530346170468	0.001936815994138934
6	24.48979518367346	6.66768088084247	-1.18365592525146	-0.018194182475718	0.002431840227104987
7	28.574285742857	11.529615549621965	-1.1586190510431171	0.01795821713816887	0.0029026687847148225
8	32.6530612244898	15.78927533722063	-0.886130057668836	0.0486031410210178	0.002108648949438495
9	36.73463877051024	-18.45629462227272	-0.3347812971796835	0.0744230774895613	0.00128792833771232

Рис. 14 – Вывод решения обратной и прямой задач кинематики

Затем происходит интерполирование каждого набора координат с последующим выводом коэффициентов интерполяционных полиномов (Рис. 15).

ДЕКАРТОВЫЕ КООРДИНАТЫ		ОБЩЕНЫЕ КООРДИНАТЫ	
X	Y	ALPHA	BETA
c	0	a	β
0	0	0.5192271775120411	0
4.081632633061225	1.952385329959332	0.4146066472412876	-0.02343795143504025
8.16326530612245	3.1517790410562982	0.14785915107181752	-0.04191519026479934
12.24489759182675	3.0043238943668155	-0.23289696399203802	-0.03136270813969957
16.3265306122449	1.306507973462113	-0.643923780183318	-0.04934621183217762
20.40918265306122	-2.164162637962605	-0.9302769794238529	-0.0353530346170468
24.48979518367346	6.66768088084247	-1.18365592525146	-0.018194182475718
28.574285742857	11.529615549621965	-1.1586190510431171	0.01795821713816887
32.6530612244898	15.78927533722063	-0.886130057668836	0.0486031410210178
36.73463877051024	-18.45629462227272	-0.3347812971796835	0.0744230774895613

Рис. 15 – Коэффициенты интерполяционных полиномов
Заключительным этапом является визуализация траектории (Рис. 16).

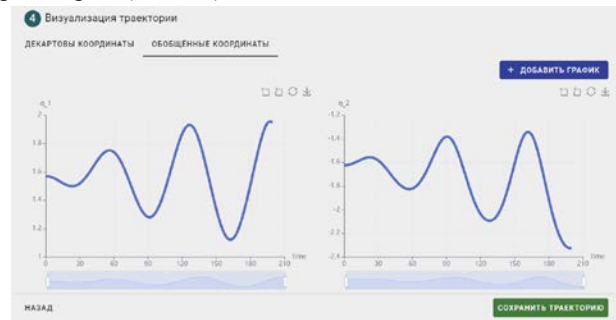


Рис. 16 – Визуализация траектории

Для сохранения траектории в базу данных необходимо ввести её название, а также краткое описание (Рис. 17).

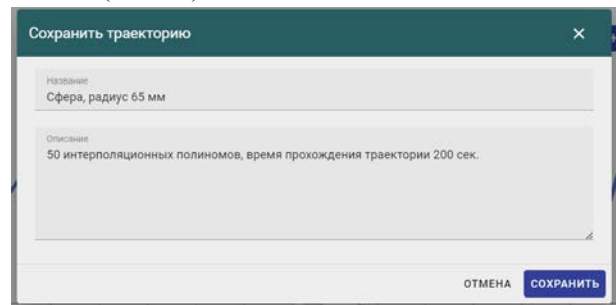


Рис. 17 – Форма сохранения траектории

Обозреватель траекторий механизма (Рис. 18) позволяет визуализировать траектории, а также импортировать их в файл.



Рис. 18 – Обозреватель траекторий механизма

C. Результат планирования траектории

Для проверки работы системы была задана траектория в параметрической форме с временем прохождения $T = 100$ и количеством опорных точек $N = 50$:

$$x(t) = 50 \sin\left(\frac{9}{20}t\right) \cos(5t); \quad y(t) = 50 \sin\left(\frac{9}{20}t\right) \sin(5t);$$

$$z(t) = 50 \cos\left(\frac{9}{20}t\right) + 10; \quad \alpha(t) = 0 \cdot t; \quad \beta = 0.$$

Законы изменения обобщённых координат, полученные в результате формирования траектории, были заданы на вход системы управления манипуляционного механизма. Результирующие перемещения выходного звена рассматриваемого стенда представлены на рисунках 19-21.

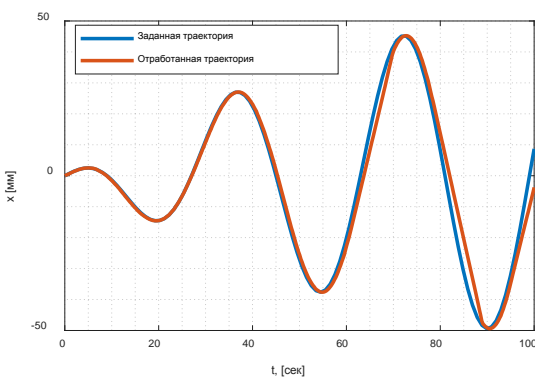


Рис. 19 – Изменение $x(t)$

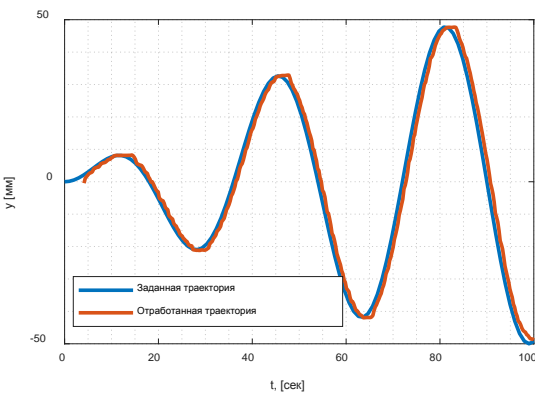


Рис. 20 – Изменение $y(t)$

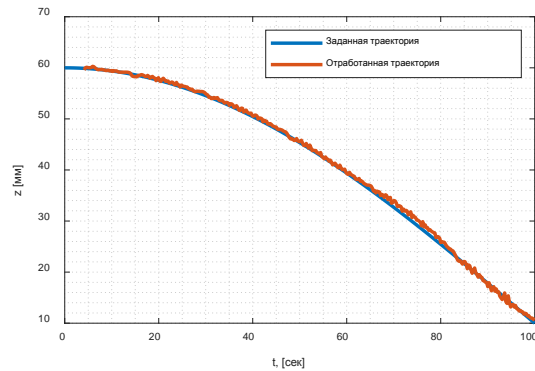


Рис. 21 – Изменение $z(t)$

D. Наблюдение за параметрами механизма

Для проведения сеанса наблюдений необходимо выбрать интересующий механизм, а затем для каждого из приводов указать параметры для наблюдения (Рис. 22).

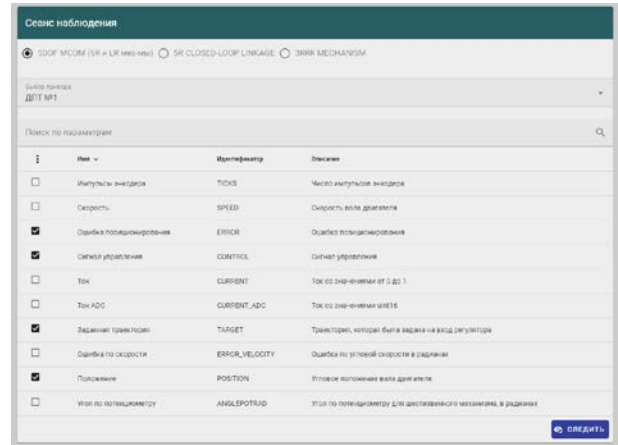


Рис. 22 – Выбор параметров для наблюдений при сеансе испытаний механизма

После нажатия на кнопку «следить» откроется окно с графиками параметров, изменяющихся по мере обработки сообщений, получаемых по веб-сокету (Рис. 23).

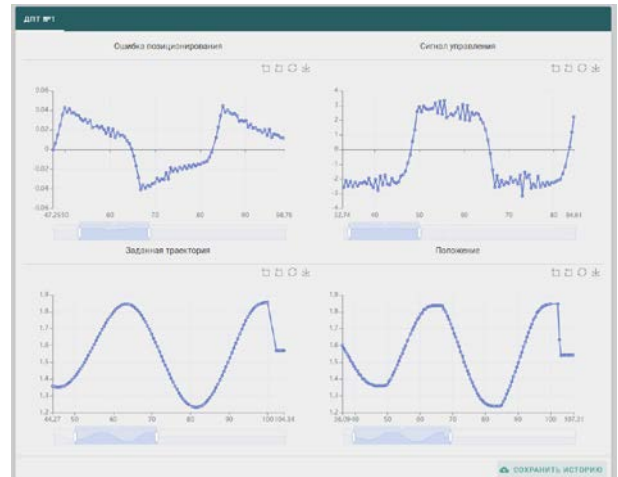


Рис. 23 – Проведение наблюдений за параметрами механизма

E. История наблюдений

Пользователь имеет возможность визуализировать историю, скачивать данные параметров, редактировать описание истории, а также удалять запись с историей наблюдений из базы данных (Рис. 24).

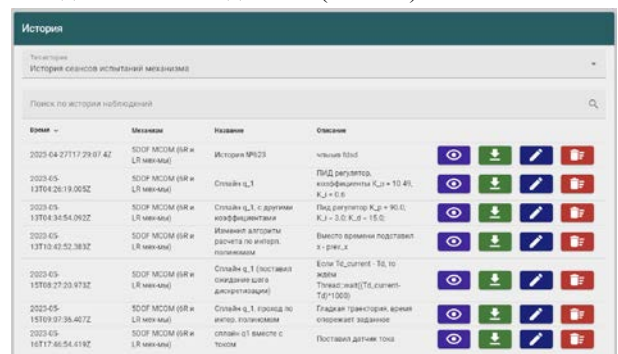


Рис. 24 – Список сохранённых историй

Пример повторной визуализации сохранённой истории представлен на рисунке 25.

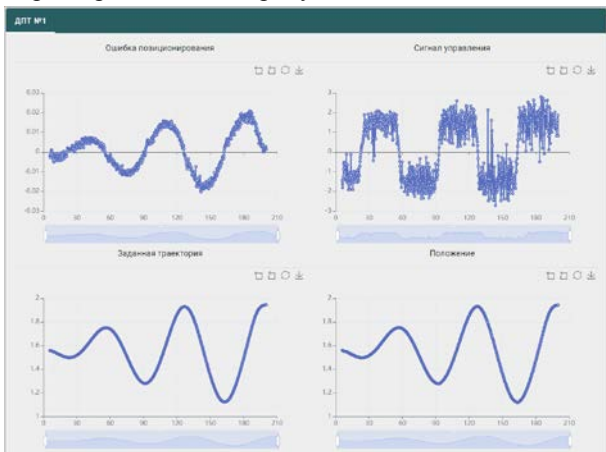


Рис. 25 – Визуализация сохранённой истории

VII. ЗАКЛЮЧЕНИЕ

Предложенный комплекс планирования траектории позволяет осуществлять управление манипуляционными механизмами различной кинематической структуры. Апробация результатов работы была произведена на механизме совместного относительного манипулирования, объединяющий последовательный и параллельный механизмы. Алгоритм формирования траектории включает решение прямой и обратной задач кинематики для получения наборов декартовых и обобщённых координат с последующей интерполяцией для восстановления характеристик траектории в промежуточных положениях. Результирующий программный комплекс управления был внесен в государственный реестр программ для ЭВМ и баз данных с идентификационными номерами №2023621249, №2023618176 и №2023663025.

БИБЛИОГРАФИЯ

- [1] Liu, Liang, et al. "Smooth trajectory planning for a parallel manipulator with joint friction and jerk constraints." *International Journal of Control, Automation and Systems* 14 (2016): 1022-1036. <https://doi.org/10.1007/s12555-014-0495-4>
- [2] Liu, Huashan, Xiaobo Lai, and Wenxiang Wu. "Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints." *Robotics and Computer-Integrated Manufacturing* 29.2 (2013): 309-317. <https://doi.org/10.1016/j.rcim.2012.08.002>
- [3] Yu, Xiuli, Mingshuai Dong, and Weimin Yin. "Time-optimal trajectory planning of manipulator with simultaneously searching the optimal path." *Computer Communications* 181 (2022): 446-453. <https://doi.org/10.1016/j.comcom.2021.10.005>
- [4] Luo, LP., Yuan, C., Yan, RJ. et al. Trajectory planning for energy minimization of industry robotic manipulators using the Lagrange interpolation method. *Int. J. Precis. Eng. Manuf.* 16, 911–917 (2015). <https://doi.org/10.1007/s12541-015-0119-9>
- [5] Chembuly, VVMJ Satish, and Hari Kumar Voruganti. "Trajectory planning of redundant manipulators moving along constrained path and avoiding obstacles." *Procedia computer science* 133 (2018): 627-634. <https://doi.org/10.1016/j.procs.2018.07.094>
- [6] Svenstrup, Mikael, Thomas Bak, and Hans Jørgen Andersen. "Trajectory planning for robots in dynamic human environments."

- 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2010. <https://doi.org/10.1109/IROS.2010.5651531>
- [7] Крутько, П.Д., 2004. Обратные задачи динамики в теории автоматического управления. М.: Машиностроение, 576.
- [8] Gasparetto, Alessandro, et al. "Path planning and trajectory planning algorithms: A general overview." *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches* (2015): 3-27. https://doi.org/10.1007/978-3-319-14705-5_1
- [9] Rossi, Cesare, and Sergio Savino. "Robot trajectory planning by assigning positions and tangential velocities." *Robotics and Computer-Integrated Manufacturing* 29.1 (2013): 139-156. <https://doi.org/10.1016/j.rcim.2012.04.003>
- [10] Guan, Y., Yokoi, K., Stasse, O., & Kheddar, A. (2005, July). On robotic trajectory planning using polynomial interpolations. In 2005 IEEE international conference on robotics and biomimetics-ROBIO (pp. 111-116). IEEE. <https://doi.org/10.1109/ROBIO.2005.246411>
- [11] Švejsda, Martin, and Tomáš Čechura. "Interpolation method for robot trajectory planning." 2015 20th International Conference on Process Control (PC). IEEE, 2015. <https://doi.org/10.1109/PC.2015.7169997>
- [12] Bielak, Konrad, Bartłomiej Borek, and Małgorzata Plechawska-Wójcik. "Web application performance analysis using Angular, React and Vue. js frameworks." *Journal of Computer Sciences Institute* 23 (2022): 77-83. <https://doi.org/10.35784/jcsi.2827>
- [13] Yellavula, Naren. Building RESTful Web services with Go: Learn how to build powerful RESTful APIs with Golang that scale gracefully. Packt Publishing Ltd, 2017.
- [14] Maier, Alexander, Andrew Sharp, and Yuriy Vagapov. "Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things." 2017 Internet Technologies and Applications (ITA). IEEE, 2017. <https://doi.org/10.1109/ITECHA.2017.8101926>
- [15] Obe, Regina O., and Leo S. Hsu. PostgreSQL: up and running: a practical guide to the advanced open source database. " O'Reilly Media, Inc.", 2017.
- [16] Rahman, N. N. A., & Yahya, N. M. (2021). A mathematical model of a brushed DC motor system. *Data Analytics and Applied Mathematics (DAAM)*, 2(2), 60-68. <https://doi.org/10.15282/daam.v2i2.6830>

Development of a software package for solving the problem of trajectory planning with motion control of the manipulation mechanism

Maxim O. Chaikin

Abstract — Trajectory planning of the end effector of manipulation mechanisms is one of the most important tasks in the practical application of robots in various fields of technology. This paper describes the development of a software package for automating the trajectory setting process with the condition of ensuring its accurate execution. The developed software allows to form a trajectory based on various source data, solve the direct and inverse kinematics problem, as well as perform an interpolation procedure in order to obtain the laws of change of the generalized coordinates of the manipulation mechanism. The interface for interacting with a human operator is represented by a WEB client written in Vue JS. Interaction with the PostgreSQL database for storing information is carried out through the REST API implemented in the Go programming language. To ensure passage along a given trajectory, a control algorithm based on inverse dynamics problems was considered. The control law was calculated using STM32 family microcontrollers for DC motors with a Hall sensor for position and speed feedback. Data transfer between the WEB client and the manipulation mechanism is carried out using the ESP32 microcontroller, which implements websocket, as well as a REST API for setting control commands. The results of the trajectory planning complex are demonstrated by the example of joint relative manipulation of a mechanism with five degrees of freedom.

Keywords — trajectory planning, manipulation mechanism control, REST API, Golang, PostgreSQL, Vue JS, ESP32, STM32

REFERENCES

- [1] Liu, Liang, et al. "Smooth trajectory planning for a parallel manipulator with joint friction and jerk constraints." *International Journal of Control, Automation and Systems* 14 (2016): 1022-1036. <https://doi.org/10.1007/s12555-014-0495-4>
- [2] Liu, Huashan, Xiaobo Lai, and Wenxiang Wu. "Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints." *Robotics and Computer-Integrated Manufacturing* 29.2 (2013): 309-317. <https://doi.org/10.1016/j.rcim.2012.08.002>
- [3] Yu, Xiuli, Mingshuai Dong, and Weimin Yin. "Time-optimal trajectory planning of manipulator with simultaneously searching the optimal path." *Computer Communications* 181 (2022): 446-453. <https://doi.org/10.1016/j.comcom.2021.10.005>
- [4] Luo, LP., Yuan, C., Yan, RJ, et al. Trajectory planning for energy minimization of industry robotic manipulators using the Lagrange interpolation method. *Int. J. Precis. Eng. Manuf.* 16, 911-917 (2015). <https://doi.org/10.1007/s12541-015-0119-9>
- [5] Chembuly, VVMJ Satish, and Hari Kumar Voruganti. "Trajectory planning of redundant manipulators moving along constrained path and avoiding obstacles." *Procedia computer science* 133 (2018): 627-634. <https://doi.org/10.1016/j.procs.2018.07.094>
- [6] Svenstrup, Mikael, Thomas Bak, and Hans Jørgen Andersen. "Trajectory planning for robots in dynamic human environments." 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2010. <https://doi.org/10.1109/IROS.2010.5651531>
- [7] Krutko, P.D., 2004. Inverse problems of dynamics in the theory of automatic control. Moscow: Mashinostroenie, 576.
- [8] Gasparetto, Alessandro, et al. "Path planning and trajectory planning algorithms: A general overview." *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches* (2015): 3-27. https://doi.org/10.1007/978-3-319-14705-5_1
- [9] Rossi, Cesare, and Sergio Savino. "Robot trajectory planning by assigning positions and tangential velocities." *Robotics and Computer-Integrated Manufacturing* 29.1 (2013): 139-156. <https://doi.org/10.1016/j.rcim.2012.04.003>
- [10] Guan, Y., Yokoi, K., Stasse, O., & Kheddar, A. (2005, July). On robotic trajectory planning using polynomial interpolations. In 2005 IEEE international conference on robotics and biomimetics-ROBIO (pp. 111-116). IEEE. <https://doi.org/10.1109/ROBIO.2005.246411>
- [11] Švejda, Martin, and Tomáš Čechura. "Interpolation method for robot trajectory planning." 2015 20th International Conference on Process Control (PC). IEEE, 2015. <https://doi.org/10.1109/PC.2015.7169997>
- [12] Bielak, Konrad, Bartłomiej Borek, and Małgorzata Plechawska-Wójcik. "Web application performance analysis using Angular, React and Vue.js frameworks." *Journal of Computer Sciences Institute* 23 (2022): 77-83. <https://doi.org/10.35784/jcsi.2827>
- [13] Yellavula, Naren. Building RESTful Web services with Go: Learn how to build powerful RESTful APIs with Golang that scale gracefully. Packt Publishing Ltd, 2017.
- [14] Maier, Alexander, Andrew Sharp, and Yuriy Vagapov. "Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things." 2017 *Internet Technologies and Applications (ITA)*. IEEE, 2017. <https://doi.org/10.1109/ITECHA.2017.8101926>
- [15] Obe, Regina O., and Leo S. Hsu. PostgreSQL: up and running: a practical guide to the advanced open source database. "O'Reilly Media, Inc.", 2017.
- [16] Rahman, N. N. A., & Yahya, N. M. (2021). A mathematical model of a brushed DC motor system. *Data Analytics and Applied Mathematics (DAAM)*, 2(2), 60-68. <https://doi.org/10.15282/daam.v2i2.6830>