

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Локтионова Оксана Геннадьевна  
Должность: проректор по учебной работе  
Дата подписания: 31.01.2025 16:00:50  
Уникальный программный ключ:  
0b817ca911e6668abb13a5d44e91611176e410c1e8911a1c118

## МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра механики, мехатроники и робототехники

УТВЕРЖДАЮ

проректор по учебной работе

  
О.Г. Локтионова  
« 30 » 01 2025 г.



## МОДЕЛИРОВАНИЕ И ИССЛЕДОВАНИЕ МЕХАТРОННЫХ СИСТЕМ И РОБОТОВ

методические указания по выполнению практической  
и самостоятельной работы  
для студентов направления 15.04.06 «Мехатроника и  
робототехника»

Курск 2025

УДК 621

Составители: С. Ф. Яцун, А. С. Печурин, Е.Н. Политов

Рецензент

Кандидат технических наук, доцент *А.Н. Рукавицын*

**Моделирование и исследование мехатронных систем и роботов:** методические указания по выполнению практической и самостоятельной работы для студентов направления 15.04.06 «Мехатроника и робототехника» / Юго-Зап. гос. ун-т; сост. С. Ф. Яцун, А. С. Печурин, Е.Н. Политов. Курск, 2025. - 46 с.

Методические указания содержат сведения о подходах к моделированию мехатронных и робототехнических систем, составлению математических моделей мехатронных систем и их исследованию, а также примеры решения задач моделирования роботов.

Приведены задания для самостоятельной работы студентов.

Предназначены для студентов направления подготовки 15.04.06 – Мехатроника и робототехника.

Текст печатается в авторской редакции

Подписано в печать *30.09.25*. Формат 60x84 1/16  
Усл.печ.л. 2,73. Уч.-изд.л. 2,61 Тираж 15 экз. Заказ. *53* Бесплатно.

Юго-Западный государственный университет.

305040 Курск, ул. 50 лет Октября, 94

## СОДЕРЖАНИЕ

1 Математическое моделирование электропривода на базе двигателя постоянного тока с редуктором	4
2 Математическое моделирование манипулятора	15
3. Математическое моделирование кинематики колёсного робота с дифференциальным приводом	29
Приложение	44
Список рекомендуемых источников	45

# 1 МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ЭЛЕКТРОПРИВОДА НА БАЗЕ ДВИГАТЕЛЯ ПОСТОЯННОГО ТОКА С РЕДУКТОРОМ

## **Цель работы:**

Основываясь на уравнениях электромеханического равновесия двигателя постоянного тока (ДПТ) независимого возбуждения (НВ), создать математическую модель и провести моделирование работы двигателя постоянного тока в программе MATLAB.

Ознакомиться с методом решения систем дифференциальных уравнений в программе *MATLAB* средствами встроенного решателя дифференциальных уравнений «ode» для моделирования механических систем на примере ДПТ. Освоить основные принципы компьютерного математического моделирования.

## **Задача 1:**

Создать файл рабочей программы с математической моделью работы ДПТ НВ, обеспечить выходную характеристику  $\omega = N$  - скорость вращения якоря при заданном  $m_c = 0.2N$  - статическом моменте, где  $N$  – номер варианта.

## ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Математическая модель двигателя состоит из электрической и механической частей, описываемых системой уравнений (1):

$$\begin{cases} u_a = e_a + i_a \cdot r_a + T_a \cdot r_a \cdot \frac{di_a}{dt}; \\ m - m_c = T_j \cdot \frac{d\omega}{dt}; \\ m = \varphi \cdot i_a; \\ e_a = \varphi \cdot \omega. \end{cases}, \quad (1)$$

где:

$u_a$  - напряжение источника питания, подаваемое на электрическую часть якоря двигателя (1 – подано номинальное напряжение, 0 – отключено);

$e_a$  - ЭДС, наводимая в обмотке якоря при пересечении проводников потока  $\Phi$ , создаваемого неподвижными полюсами;  $i_a$  - электрический ток в якорной цепи;  $r_a$  - полное электрическое сопротивление якорной цепи, включая сопротивление дополнительных полюсов и щеток;  $T_a = \frac{L_a}{r_a}$  - постоянная времени якорной цепи;  $m, m_c$  - электромагнитный и статический моменты;  $\omega$  - скорость вращения якоря;  $\Phi$  - магнитная постоянная;  $T_j$  - инерционная постоянная времени двигателя.

Параметры двигателя:  $r_a = 0,107$ ;  $T_a = 0,034$  с;  $T_j = 0,72$  с;  $\Phi = 1$ .

Первое уравнение системы (1) - баланс напряжений в электрической части двигателя. Второе - уравнение движения для механической части. Третье и четвертое - уравнения связи электрической и механической частей двигателя.

Из системы уравнений (1) выразим производные через конечные разности:

$$\frac{di_a}{dt} = (u_a - e_a - i_a \cdot r_a) \cdot \frac{1}{T_a \cdot r_a}; \quad (2)$$

$$\frac{i_a(i+1) - i_a(i)}{dt} = (u_a - e_a(i) - i_a(i) \cdot r_a) \cdot \frac{1}{T_a \cdot r_a}. \quad (3)$$

Тогда ток в якоре в  $(i+1)$  момент времени определится:

$$i_a(i+1) = i_a(i) + (u_a - e_a(i) - i_a(i) \cdot r_a) \cdot \frac{dt}{T_a \cdot r_a}. \quad (4)$$

Аналогично, для скорости  $\omega$ :

$$\frac{d\omega}{dt} = (m - m_c) \cdot \frac{1}{T_j}; \quad (5)$$

$$\frac{\omega(i+1) - \omega(i)}{dt} = (m - m_c) \cdot \frac{1}{T_j}; \quad (6)$$

$$\omega(i+1) = \omega(i) + (m - m_c) \cdot \frac{dt}{T_j}. \quad (7)$$

### Ход решения задачи 1:

1. Ознакомиться с теоретической частью
2. Изучить пример реализации.

Пример файла рабочей программы с математической моделью работы ДПТ НВ представлен на рисунках 1 – 3.

```
function simple_DC_motor_model()
    % DC Motor parameters and state
    [DC_mtr_params, DC_motor_state] = retDCmotorParameters;

    % Time settings
    t_start = 0;
    t_end = 5;
    dt = 0.001; %
    count = floor((t_end-t_start)/dt);

    % Определение времени воздействия
    DC_motor_start_time = 1;
    DC_motor_start_resistance_torque_time = 3;
```

Рис. 1. Пример кода объявления главной функции и используемых переменных

```
% Основной цикл
for i = 1:count
    t = i * dt;

    % Подача управляющего воздействия
    if(t >= DC_motor_start_time) |
        DC_motor_state.uy = 1;
    else
        DC_motor_state.uy = 0;
    end;

    % Подача возмущающего воздействия
    if(t >= DC_motor_start_resistance_torque_time)
        DC_motor_state.mc = 0.5;
    else
        DC_motor_state.mc = 0;
    end

    % Дифференциальное уравнение двигателя
    DC_motor_state.ia = DC_motor_state.ia + dia_dt(DC_mtr_params, DC_motor_state) * dt; % Численное интегрирование

    DC_motor_state.m = DC_mtr_params.phi * DC_motor_state.ia;
    DC_motor_state.w = DC_motor_state.w + dw_dt(DC_mtr_params, DC_motor_state) * dt;
    DC_motor_state.ea = DC_mtr_params.phi * DC_motor_state.w;

    % Сохранение данных в структуру
    result.t(i+1,1) = t;
    result.DC_motor_state.w(i+1,1) = DC_motor_state.w;
    result.DC_motor_state.ia(i+1,1) = DC_motor_state.ia;
    result.DC_motor_state.m(i+1,1) = DC_motor_state.m;
    result.DC_motor_state.ea(i+1,1) = DC_motor_state.ea;
end

% Вывод результатов
plotGraphs(result);
end
```

Рис. 2. Пример кода основного цикла программы

```

function [DC_mtr_params, DC_motor_state] = retDCmotorParameters
    DC_mtr_params.ra = 0.107; % Сопротивление обмотки якоря, О. е.
    DC_mtr_params.phi = 1; % Номинальный магнитный поток
    DC_mtr_params.Ta = 0.034; % Электромагнитная постоянная времени цепи якоря двигателя
    DC_mtr_params.Tj = 0.72; % Инерционная постоянная времени

    DC_motor_state.ia = 0; % Ток якоря
    DC_motor_state.w = 0; % Скорость вращения ротора
    DC_motor_state.ea = 0; % ЭДС двигателя
    DC_motor_state.m = 0; % Электромагнитный момент двигателя
    DC_motor_state.mc = 0; % Момент статического сопротивления
    DC_motor_state.uy = 0; % Управляющее воздействие
end

function ret = dia_dt(DC_mtr_params, DC_motor_state)
    ret = (DC_motor_state.uy - DC_motor_state.ea - DC_motor_state.ia * DC_mtr_params.ra) / (DC_mtr_params.Ta * DC_mtr_params.ra);
end

function ret = dw_dt(DC_mtr_params, DC_motor_state)
    ret = (DC_motor_state.m - DC_motor_state.mc) / DC_mtr_params.Tj;
end

function plotGraphs(result)
    figure(3)
    subplot(2,1,1)
    plot(result.t, result.DC_motor_state.w, 'b')
    xlabel('t, s');
    ylabel('\omega, rad/s');
    grid on;

    subplot(2,1,2)
    plot(result.t, result.DC_motor_state.ia, 'r')
    xlabel('t, s');
    ylabel('i_a, A');
    grid on;
end

```

Рис. 3. Пример кода вспомогательных функций

3. Создать файл рабочей программы с математической моделью работы ДПТ НВ и представить результаты.

Пример вывода результатов представлен на рисунке 4.

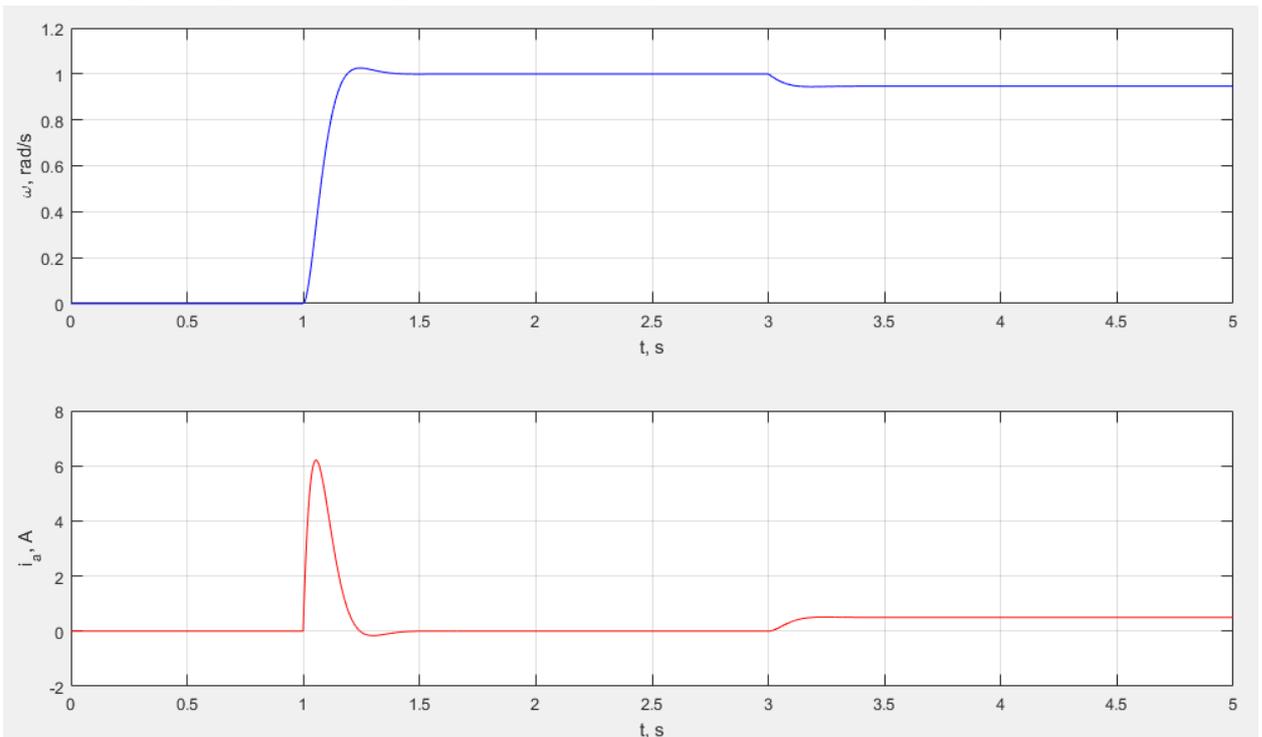


Рис. 4. Пример вывода результатов

### Задача 2:

Создать файл рабочей программы с математической моделью работы ДПТ НВ, решение дифференциальных уравнений которого, обеспечивается с помощью встроенной в MATLAB функции «ode».

Необходимо выбрать модель существующего ДПТ, использовать его характеристики для моделирования и сравнить результаты моделирования с номинальными характеристиками.

## ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

### Математическое описание ДПТ НВ

Схема подключения ДПТ НВ к источнику постоянного напряжения  $U$  представлена на рис. 5.

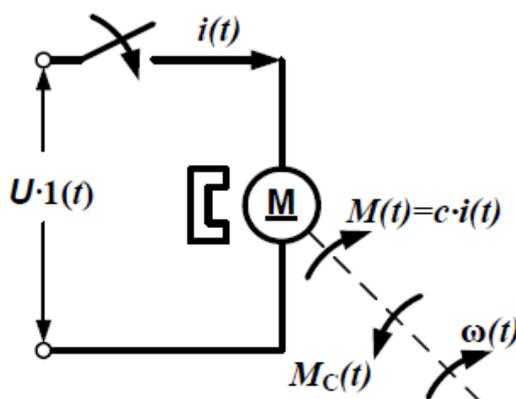


Рис. 5. Схема подключения ДПТ НВ к источнику постоянного напряжения

Схема замещения якорной цепи ДПТ НВ показана на рис. 6.

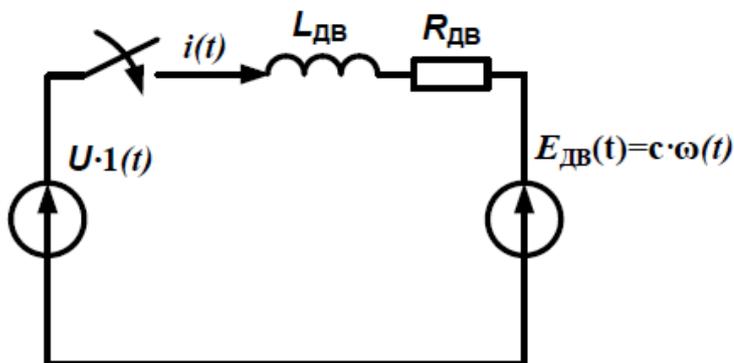


Рис. 6. Схема замещения якорной цепи ДПТ НВ

При составлении математической модели ДПТ НВ примем следующие допущения. Считаем, что реакция якоря полностью скомпенсирована (в реальном ДПТ всегда есть компенсационная обмотка либо добавочные

полюса), поток возбуждения постоянен, а активное сопротивление якорной цепи не изменяется во время работы двигателя.

Запишем дифференциальное уравнение электрического равновесия якорной цепи двигателя (рис. 3):

$$U = R_{\text{дв}} \cdot i(t) + L_{\text{дв}} \cdot \frac{di(t)}{dt} + E_{\text{дв}}(t),$$

где  $R_{\text{дв}}$  – суммарное активное сопротивление последовательно включенных обмотки якоря и добавочных полюсов в горячем состоянии (при  $t = 75^\circ \text{C}$ );  $L_{\text{дв}}$  – суммарная индуктивность якорной цепи;  $E_{\text{дв}}(t)$  – противо-ЭДС двигателя;  $U \cdot 1(t)$  – напряжение, приложенное к якорной цепи;  $i(t)$  – ток якорной цепи.

Уравнение механического равновесия двигателя:

$$M(t) - M_c = J_{\text{дв}} \cdot \frac{d\omega(t)}{dt},$$

где  $M(t)$  – электромагнитный момент ДПТ НВ;  $M_c \cdot 1(t)$  – момент сопротивления нагрузки;  $J_{\text{дв}}$  – суммарный момент инерции, приведенный к валу двигателя;  $\omega(t)$  – скорость двигателя.

Учитывая, что  $E_{\text{дв}}(t) = c \cdot \omega(t)$  и  $M(t) = c \cdot i(t)$  ( $c$  – коэффициент ЭДС и момента ДПТ НВ), запишем систему дифференциальных уравнений:

$$\begin{cases} U = R_{\text{дв}} \cdot i(t) + L_{\text{дв}} \cdot \frac{di(t)}{dt} + c \cdot \omega(t) \\ c \cdot i(t) - M_c = J_{\text{дв}} \cdot \frac{d\omega(t)}{dt} \end{cases}$$

СДУ в нормальной форме Коши:

$$\begin{cases} \frac{di(t)}{dt} = \frac{1}{L_{\text{дв}}} \cdot [U - R_{\text{дв}} \cdot i(t) - c \cdot \omega(t)] \\ \frac{d\omega(t)}{dt} = \frac{1}{J_{\text{дв}}} \cdot [c \cdot i(t) - M_c] \end{cases}$$

СДУ в матричном виде:

$$\frac{d}{dt} \begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix} = \begin{pmatrix} -\frac{R_{\text{дв}}}{L_{\text{дв}}} & -\frac{c}{L_{\text{дв}}} \\ \frac{c}{J_{\text{дв}}} & 0 \end{pmatrix} \cdot \begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix} + \begin{pmatrix} \frac{U}{L_{\text{дв}}} \\ -\frac{M_c}{J_{\text{дв}}} \end{pmatrix}$$

Здесь:

$$A = \begin{pmatrix} -\frac{R_{\text{дв}}}{L_{\text{дв}}} & -\frac{c}{L_{\text{дв}}} \\ \frac{c}{J_{\text{дв}}} & 0 \end{pmatrix} \quad - \text{ матрица коэффициентов перед переменными}$$

состояния;

$$B = \begin{pmatrix} \frac{U}{L_{\text{дв}}} \\ -\frac{M_c}{J_{\text{дв}}} \end{pmatrix} \quad - \text{ вектор свободных членов СДУ (вектор входных воздействий);}$$

$$x(t) = \begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix} \quad - \text{ вектор переменных состояния.}$$

Из полученной математической модели ДПТ НВ видно, что переменными состояниями в нем являются скорость вала и ток в якорной цепи. Эти переменные состояния соответственно связаны с массой вала и индуктивностью обмотки якоря, то есть с механической и электрической инерционностями двигателя.

### *Ход решения задачи 2:*

4. Ознакомиться с теоретической частью
5. Изучить пример реализации.

На рисунках 2 – 4 представлен программный код для моделирования ДПТ НВ помощью решателя дифференциальных уравнений “ode”.

```

function my_simple_DC_motor_model()
    % Time settings
    t_start = 0;
    t_end = 1;
    dt = 0.0001; %
    count = floor((t_end-t_start)/dt);

    % Motor parameters
    % DCX 26 L Ø 26mm, graphite brushes, sintered bearings
    R = 0.242;
    L = 0.0321/1000;
    Ci = 10.7/1000;
    J = 0.00000214;
    Mc = 0;
    U = 12;

    % Motor state
    I = 0;
    W = 0;
    phi = 0;

    q = [0; 0]; % [I, W]

```

Рис. 7. Пример кода объявления главной функции и используемых переменных

```

    % Основной цикл
    for i = 1:count
        t = i * dt;

        % Скалярный вид
        dIdt = (U - I*R - Ci*W)/L;
        I = I + dIdt * dt; % Интегрирование
        dWdt = (-Mc + Ci * I) / J;
        W = W + dWdt*dt; % Интегрирование
        phi = phi + W*dt;

        % Сохранение данных
        r.t(i,1) = t;
        r.I(i,1) = I;
        r.W(i,1) = W;
        r.phi(i,1) = phi;
        r.tau(i,1) = Ci * I*1000;
        r.rpm(i,1) = W * 9.5493;
        r.dIdt(i,1) = dIdt;
        r.dWdt(i,1) = dWdt;

        % Векторный вид
        [A, B] = DC_motr_vector_eq(R, L, Ci, J, U, Mc);

        dqdt = A*q + B;
        q = q + dqdt*dt;

        r.q1(i,1) = q(1);
        r.q2(i,1) = q(2);
    end

    % Графики при решении методом Эйлера
    getGraphs1(r);

```

Рис. 8. Пример кода основного цикла программы

```

% Математическое моделирование с помощью использования функции "ode"
[A, B] = DC_mottr_vector_eq(R, L, Ci, J, U, Mc);

y = [0; 0]; % [I; W]

dydt = @(t, y) A*y+ B;

% Начальные условия
initial_conditions = [0; 0.0]; % [ток якоря; угловая скорость]

% Время симуляции
tspan = [0, 1];

% Решение дифференциального уравнения
[t, y] = ode45(dydt, tspan, initial_conditions);

% Графики при решении методом "ode"
getGraphs2(t, y, Ci);
end

```

Рис. 9. Пример программного кода математической модели ДПТ НВ, где решение происходит с помощью функции «ode»

```

function [A, B] = DC_mottr_vector_eq(R, L, Ci, J, U, Mc)
    A = [-R/L, -Ci/L;
         Ci/J,    0];

    B = [U/L;
         -Mc/J];
end

function getGraphs1(r)
    figure(1)
    subplot(2,1,2);
    plot(r.t, r.rpm);
    title('Скорость вращения');
    xlabel('Время (с)');
    ylabel('Скорость (rpm)');
    grid on;

    subplot(2,1,1);
    plot(r.t, r.I);
    title('Ток якоря');
    xlabel('Время (с)');
    ylabel('Ток (А)');
    grid on;

    figure(2)
    plot(r.tau, r.rpm, r.I, r.rpm)
    xlabel('M [mNm]')
    ylabel('n [rpm]')
    legend('n [rpm] (M [mNm])', 'n [rpm] (I [A])')
    grid on;
end

function getGraphs2(t, y, Ci)
    figure(3)
    subplot(2,1,1);
    plot(t, y(:,1));
    title('Ток якоря');
    xlabel('Время (с)');
    ylabel('Ток (А)');
    grid on;

    subplot(2,1,2);
    plot(t, y(:,2));
    title('Скорость вращения');
    xlabel('Время (с)');
    ylabel('Скорость (об/мин)');
    grid on;

    figure(4)
    plot(Ci * y(:,1) * 1000, y(:,2) * 9.5493, ...
         y(:,1), y(:,2) * 9.5493)
    |
    xlabel('M [mNm]')
    ylabel('n [rpm]')
    legend('n [rpm] (M [mNm])', 'n [rpm] (I [A])')
    grid on;
end

```

Рис. 10. Программный код вспомогательных функции

Характеристики двигателя приведены в файле приложения 1.

6. Создать файл рабочей программы с математической моделью работы ДПТ НВ и представить результаты.

Пример вывода результатов представлен на рисунке 11.

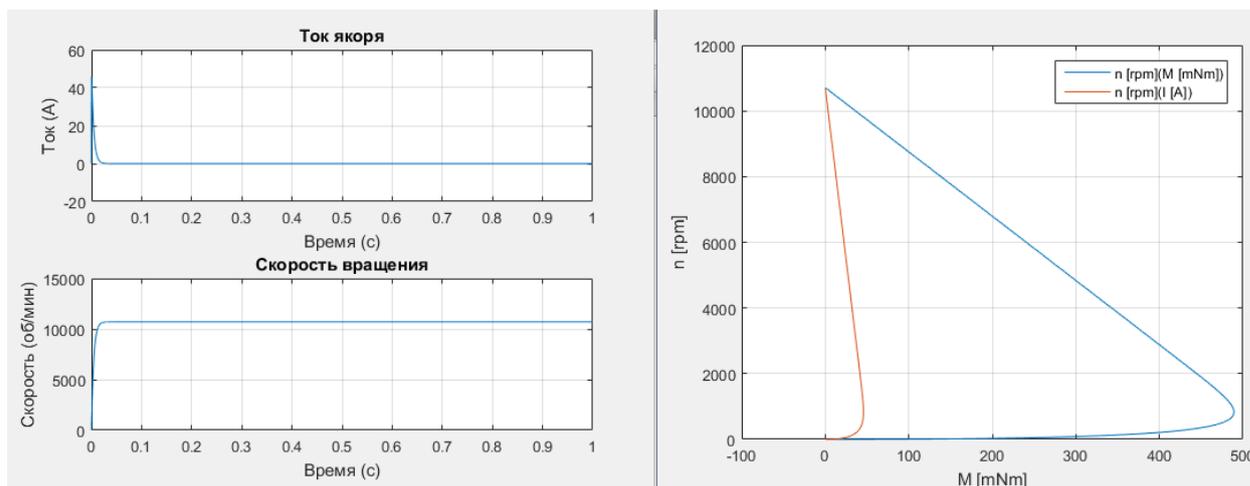


Рис. 11. Пример вывода результатов

### МАТЕРИАЛЫ ДЛЯ СПРАВКИ

« $\text{dydt} = @(t, y) \dots$ »: Это определение анонимной функции MATLAB, « $t$ » - это время, а « $y$ » - это вектор состояния системы. Функция возвращает производные состояний по времени.

« $A*y + B$ »: Это выражение определяет производную по времени для обоих состояний.

### СОДЕРЖАНИЕ ОТЧЁТА

В отчёте о проделанной работе должно содержаться следующая информация:

- Название лабораторной работы и её цель.
- Расчетная схема объекта моделирования;
- Программный код;
- Рабочий файл программы;
- Графики изменения характеристик объектов;

- Проверка адекватности разработанной модели путем анализа получаемых результатов, выводы по работе;
- Список использованной литературы.

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие функции используются для описания математической модели ДПТ.
2. Каковы особенности математического моделирования механизмов в MATLAB.
3. Что такое численное интегрирование, дифференцирование и решатель дифференциальных уравнений, их формулы
4. Как проверить адекватность построенной модели?

## 2 МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ МАНИПУЛЯТОРА

### *Цель работы:*

Провести кинематический анализ манипулятора, заданной конфигурации.

### *Задачи:*

1. Проанализировать заданный манипулятор и определить:
  - 1.1. конфигурацию манипулятора;
  - 1.2. количество твёрдых тел, из которых состоит манипулятор;
  - 1.3. количество кинематических пар и их типы.
2. Составить расчётную схему манипулятора, заданной конфигурации.
3. Составить аналитическое описание модели манипулятора, заданной конфигурации:
  - 3.1. Написать уравнения решения прямой задачи кинематики;
  - 3.2. Написать уравнения решения обратной задачи кинематики.
4. Описать желаемый закон движения исполнительного звена манипулятора.
5. Составить блок-схему алгоритма численного моделирования.
6. Провести численное математическое моделирование движения.
7. Представить результаты и провести их критический анализ.

### ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

На рисунке 1 изображена твердотельная модель манипулятора.

Решение задачи 1

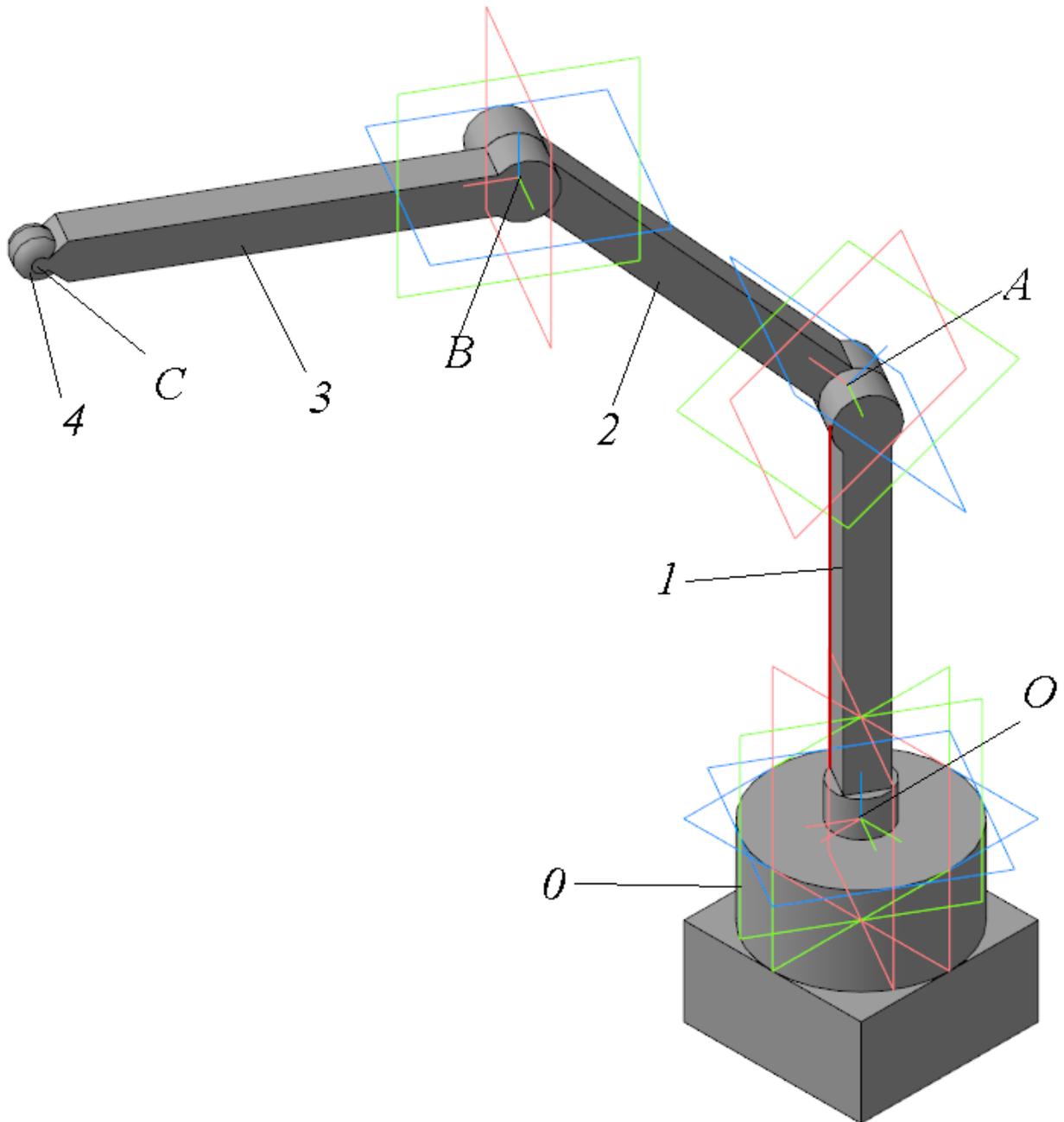


Рис. 1. Твёрдотельная модель манипулятора

В ходе анализа объекта, представленного на рисунке 1, было установлено, что:

- объект является **пространственным четырёхзвенным манипулятором**;
- манипулятор состоит из **4 твёрдых тел** (рис. 1): основания – 0, вертикального звена – 1, двух горизонтальных звеньев – 2, 3

и исполнительного звена – 4, установленного на конце звена 3 в точке  $C$ ;

- манипулятор состоит из **трёх кинематических пар**: в точке  $O$  вращательная пара относительно вертикальной оси, в точках  $A$  и  $B$  вращательные пары относительно фронтальной оси.

### Решение задачи 2

По результатам анализа, составим расчётную схему пространственного четырёхзвенного манипулятора (рис. 2).

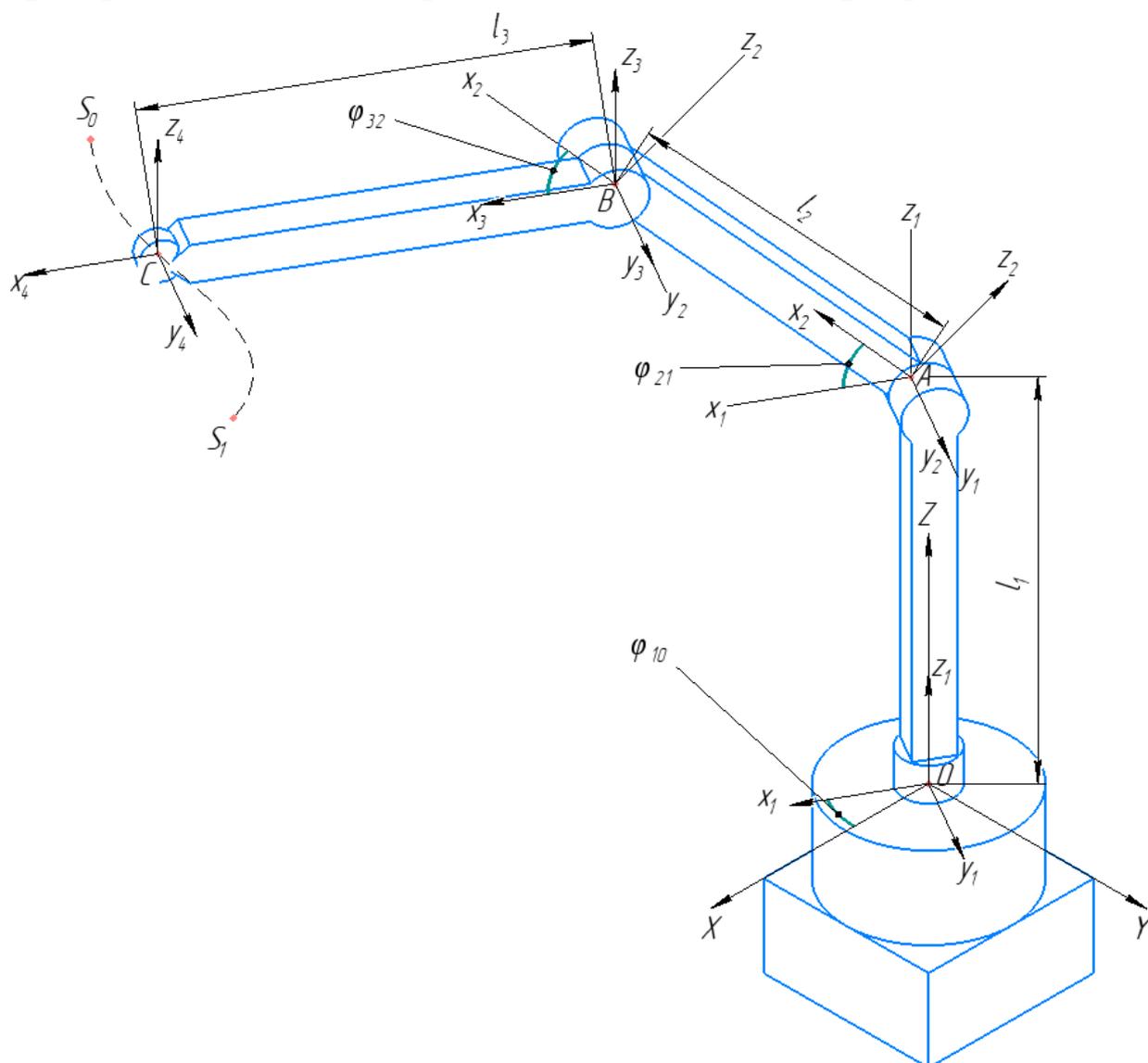


Рис. 2. Расчётная схема манипулятора

Система координат (СК)  $XYZ$  с центром в точке  $O$ , является абсолютной системой отчёта. Векторы и переменные, описываемые

в СК  $XYZ$ , помечаются верхним индексом  $(0)$ . СК  $x_1y_1z_1$  с центром в точке  $O$ , СК  $x_2y_2z_2$  с центром в точке  $A$ , СК  $x_3y_3z_3$  с центром в точке  $B$  и СК  $x_4y_4z_4$  с центром в точке  $C$  являются относительными СК, связанными с телами 1, 2, 3 и 4 соответственно. Векторы и переменные, описываемые в этих СК, помечаются индексами:  $(1)$ ,  $(2)$ ,  $(3)$ ,  $(4)$  в верхнем регистре.

Пусть точка  $C$  движется по заданной траектории из точки  $S_0$  в точку  $S_1$ . Вектор  $\vec{r}_{CO}^{(0)}(\bar{q})$ , описывающий координаты центра исполнительного звена, рассчитывается в зависимости от значений вектора собственных обобщённых координат, который имеет вид:  $\bar{q} = (\varphi_{10} \ \varphi_{21} \ \varphi_{32})^T$ . состоящий из  $\varphi_{10}$  - угол поворота звена 1 относительно звена 0, вокруг оси  $OZ^{(0)}$ .  $\varphi_{21}$  - угол поворота звена 2 относительно звена 1, вокруг оси  $Ay_2^{(2)}$ .  $\varphi_{32}$  - угол поворота звена 3 относительно звена 2, вокруг оси  $Bz_3^{(3)}$ . Длины звеньев обозначены как:  $l_1$  – длина 1-го звена;  $l_2$  – длина 2-го звена;  $l_3$  – длина 3-го звена.

### Решение задачи 3

#### Прямая задача кинематики

Прямая задача кинематики робота-манипулятора заключается в расчёте положения и ориентации исполнительного звена на основе известных углов или положений суставов робота. Суть этой задачи заключается в том, чтобы рассчитать, в каких координатах и как будет ориентировано исполнительное звено (численные значения вектора  $\vec{r}_{CO}^{(0)}$ ), если известна конфигурация манипулятора и какие значения углов в суставах робота (численные значения вектора  $\bar{q}$ ). Это необходимо для управления роботом и планирования траекторий его движений.

Уравнение прямой задачи кинематики:

$$\vec{r}_{CO}^{(0)} = R_{10}(\varphi_{10}) \cdot \vec{r}_{AO}^{(1)} + (R_{10}(\varphi_{10}) \cdot R_{21}(\varphi_{21})) \cdot \vec{r}_{BA}^{(2)} + (R_{10}(\varphi_{10}) \cdot R_{21}(\varphi_{21}) \cdot R_{32}(\varphi_{32})) \cdot \vec{r}_{CB}^{(3)}. \quad (0)$$

Матрицы  $R_{10}(\varphi_{10})$ ,  $R_{21}(\varphi_{21})$  и  $R_{32}(\varphi_{32})$  используются для расчёта численных значений векторов каждой из точек в  $(0)$  СК в трёхмерном пространстве. Эти матрицы называются матрицами

вращения вокруг осей декартовой системы координат и записываются как:

- Вращение вокруг оси  $OZ$  на угол  $\varphi_{10}$ :

$$R_{10}(\varphi_{10}) = \begin{pmatrix} \cos \varphi_{10} & -\sin \varphi_{10} & 0 \\ \sin \varphi_{10} & \cos \varphi_{10} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (6)$$

- Вращение вокруг оси  $Ay_2$  на угол  $\varphi_{21}$ :

$$R_{21}(\varphi_{21}) = \begin{pmatrix} \cos \varphi_{21} & 0 & \sin \varphi_{21} \\ 0 & 1 & 0 \\ -\sin \varphi_{21} & 0 & \cos \varphi_{21} \end{pmatrix}. \quad (5)$$

- Вращение вокруг оси  $Bu_3$  на угол  $\varphi_{32}$ :

$$R_{32}(\varphi_{32}) = \begin{pmatrix} \cos \varphi_{32} & 0 & \sin \varphi_{32} \\ 0 & 1 & 0 \\ -\sin \varphi_{32} & 0 & \cos \varphi_{32} \end{pmatrix}. \quad (5)$$

Любое вращение в трёхмерном пространстве может быть представлено как композиция поворотов вокруг трёх ортогональных осей (например, вокруг осей декартовых координат).

Определим координаты точки  $O$  и её вектора  $\vec{r}_O^{(0)}$  в  $^{(0)}$  СК:

$$O^{(0)} = (0, 0, 0), \quad (6)$$

$$\vec{r}_O^{(0)} = (0 \ 0 \ 0)^T. \quad (7)$$

Определим координаты точек  $O, A$  и их вектора  $\vec{r}_{AO}^{(1)}$  в  $^{(1)}$  СК:

$$O^{(1)} = (0, 0, 0), \quad (8)$$

$$A^{(1)} = (0, 0, l_1), \quad (9)$$

$$\vec{r}_{AO}^{(1)} = A^{(1)} - O^{(1)} = (0 \ 0 \ l_1)^T. \quad (10)$$

Определим координаты точек  $A, B$  и их вектора  $\vec{r}_{BA}^{(2)}$  в  $^{(2)}$  СК:

$$A^{(2)} = (0, 0, 0), \quad (11)$$

$$B^{(2)} = (l_2, 0, 0), \quad (12)$$

$$\vec{r}_{BA}^{(2)} = B^{(2)} - A^{(2)} = (l_2 \ 0 \ 0)^T. \quad (13)$$

Определим координаты точек  $B, C$  и их вектора  $\vec{r}_{CB}^{(3)}$  в  $^{(3)}$  СК:

$$B^{(3)} = (0, 0, 0), \quad (14)$$

$$C^{(3)} = (l_3, 0, 0), \quad (15)$$

$$\vec{r}_{CB}^{(3)} = C^{(3)} - B^{(3)} = (l_3 \ 0 \ 0)^T. \quad (16)$$

Определим вектор каждой из точек в  ${}^{(0)}$  СК.

Для расчёта ориентации звена 1 в глобальной СК используется матрица поворота вокруг оси  $OZ$ , согласно (6) рассчитаем положение вектора  $\bar{r}_{AO}^{(0)}$ , зависящее от угла поворота  $\varphi_{10}$  как:

$$\bar{r}_{AO}^{(0)} = \bar{r}_O^{(0)} + R_{10}(\varphi_{10}) \cdot \bar{r}_{AO}^{(1)}; \quad (17)$$

$$\bar{r}_{AO}^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \cos \varphi_{10} & -\sin \varphi_{10} & 0 \\ \sin \varphi_{10} & \cos \varphi_{10} & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ l_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ l_1 \end{pmatrix}. \quad (18)$$

Для расчёта ориентации звена 2 в глобальной СК используются матрицы поворота вокруг осей  $Ay_2$  и  $OZ$ , согласно (5, б) рассчитаем положение вектора  $\bar{r}_{BO}^{(0)}$ , зависящее от углов поворота  $\varphi_{10}$  и  $\varphi_{21}$  как:

$$\bar{r}_{BO}^{(0)} = \bar{r}_{AO}^{(0)} + (R_{10}(\varphi_{10}) \cdot R_{21}(\varphi_{21})) \cdot \bar{r}_{BA}^{(2)}; \quad (19)$$

$$\bar{r}_{BO}^{(0)} = \begin{pmatrix} 0 \\ 0 \\ l_1 \end{pmatrix} + \left( \begin{pmatrix} \cos \varphi_{10} & -\sin \varphi_{10} & 0 \\ \sin \varphi_{10} & \cos \varphi_{10} & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos \varphi_{21} & 0 & \sin \varphi_{21} \\ 0 & 1 & 0 \\ -\sin \varphi_{21} & 0 & \cos \varphi_{21} \end{pmatrix} \right) \cdot \begin{pmatrix} l_2 \\ 0 \\ 0 \end{pmatrix}; \quad (20)$$

$$\bar{r}_{BO}^{(0)} = \begin{pmatrix} 0 \\ 0 \\ l_1 \end{pmatrix} + \begin{pmatrix} \cos \varphi_{10} \cdot \cos \varphi_{21} & -\sin \varphi_{10} & \cos \varphi_{10} \cdot \sin \varphi_{21} \\ \cos \varphi_{21} \cdot \sin \varphi_{10} & \cos \varphi_{10} & \sin \varphi_{10} \cdot \sin \varphi_{21} \\ -\sin \varphi_{21} & 0 & \cos \varphi_{21} \end{pmatrix} \cdot \begin{pmatrix} l_2 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} l_2 \cdot \cos \varphi_{10} \cdot \cos \varphi_{21} \\ l_2 \cdot \cos \varphi_{21} \cdot \sin \varphi_{10} \\ l_1 - l_2 \cdot \sin \varphi_{21} \end{pmatrix}. \quad (21)$$

Обратите внимание, что порядок перемножения матриц имеет важное значение! В данном примере порядок перемножения матриц следующий:  $R_{10}(\varphi_{10})$ , затем  $R_{21}(\varphi_{21})$ , поскольку звенья вращаются сначала вокруг оси  $OZ$ , а затем вокруг оси  $Ay_2$ .

Для расчёта ориентации исполнительного звена, находящегося на звене 3 в глобальной СК используются матрицы поворота вокруг осей  $OZ$ ,  $Ay_2$  и  $Bu_3$ , согласно (5, б) рассчитаем положение вектора  $\bar{r}_{CO}^{(0)}$ , зависящее от углов поворота  $\varphi_{10}$ ,  $\varphi_{21}$  и  $\varphi_{32}$  как:

$$\bar{r}_{CO}^{(0)} = \bar{r}_{BO}^{(0)} + (R_{10}(\varphi_{10}) \cdot R_{21}(\varphi_{21}) \cdot R_{32}(\varphi_{32})) \cdot \bar{r}_{CB}^{(3)}; \quad (22)$$

$$\bar{r}_{CO}^{(0)} = \bar{r}_{BO}^{(0)} + \left( \begin{pmatrix} \cos \varphi_{10} & -\sin \varphi_{10} & 0 \\ \sin \varphi_{10} & \cos \varphi_{10} & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos \varphi_{21} & 0 & \sin \varphi_{21} \\ 0 & 1 & 0 \\ -\sin \varphi_{21} & 0 & \cos \varphi_{21} \end{pmatrix} \cdot \begin{pmatrix} \cos \varphi_{32} & 0 & \sin \varphi_{32} \\ 0 & 1 & 0 \\ -\sin \varphi_{32} & 0 & \cos \varphi_{32} \end{pmatrix} \right) \cdot \begin{pmatrix} l_3 \\ 0 \\ 0 \end{pmatrix}; \quad (23)$$

$$\bar{r}_{CO}^{(0)} = \begin{pmatrix} l_2 \cdot \cos \varphi_{10} \cdot \cos \varphi_{21} \\ l_2 \cdot \cos \varphi_{21} \cdot \sin \varphi_{10} \\ l_1 - l_2 \cdot \sin \varphi_{21} \end{pmatrix} + \begin{pmatrix} \cos \varphi_{10} \cdot \cos(\varphi_{21} + \varphi_{32}) & -\sin \varphi_{10} & \cos \varphi_{10} \cdot \sin(\varphi_{21} + \varphi_{32}) \\ \cos(\varphi_{21} + \varphi_{32}) \cdot \sin \varphi_{10} & \cos \varphi_{10} & \sin \varphi_{10} \cdot \sin(\varphi_{21} + \varphi_{32}) \\ -\sin(\varphi_{21} + \varphi_{32}) & 0 & \cos(\varphi_{21} + \varphi_{32}) \end{pmatrix} \cdot \begin{pmatrix} l_3 \\ 0 \\ 0 \end{pmatrix}; \quad (24)$$

$$\bar{r}_{CO}^{(0)} = \begin{pmatrix} X_{CO}^{(0)} \\ Y_{CO}^{(0)} \\ Z_{CO}^{(0)} \end{pmatrix} = \begin{pmatrix} \cos \varphi_{10} \cdot (l_3 \cdot \cos(\varphi_{21} + \varphi_{32}) + l_2 \cdot \cos \varphi_{21}) \\ \sin \varphi_{10} \cdot (l_3 \cdot \cos(\varphi_{21} + \varphi_{32}) + l_2 \cdot \cos \varphi_{21}) \\ l_1 - l_2 \cdot \sin \varphi_{21} - l_3 \cdot \sin(\varphi_{21} + \varphi_{32}) \end{pmatrix}. \quad (25)$$

### Обратная задача кинематики

Обратная задача кинематики робота-манипулятора заключается в расчёте углов или положений суставов робота, которые необходимы для достижения заданного положения и ориентации исполнительного звена (например, захвата или инструмента).

Условимся, что далее говоря про вектор  $\bar{r}_{CO}^{(0)}$  будем иметь ввиду его описание в СК  $XYZ$ , тогда :  $\bar{r}_{CO} = \bar{r}_{CO}^{(0)}$ .

Вектор  $\bar{r}_{CO}$  зависит как от времени, так и от обобщённых координат ( ):

$$\bar{r}_{CO}(t) = \bar{r}_{CO}(\bar{q}). \quad (26)$$

Взяв частную производную по времени от ( ) получим ( )

$$\frac{\partial \bar{r}_{CO}(t)}{\partial t} = \frac{\partial \bar{r}_{CO}(\bar{q})}{\partial t} \cdot \frac{\partial \bar{q}}{\partial t}. \quad (27)$$

Выделим вектор производных обобщённых координат из (26) и получим (27):

$$\dot{\bar{r}}_{CO} = J \cdot \dot{\bar{q}}, \quad (28)$$

где:  $J$  - **матрица Якоби**, являющаяся производной векторной функции от векторного аргумента (матрица частных производных).

Матрица Якоби  $J(\bar{q})$  - матрица частных производных функции прямой кинематики по каждой из обобщённых координат, рассчитывается как ( ):

$$J = \begin{pmatrix} \frac{\partial X_{CO}^{(0)}}{\partial \varphi_{10}} & \frac{\partial X_{CO}^{(0)}}{\partial \varphi_{21}} & \frac{\partial X_{CO}^{(0)}}{\partial \varphi_{32}} \\ \frac{\partial Y_{CO}^{(0)}}{\partial \varphi_{10}} & \frac{\partial Y_{CO}^{(0)}}{\partial \varphi_{21}} & \frac{\partial Y_{CO}^{(0)}}{\partial \varphi_{32}} \\ \frac{\partial Z_{CO}^{(0)}}{\partial \varphi_{10}} & \frac{\partial Z_{CO}^{(0)}}{\partial \varphi_{21}} & \frac{\partial Z_{CO}^{(0)}}{\partial \varphi_{32}} \end{pmatrix}$$

$$J = \begin{pmatrix} -\sin \varphi_{10} \cdot (l_3 \cdot \cos(\varphi_{21} + \varphi_{32}) + l_2 \cdot \cos \varphi_{21}) & -\cos \varphi_{10} \cdot (l_3 \cdot \sin(\varphi_{21} + \varphi_{32}) + l_2 \cdot \sin \varphi_{21}) & -l_3 \cdot \sin(\varphi_{21} + \varphi_{32}) \cdot \cos \varphi_{10} \\ \cos \varphi_{10} \cdot (l_3 \cdot \cos(\varphi_{21} + \varphi_{32}) + l_2 \cdot \cos \varphi_{21}) & -\sin \varphi_{10} \cdot (l_3 \cdot \sin(\varphi_{21} + \varphi_{32}) + l_2 \cdot \sin \varphi_{21}) & -l_3 \cdot \sin(\varphi_{21} + \varphi_{32}) \cdot \sin \varphi_{10} \\ 0 & -l_3 \cdot \cos(\varphi_{21} + \varphi_{32}) - l_2 \cdot \cos \varphi_{21} & -l_3 \cdot \cos(\varphi_{21} + \varphi_{32}) \end{pmatrix}$$

Выразим  $\dot{\bar{q}}$  из (29) и получим :

$$\dot{\bar{q}} = J^{-1} \cdot \dot{\bar{r}}_{CO}. \quad (29)$$

Для численного решения обратной задачи кинематики необходимо перейти от непрерывного времени описания времени к дискретному ():

$$t = t + \Delta t, \quad (30)$$

где:  $\Delta t$  - малое, постоянное приращение (рис. 3).

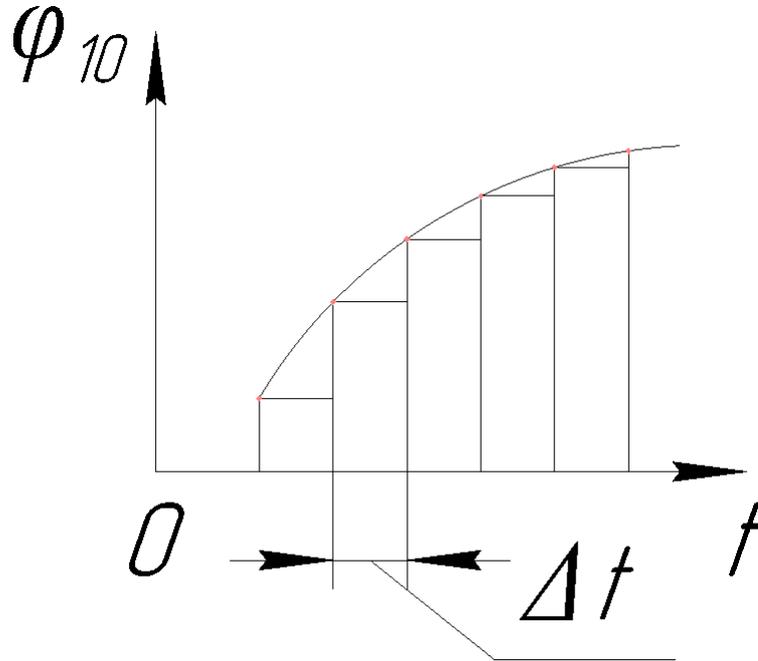


Рис. 3. Дискретизация непрерывного сигнала

Тогда численно, производную вектора обобщённых координат можно записать в виде ():

$$\dot{\bar{q}} = \frac{\bar{q}^k - \bar{q}^{k-1}}{\Delta t}. \quad (31)$$

Выразим  $\bar{q}$  на  $k$ -м как ( ):

$$\bar{q}^k = \bar{q}^{k-1} + J^{-1} \cdot \Delta \bar{r}_{CO},$$

где:  $\Delta \bar{r}_{CO} = \bar{r}_{CO}^{k-1} - \bar{r}_{CO}^{k-2}$  - приращение функции  $\bar{r}_{CO}$  на временном шаге  $\Delta t$ .

Полученное соотношение позволяет находить вектор обобщенных координат  $\bar{q}$  на  $k$ -м временном шаге по известному значению  $\bar{q}$  на  $(k-1)$ -м шаге.

Обратная задача кинематики позволяет планировать траекторию движения робота. Если известно, куда должно переместиться исполнительное звено, можно вычислить, как должны измениться углы суставов, чтобы достичь этой цели.

## Решение задачи 4

### Планирование траектории

Определим простую траекторию движения, определяемую ломанной кривой (рис. 4 а)) на временном промежутке  $t \in (t_0 \ T)$ . Начальная и конечная точки траектории  $S_0$  и  $S_1$  описываются координатами:  $S_0^{(0)} = (X_{S_0}, Y_{S_0}, Z_{S_0})$  и  $S_1^{(0)} = (X_{S_1}, Y_{S_1}, Z_{S_1})$ .

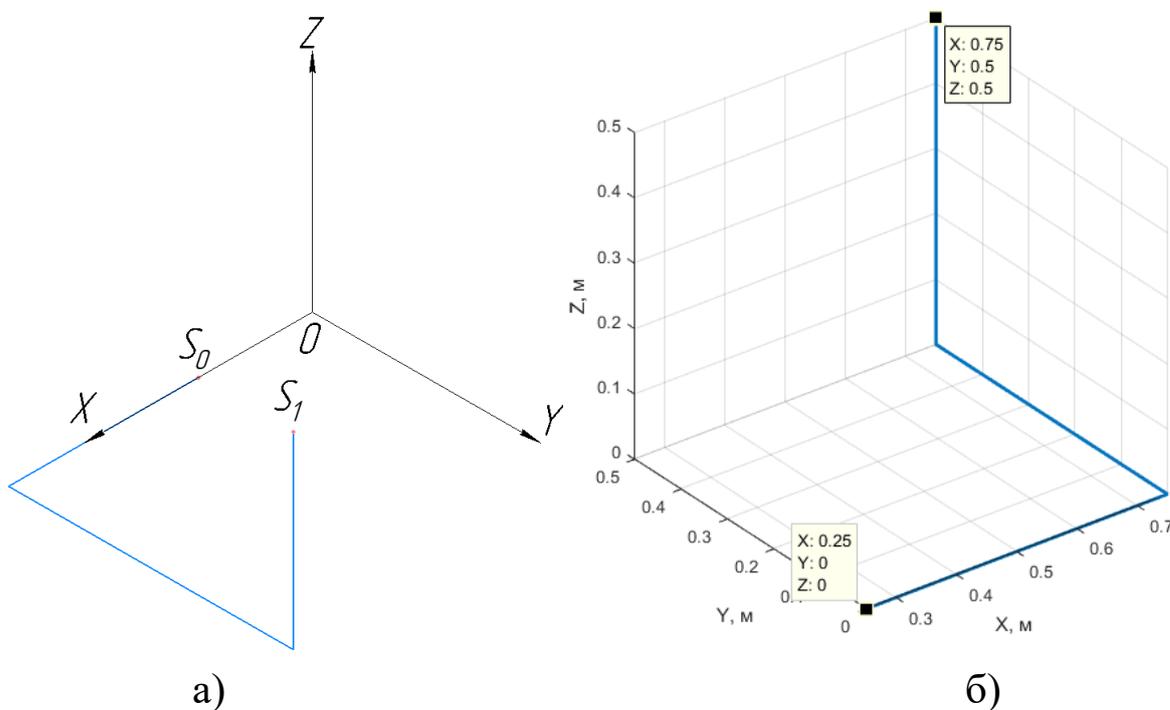


Рис. 4. Заданная траектория движения: а) схема; б) график с численными значениями

Установим зависимости траектории от времени.

$$\text{Если } t \in \left( t_0 \quad \frac{T}{3} \right), \text{ то: } S^{(0)} = \left( X_{s_0} + \left( t - \frac{T}{3} \right) \cdot \frac{1}{2}, Y_{s_0}, Z_{s_0} \right).$$

$$\text{Если } t \in \left( \frac{T}{3} \quad \frac{2 \cdot T}{3} \right), \text{ то: } S^{(0)} = \left( X_{s_0} + \frac{T}{3} \cdot \frac{1}{2}, Y_{s_0} + \left( \left( t - \frac{T}{3} \right) \cdot \frac{T}{3} \right) \cdot \frac{1}{2}, Z_{s_0} \right).$$

$$\text{Если } t \in \left( \frac{2 \cdot T}{3} \quad T \right), \text{ то: } S^{(0)} = \left( X_{s_0} + \frac{T}{3} \cdot \frac{1}{2}, Y_{s_0} + \frac{T}{3} \cdot \frac{1}{2}, Z_{s_0} + \left( \left( t - \frac{2 \cdot T}{3} \right) \cdot \frac{T}{3} \right) \cdot \frac{1}{2} \right).$$

Определим временные интервалы:  $t_0 = 0$  сек,  $T = 3$  сек,  $S_0^{(0)} = (0.25, 0, 0)$  м, тогда следуя условиям выше:  $S_1^{(0)} = (0.75, 0.5, 0.5)$  м. График изменения заданной траектории представлен на рисунке 4 б).

### Решение задачи 5

На рисунке 5 представлена блок-схема алгоритма численного математического моделирования прямой и обратной задач кинематики пространственного трёхзвенного манипулятора.

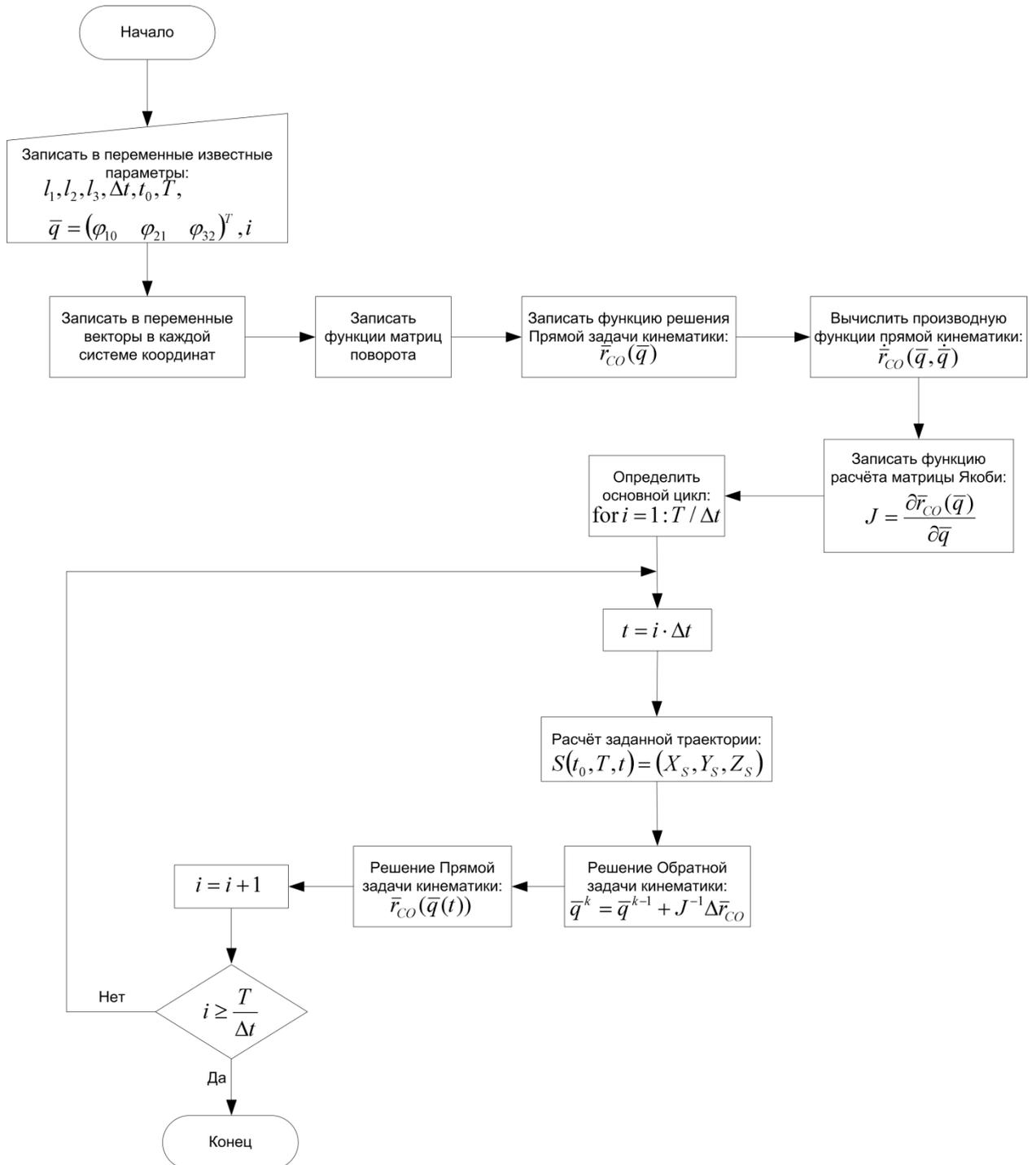


Рис. 5. Блок-схема алгоритма численного математического моделирования

## Решение задачи 6

На рисунке 6 представлен пример программного кода MATLAB-программы для численной математической модели пространственного движения трёхзвенного манипулятора, согласно блок – схеме (рис. 5).

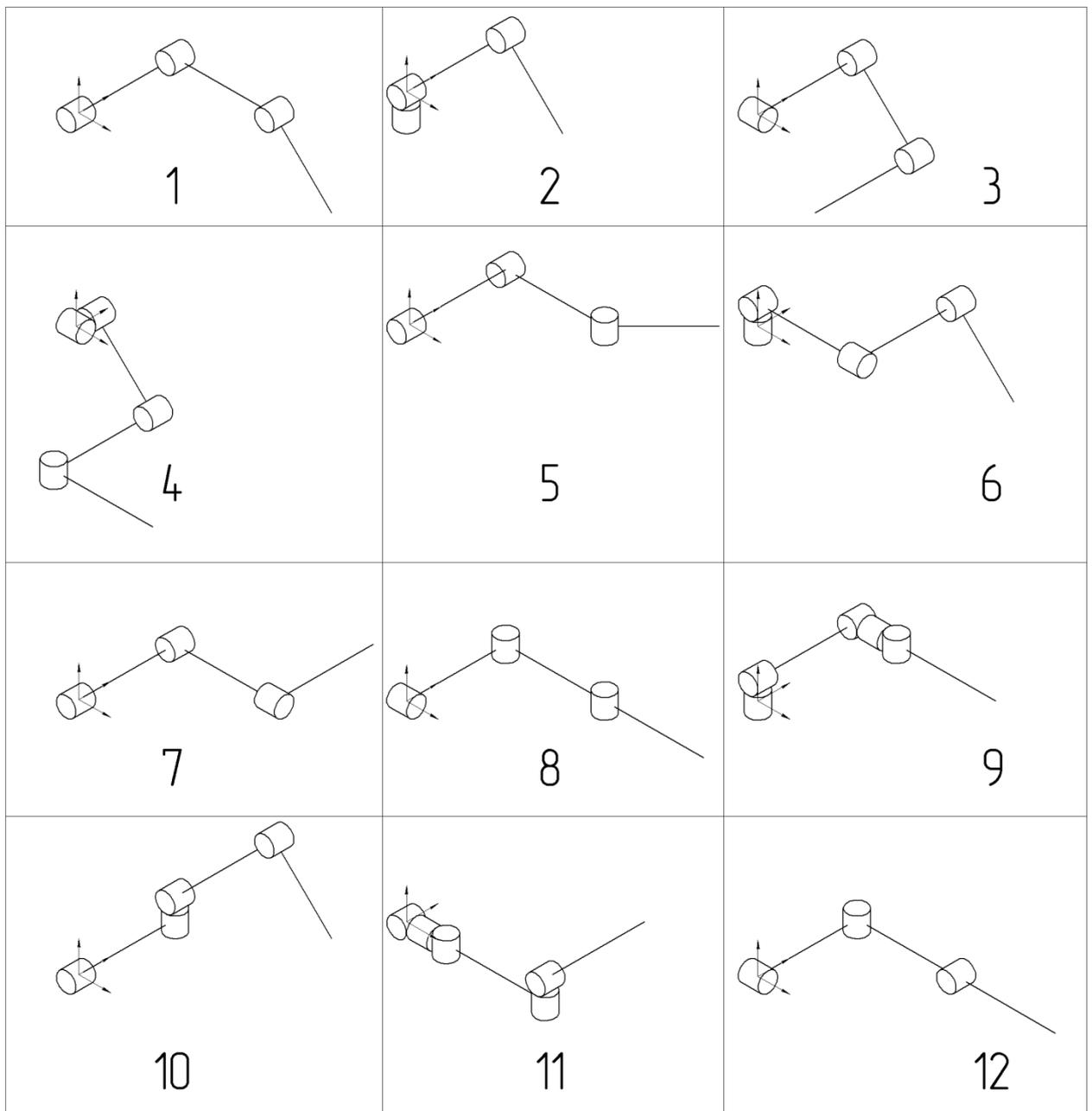


Рис. 6. Пример программного кода численной модели манипулятора

### Решение задачи 7

На рисунках представлен пример вывода результатов численного моделирования прямой и обратной задач кинематики манипулятора.

### ВАРИАНТЫ ЗАДАНИЙ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ



## СОДЕРЖАНИЕ ОТЧЁТА

В отчёте о проделанной работе должно содержаться следующая информация:

- Название лабораторной работы и её цель.
- Расчетная схема объекта моделирования;
- Программный код;
- Рабочий файл программы;
- Графики изменения характеристик объектов;
- Проверка адекватности разработанной модели путем анализа получаемых результатов, выводы по работе;
- Список использованных источников.

## КОНТРОЛЬНЫЕ ВОПРОСЫ

5. Какие функции используются для описания численной математической модели кинематики манипулятора?

6. Как нарисовать расчётную схему манипулятора? Какие основные элементы должны быть на расчётной схеме манипулятора?

7. Что такое фреймы? Как описывается их взаимное расположение?

8. Что такое численное интегрирование, дифференцирование и решатель дифференциальных уравнений, их формулы

9. Как проверить адекватность построенной модели?

### **3. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ КИНЕМАТИКИ КОЛЁСНОГО РОБОТА С ДИФФЕРЕНЦИАЛЬНЫМ ПРИВОДОМ**

#### ***Цель работы:***

Провести кинематический анализ колёсного робота с дифференциальным приводом.

#### ***Задачи:***

1. Проанализировать заданный колёсный робот и определить:
  - 1.1. конфигурацию колёсного робота;
  - 1.2. количество твёрдых тел, из которых состоит колёсный робот;
  - 1.3. количество кинематических пар и их типы.
2. Составить расчётную схему колёсного робота, заданной конфигурации.
3. Составить аналитическое описание модели колёсного робота, заданной конфигурации:
  - 3.1. Написать уравнения решения прямой задачи кинематики;
  - 3.2. Написать уравнения решения обратной задачи кинематики.
4. Описать желаемый закон движения колёсного робота.
5. Составить блок-схему алгоритма численного моделирования.
6. Провести численное математическое моделирование движения.
7. Представить результаты и провести их критический анализ.

### **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

#### **Общее описание**

Дифференциальный колесный робот - это мобильный робот, движение которого основано на работе двух колес с независимым приводом, расположенных по обеим сторонам корпуса робота (рис. 1). Таким образом, он может менять направление движения,

изменяя относительную скорость вращения колес, и, следовательно, ему не требуется дополнительное рулевое управление. Роботы с таким приводом обычно имеют одно или несколько поворотных колес для предотвращения опрокидывания транспортного средства.

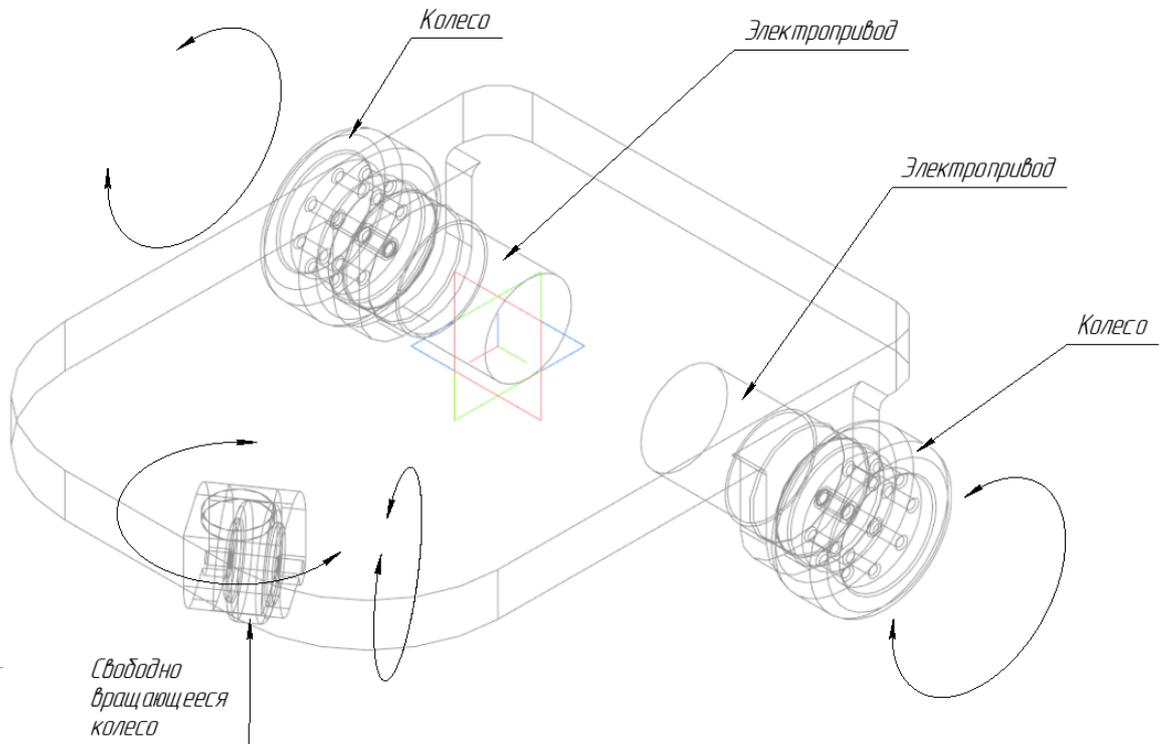


Рис. 1. Схема расположения приводов и колёс трехколесного робота с дифференциальным управлением

## Общее описание кинематики колёсного робота с дифференциальным приводом

### Определение систем координат

#### 1. Анализ робота

Рассмотрим схему колёсного робота, представленную на рисунке 1. Заданный колёсный робот **состоит из:** корпуса, независимого поворотного колеса и двух независимых ведущих колёс с электроприводами, что указывает на то, что это **колёсный робот с дифференциальным приводом (КРсДП)**. Робот состоит из **7-ми твёрдых тел**, движение в пространстве осуществляется с

помощью **четырёх кинематических пар вращения и трёх цилиндр-плоскость.**

## **2. Расчётная схема робота**

Рассмотрим расчётную схему кинематики КРсДП, представленную на рисунке 2. Для описания положения КРсДП в пространстве, необходимо определить две системы координат (СК):

1. Инерциальная система координат: это система координат, представляющая собой глобальную систему отсчёта (СО), зафиксированную в среде или плоскости, в которой происходит движение КРсДП. Эта СК рассматривается как СК в глобальной СО и обозначается как  $XOY$ . Векторы и переменные, описываемые в СК  $XOY$ , помечаются верхним индексом <sup>(0)</sup>.
2. Система координат робота: это СК являющаяся локальной СО, связанной с КРсДП и, таким образом, движущейся вместе с ним. Эта СК обозначается как  $x_1Cy_1$ . Векторы и переменные, описываемые в СК  $x_1Cy_1$ , помечаются верхним индексом <sup>(1)</sup>.

Две, определенные выше, системы координат показаны на рисунке 2, примем, что начало СК робота совпадает с его центром масс.

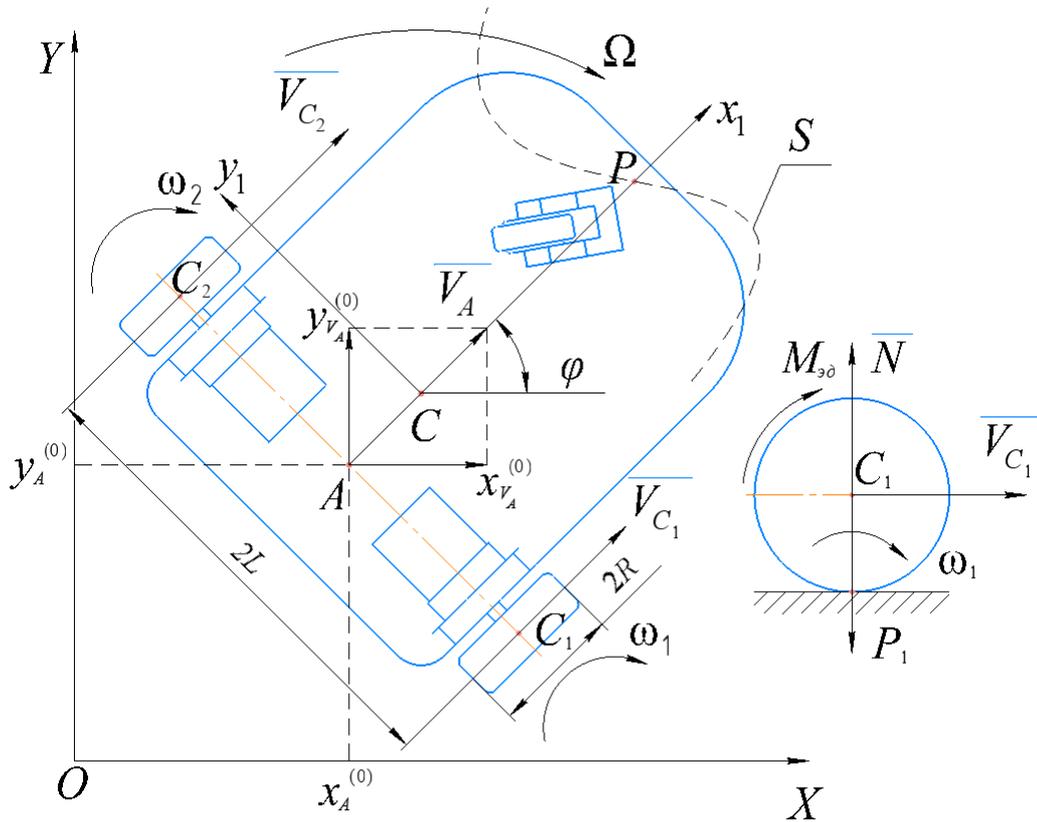


Рис. 2.

Согласно расчётной схеме (рис. 2), положение и ориентация робота в инерциальной СК могут быть определены как вектор (1):

$$\bar{q}^{(0)} = (x_C^{(0)} \quad y_C^{(0)} \quad \varphi)^T, \quad (1)$$

где:  $x_C^{(0)}$  и  $y_C^{(0)}$  - координаты точки  $C$  в СК  $XOY$ ,  $\varphi$  - угол поворота корпуса КРсДП.

Положение любой точки на корпусе робота может быть определено в СК робота  $^{(1)}$  и инерциальной СК  $^{(0)}$ , следующим образом:

Пусть  $\bar{\chi}^{(1)} = (x_C^{(1)} \quad y_C^{(1)} \quad \varphi^{(1)})^T$ , и  $\bar{\chi}^{(0)} = (x_C^{(0)} \quad y_C^{(0)} \quad \varphi^{(0)})^T$  — векторы положения и ориентации точки  $C$  в  $^{(1)}$  и  $^{(0)}$  СК соответственно.

Тогда, связь двух систем координат можно описать следующим преобразованием (2):

$$\bar{\chi}^{(0)} = R_z(\varphi) \bar{\chi}^{(1)} = \begin{pmatrix} x_C^{(0)} \\ y_C^{(0)} \\ \varphi^{(0)} \end{pmatrix} = \begin{pmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_C^{(1)} \\ y_C^{(1)} \\ \varphi^{(1)} \end{pmatrix}, \quad (2)$$

где:  $R_z(\varphi)$  - ортогональная матрица вращения.

Это преобразование также позволит рассчитывать движение между системами координат (3):

$$\dot{\bar{\chi}}^{(0)} = R_z(\varphi)\dot{\bar{\chi}}^{(1)}. \quad (3)$$

Уравнение (3) очень важно для вывода кинематических и динамических моделей КРСДП, поскольку оно описывает взаимосвязь между скоростями в инерциальной системе отсчета и системе отсчета робота.

Для КРСДП основной целью кинематического моделирования является представление скоростей робота как функции скоростей ведущих колес вместе с геометрическими параметрами робота.

### 3. Аналитическая модель

#### 3.1. Решение прямой задачи кинематики

#### Аналитическая модель кинематики колёсного робота с дифференциальным приводом

Вектор линейной скорости КРСДП в СК <sup>(1)</sup>, связанной с роботом является средним значением линейных скоростей каждого ведущего колеса в СК робота, и рассчитывается согласно:

$$\bar{V}_A^{(1)} = \begin{pmatrix} x_{V_A}^{(1)} \\ y_{V_A}^{(1)} \\ z_{V_A}^{(1)} \end{pmatrix} = \begin{pmatrix} (V_{C_1} + V_{C_2})/2 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} R \cdot (\omega_1 + \omega_2)/2 \\ 0 \\ 0 \end{pmatrix}. \quad (5)$$

Вектор скоростей вращения КРСДП рассчитывается согласно:

$$\bar{\Omega} = \begin{pmatrix} 0 \\ 0 \\ \Omega \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ (V_{C_1} - V_{C_2})/(2 \cdot L) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ R \cdot (\omega_1 - \omega_2)/(2 \cdot L) \end{pmatrix}. \quad (6)$$

Вектор обобщённых скоростей КРСДП в СК <sup>(1)</sup> можно представить через скорости точки *A* робота в СК <sup>(1)</sup> следующим образом:

$$\dot{\bar{q}}_A^{(1)} = \begin{pmatrix} x_{V_A}^{(1)} \\ y_{V_A}^{(1)} \\ \Omega \end{pmatrix} = \begin{pmatrix} R \cdot (\omega_1 + \omega_2)/2 \\ 0 \\ R \cdot (\omega_1 - \omega_2)/(2 \cdot L) \end{pmatrix}. \quad (7)$$

Либо в матрично-векторной форме:

$$\dot{\bar{q}}_A^{(1)} = \begin{pmatrix} x_{V_A}^{(1)} \\ y_{V_A}^{(1)} \\ \Omega \end{pmatrix} = \begin{pmatrix} R/2 & R/2 \\ 0 & 0 \\ R/(2 \cdot L) & -R/(2 \cdot L) \end{pmatrix} \cdot \begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix}, \quad (8)$$

После нахождения вектора обобщённых скоростей  $\dot{\bar{q}}_A^{(1)}$  робота в СК  $^{(1)}$  согласно уравнению (3) можно вычислить вектор обобщённых скоростей КРсДП в СК  $^{(0)}$  следующим образом (9):

$$\dot{\bar{q}}_A^{(0)} = R_Z(\varphi) \dot{\bar{q}}_A^{(1)}. \quad (9)$$

Для нахождения вектора обобщённых координат воспользуемся численным интегрированием методом Эйлера. Условимся, что дальнейшие действия происходят в СК  $^{(0)}$ , тогда представим (9) как (10):

$$\bar{q}_A^k = \bar{q}_A^{k-1} + \dot{\bar{q}}_A^k \cdot \Delta t. \quad (10)$$

Альтернативная форма кинематической модели (11) может быть получена путем представления скоростей КРсДП через линейные и угловые скорости КРсДП в системе координат робота:

$$\dot{\bar{q}}_A^{(0)} = \begin{pmatrix} x_{V_A}^{(0)} \\ y_{V_A}^{(0)} \\ \Omega \end{pmatrix} = \begin{bmatrix} \frac{R}{2} \cdot \cos \phi & \frac{R}{2} \cos \phi \\ \frac{R}{2} \cdot \sin \phi & \frac{R}{2} \cdot \sin \phi \\ \frac{R}{2 \cdot L} & -\frac{R}{2 \cdot L} \end{bmatrix} \cdot \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix}. \quad (11)$$

Уравнение (11) отражает аналитическую модель кинематики КРсДП.

Форма записи кинематики (12) может использоваться для решения обратной задачи кинематики КРсДП.

$$\dot{\bar{q}}_A^{(0)} = \begin{pmatrix} x_{V_A}^{(0)} \\ y_{V_A}^{(0)} \\ \Omega \end{pmatrix} = \begin{pmatrix} \cos \phi & 0 \\ \sin \phi & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{V_A}^{(1)} \\ \Omega \end{pmatrix}. \quad (12)$$

### 3.2. Решение обратной задачи кинематики

Для управления мы можем использовать параметры:  $x_{V_A}^{(1)}$ ,  $\Omega$ , применяемые в канонической модели (12) для непосредственного управления двумя компонентами линейной скорости опорной точки  $P$ , закрепленной на шасси робота. Это полезно, например, когда датчик смещён от оси колёс и расположен в точке  $P$ .

Условимся, что  $P^{(0)} = (x_P^{(0)} \ y_P^{(0)})^T$  будут координатами точки  $P$  в мировой СК, а  $P^{(1)} = (x_P^{(1)} \ y_P^{(1)})^T$  будут ее (постоянными) координатами в СК, связанной с роботом. Чтобы вычислить значения  $x_{V_A}^{(1)}$ ,  $\Omega$ , необходимые для достижения желаемого движения в мировой СК  $\dot{P}^{(0)} = (\dot{x}_P^{(0)} \ \dot{y}_P^{(0)})^T$ , мы сначала запишем (13):

$$P^{(0)} = \begin{pmatrix} x_P^{(0)} \\ y_P^{(0)} \end{pmatrix} = \begin{pmatrix} x_A^{(0)} \\ y_A^{(0)} \end{pmatrix} + \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} \cdot \begin{pmatrix} x_P^{(1)} \\ y_P^{(1)} \end{pmatrix}. \quad (13)$$

Продифференцировав, получим (14):

$$\dot{P}^{(0)} = \begin{pmatrix} \dot{x}_P^{(0)} \\ \dot{y}_P^{(0)} \end{pmatrix} = \begin{pmatrix} \dot{x}_{V_A}^{(0)} \\ \dot{y}_{V_A}^{(0)} \end{pmatrix} + \dot{\varphi} \cdot \begin{pmatrix} -\sin \varphi & -\cos \varphi \\ \cos \varphi & -\sin \varphi \end{pmatrix} \cdot \begin{pmatrix} x_P^{(1)} \\ y_P^{(1)} \end{pmatrix}. \quad (14)$$

Подставляя  $\Omega$  вместо  $\dot{\varphi}$  и  $(x_{V_A}^{(1)} \cdot \cos \varphi \ x_{V_A}^{(1)} \cdot \sin \varphi)$  вместо  $(x_P^{(1)} \ y_P^{(1)})$  и решая, получаем (15):

$$\begin{pmatrix} x_{V_A}^{(1)} \\ \Omega \end{pmatrix} = \frac{1}{x_P^{(1)}} \cdot \begin{pmatrix} x_P^{(1)} \cdot \cos \varphi - y_P^{(1)} \cdot \sin \varphi & x_P^{(1)} \cdot \sin \varphi + y_P^{(1)} \cdot \cos \varphi \\ -\sin \varphi & \cos \varphi \end{pmatrix} \cdot \begin{pmatrix} \dot{x}_P^{(0)} \\ \dot{y}_P^{(0)} \end{pmatrix}. \quad (15)$$

Уравнение (15) можно интерпретировать как (16):

$$\begin{pmatrix} x_{V_A}^{(1)} \\ \Omega \end{pmatrix} = J^{-1}(\bar{q}) \cdot \begin{pmatrix} \dot{x}_P^{(0)} \\ \dot{y}_P^{(0)} \end{pmatrix}, \quad (16)$$

где:  $J(\bar{q}) = \begin{pmatrix} \cos \varphi & -y_P^{(1)} \cdot \cos \varphi - x_P^{(1)} \cdot \sin \varphi \\ \sin \varphi & x_P^{(1)} \cdot \cos \varphi - y_P^{(1)} \cdot \sin \varphi \end{pmatrix}$  - матрица Якоби, описывающая между движением точки  $P$  и  $(x_{V_A}^{(1)} \ \Omega)$  в мировой системе отсчёта.

Расчёт желаемых угловых скоростей ведущих колёс КРСДП происходит согласно (17):

$$\begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix} = \begin{pmatrix} 1/R & L/R \\ 1/R & -L/R \end{pmatrix} \cdot \begin{pmatrix} x_{V_A}^{(1)} \\ \Omega \end{pmatrix}. \quad (17)$$

#### 4. Желаемый закон движения

Определим простую траекторию движения, зависящую от времени  $S(t)$  (рис. 3) на временном промежутке  $t \in (t_0 \ T)$ . Начальная и конечная точки траектории  $S_0$  и  $S_1$  описываются координатами:  $S_0^{(0)} = (X_{S_0}, Y_{S_0})$  и  $S_1^{(0)} = (X_{S_1}, Y_{S_1})$ .

Уравнения, описывающие траекторию имеет вид (18):

$$S = \left( 0.6 + \frac{t}{2} \quad \frac{3}{4} \cdot \sin\left(\frac{\pi \cdot t}{4}\right) \right)^T. \quad (18)$$

Определим временные интервалы:  $t_0$  сек,  $T = 10$  сек,  $S_0^{(0)} = (0.6, 0)$  м, тогда следуя условиям выше:  $S_1^{(0)} = (5.6, 0.75)$  м.

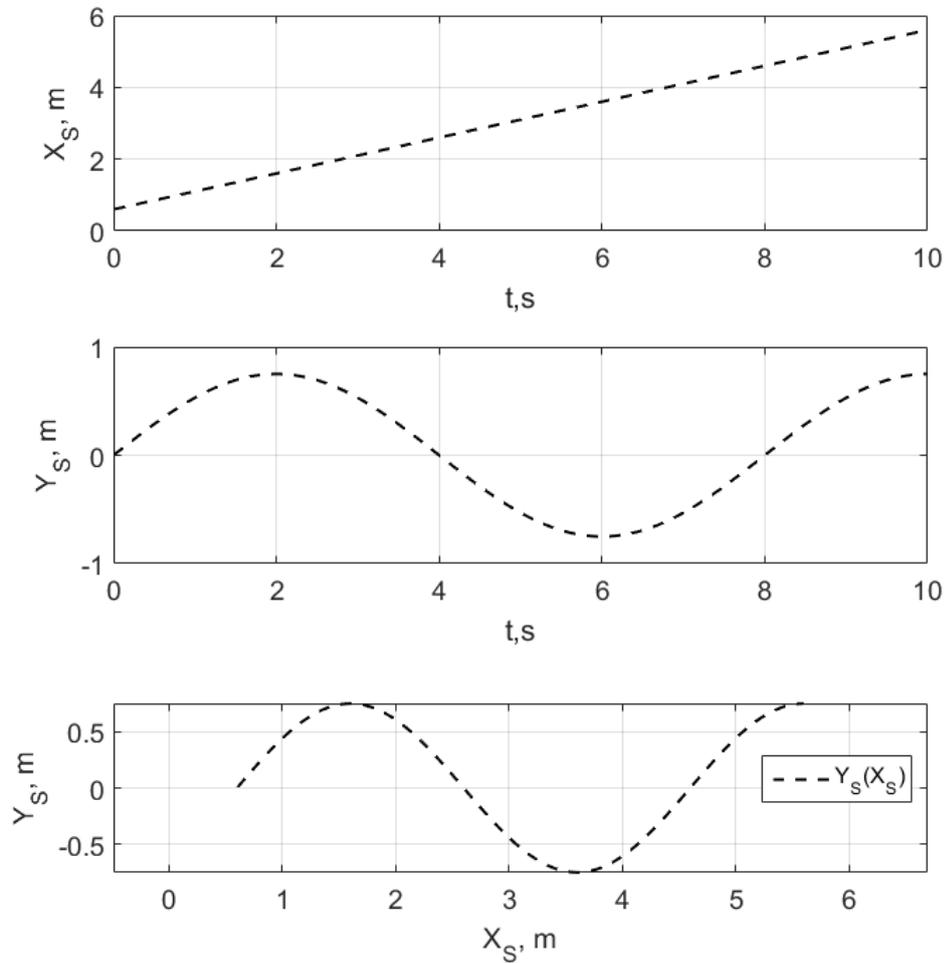


Рис. 3. Заданная траектория движения

## 5. Алгоритм моделирования

На рисунке 4 представлена блок-схема алгоритма численного математического моделирования прямой и обратной задач кинематики КРСДП.

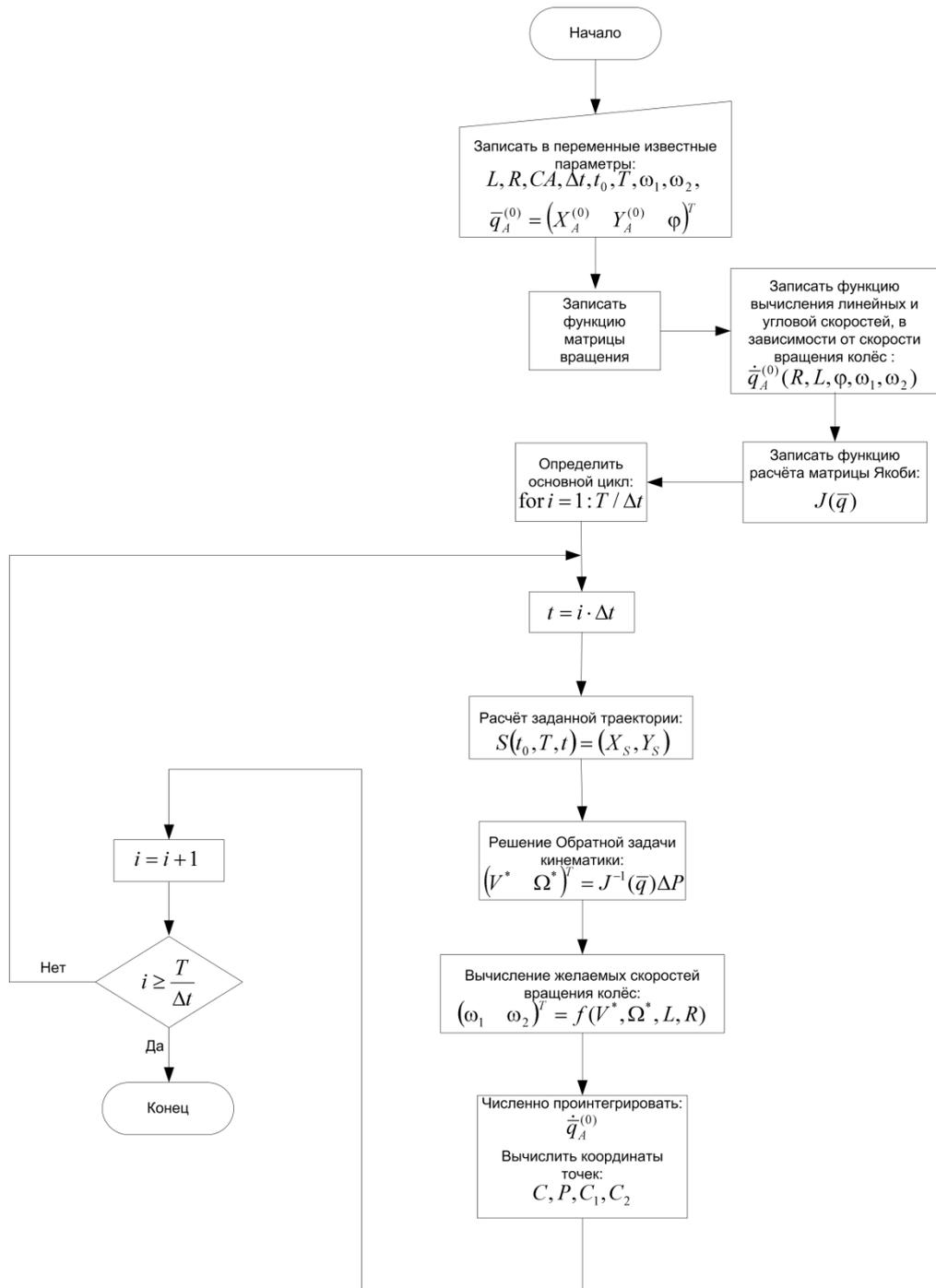


Рис. 4. Блок-схема алгоритма численного математического моделирования

## 6. Численное математическое моделирование

На рисунке 5 представлен пример программного кода MATLAB-программы для численной математической модели пространственного движения трёхзвенного манипулятора, согласно блок – схеме (рис. 4).

Программный код функций, отвечающих за анимацию, представлен в приложении.

```

1  function MMSiR_LR_3_it2
2  -     anim = crtAnimFig; % get animation figure
3
4  -     time = 10; dt = 0.01; count = floor(time/dt); % time
5  -     L = 0.2; R = 0.1; % KRSDP parameters
6  -     q_A_0 = [0; 0; 0]; phi = q_A_0(3);
7  -     CA = 0.25; CA1 = [0; -L]; CA2 = [0; L];
8  -     w1 = 0; w2 = 0;
9
10 -     r = [];
182  % Forward kinematics matrix
183  function ret = matrix(R, L, q)
184  -     ret = [(R*cos(q))/2, (R*cos(q))/2;
185  -           (R*sin(q))/2, (R*sin(q))/2;
186  -           R/(2*L),      -R/(2*L)];
187  -     end
188  |
189  % Rotation matrix
190  function ret = matrix2(q)
191  -     ret = [cos(q), -sin(q);
192  -           sin(q),  cos(q)];
193  -     end
194
195  % Jacobian matrix o kinematics of KRSDP
196  function ret = mJ(q, x, y)
197  -     ret = [cos(q), -y*cos(q)-x*sin(q);
198  -           sin(q), x*cos(q) - y*cos(q)];
199  -     end

```

```

12 - for i = 1:count
13 -     t = i*dt;
14
15 -     clearingAnimPoints(anim); % clearing anim points
16
17 -     w1 = W1*25; w2 = W2*25; % Control
18
19 -     % Forward kinematics task solution
20 -     dq_A_0 = matrix(R, L, phi) * [w1; w2];
21 -     q_A_0 = q_A_0 + (dq_A_0 * dt);
22 -     A_0 = [q_A_0(1); q_A_0(2)];
23
24 -     % Another points of robot
25 -     C = A_0 + matrix2(phi) * [CA; 0];
26 -     C1 = A_0 + matrix2(phi) * CA1; C2 = A_0 + matrix2(phi) * CA2;
27 -     P = A_0 + matrix2(phi) * [0.45; 0]; % Point with sensor
28
29 -     s = [0.6 + (t*0.5); sin(0.25*pi*t)*0.75]; % Desired trajectory
30
31 -     X_A_0 = q_A_0(1); Y_A_0 = q_A_0(2); phi = q_A_0(3);
32
33 -     % inverse kinematics task solution
34 -     J = mJ(phi, P(1), P(2)); % Jacobian matrix
35 -     Q = inv(J) * (s-P); % X_V_A_0 and Omega finding
36
37 -     % Desired wheels velocitys finding
38 -     W1 = (Q(1) + Q(2)*L) / R; W2 = (Q(1) - Q(2)*L) / R;
39
40 -     % Animation
41 -     dopsForAnimation(anim, A_0, phi, L, s, C, C1, C2, P);
42 -     % Data saving
43 -     r = dataSaving(r, i, t, X_A_0, Y_A_0, phi, C, C1, C2, s, P, w1, w2);
44 - end

```

Рис. 5. Пример программного кода численной математической модели кинематики КРСДП

## 7. Результаты численного математического моделирования

На рисунках 6 - 9 представлен пример вывода результатов численного моделирования прямой и обратной задач кинематики КРСДП.

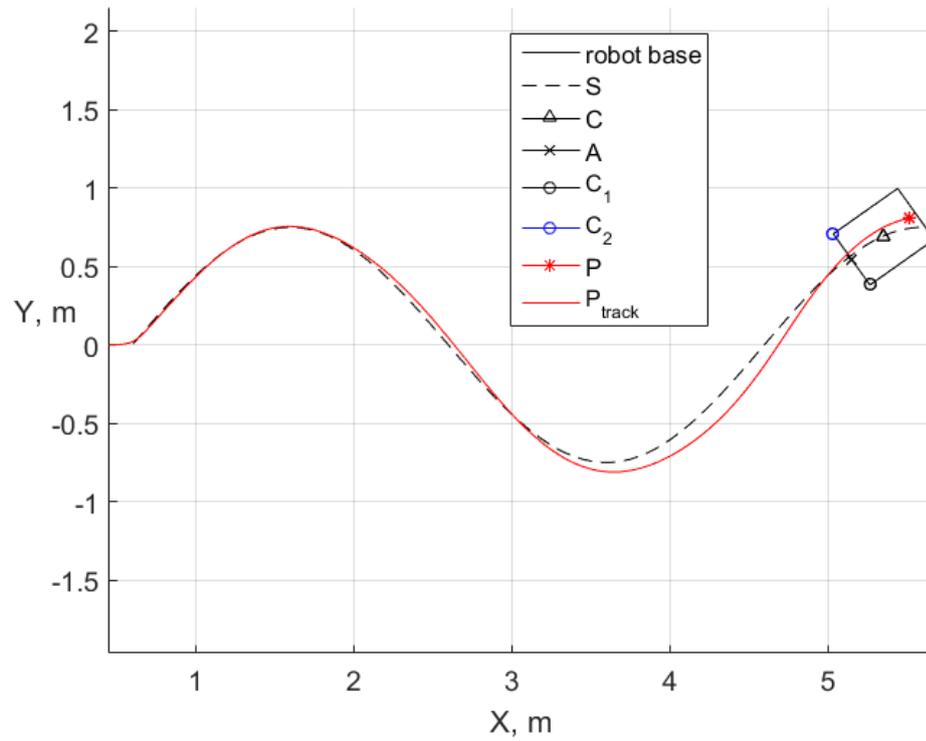


Рис. 6. Кадр из анимации движения модели кинематики КРсДП

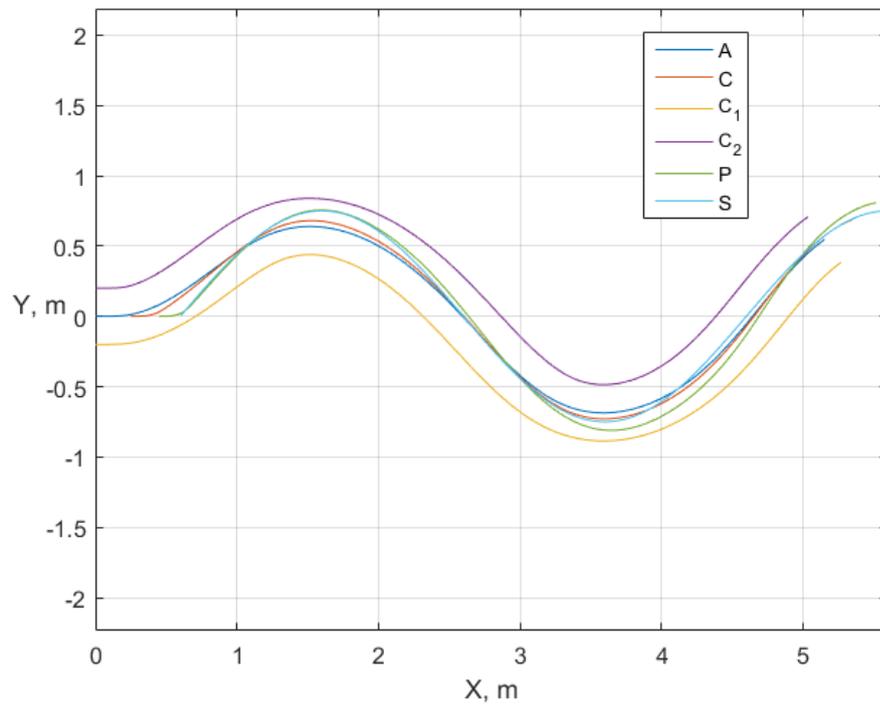


Рис. 7. Графики изменения траекторий движения ключевых точек КРсДП

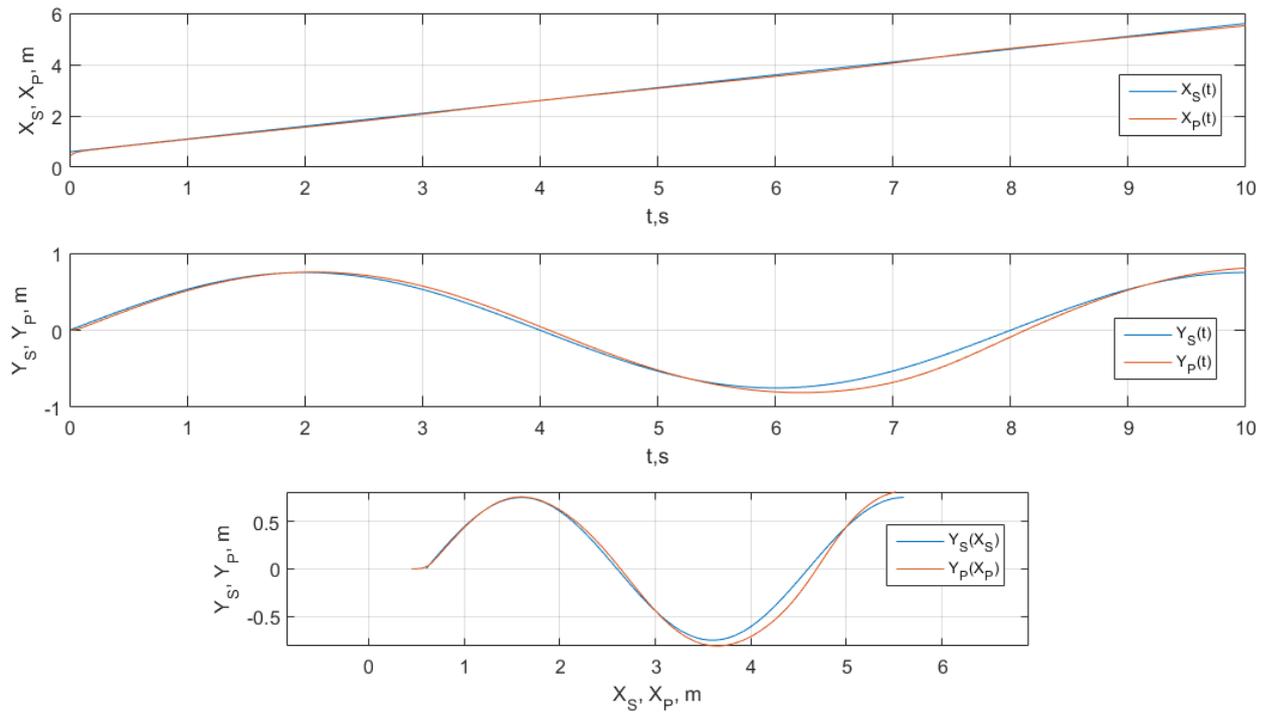


Рис. 8. Графики изменения траекторий движения точек  $S$  и  $P$

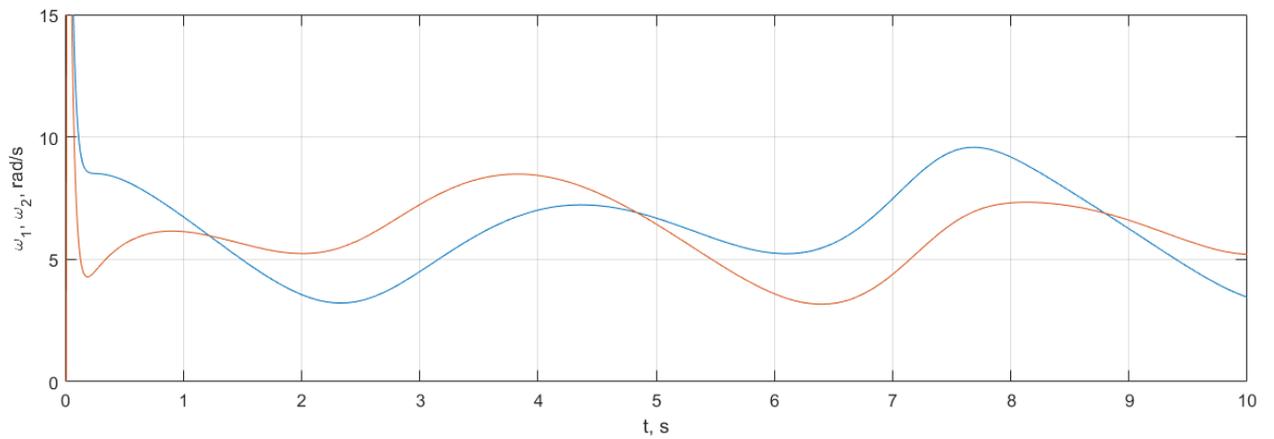


Рис. 9. Графики изменения вычисленных угловых скоростей колёс

$\omega_1$  и  $\omega_2$

## ВАРИАНТЫ ЗАДАНИЙ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Таблица 1, варианты заданий

<b>Вариант</b>	<b>R колеса, м</b>	<b>L колёсной оси, м</b>	<b>СА смещение центра масс, м</b>
1	0.1	0.25	0
2	0.15	0.5	0.125
3	0.25	0.65	-0.2
4	0.37	0.7	-0.25
5	0.5	0.3	0.15
6	0.05	0.28	0
7	0.175	0.9	0.45
8	0.2	1.0	0.3
9	0.18	1.5	-0.29
10	0.3	2.0	0.37
11	0.19	1	0
12	0.21	1.25	0.2
13	0.25	0.75	-0.22
14	0.27	0.95	-0.56
15	0.39	0.65	0
16	0.55	0.9	-0.14

### СОДЕРЖАНИЕ ОТЧЁТА

В отчёте о проделанной работе должно содержаться следующая информация:

- Название лабораторной работы и её цель.
- Расчетная схема объекта моделирования;
- Программный код;
- Рабочий файл программы;
- Графики изменения характеристик объектов;
- Проверка адекватности разработанной модели путем анализа получаемых результатов, выводы по работе;

- Список использованных источников.

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие функции используются для описания численной математической модели кинематики колёсного робота с дифференциальным приводом?
2. Как нарисовать расчётную схему кинематики колёсного робота с дифференциальным приводом? Какие основные элементы должны быть на расчётной схеме кинематики колёсного робота с дифференциальным приводом?
3. Что такое фреймы? Как описывается их взаимное расположение?
4. Что такое матрица / вектор? Какие действия с векторами и матрицами вы можете назвать? Запишите основные действия с матрицами и векторами.
5. Что такое численное интегрирование, дифференцирование и решатель дифференциальных уравнений, их формулы
6. Как проверить адекватность построенной модели?

## ПРИЛОЖЕНИЕ

### Программный код функций, отвечающих за анимацию

```

201     % Animation KRSDP base
202     function ret = animBody(A, q, L)
203 -         C2 = A + matrix2(q) * [0; L];
204 -         P1 = C2 + matrix2(q) * [0.5; 0];
205 -         P2 = P1 + matrix2(q) * [0; -2*L];
206 -         C1 = A + matrix2(q) * [0; -L];
207 -         P3 = C1 + matrix2(q) * [0; 2*L];
208
209 -         ret = [A'; C2'; P1'; P2'; C1'; P3'];
210 -     end

76     % Animation functions
77     function anim = crtAnimFig
78 -         figure(1)
79 -         clf;
80 -         anim.body = animatedline;
81 -         anim.s = animatedline('linestyle', '--');
82 -         anim.c = animatedline('marker', '^');
83 -         anim.a = animatedline('marker', 'x');
84 -         anim.c1 = animatedline('marker', 'o');
85 -         anim.c2 = animatedline('marker', 'o', 'color', 'b');
86 -         anim.p = animatedline('marker', '*', 'color', 'r');
87 -         anim.p_trj = animatedline('color', 'r');
88 -         xlabel('X, m');
89 -         ylabel('Y, m', 'rotation', 0);
90 -         legend('robot base', 'S', 'C', 'A', 'C_1', 'C_2', 'P', 'P_{track}')
91 -         axis equal;
92 -         grid on;
93 -     end

94
95     function clearingAnimPoints(anim)
96 -         % Clear animation points
97 -         clearpoints(anim.body);
98 -         clearpoints(anim.c);
99 -         clearpoints(anim.c1);
100 -         clearpoints(anim.c2);
101 -         clearpoints(anim.p);
102 -         clearpoints(anim.a);
103 -     end

165     function dopsForAnimation(anim, A_0, phi, L, S, C, C1, C2, P)
166 -         % Animation
167 -         animMass = animBody(A_0, phi, L);
168
169 -         addpoints(anim.body, animMass(:,1), animMass(:,2));
170 -         addpoints(anim.s, S(1), S(2))
171 -         addpoints(anim.c, C(1), C(2))
172
173 -         addpoints(anim.c1, C1(1), C1(2))
174 -         addpoints(anim.c2, C2(1), C2(2))
175 -         addpoints(anim.p, P(1), P(2))
176 -         addpoints(anim.a, A_0(1), A_0(2))
177 -         addpoints(anim.p_trj, P(1), P(2))
178 -         drawnow;
179 -     end

```

## СПИСОК РЕКОМЕНДУЕМЫХ ИСТОЧНИКОВ

1. MATLAB 6/6.1/6.5 + Simulink 4/5 в математике и моделировании / Дьяконов В.П. Полное руководство пользователя - М.: СОЛОН-Пресс. – 2003. – 576 с.

2. Моделирование и визуализация движений механических систем в MATLAB: Учебное пособие / В.С.Щербаков, М. С. Корилов, А.А. Руппель и др.– Омск: СибАДИ, 2007. – 84с.

3. Моделирование мехатронных систем в среде MATLAB (Simulink / SimMechanics): учебное пособие для высших учебных заведений / Мусалимов В.М., Г.Б. Заморуев, И.И. Калапышина и др. – СПб: НИУ ИТМО, 2013. – 114 с.

4. Моделирование процессов и систем в MATLAB. Учебный курс / Лазарев Ю.П. - СПб.: Питер, 2005, 511с.

5. Математическое моделирование двигателя постоянного тока в системе относительных единиц в Matlab и Си / А. А. Емельянов, В. В. Бесклеткин, И. М. Агзамов [и др.]. — Текст : непосредственный // Молодой ученый. — 2019. — № 11 (249). — С. 1-7. — URL:

6. Иванов, В. К. Моделирование мехатронных систем : учебное пособие / В. К. Иванов, В. Е. Макаров, К. Н. Никоноров ; под общ.ред. В. К. Иванова. - Йошкар-Ола : Поволжский государственный технологический университет, 2021. - 122 с. - URL: <https://biblioclub.ru/index.php?page=book&id=690797> (дата обращения: 29.09.2024) . – Режим доступа : по подписке. – Текст : электронный.

7. Данилов, Н. Н. Математическое моделирование : учебное пособие / Н. Н. Данилов. – Кемерово: Кемеровский государственный университет, 2014. – 98 с. – URL: <https://biblioclub.ru/index.php?page=book&id=278827> (дата обращения: 29.09.2024). – Режим доступа : по подписке. – Текст : электронный.

8. Лыкин, А. В. Математическое моделирование электрических систем и их элементов: учебное пособие / А. В. Лыкин. – 3-е изд. –

Новосибирск : Новосибирский государственный технический университет, 2013. – 227 с. – URL: <https://biblioclub.ru/index.php?page=book&id=228767> (дата обращения: 29.09.2024). – Режим доступа : по подписке. – Текст : электронный.

9. Моделирование мехатронных систем в среде MATLAB (Simulink/SimMechanics): учебное пособие для высших учебных заведений. – СПб. : НИУ ИТМО, 2013. – 114 с. – Текст : непосредственный.

10. Введение в математическое моделирование : учебное пособие / В. Н. Ашихмин [и др.] ; под ред. П. В. Трусова. - Москва : Логос : Университетская книга, 2015. - 440 с. - (Новая университетская библиотека). - Текст : непосредственный.

11. Бенькович, Е.С. Практическое моделирование динамических систем / Е. С. Бенькович. - СПб.: БХВ-Петербург, 2002. - 223 с. – Текст : непосредственный.

12. Глазырин, А.С. Математическое моделирование электромеханических систем. Аналитические методы: учебное пособие / А.С. Глазырин. –Томск: Изд-во Томского политехнического университета, 2009. - 216 с. – Текст : непосредственный.

13. Голубева, Н. В. Математическое моделирование систем и процессов : учебное пособие. - СПб.: Лань, 2013. - 192 с. – Текст : непосредственный.

14. Яцун, С. Ф. Применение мехатронных систем : учебно-практическое пособие / С. Ф. Яцун, А. Н. Рукавицын ; Юго-Западный государственный университет. - Курск : ЮЗГУ, 2011. - 178 с. – Текст : электронный.

15. Математическое моделирование мехатронных систем и роботов средствами программного пакета Mathcad: учебное пособие: [для студентов, обуч. по направлению подготовки "Мехатроника и робототехника"] / С. Ф. Яцун [и др.]; Юго-Зап. гос. ун-т. - Курск: ЮЗГУ, 2017. - 180 с. – Текст : электронный.

