

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**  
**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Практическая работа №4**  
**«Планирование движения»**  
по дисциплине «Моделирование и управление робототехническими системами»

Выполнил:  
студент гр. R41341с  
Борисов М. В.

Преподаватель:  
Каканов М. А.

Санкт-Петербург  
2021 г.

# Дано

## Задание

Выполнить планирование составной траектории с четырьмя опорными точками для шестизвеного манипулятора.

1. Определить количество участков траектории
2. Нормировать время
3. Задать ограничения на траекторию
4. Сформировать полиномы соответствующей степени для описания обобщённых координат, скоростей и ускорений
5. Сформировать матричное уравнение на основе заданных полиномов и решить его относительно неизвестных коэффициентов

## Решение

В приложении А приведён скрипт планирования траектории инструмента. Траектория должна проходить через четыре опорные точки, ограничивающие собой три участка — сегмент ухода, сегмент собственного перемещения между рабочими зонами и сегмент подхода. Скорости и ускорения в опорных точках на разных участках должны совпадать из условия непрерывного движения.

В качестве опорных точек зададим следующие:

$$\begin{aligned}\chi_1 &= [2 \ 0 \ 0 \ 0 \ 1.3090 \ 3.1416] ; \\ \chi_2 &= [2 \ 0 \ 2 \ 0 \ 0.5236 \ 3.1416] ; \\ \chi_3 &= [0 \ 2 \ 2 \ 1.5708 \ 0.5236 \ 3.1416] ; \\ \chi_4 &= [0 \ 2 \ 0 \ 1.5708 \ 1.3090 \ 3.1416] ;\end{aligned}\tag{1}$$

Начальные и конечные скорости и ускорения схвата манипулятора приняты нулевыми. Опорные значения времени примем  $t = [1, 2, 3, 4]$ , точек на траектории 600.

Затем для каждой опорной позиции решается ОЗК для нахождения конфигурации робота в каждом из положений. После вводится матрица  $M$ , составленная из моментов времени и коэффициентов дифференцирования. После чего для каждого звена, используя данную матрицу и известные ограничения, рассчитываются коэффициенты полиномов.

Имея коэффициенты рассчитывается траектория движения манипулятора. Расчёт производится для каждого отрезка времени по положению, скорости и ускорению с помощью соответствующих полиномов.

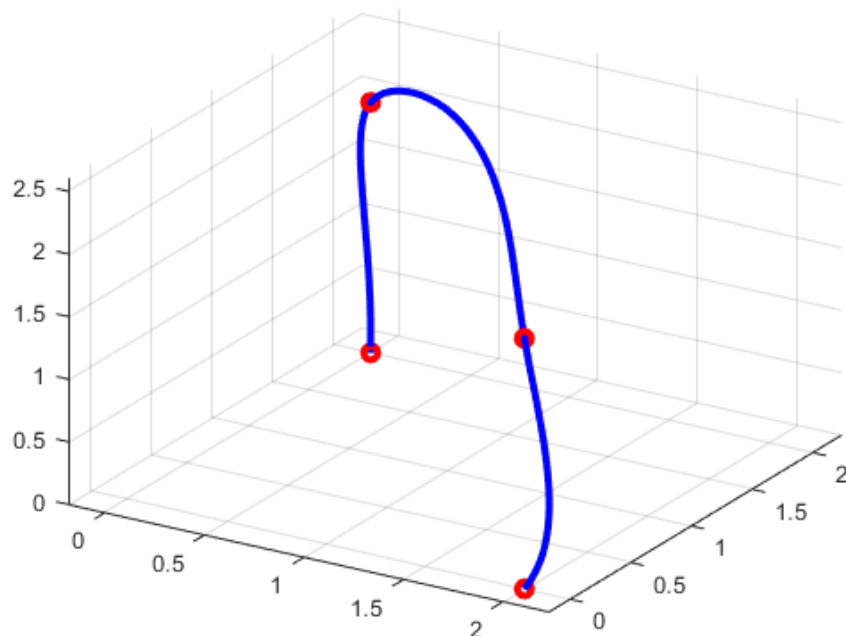


Рис. 1: Траектория движения манипулятора

По рисунку 1 видно, что траектория проходит через все обозначенные точки и плавная, что доказывает верность реализации.

## Вывод

В ходе работы реализован скрипт, строящий траекторию движения схвата манипулятора через опорные точки используя данные кинематики манипулятора. Полученная кривая является гладкой и не имеет точек разрыва.

## А. Функция решения прямой задачи кинематики

```
1  %опорные точки
2  xi1 = [2 0 0 0 1.3090 3.1416];
3  xi2 = [2 0 2 0 0.5236 3.1416];
4  xi3 = [0 2 2 1.5708 0.5236 3.1416];
5  xi4 = [0 2 0 1.5708 1.3090 3.1416];
6
7  %начальная и конечная скорости и ускорения
8  dq1 = [0 0 0 0 0 0];
9  ddq1 = [0 0 0 0 0 0];
10
11 dq4 = [0 0 0 0 0 0];
12 ddq4 = [0 0 0 0 0 0];
13
14 %отрезки времени
15 t1 = 0; t2 = 2; t3 = 4; t4 = 6;
16
17 inc = 600; %детализация траектории
18
19 q1 = ik(xi1);
20 q2 = ik(xi2);
21 q3 = ik(xi3);
22 q4 = ik(xi4);
23
24 M = [0 0 0 0 1 0 0 0 0 0 0 0 0 0;...
25      0 0 0 1 0 0 0 0 0 0 0 0 0 0;...
26      0 0 2 0 0 0 0 0 0 0 0 0 0 0;...
27      1 1 1 1 1 0 0 0 0 0 0 0 0 0;...
28      0 0 0 0 0 0 0 0 1 0 0 0 0 0;...
29      4 3 2 1 0 0 0 -1 0 0 0 0 0 0;...
30      12 6 2 0 0 0 -2 0 0 0 0 0 0 0;...
31      0 0 0 0 0 1 1 1 1 0 0 0 0 0;...
32      0 0 0 0 0 0 0 0 0 0 0 0 0 1;...
33      0 0 0 0 0 3 2 1 0 0 0 0 -1 0;...
34      0 0 0 0 0 6 2 0 0 0 0 -2 0 0;...
35      0 0 0 0 0 0 0 0 0 1 1 1 1 1;...
36      0 0 0 0 0 0 0 0 0 4 3 2 1 0;...
37      0 0 0 0 0 0 0 0 0 12 6 2 0 0];
38
39 k1 = inv(M)*[q1(1) dq1(1) ddq1(1) q2(1) q2(1) 0 0 q3(1) q3(1) 0 0
40   ↪ q4(1) dq4(1) ddq4(1)]';
41 k2 = inv(M)*[q1(2) dq1(2) ddq1(2) q2(2) q2(2) 0 0 q3(2) q3(2) 0 0
42   ↪ q4(2) dq4(2) ddq4(2)]';
43 k3 = inv(M)*[q1(3) dq1(3) ddq1(3) q2(3) q2(3) 0 0 q3(3) q3(3) 0 0
44   ↪ q4(3) dq4(3) ddq4(3)]';
45 k4 = inv(M)*[q1(4) dq1(4) ddq1(4) q2(4) q2(4) 0 0 q3(4) q3(4) 0 0
46   ↪ q4(4) dq4(4) ddq4(4)]';
```

```

43 k5 = inv(M)*[q1(5) dq1(5) ddq1(5) q2(5) q2(5) 0 0 q3(5) q3(5) 0 0
    ↪ q4(5) dq4(5) ddq4(5)]';
44 k6 = inv(M)*[q1(6) dq1(6) ddq1(6) q2(6) q2(6) 0 0 q3(6) q3(6) 0 0
    ↪ q4(6) dq4(6) ddq4(6)]';
45
46 q = zeros(6, inc);
47 dq = zeros(6, inc);
48 ddq = zeros(6, inc);
49 i = 1;
50 for t=linspace(t1,t4,inc)
51     tau1 = (t-t1)/(t2-t1);
52     tau2 = (t-t2)/(t3-t2);
53     tau3 = (t-t3)/(t4-t3);
54     if (t >= t1) && (t < t2)
55         q(1, i) = k1(1)*tau1^4 + k1(2)*tau1^3 + k1(3)*tau1^2 +
            ↪ k1(4)*tau1+k1(5);
56         q(2, i) = k2(1)*tau1^4 + k2(2)*tau1^3 + k2(3)*tau1^2 +
            ↪ k2(4)*tau1+k2(5);
57         q(3, i) = k3(1)*tau1^4 + k3(2)*tau1^3 + k3(3)*tau1^2 +
            ↪ k3(4)*tau1+k3(5);
58         q(4, i) = k4(1)*tau1^4 + k4(2)*tau1^3 + k4(3)*tau1^2 +
            ↪ k4(4)*tau1+k4(5);
59         q(5, i) = k5(1)*tau1^4 + k5(2)*tau1^3 + k5(3)*tau1^2 +
            ↪ k5(4)*tau1+k5(5);
60         q(6, i) = k6(1)*tau1^4 + k6(2)*tau1^3 + k6(3)*tau1^2 +
            ↪ k6(4)*tau1+k6(5);
61         dq(1, i) = 4*k1(1)*tau1^3 + 3*k1(2)*tau1^2 + 2*k1(3)*tau1 +
            ↪ k1(4);
62         dq(2, i) = 4*k2(1)*tau1^3 + 3*k2(2)*tau1^2 + 2*k2(3)*tau1 +
            ↪ k2(4);
63         dq(3, i) = 4*k3(1)*tau1^3 + 3*k3(2)*tau1^2 + 2*k3(3)*tau1 +
            ↪ k3(4);
64         dq(4, i) = 4*k4(1)*tau1^3 + 3*k4(2)*tau1^2 + 2*k4(3)*tau1 +
            ↪ k4(4);
65         dq(5, i) = 4*k5(1)*tau1^3 + 3*k5(2)*tau1^2 + 2*k5(3)*tau1 +
            ↪ k5(4);
66         dq(6, i) = 4*k6(1)*tau1^3 + 3*k6(2)*tau1^2 + 2*k6(3)*tau1 +
            ↪ k6(4);
67         ddq(1,i) = 12*k1(1)*tau1^2 + 6*k1(2)*tau1 + 2*k1(3);
68         ddq(2,i) = 12*k2(1)*tau1^2 + 6*k2(2)*tau1 + 2*k2(3);
69         ddq(3,i) = 12*k3(1)*tau1^2 + 6*k3(2)*tau1 + 2*k3(3);
70         ddq(4,i) = 12*k4(1)*tau1^2 + 6*k4(2)*tau1 + 2*k4(3);
71         ddq(5,i) = 12*k5(1)*tau1^2 + 6*k5(2)*tau1 + 2*k5(3);
72         ddq(6,i) = 12*k6(1)*tau1^2 + 6*k6(2)*tau1 + 2*k6(3);
73     elseif (t >= t2) && (t < t3)
74         q(1, i) = k1(6)*tau2^3 + k1(7)*tau2^2 + k1(8)*tau2+k1(9);
75         q(2, i) = k2(6)*tau2^3 + k2(7)*tau2^2 + k2(8)*tau2+k2(9);
76         q(3, i) = k3(6)*tau2^3 + k3(7)*tau2^2 + k3(8)*tau2+k3(9);

```

```

77     q(4, i) = k4(6)*tau2^3 + k4(7)*tau2^2 + k4(8)*tau2+k4(9);
78     q(5, i) = k5(6)*tau2^3 + k5(7)*tau2^2 + k5(8)*tau2+k5(9);
79     q(6, i) = k6(6)*tau2^3 + k6(7)*tau2^2 + k6(8)*tau2+k6(9);
80     dq(1, i) = 3*k1(6)*tau2^2 + 2*k1(7)*tau2 + k1(8);
81     dq(2, i) = 3*k2(6)*tau2^2 + 2*k2(7)*tau2 + k2(8);
82     dq(3, i) = 3*k3(6)*tau2^2 + 2*k3(7)*tau2 + k3(8);
83     dq(4, i) = 3*k4(6)*tau2^2 + 2*k4(7)*tau2 + k4(8);
84     dq(5, i) = 3*k5(6)*tau2^2 + 2*k5(7)*tau2 + k5(8);
85     dq(6, i) = 3*k6(6)*tau2^2 + 2*k6(7)*tau2 + k6(8);
86     ddq(1,i) = 6*k1(6)*tau2 + 2*k1(7);
87     ddq(2,i) = 6*k2(6)*tau2 + 2*k2(7);
88     ddq(3,i) = 6*k3(6)*tau2 + 2*k3(7);
89     ddq(4,i) = 6*k4(6)*tau2 + 2*k4(7);
90     ddq(5,i) = 6*k5(6)*tau2 + 2*k5(7);
91     ddq(6,i) = 6*k6(6)*tau2 + 2*k6(7);
92     elseif (t >= t3) && (t <= t4)
93         q(1, i) = k1(10)*tau3^4 + k1(11)*tau3^3 + k1(12)*tau3^2 +
94             ↪ k1(13)*tau3 + k1(14);
95         q(2, i) = k2(10)*tau3^4 + k2(11)*tau3^3 + k2(12)*tau3^2 +
96             ↪ k2(13)*tau3 + k2(14);
97         q(3, i) = k3(10)*tau3^4 + k3(11)*tau3^3 + k3(12)*tau3^2 +
98             ↪ k3(13)*tau3 + k3(14);
99         q(4, i) = k4(10)*tau3^4 + k4(11)*tau3^3 + k4(12)*tau3^2 +
100             ↪ k4(13)*tau3 + k4(14);
101         q(5, i) = k5(10)*tau3^4 + k5(11)*tau3^3 + k5(12)*tau3^2 +
102             ↪ k5(13)*tau3 + k5(14);
103         q(6, i) = k6(10)*tau3^4 + k6(11)*tau3^3 + k6(12)*tau3^2 +
104             ↪ k6(13)*tau3 + k6(14);
105         dq(1, i) = 4*k1(10)*tau3^3 + 3*k1(11)*tau3^2 + 2*k1(12)*tau3
106             ↪ + k1(13);
107         dq(2, i) = 4*k2(10)*tau3^3 + 3*k2(11)*tau3^2 + 2*k2(12)*tau3
108             ↪ + k2(13);
109         dq(3, i) = 4*k3(10)*tau3^3 + 3*k3(11)*tau3^2 + 2*k3(12)*tau3
110             ↪ + k3(13);
111         dq(4, i) = 4*k4(10)*tau3^3 + 3*k4(11)*tau3^2 + 2*k4(12)*tau3
112             ↪ + k4(13);
113         dq(5, i) = 4*k5(10)*tau3^3 + 3*k5(11)*tau3^2 + 2*k5(12)*tau3
114             ↪ + k5(13);
115         dq(6, i) = 4*k6(10)*tau3^3 + 3*k6(11)*tau3^2 + 2*k6(12)*tau3
116             ↪ + k6(13);
117         ddq(1,i) = 12*k1(10)*tau3^2 + 6*k1(11)*tau3 + 2*k1(12);
118         ddq(2,i) = 12*k2(10)*tau3^2 + 6*k2(11)*tau3 + 2*k2(12);
119         ddq(3,i) = 12*k3(10)*tau3^2 + 6*k3(11)*tau3 + 2*k3(12);
120         ddq(4,i) = 12*k4(10)*tau3^2 + 6*k4(11)*tau3 + 2*k4(12);
121         ddq(5,i) = 12*k5(10)*tau3^2 + 6*k5(11)*tau3 + 2*k5(12);
122         ddq(6,i) = 12*k6(10)*tau3^2 + 6*k6(11)*tau3 + 2*k6(12);
123     end
124     i = i + 1;

```

```

113 end
114
115 xi = zeros(6,inc);
116 for i=1:inc
117     xi(:,i) = fk(q(:,i)');
118 end
119
120 figure
121 plot3(xi(1,:), xi(2,:), xi(3,:), 'Color','blue','Linewidth',3)
122 hold on
123 grid
124 plot3(xi1(1), xi1(2), xi1(3), 'Color','red','Linewidth', 3,
    ↪ 'Marker', 'o')
125 plot3(xi2(1), xi2(2), xi2(3), 'Color','red','Linewidth', 3,
    ↪ 'Marker', 'o')
126 plot3(xi3(1), xi3(2), xi3(3), 'Color','red','Linewidth', 3,
    ↪ 'Marker', 'o')
127 plot3(xi4(1), xi4(2), xi4(3), 'Color','red','Linewidth', 3,
    ↪ 'Marker', 'o')

```