

**Министерство науки и высшего образования**

**Российской Федерации**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технологический университет «СТАНКИН»**

**(ФГБОУ ВО «МГТУ «СТАНКИН»)**

Институт автоматизации и робототехники

Дисциплина: «Методы оптимизации»

**ЛАБОРАТОРНАЯ РАБОТА 4**

«Применение метода замены переменных для минимизации с ограничениями»

Выполнил:

студент группы АДБ-17-11 \_\_\_\_\_\_\_\_\_\_ Абдулзагиров М.М.

(подпись) (ФИО)

Принял

преподаватель: \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (ФИО)

Дата:\_\_\_\_\_\_\_\_\_

Москва 2020.

# 4.1. Рассмотрим задачу управления роботом с учётом ограничений на его обобщённые координаты.

Пусть для каждой координаты задан интервал возможных значений, записанных в матрицу:

dq=[-2 2;-2 2;-2 2]

Проведём преобразование переменных так, чтобы новые переменные могли меняться без ограничений:

q(i) = (b – a)/pi\*arctg(z(i)) + (a + b)/2.

При указанных ограничениях формула примет вид:

q(i) = 4/pi\*arctg(z(i)) .

Для z(i) справедлива формула:

z(i) = tg((q(i) –(a + b)/2) \*pi / (b – a) ) .

Что для начальных значений даёт при заданных ограничениях:

z0(i) = tg(q0(i) \*pi / 4 ).

Тогда целевая функция будет:

function d=distz(z,q,l,xc,q0)

q = 4/pi\*atan(z) x=coord(q,l);

d=norm(x-xc);

Вызов функции минимизации:

[z,fs, e, inf]=fminsearch(@distz,z0, [], l, xc)

В итоге код программы выглядит так:

function prog15\_7\_2

%открываем граффическое окно задаём начальные параметры

axes('Xlim',[-1 15], 'Ylim',[-1 10]);

axis equal; grid on; hold on;

l =[5 5 5];

xc=[12; 0]; %задаём новые координаты

q0=[0.7 -1 0.5]; %Изменённый начальный вектор

dq=[-2 2; -2 2; -2 2]; %интервалы перемещений

%массывы чисел, нужных для вычисления

a=transpose(dq(:,1))

b=transpose(dq(:,2))

c=[pi/(b(1)-a(1)) pi/(b(2)-a(2)) pi/(b(3)-a(3))];

z0 = tan((q0 -(a + b)/2).\* c) %tan((q0) \*pi /4 )

%рисуем начальное полжение пунктиром

[XInit,YInit]=hinges(q0,l);

plot(XInit, YInit,'k:');

%вычисляем значения углов и рисуем конечное полжение

[z,fs, e, inf]=fminsearch(@distz,z0, [], l, xc,dq)

q = (b - a)/pi.\*atan(z) + (a + b)/2;

[XOptim,YOptim]=hinges(q,l);

plot(XOptim, YOptim,'k-','LineWidt',2);

%отрисовываем элементы оформления

plot([0],[-0.3],'k^','MarkerSize',15);

plot(XInit(1:3), YInit(1:3),'ko');

plot([XInit(4)], [YInit(4)],'ks', 'MarkerSize', 15);

plot(XOptim(1:3), YOptim(1:3),'ko');

plot([XOptim(4)], [YOptim(4)],'ks', 'MarkerSize', 15);

xlabel('x1'); ylabel('x2');

legend('Начальное','Конечное');

title('Два положения манипулятора');

end

function [X,Y] = hinges(q,l)

Q=cumsum(q);

x=l.\*cos(Q);

y=l.\*sin(Q);

x=[0 x]; y=[0 y]; X=cumsum(x); Y=cumsum(y);

end

function d=distz(z,l,xc,dq)

%q = 4/pi\*atan(z);

a=transpose(dq(:,1))

b=transpose(dq(:,2))

q = (b - a)/pi.\*atan(z) + (a + b)/2;

x=coord(q,l);

d=norm(x-xc);

end

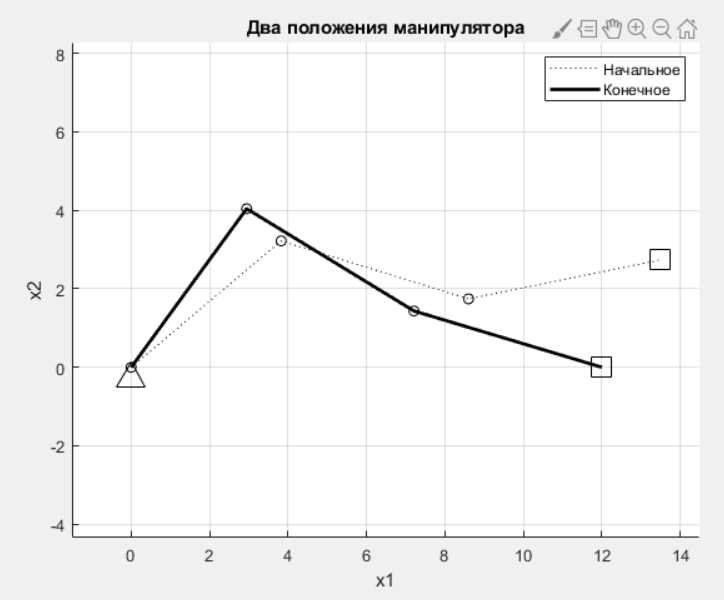
function x=coord(q,l)

Q=cumsum(q);

x=[sum(l.\*cos(Q)); sum(l.\*sin(Q))];

end

По полученным значениям переменных z необходимо рассчитать значения *q* и использовать их в программе *prog15\_7*. Получаем рисунок следующего вида:



**Рис. 1** кинематическая схема по полученным значения q.

**Вывод**: робот достигает точки, установив значение обобщенных координат, не превышающих предельные значения.

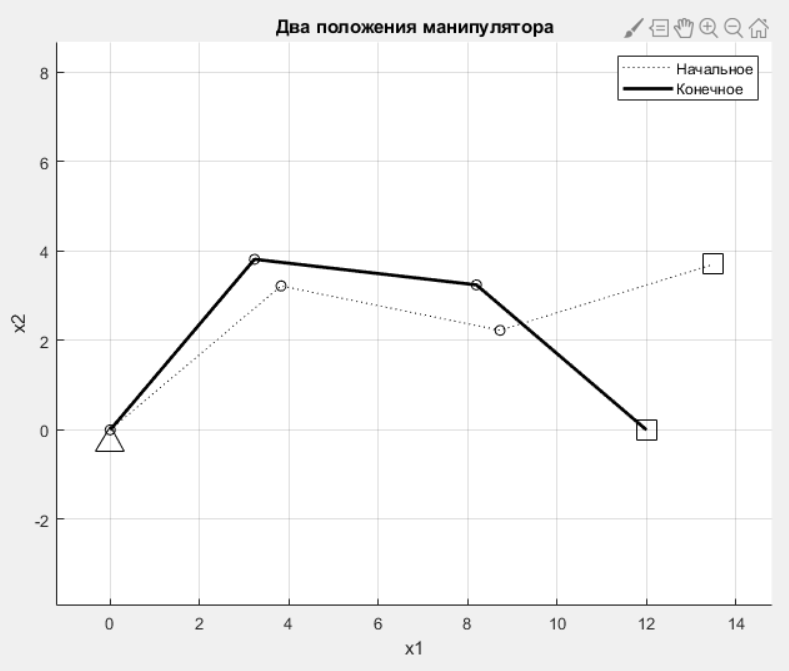
4.2. Изменим допустимые пределы изменения обобщённых координат:

dq=[-1 1; -1 1; -1 1]; %интервалы перемещений

Изменим их начальные значения, так чтобы они оказались внутри допустимой области:

q0=[0.7 -0.9 0.5]; %Изменённый начальный вектор

Реализованная конфигурация робота приведена на следующем рисунке.



**Рис. 2**. Реализованная конфигурация робота

**Вывод:** робот так же достигает точки, установив значение обобщенных координат, не превышающих предельные значения, но в данном случае они отличны от прошлых, т.к. прошлые значения превышали допустимые значения подвижностей.

4.3. Выполнить аналогичные исследования для своего варианта кинематических схем, заданных в табл. 3. Допустимые диапазоны обобщённых координат заданы в табл. 4:

Таблица. 4

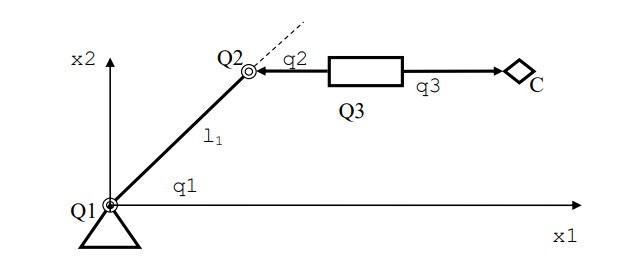
|  |  |  |  |
| --- | --- | --- | --- |
| Вариант № | Схема № | Обобщённые координаты робота | Допустимые диапазоны изменения обобщённых координат |
| 1 | 1 | q1 q2 q3 | [-1.5, 1.5]  [-1.5, 1.5]  [1, 5] |

Таблица. 3

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант № | Схема № | Параметры, координаты целевой точки | Начальный вектор q0 |
| 1 | 1 | l1=5 xc=[12; 0]; xc=[5;5]; | [0.7 -1 1] |

Два вращательных шарнира (q1, q2) и одна поступательная координата (q3)

(схема ВВП).



**Рис. 9**. Кинематическая схема робота.

Изменим начальный вектор q0 на q0=[0.7 -1 1.1] чтобы они оказались внутри допустимой области и конечную точку на xc=[9; 0], т.к. координаты xc=[12; 0] выходят за пределы рабочей зоны робота.

Код программы задания с пояснениями:

function task\_1\_1\_2v

%открываем граффическое окно задаём начальные параметры

axes('Xlim',[-1 15], 'Ylim',[-1 10]);

axis equal; grid on; hold on;

l=5;

xc=[9; 0]; %задаём новые координаты

q0=[0.7 -1 1.1] ; %Изменённый начальный вектор

dq=[-1.5 1.5; -1.5 1.5; 1 5]; %интервалы перемещений

%массывы чисел, нужных для вычисления

a=transpose(dq(:,1));

b=transpose(dq(:,2));

c=[pi/(b(1)-a(1)) pi/(b(2)-a(2)) pi/(b(3)-a(3))];

z0 = tan((q0 -(a + b)/2).\* c);

%рисуем начальное полжение пунктиром

[XInit,YInit]=hinges(q0,l);

plot(XInit, YInit,'k:');

%вычисляем значения углов и рисуем конечное полжение

z=fminsearch(@distz,z0, [], l, xc,dq)

%[q,fs, e, inf]=fminsearch(@dist,q0, [], l, xc)

q = (b - a)/pi.\*atan(z) + (a + b)/2;

[XOptim,YOptim]=hinges(q,l);

plot(XOptim, YOptim,'k-','LineWidt',2);

%отрисовываем элементы оформления

plot([0],[-0.3],'k^','MarkerSize',15);

plot(XInit(1:2), YInit(1:2),'ko');

plot([XInit(3)], [YInit(3)],'ks', 'MarkerSize', 15);

plot(XOptim(1:2), YOptim(1:2),'ko');

plot([XOptim(3)], [YOptim(3)],'ks', 'MarkerSize', 15);

xlabel('x1'); ylabel('x2');

legend('Начальное','Конечное');

title('Два положения манипулятора');

end

function [X,Y] = hinges(q,l)

x=[l\*cos(q(1)) q(3)\*cos(q(1)+q(2))];

y=[l\*sin(q(1)) q(3)\*sin(q(1)+q(2))];

x=[0 x]; y=[0 y]; X=cumsum(x); Y=cumsum(y);

end

function d=distz(z,l,xc,dq)

a=transpose(dq(:,1));

b=transpose(dq(:,2));

q = (b - a)/pi.\*atan(z) + (a + b)/2;

x=coord(q,l);

d=norm(x-xc)

end

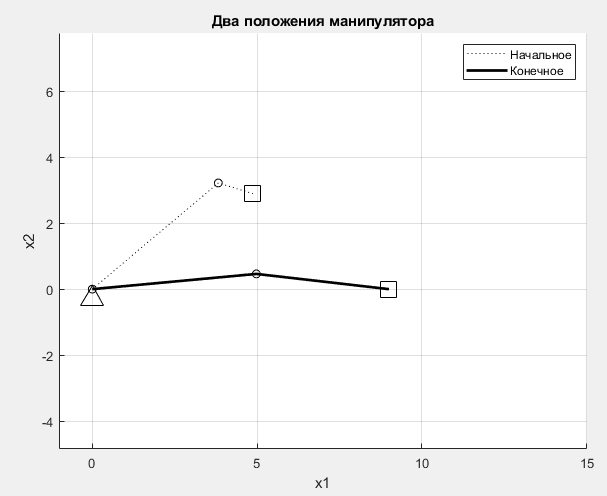
function x=coord(q,l)

x1=l\*cos(q(1))+q(3)\*cos(q(1)+q(2));

y1=l\*sin(q(1))+q(3)\*sin(q(1)+q(2));

x=[x1; y1];

end



**Рис. 4**. Реализованная конфигурация робота при xc=[9; 0] .

Если всё же установить конечные координаты xc=[12; 0], то робот доходит лишь до точки 10, при котором расстояние от начала координат до рабочего органа не превышает суммарной длины первого звена (5) и максимальной длины обобщенной поступательной координаты q3 (5). Так же выводится сообщение о том, что превышен лимит попыток достичь минимального значения:

>> task\_1\_1\_2v

Exiting: Maximum number of function evaluations has been exceeded

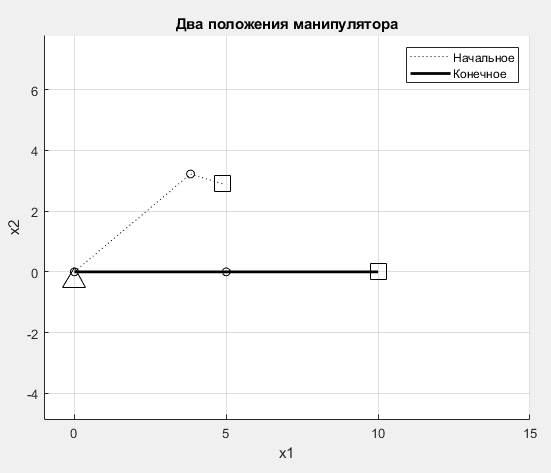
- increase MaxFunEvals option.

Current function value: 2.000000

z =

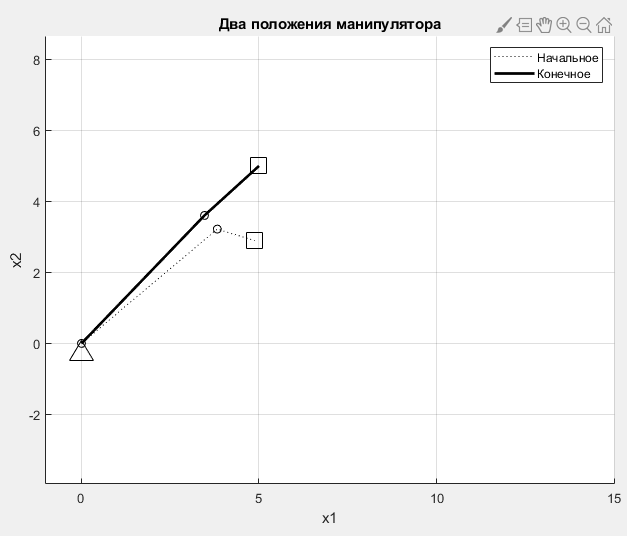
1.0e+11 \*

-0.0000 0.0000 5.5012



**Рис. 5**. Реализованная конфигурация робота при xc=[12; 0] .

Точка xc=[5; 5] входит в рабочую зону робота:



**Рис. 4**. Реализованная конфигурация робота при xc=[5; 5] .

**Вывод:** робот достигает конечной точки, если она входит в рабочую зону робота с ограниченными подвижностями