ЛАБОРАТОРНАЯ РАБОТА 3

3. Использование функции fminunc

Для ускорения поиска в функцию необходимо включить формулы для вычисления градиента и гессиана. Это необходимо указать в списке управляющих параметров для функции минимизации:

```
Options=optimset('Display', 'final','GradObj', 'on',
'Hessian', 'on');

[x,f1,e_flag,out,grad,hes]=fminunc(@Descartes,x0,options)
```

В режиме функции `help` изучить параметры и способы использования функции `fminunc`. Записать пояснения в отчёт.

3.1. Текст следующей функции записать в соответствующий `m`-файл:

```
function prog15_61
axes('Xlim',[-1.5 2.5], 'Ylim',[-1.5 2.5]);
axis equal; grid off; hold on;
xlabel('x1'); ylabel('x2');
colormap copper;

X0=-1.5:0.05:2.5;
 [X Y]=meshgrid(X0);
s=size(X); Z=zeros(s);
for i=1:s(1)
    for j=1:s(2)
        Z(i,j) = Descartes([X(i,j); Y(i,j)]);
    end
end

V=-0.8:0.2:1;
contour(X,Y,Z,V);
Options=optimset('Display', 'final','GradObj', 'on',
'Hessian', 'on');
x0=[2; 2];
line(x0(1),x0(2), 'Marker', '.', 'MarkerSize', 10);
[x,f1,e_flag,out]= fminunc(@Descartes,x0,options)
line(x(1),x(2), 'Marker', '.', 'MarkerSize', 20);

plot([x0(1),x(1)], [x0(2),x(2)],'k-')

function f = Descartes(x)
f = x(1)^3+x(2)^3-3*x(1)*x(2);
if nargout > 1
    g = [3*(x(1)^2-x(2)); 3*(x(2)^2-x(1))];
```

```
end
if nargout > 2
   H = [6*(x(1) -3; -3 6*x(2)];
end
end
```

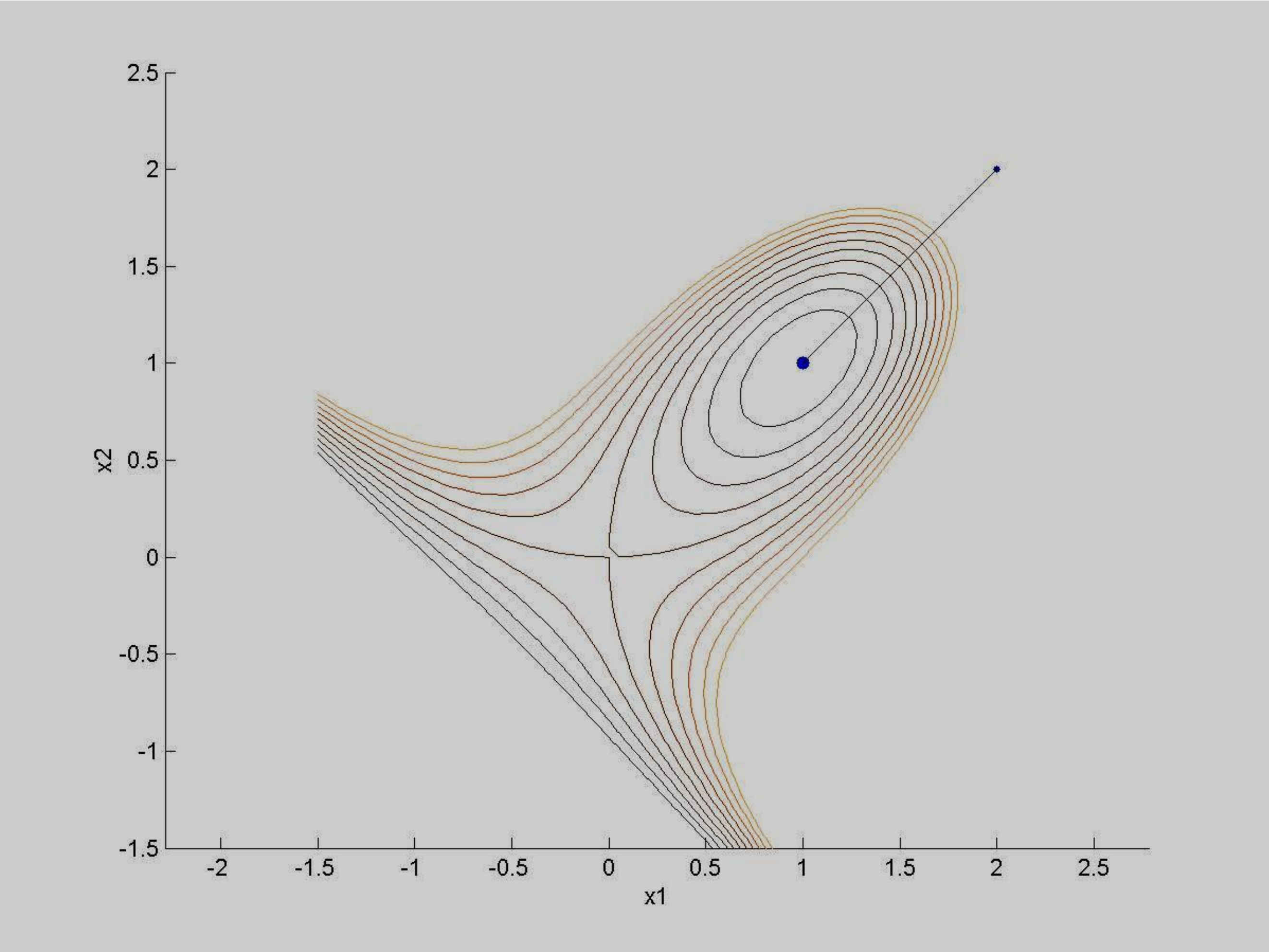
Внимательно изучите текст программы, в отчёте кратко опишите алгоритм её работы.
Вызов функции имеет вид:

```
>> prog15_61
```

Результат её выполнения:

```
Optimization terminated successfully:
 Relative function value changing by less than
OPTIONS.TolFun

x =
    1.0000
    1.0000
f1 =    -1
e_flag =     1
out =
       iterations: 6
        funcCount: 6
      cgiterations: 5
     firstorderopt: 6.9849e-010
        algorithm: [1x32 char]
grad =  1.0e-009 *
    0.6985
    0.6985
hes =
    6.0000   -3.0000
   -3.0000    6.0000
```

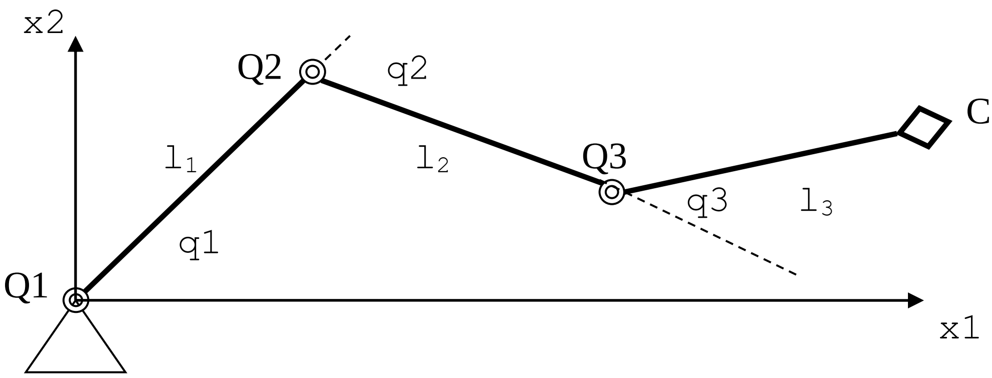
Поясните полученный результат.



Декартов лист.

3.2. <u>Задание</u>: По образцу рассмотренного примера применить функцию `fminunc` к минимизации функций предыдущего пункта с использованием градиента и гессиана. Выражения первых и вторых производных привести в отчёте. Сравнить полученные результаты.

3.3. Задача об управлении роботом
На следующем рисунке представлена кинематическая схема плоского трёхзвенного манипулятора.



В точке Q1 неподвижно закреплён центр первого шарнира, точки Q2 и Q3 соответствуют центрам двух других шарниров, соединяющих звенья механизма между собой. Буквой C обозначен центр захватного устройства робота, координаты которого образуют вектор `x=[x1; x2]`. Каждое звено может поворачиваться вокруг центров указанных шарниров на углы `q1`, `q2`, `q3`, являющиеся обобщёнными координатами робота.
Длины звеньев постоянны и равны `l1`, `l2`, `l3`, соответственно.
Требуемое положение схвата задаётся вектором `xc=[xc1; xc2]`, целью управления является минимизация нормы разности `x-xc`.
Координаты схвата выражаются через длины звеньев `l1`, `l2`, `l3` и углы `q1`, `q2`, `q3` по формулам:

$$x_1=l_1*\cos(q_1)+l_2*\cos(q_1+q_2)+l_3*\cos(q_1+q_2+q_3)$$
$$x_2=l_1*\sin(q_1)+l_2*\sin(q_1+q_2)+l_3*\sin(q_1+q_2+q_3)$$

Вычисление координат схвата выполняет функция:

```
function x=coord(q,l)
Q=cumsum(q);
x=[sum(l.*cos(Q)); sum(l.*sin(Q))];
```

Целевая функция вычисляется следующим образом:

```
function d=dist(q,l,xc)
x=coord(q,l);
d=norm(x-xc);
```

Исходные данные задают начальный вектор обобщённых координат `q0`, вектор длин стержней `l`, требуемое положение схвата `xc`:

```
q0=[0.7 -1 0.5];
l=[5 5 5];
xc=[12; 0];
```

Обращение к функции минимизации имеет вид:

```
[q,f]=fminsearch(@dist,q0, [], l, xc)
```

Результат расчёта:

```
q =    0.8924   -1.5601    0.5078
f =  8.6000e-005
```

Графическая иллюстрация получается с помощью программы:

```
function prog15_7
axes('Xlim',[-1 15], 'Ylim',[-1 10]);
axis equal; grid on; hold on;
```

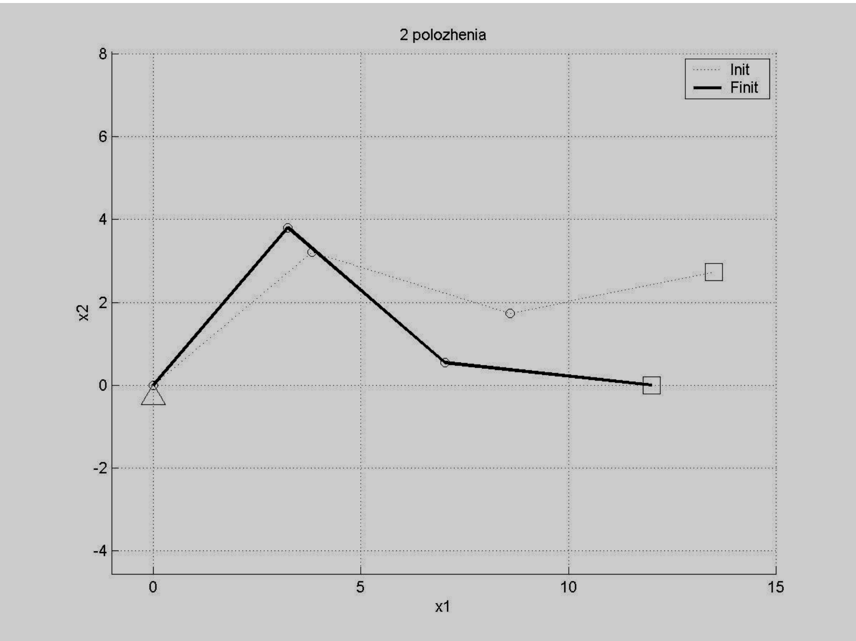
```
l=[5 5 5];
xc=[12; 0];
qInit=[0.7 -1 0.5];
[XInit,YInit]=hinges(qInit,l);

plot(XInit, YInit,'k:');
qOptim=[0.8924   -1.5601    0.5078];
[XOptim,YOptim]=hinges(qOptim,l);
plot(XOptim, YOptim,'k-','LineWidt',2);

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');

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);
```

Вызов:

```
>> prog15_7
```

Результат представлен на следующем рисунке.

Сделать пояснения к рисунку.



Изменить целевую точку схвата:
`xc=[10; 5];`
Получить соответствующие значения обобщённых координат робота, а также дополнительную информацию о ходе вычислений.
Внести необходимые изменения в программу `prog15_7`, чтобы отобразить полученное новое решение.

3.4. Изменить начальный вектор

`q0=[-0.7 1 -0.5].`

Провести аналогичное исследование:

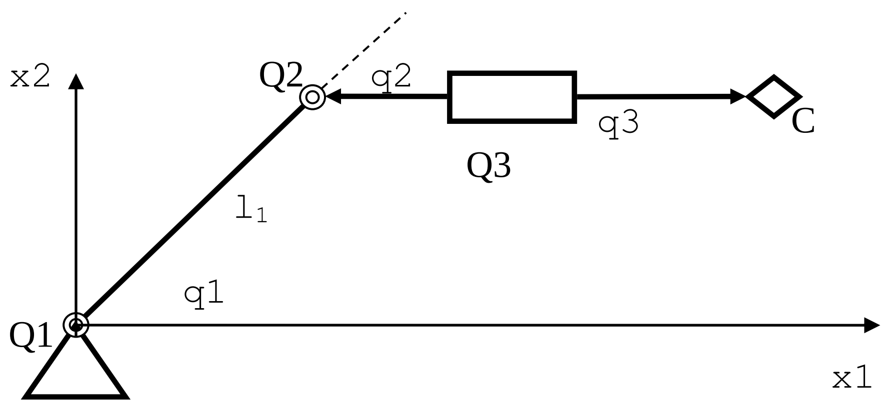
– Найти значение `q` для целевой точки схвата
`xc=[12; 0];`

– Получить графическую иллюстрацию, изменив соответствующим образом программу `prog15_7;`

– Дать объяснение полученным результатам.

3.5. Решить аналогичную задачу для робота, кинематическая схема которого соответствует одному из следующих вариантов (Таблица 3).

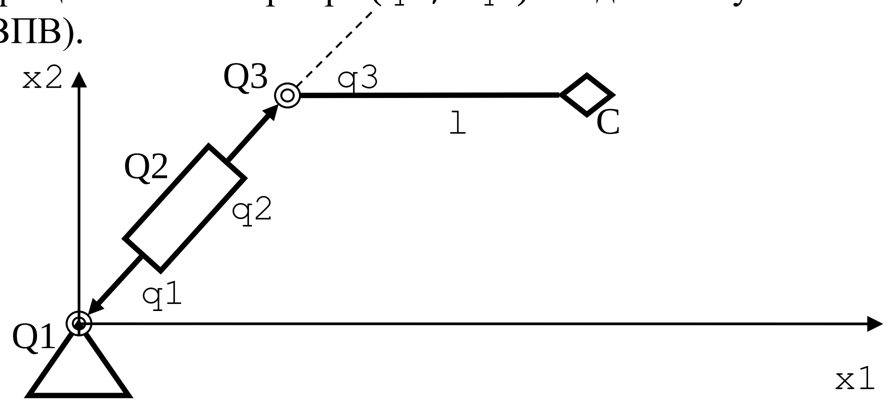
1) Два вращательных шарнира (`q1, q2`) и одна поступательная координата (`q3`) (схема ВВП).



Координаты схвата вычисляются по формулам:

$$x_1=l_1*\cos(q_1)+q_3*\cos(q_1+q_2)$$
$$x_2=l_1*\sin(q_1)+q_3*\sin(q_1+q_2)$$

2) Два вращательных шарнира (q1, q3) и одна поступательная координата (q2) (схема ВПВ).

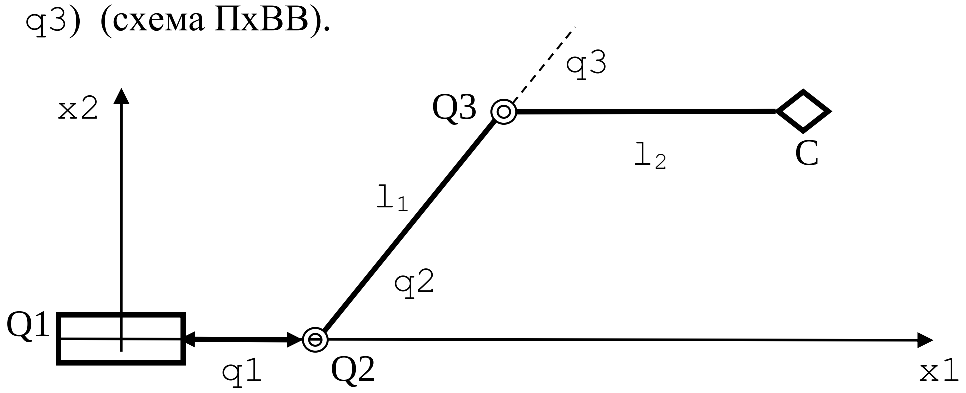


Координаты схвата вычисляются по формулам:

$x_1=q_2*\cos(q_1)+l*\cos(q_1+q_3)$

$x_2=q_2*\sin(q_1)+l*\sin(q_1+q_3)$

3) Одна поступательная координата по оси x_1 (q1) и два вращательных шарнира (q2, q3) (схема ПхВВ).

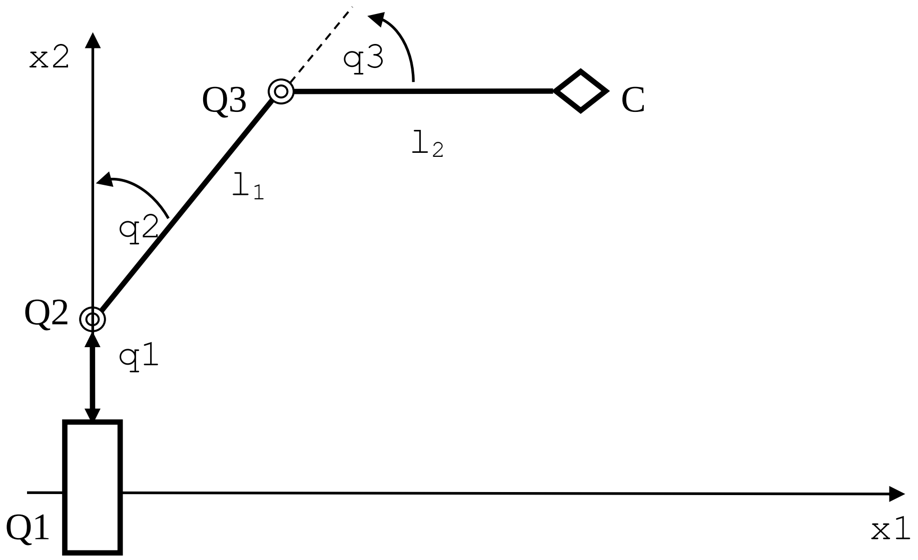


Координаты схвата вычисляются по формулам:

$x_1=q_1+l_1*\cos(q_2)+l_2*\cos(q_2+q_3)$

$x_2=l_1*\sin(q_2)+l_2*\sin(q_2+q_3)$

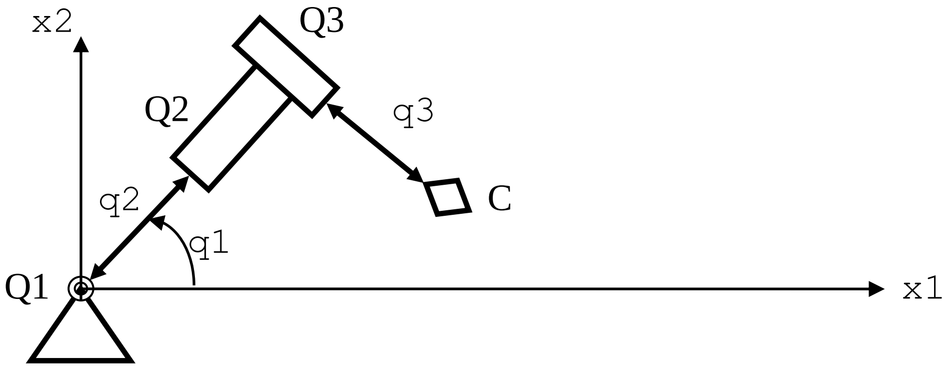
4) Одна поступательная координата по оси x_2 (q1) и два вращательных шарнира (q2, q3) (схема ПуВВ).



Координаты схвата вычисляются по формулам:

$$X_1=l_1*\sin(q_2)+l_2*\sin(q_2+q_3)$$
$$X_2=q_1+l_1*\cos(q_2)+l_2*\cos(q_2+q_3)$$

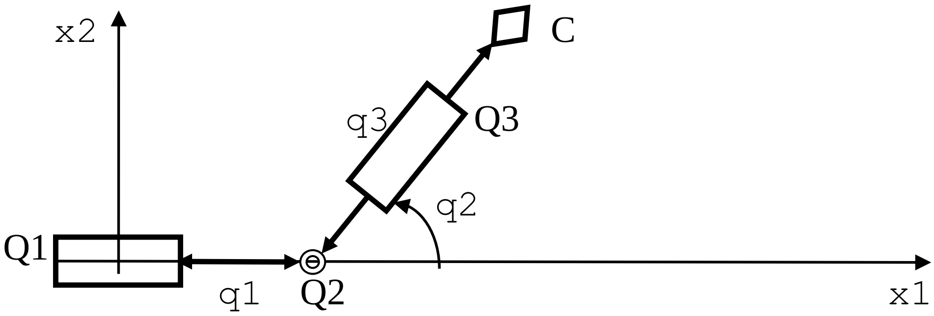
5) Один вращательный шарнир (q1) и две взаимно перпендикулярные поступательные координаты (q2, q3) (схема ВПП).



Координаты схвата вычисляются по формулам:

$$x_1=q_2*\cos(q_1)+q_3*\sin(q_1)$$
$$x_2=q_2*\sin(q_1)-q_3*\cos(q_1)$$

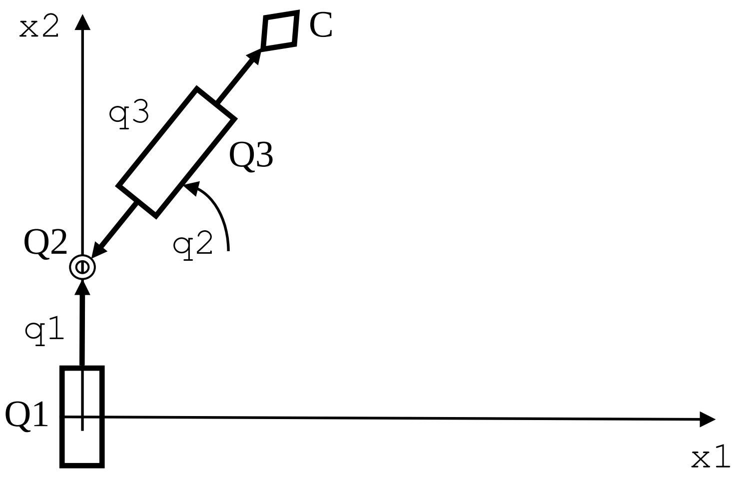
6) Две поступательные координаты (первая по оси x_1) (q1,q3) и один вращательный шарнир (q2) (схема ПхВП).



Координаты схвата вычисляются по формулам:

$$x_1=q_1+q_3*\cos(q_2)$$
$$x_2=q_3*\sin(q_2)$$

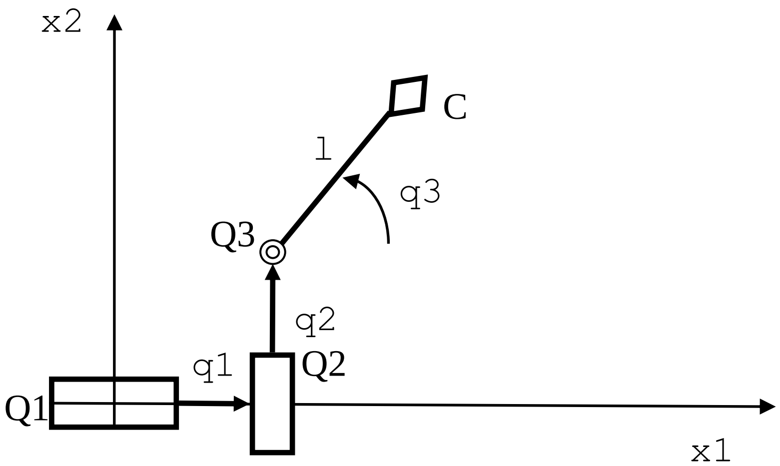
7) Две поступательные координаты (первая по оси x_2) ($q1, q3$) и один вращательный шарнир ($q2$) (схема ПуВП).



Координаты схвата вычисляются по формулам:

$$x_1 = q_3 * \cos(q_2)$$
$$x_2 = q_1 + q_3 * \sin(q_2)$$

8) Две поступательные координаты (первая по оси x_1) ($q1, q2$) и один вращательный шарнир ($q3$) (схема ППВ).



Координаты схвата вычисляются по формулам:

$$x_1 = q_1 + l * \cos(q_3)$$
$$x_2 = q_2 + l * \sin(q_2)$$

Таблица 3.

Вариант №	Схема №	Параметры, координаты целевой точки	Начальный вектор `q0`
1	1	l_1=5 xc=[12; 0]; xc=[5;5];	`[0.7 -1 1]`

2	2	l_1=5
		xc=[12; 0];
		xc=[10;10];
			`[0.7 1 0.5]`

3	3	l_1=5, l_2=5 xc=[12; 0]; xc=[10;5];	`[1 -1 0.5]`
4	4	l_1=5, l_2=5 xc=[12; 0]; xc=[5;10];	`[1 -1 -0.7]`
5	5	xc=[12; 0]; xc=[5;5];	`[0.7 1 1]`
6	6	xc=[12; 0]; xc=[10;5];	`[1 0.7 1]`
7	7	xc=[12; 0]; xc=[5;10];	`[1 0.7 1]`
8	8	l_1=5 xc=[12; 0]; xc=[0;10];	`[1 1 0.7]`
9	4	l_1=6, l_2=6 xc=[12; 0]; xc=[0;5];	`[1 -1 -0.7]`
10	3	l_1=6, l_2=6 xc=[12; 0]; xc=[12;5];	`[1 -1 0.5]`
11	2	l_1=6 xc=[12; 0]; xc=[10;3];	`[0.7 1 0.5]`
12	1	l_1=6 xc=[12; 0]; xc=[3;10];	`[0.7 -1 1]`
13	6	xc=[12; 0]; xc=[7;7];	`[2 0.5 2]`
14	5	xc=[12; 0]; xc=[10;7];	`[2 0.7 2]`
15	8	l_1=6 xc=[12; 0]; xc=[7;10];	`[2 2 0.7]`
16	7	xc=[12; 0]; xc=[0;7];	`[2 0.7 2]`
17	6	xc=[12; 0]; xc=[8;5];	`[1.5 0.5 1.5]`
18	5	xc=[12; 0]; xc=[5;8];	`[1.5 0.7 1.5]`
19	4	l_1=6, l_2=5 xc=[12; 0]; xc=[7;5];	`[2 -1 0.5]`
20	3	l_1=5, l_2=6 xc=[12; 0]; xc=[5;7];	`[2 1 -0.5]`

3.6. Изменить выражение целевой функции в постановке задачи, используя следующие определения нормы разности векторов:

а) $d = |x_1 - xc_1| + |x_2 - xc_2|$;

б) $d = \max(|x_1 - xc_1|,\ |x_2 - xc_2|)$.

Провести необходимую коррекцию программ, выполнить вычисления, сравнить полученные результаты, сделать выводы.

3.7. Модифицировать целевую функцию примера из пункта 3.3., включив условие минимизации изменения значений обобщённых координат по сравнению с начальными величинами. Тогда целевая функция должна вычисляться по формуле:

```
d=norm(x-xc)+norm(q-q0);
```

Соответствующая функция примет вид:

```
function d=distq(q,l,xc,q0)
x=coord(q,l);
d=norm(x-xc)+norm(q-q0);
```

Её текст необходимо сохранить под именем `distq.m`.

Для расчётов примем те же исходные данные:

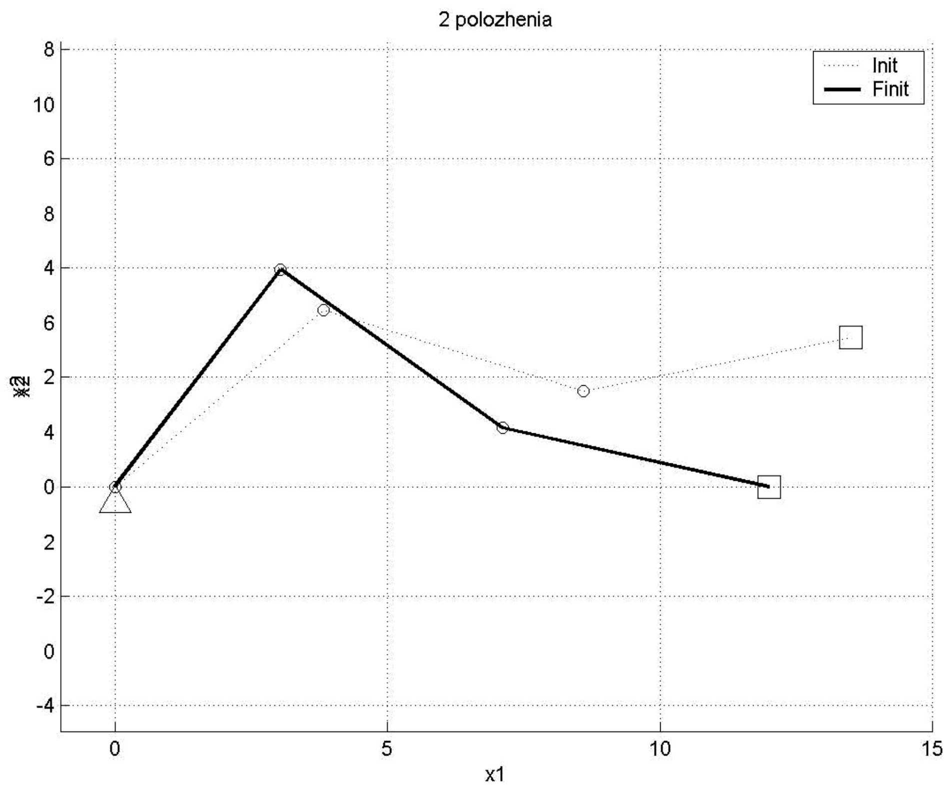
```
q0=[0.7 -1 0.5];
l=[5 5 5];
xc=[12; 0];
```

Обращение к функции минимизации примет вид:

```
[q,fs, e, inf]=fminsearch(@distq,q0, [], l, xc, q0)
```

Результаты сравнить с полученными в пункте 3.3.

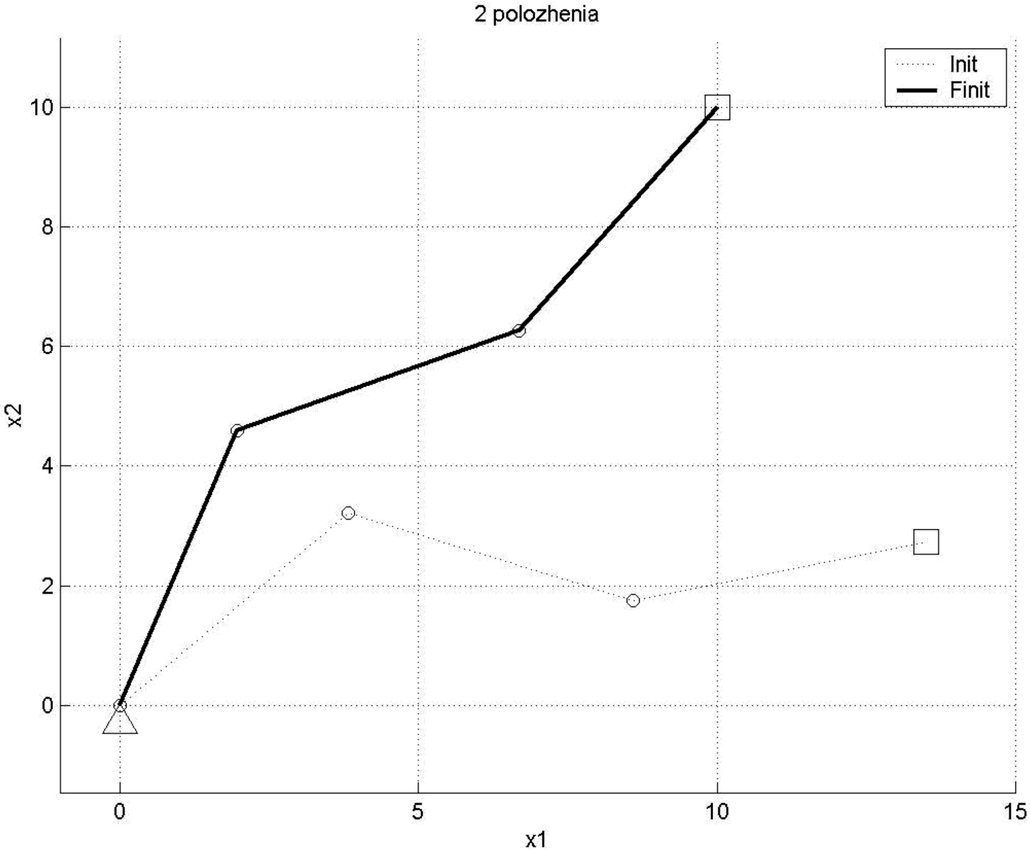
Использовать функцию `prog15_7` для графического отображения решения, задав в неё рассчитанные значения `qOptim`. Рисунок должен выглядеть примерно так:



3.8. Изменить значение целевой точки положения схвата `xc=[10; 10];`

и провести аналогичные расчёты.

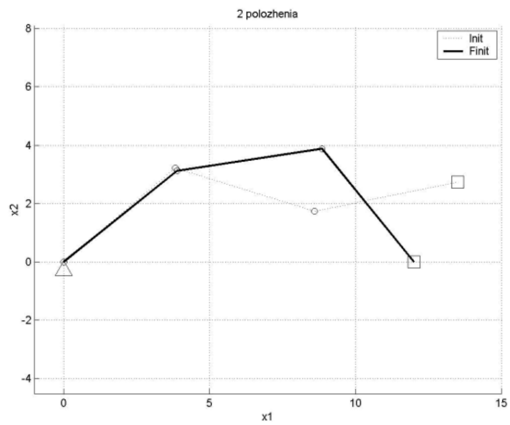
В результате должна получиться примерно такая конфигурация, как представлена на следующем рисунке:

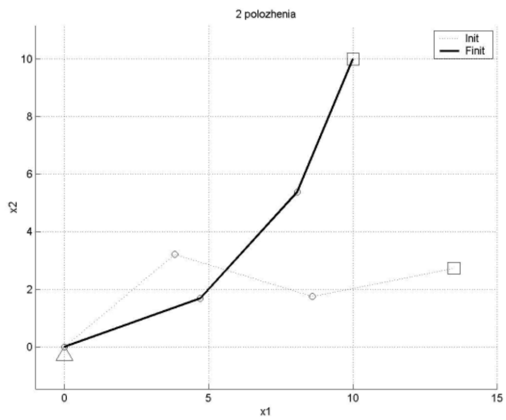


3.9. Изменить целевую функцию минимизации, которая должна обеспечить наименьшее отклонение обобщённых координат от нулевых значений: `d=norm(x-xc)+norm(q);`

Провести исследование, аналогичное пунктам 3.7, 3.8, для двух значений требуемого положения схвата: `xc=[12; 0];` `xc=[10; 10];`

Результирующие рисунки будут иметь примерный вид:





Прокомментировать полученные результаты и сделать выводы.

3.10. Для кинематических схем манипуляторов согласно вариантам из табл. 3, по примеру пунктов 3.7–3.9, исследовать влияние вида целевой функции на получаемые результаты.