

**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 3**

«Использование функции fminunc»

Выполнил:

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

(подпись) (ФИО)

Принял

преподаватель: \_\_\_\_\_\_\_\_\_\_ \_Порунов М. \_

(подпись) (ФИО)

Дата:\_\_\_\_\_\_\_\_\_

Москва 2020.

# 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

Функция fminunc находит минимум неограниченной многомерной функции.

Простой пример:

A = [23 42 37 15 52];

M = min(A)

M = 15

Варианты обращения к функции fminunc:

x = fminunc(fun,x0)

x = fminunc(fun,x0,options)

x = fminunc(problem)

[x,fval] = fminunc(...)

[x,fval,exitflag] = fminunc(...)

[x,fval,exitflag,output] = fminunc(...)

[x,fval,exitflag,output,grad] = fminunc(...)

[x,fval,exitflag,output,grad,hessian] =fminunc(...)

Входные аргументы:

**fun** – либо только целевая функция, либо функция, возвращающая значения целевой функции и ее градиента, либо функция, возвращающая значения целевой функции, ее градиента и гессиана;

**x0** – начальная точка;

**options** – используется для внесения изменений в настройки параметров процесса;

Смысл обозначений выходных аргументов:

**x** и **fval** – принимаемый аргумент x b целевой функции вектор значений искомых переменных ;

**exitflag** – указывает причину завершения алгоритма, может принимать следующие значения:

1 – величина градиента достаточно мала.

2 – Изменение в X слишком мало.

3 – Изменение целевой функции слишком мало.

5 – не удается уменьшить функцию вдоль направления поиска.

0 – слишком много оценок функций или итераций.

-1 – остановлено функцией вывода / построения.

-3 – проблема кажется безграничной.

**grad** – градиент целевой функции;

**hessian** – гессиан целевой функции;

**Output** – структура, содержащая информацию о процессе в следующих полях:

Iterations – количество итераций,

funcCount – количество оценок функций в выходных данных,

firstorderopt – условие оптимальности 1-го порядка,

algorithm – использованный алгоритм,

cgiterations – общее число итераций (для алгоритма большой размерности),

stepsize – заключительное смещение по X (для алгоритмов средней размерности),

message – сообщение о завершении.

# 3.1.

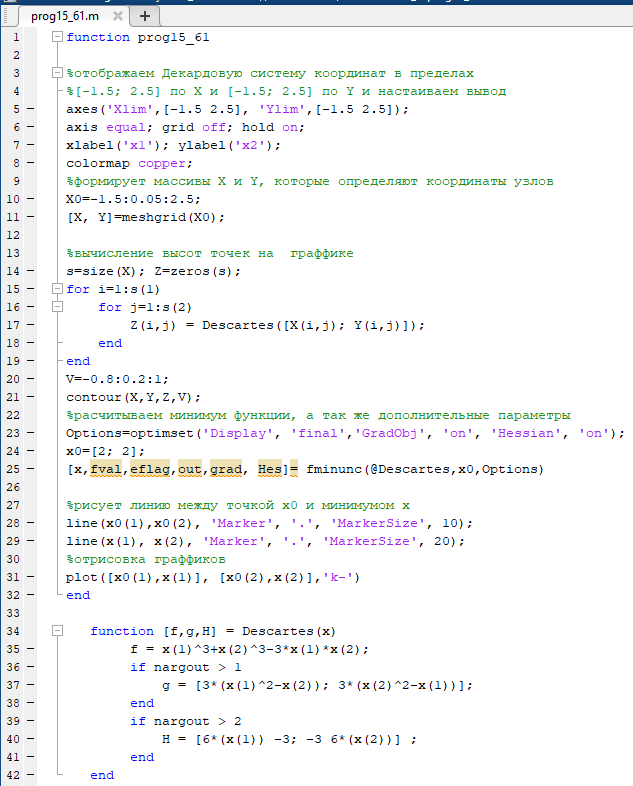


Рис. 1. Содержание m-файла prog15\_61.

Вызов функции имеет вид:

>> prog15\_61

Результат её выполнения:

Local minimum found.

Optimization completed because the size of the gradient is less than

the default value of the optimality tolerance.

<stopping criteria details>

x =

1

1

fval = -1

eflag = 1

out = struct with fields:

iterations: 1

funcCount: 2

stepsize: 1.4142

lssteplength: 0.1667

firstorderopt: 0

algorithm: 'quasi-newton'

message: 'Local minimum found.↵↵Optimization completed because the size of the gradient is less than↵the default value of the optimality tolerance.↵↵Stopping criteria details:↵↵Optimization completed: The first-order optimality measure, 0.000000e+00, is less ↵than options.OptimalityTolerance = 1.000000e-06.↵↵Optimization Metric Options↵relative norm(gradient) = 0.00e+00 OptimalityTolerance = 1e-06 (default)'

grad =

0

0

Hes =

6.0000 -3.0000

-3.0000 6.0000

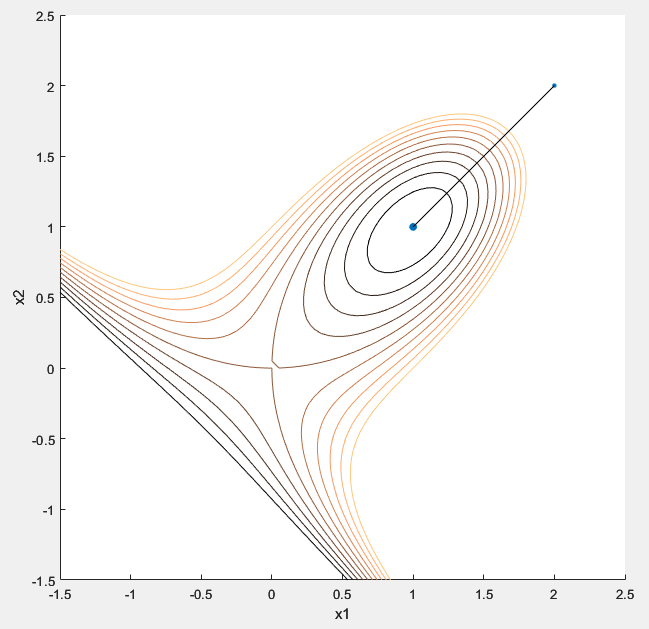


Рис. 2. Выводимый график с точкой минимума и точкой X0.

Как указанно в комментариях в программе, данный скрипт в начале отображаем Декардовую систему координат в пределах [-1.5; 2.5] по оси X и [-1.5; 2.5] по оси Y, настаивает параметры отображения, формирует массивы X и Y, которые определяют координаты узлов

3.2. Задание:

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.

>> x0=[2;2];

>> [x,fval,eflag,out,Grad, Hes]= fminunc('x(1)^3+x(2)^3-3\*x(1)\*x(2)',x0)

Local minimum found.

Optimization completed because the size of the gradient is less than

the default value of the optimality tolerance.

<stopping criteria details>

x =

1

1

fval = -1

eflag = 1

out = struct with fields:

iterations: 1

funcCount: 6

stepsize: 1.4142

lssteplength: 0.1667

firstorderopt: 5.9605e-08

algorithm: 'quasi-newton'

message: 'Local minimum found.↵↵Optimization completed because the size of the gradient is less than↵the default value of the optimality tolerance.↵↵Stopping criteria details:↵↵Optimization completed: The first-order optimality measure, 8.514949e-09, is less ↵than options.OptimalityTolerance = 1.000000e-06.↵↵Optimization Metric Options↵relative norm(gradient) = 8.51e-09 OptimalityTolerance = 1e-06 (default)'

Grad =

1.0e-07 \*

0.5960

0.5960

Hes =

6.0007 -3.0000

-3.0000 6.0007

**Вывод**: в данном случае градиент и гессиан рассчитываются автоматически, а не внутри функции, как это было в прошлом пункте (код в пункте 3,1 отличается от кода из методички т.к. иначе появлялись ошибки при выполнении функции fminunc).

# 3.3. Задача об управлении роботом

Координаты схвата выражаются через длины звеньев l1, l2, l3 и углы q1, q2, q3 по формулам:

x1=l1\*cos(q1)+l2\*cos(q1+q2)+l3\*cos(q1+q2+q3) x2=l1\*sin(q1)+l2\*sin(q1+q2)+l3\*sin(q1+q2+q3)

Вычисление координат схвата выполняет функция:

function x=coord(q,l)

Q=cumsum(q);

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

end

Целевая функция рассчитывает расстояние от начальных координат схвата до конечного положения схвата. Вычисляется следующим образом:

function d=dist(q,l,xc)

x=coord(q,l);

d=norm(x-xc);

end

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-05

Графическая иллюстрация получается с помощью программы:

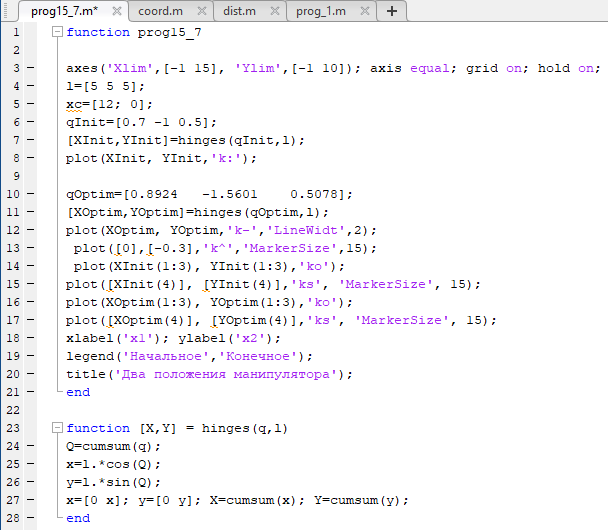


Рис. 3. Скрипт вывода графической иллюстрации.

Результат представлен на следующем рисунке

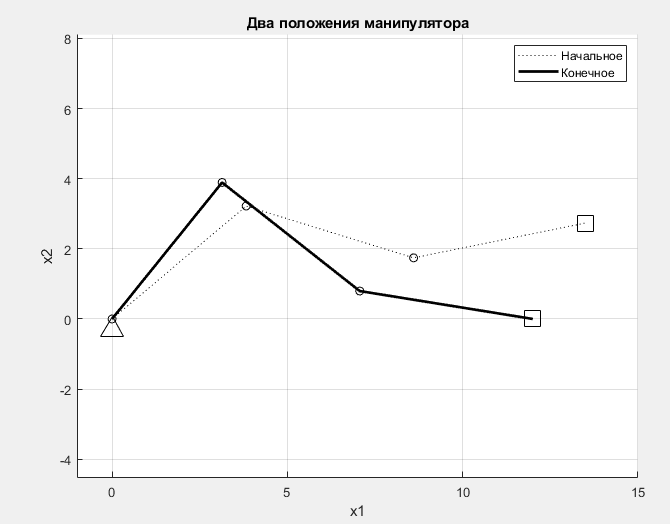


Рис. 4.Иллюстрация работы скрипта.

Изменяем целевую точку схвата:

xc=[10; 5];

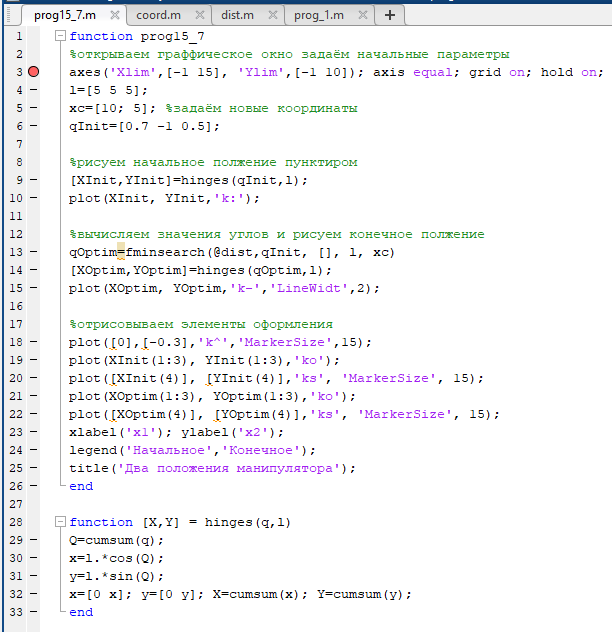


Рис. 5. Скрипт вывода графической иллюстрации.

Код программы изменён так, чтоб можно было ввести конечные координаты, и программы рассчитает и выводит углы звеньев и отображает это на иллюстрации (Рис 6).

>> prog15\_7

qOptim = 1.5644 -1.4842 -0.1604

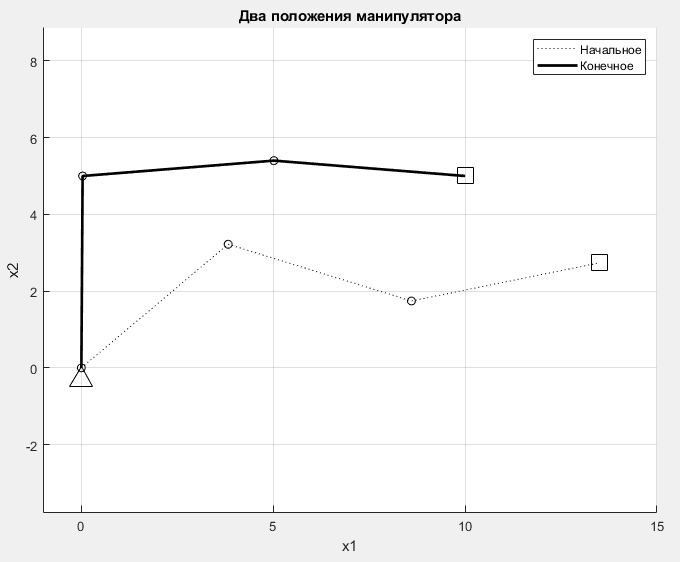


Рис. 6. Иллюстрация работы изменённого скрипта.

# 3.4. Изменяем начальный вектор

q0=[-0.7 1 0.5].

Найдём значение q для целевой точки схвата xc=[12; 0];

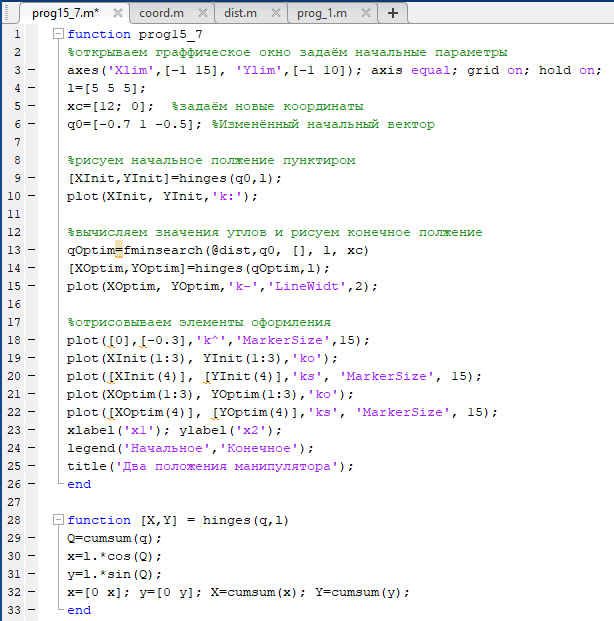


Рис. 7. Скрипт вывода графической иллюстрации.

>> prog15\_7

qOptim = -0.8924 1.5601 -0.5078

Где qOptim — значение обобщенных координат для целевой точки схвата xc.

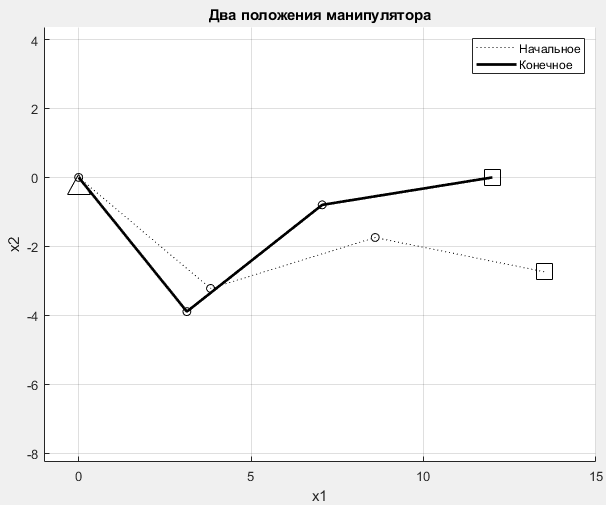


Рис. 8. Иллюстрация работы изменённого скрипта.

**Вывод**: при новых заданных начальных углах программы минимизировала суммарное изменение углов q для достижения целевой точки, при этом при одном и том же конечном положении (как и в пункте 3,3 данной лабораторной) углы имеют разное значение, т. к. при данном начальном положении для достижения конечного положения (как в пункте 3,3) требуется совершить большее движение (повернуть суммарно звенья на больший угол), что противоречит используемому алгоритму.

# 3.5. Решить аналогичную задачу для робота, кинематическая схема которого соответствует одному из следующих вариантов

|  |  |  |  |
| --- | --- | --- | --- |
| 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).

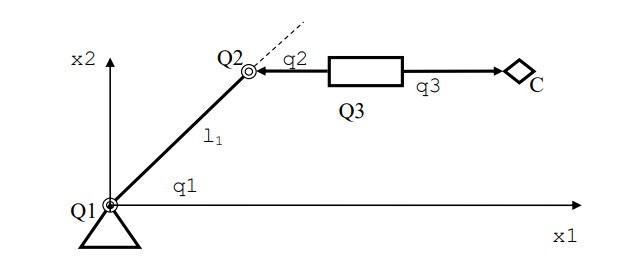


Рис. 9. Кинематическая схема робота.

Координаты схвата вычисляются по формулам: x1=l1\*cos(q1)+q3\*cos(q1+q2) x2=l1\*sin(q1)+q3\*sin(q1+q2)

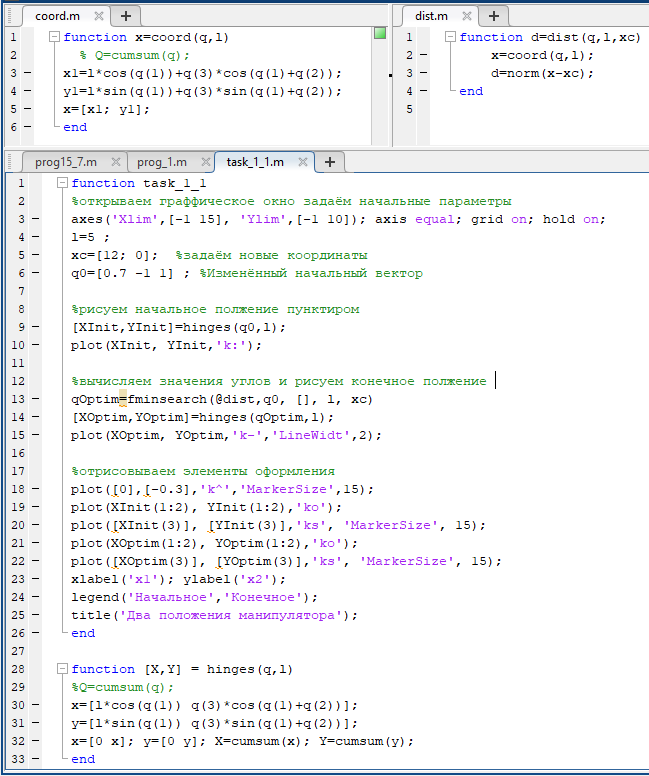


Рис. 10. Скрипт задания.

При xc=[12; 0];

>> task\_1\_1

qOptim = -0.1383 0.2359 7.0814

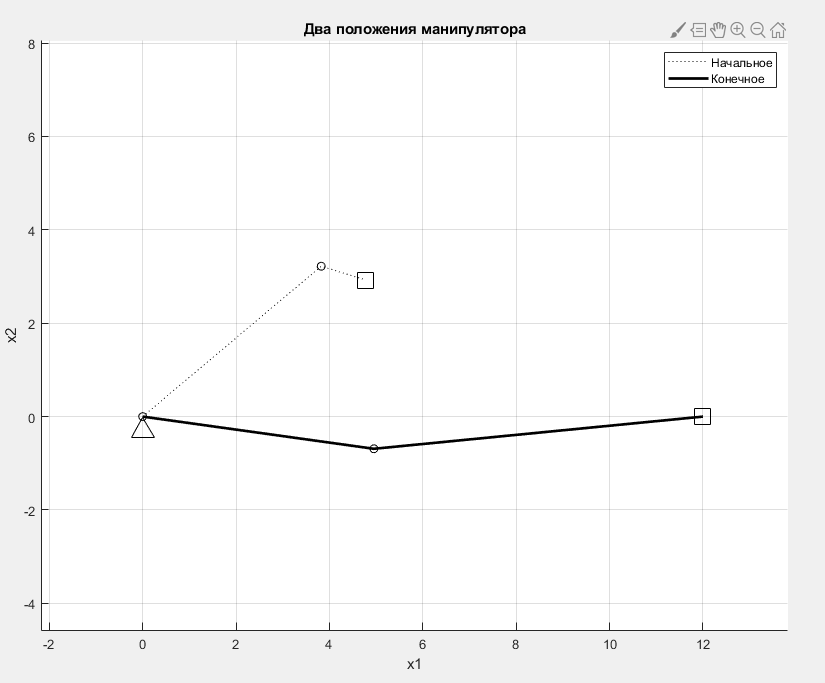


Рис. 10. Иллюстрация работы скрипта задания.

При xc=[5; 5];

>> task\_1\_1

qOptim = 0.8686 -0.2797 2.1293

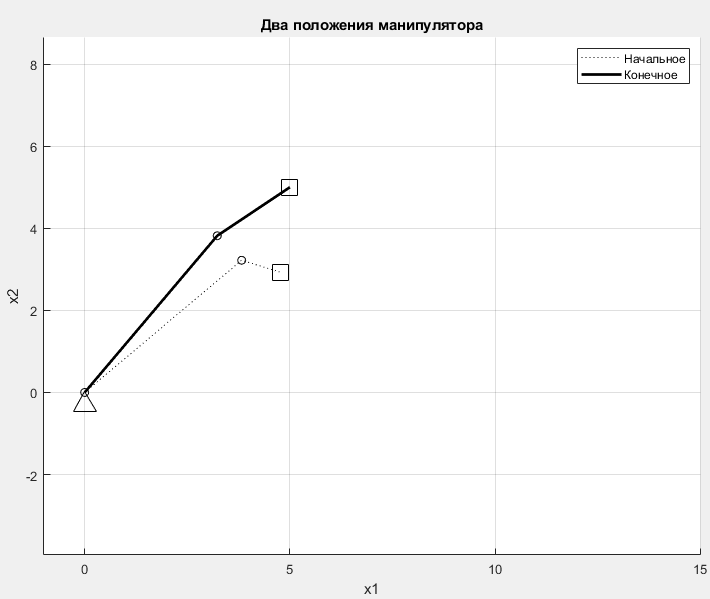


Рис. 11. Иллюстрация работы скрипта задания.

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

Для более подробного ответа воспользуемся следующим кодом:

[qOptim,fval,exitflag,output] =fminsearch(@dist,q0, [], l, xc)

До изменений при xc=[12; 0]:

>> task\_1\_1

qOptim = -0.1383 0.2359 7.0814

fval = 4.3204e-05

output = struct with fields:

iterations: 102

funcCount: 179

algorithm: 'Nelder-Mead simplex direct search'

## а) d = |x1 – xc1| + |x2 – xc2|;

Скорректирована функция dist:

function d=dist(q,l,xc)

x=coord(q,l);

d= abs(x(1)-xc(1))+abs(x(2)-xc(2));

end

При xc=[12; 0];

>> task\_1\_1

qOptim = -0.4661 0.7560 7.8613

fval = 2.5748e-05

output = struct with fields:

iterations: 192

funcCount: 350

algorithm: 'Nelder-Mead simplex direct search'

При xc=[5; 5];

>> task\_1\_1

qOptim = 0.6118 0.5561 2.3134

fval = 9.0764e-06

exitflag = 1

output = struct with fields:

iterations: 86

funcCount: 157

algorithm: 'Nelder-Mead simplex direct search'

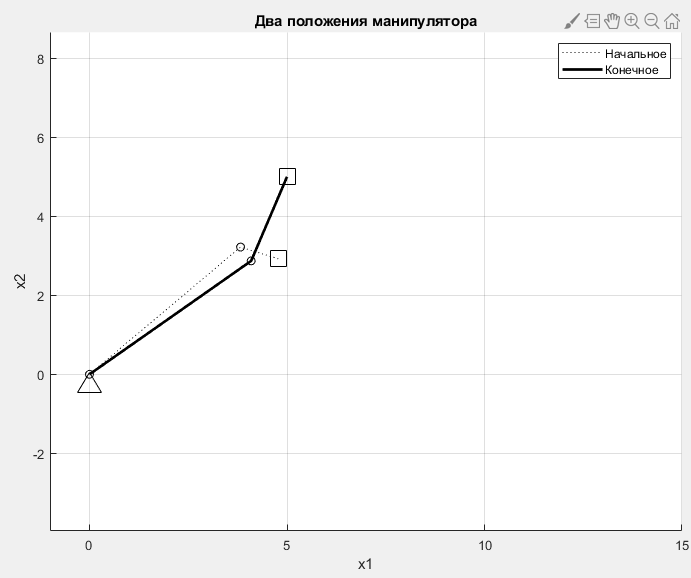
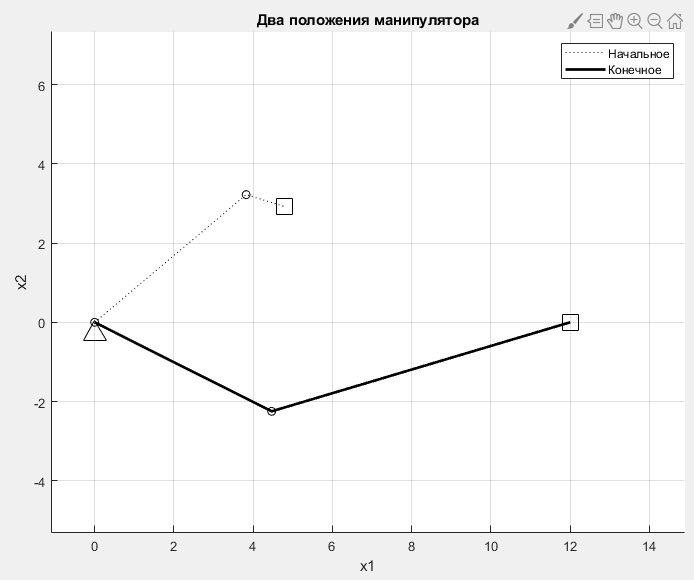


Рис. 12-13. Иллюстрация работы при xc=[12; 0] и xc=[5; 5] .

## б) d = max(|x1 – xc1|, |x2 – xc2|).

Скорректирована функция dist:

function d=dist(q,l,xc)

x=coord(q,l);

d= max(abs(x(1)-xc(1)),abs(x(2)-xc(2)));

end

При xc=[12; 0];

>> task\_1\_1

qOptim = -0.7722 1.1650 9.1123

fval = 2.4055e-06

exitflag = 1

output = struct with fields:

iterations: 105

funcCount: 194

algorithm: 'Nelder-Mead simplex direct search'

При xc=[5; 5];

>> task\_1\_1

qOptim = 0.8570 -0.2416 2.1144

fval = 1.0246e-05

exitflag = 1

output = struct with fields:

iterations: 108

funcCount: 190

algorithm: 'Nelder-Mead simplex direct search'

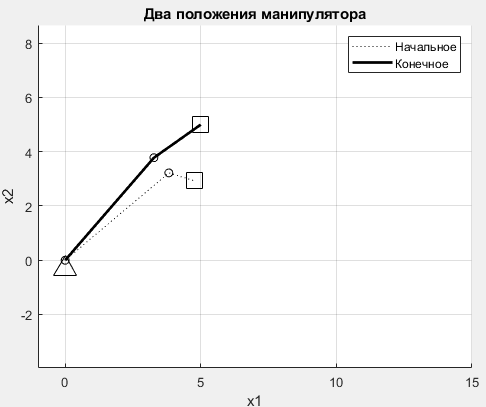
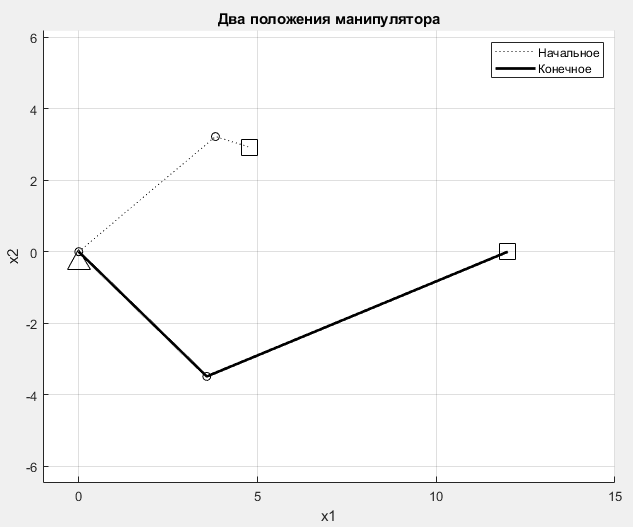


Рис. 14-15. Иллюстрация работы при xc=[12; 0] и xc=[5; 5] .

**Вывод**: при различных методах определения погрешности положение обобщённых координат отличаются, но при этом конечная точка достигается. Во всех примерах производится примерно одно и то же количество итераций и оценок функций (но в случае *а* при xc=[12; 0] их было больше).

3.7

Модифицируем 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)

>> prog15\_7

q = 0.9172 -1.5348 0.4011

fs = 0.5856

e = 1

inf = struct with fields:

iterations: 159

funcCount: 279

algorithm: 'Nelder-Mead simplex direct search'

message: 'Optimization terminated:↵ the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-04 ↵ and F(X) satisfies the convergence criteria using OPTIONS.TolFun of 1.000000e-04 ↵'

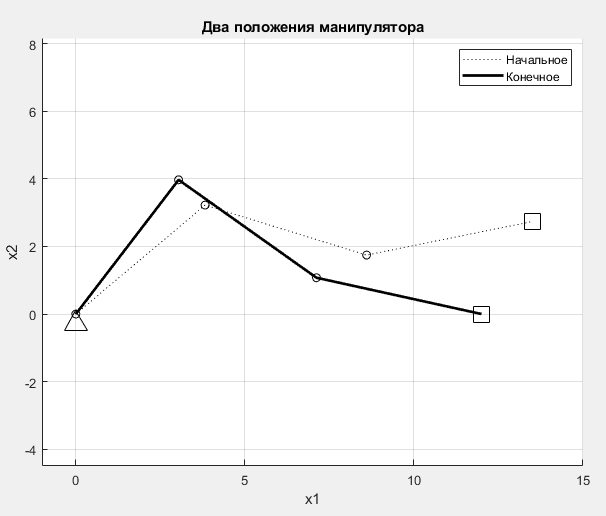


Рис. 16. Иллюстрация работы.

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

и провести аналогичные расчёты.

>> prog15\_7

qOptim = 1.1678 -0.8281 0.5035

>> prog15\_7

q = 1.1678 -0.8281 0.5035

fs = 0.4985

e = 1

inf = struct with fields:

iterations: 88

funcCount: 160

algorithm: 'Nelder-Mead simplex direct search'

message: 'Optimization terminated:↵ the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-04 ↵ and F(X) satisfies the convergence criteria using OPTIONS.TolFun of 1.000000e-04 ↵'

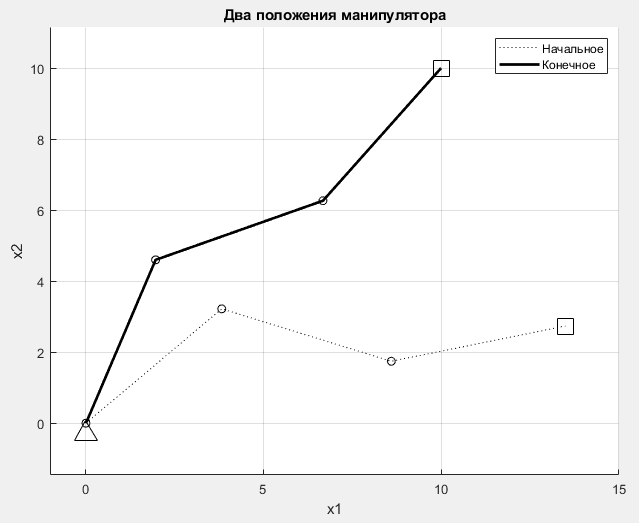


Рис. 17. Иллюстрация работы.

Вывод: в данном случае программа минимизирует так же и разницу обобщённых координат, и робот пытается сохранить похожие на начальные обобщенные координаты.

3.9. Изменим целевую функцию минимизации, которая должна обеспечить наименьшее отклонение обобщённых координат от нулевых значений:

function d=dist(q,l,xc) %,q0)

x=coord(q,l);

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

end

Провести исследование, аналогичное пунктам 3.7, 3.8, для двух значений требуемого положения схвата:

При xc=[12; 0]:

>> prog15\_7

Exiting: Maximum number of function evaluations has been exceeded

- increase MaxFunEvals option.

Current function value: 1.345703

q = 0.6759 -0.5253 -1.0384

fs = 1.3457

e = 0

inf = struct with fields:

iterations: 352

funcCount: 600

algorithm: 'Nelder-Mead simplex direct search'

message: 'Exiting: Maximum number of function evaluations has been exceeded↵ - increase MaxFunEvals option.↵ Current function value: 1.345703 ↵'

При xc=[10; 10]:

>> prog15\_7

q = 0.3439 0.4899 0.3400

fs = 0.6884

e = 1

inf = struct with fields:

iterations: 224

funcCount: 398

algorithm: 'Nelder-Mead simplex direct search'

message: 'Optimization terminated:↵ the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-04 ↵ and F(X) satisfies the convergence criteria using OPTIONS.TolFun of 1.000000e-04 ↵'

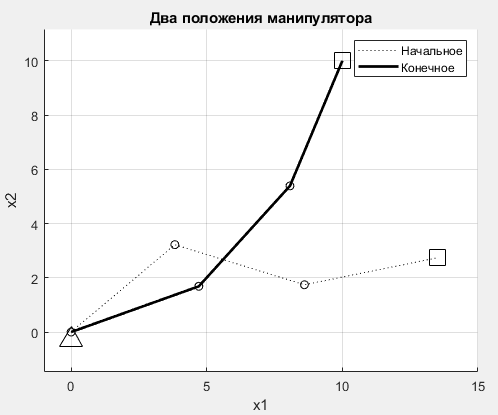
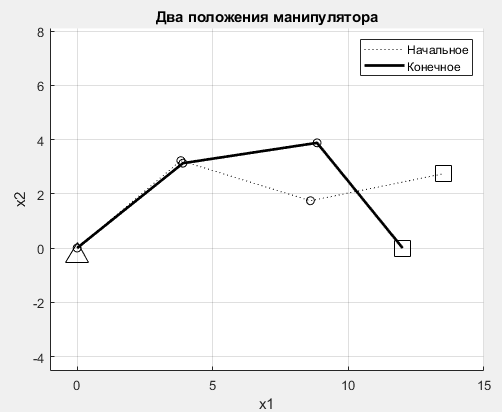


Рис. 18-19. Иллюстрация работы при xc=[12; 0] и xc=[10; 10] .

**Вывод**: в данном случае совершается больше итераций и оценок, но при этом обобщенные координаты пытаются достичь одинаковых значений (угла в данном случае).

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.

d=norm(x-xc)+norm(q–q0);

Соответствующая функция примет вид:

function d=dist(q,l,xc,q0)

x=coord(q,l);

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

end

Для расчётов примем те же исходные данные:

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

Обращение к функции минимизации примет вид:

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

>> task\_1\_1

q = 0.0436 -0.0747 7.0081

fs = 6.1143

e = 1

inf = struct with fields:

iterations: 130

funcCount: 229

algorithm: 'Nelder-Mead simplex direct search'

message: 'Optimization terminated:↵ the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-04 ↵ and F(X) satisfies the convergence criteria using OPTIONS.TolFun of 1.000000e-04 ↵'

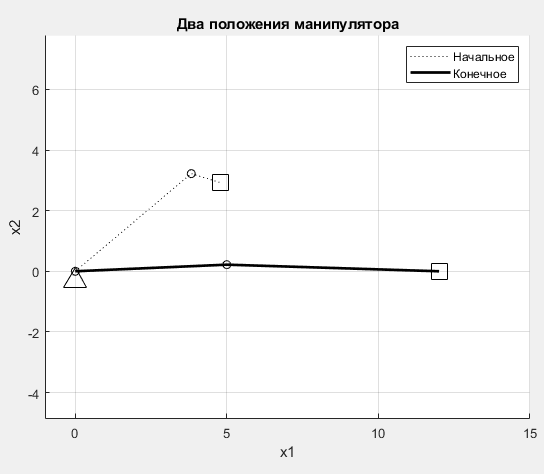


Рис. 20. Иллюстрация работы.

Изменим значение целевой точки положения схвата xc=[10; 10];

и провести аналогичные расчёты.

>> task\_1\_1

q = 0.8074 -0.0341 9.1440

fs = 8.2018

e = 1

inf = struct with fields:

iterations: 151

funcCount: 264

algorithm: 'Nelder-Mead simplex direct search'

message: 'Optimization terminated:↵ the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-04 ↵ and F(X) satisfies the convergence criteria using OPTIONS.TolFun of 1.000000e-04 ↵'

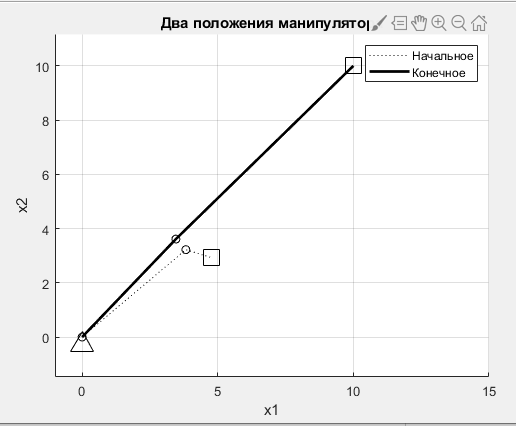


Рис. 21. Иллюстрация работы.

Изменим целевую функцию минимизации, которая должна обеспечить наименьшее отклонение обобщённых координат от нулевых значений:

function d=dist(q,l,xc)

x=coord(q,l);

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

end

Провести исследование, аналогичное пунктам 3.7, 3.8, для двух значений требуемого положения схвата:

При xc=[12; 0]:

>> task\_1\_1

q = -0.0000 -0.0000 3.0836

fs = 7.0000

e = 1

inf = struct with fields:

iterations: 66

funcCount: 121

algorithm: 'Nelder-Mead simplex direct search'

message: 'Optimization terminated:↵ the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-04 ↵ and F(X) satisfies the convergence criteria using OPTIONS.TolFun of 1.000000e-04 ↵'

При xc=[10; 10]:

>> task\_1\_1

q = 0.7752 0.0157 9.1425

fs = 9.1754

e = 1

inf = struct with fields:

iterations: 130

funcCount: 225

algorithm: 'Nelder-Mead simplex direct search'

message: 'Optimization terminated:↵ the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-04 ↵ and F(X) satisfies the convergence criteria using OPTIONS.TolFun of 1.000000e-04 ↵'

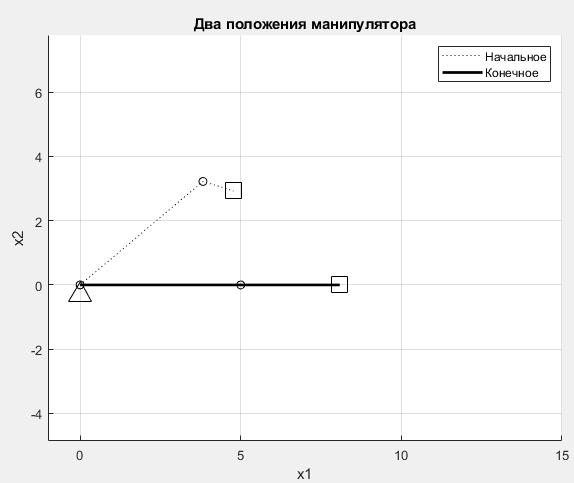
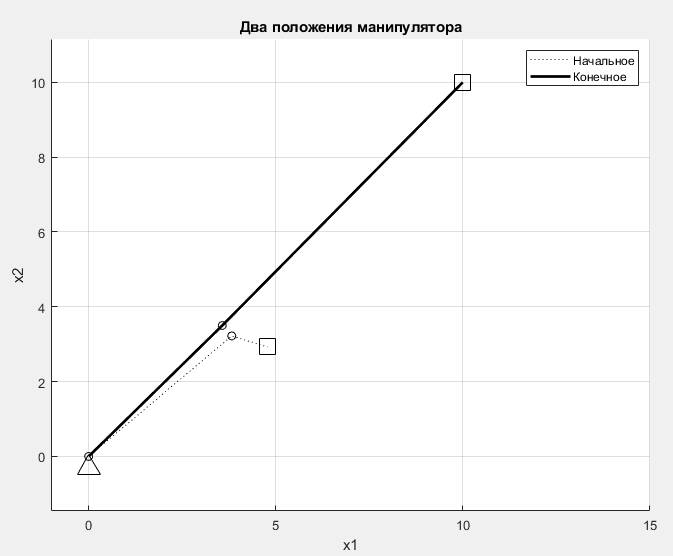
 

Рис. 18-19. Иллюстрация работы при xc=[12; 0] и xc=[10; 10] .

**Вывод**: данные расчёты так же подтверждают прошлые ответы по пунктам 3.7-3.9, но в данном случае при значениях xc=[12; 0] программа не достигла конечного положения, пытаясь усреднить все обобщенные координаты