Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет информационных технологий и управления

Кафедра систем управления

Дисциплина: Математические основы теории систем

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

на тему

Математические модели систем управления и методы оптимизации

БГУИР КР1-53 01 07 24 ПЗ

Студент: гр.222401 Мурашко К.Н.

Руководитель: кандидат технических наук, доцент Павлова А.В.

Минск 2023

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Содерж66ание**

[Введение 3](#_Toc150855845)

[1 ИССЛЕДОВАНИЕ СИСТЕМ УПРАВЛЕНИЯ 4](#_Toc150855846)

[***1.1*** ***Описание tf-, zpk-, ss- форм. Вычисление и построение в MATLAB временных характеристик системы.*** 4](#_Toc150855847)

[***1.2*** ***Построение в МАТLAB частотных характеристик и вид асимптотических ЛАЧХ и ЛФЧХ*** 6](#_Toc150855848)

[***1.3*** ***Составление уравнений состояния в нормальной и канонической формах. Результаты моделирования системы.*** 7](#_Toc150855849)

[***1.4*** ***Решение уравнения состояния в канонической форме*** 11](#_Toc150855850)

[2 ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ 12](#_Toc150855851)

[***2.1*** ***Расчет оптимального плана и экстремального значения функции цели*** 12](#_Toc150855852)

[***2.2*** ***Исследование двойственной задачи линейного программирования*** 14](#_Toc150855853)

[***2.3*** ***Нахождение целочисленного решения задачи*** 16](#_Toc150855854)

[3 неЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ 17](#_Toc150855855)

[***3.1*** ***Построение ОДЗП, выбор начальной точки поиска*** 17](#_Toc150855856)

[***3.2*** ***Нахождение экстремального значения функции F(x) без учета ограничений на переменные*** 20](#_Toc150855857)

[***3.3*** ***Нахождение экстремума функции F(x) с учетом системы ограничений*** 23](#_Toc150855858)

[Заключение 28](#_Toc150855859)

[Список использованных источников 29](#_Toc150855860)

# Введение

Методы оптимизации находят широкое применение в различных областях науки и техники. Эти методы успешно применяются в решении задач технического проектирования устройств и систем, организационно-экономических и других задач.

В наиболее общем смысле теория оптимизации представляет собой совокупность фундаментальных математических результатов и численных методов, которые позволяют найти наилучший вариант из множества альтернатив и избежать при этом полного перебора и оценивания возможных решений. Знание методов оптимизации является необходимым для инженерной деятельности при создании новых, более эффективных и менее дорогостоящих систем, а также при разработке методов повышения качества функционирования существующих систем [2].

Целью курсовой работы является построение математических моделей линейных систем управления и их моделирование, а также изучение методов оптимизации задач линейного и нелинейного программирования.

Первый раздел посвящен анализу заданной с помощью передаточной функции системы. В этом разделе для этой функции построены переходные и логарифмические амплитудно- и фазочастотная характеристики, а также построены схемы модели в пространстве состояний в нормальной и канонической формах и решено уравнение состояния в канонической форме.

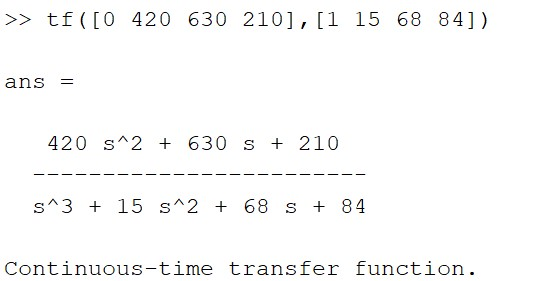
Второй раздел посвящен решению задач линейного программирования. В этом разделе приведено решение прямой задачи линейного программирования и соответствующей ей двойственной задачи, а также целочисленной задачи с помощью симплекс-таблиц.

Третий раздел посвящен решению задач нелинейного программирования. В этом разделе приведено решение такой задачи без ограничений методами Ньютона-Рафсона и наискорейшего спуска, а также с ограничениями методами допустимых направлений Зойтендейка, Куна-Таккера и линейных комбинаций. Результаты решения различными методами сравнены между собой.

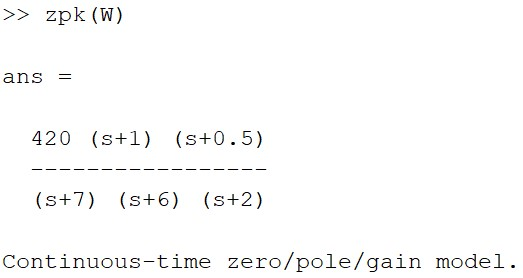
# 1 ИССЛЕДОВАНИЕ СИСТЕМ УПРАВЛЕНИЯ

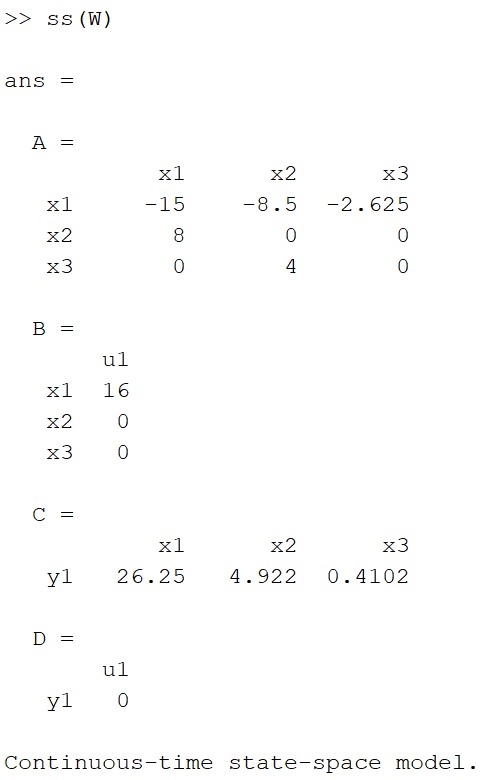
## ***Описание tf-, zpk-, ss- форм. Вычисление и построение в MATLAB временных характеристик системы.***

tf-форма задается двумя векторами строками, составленными из коэффициентов многочленов числителя и знаменателя в порядке убывания степеней s.

Для её записи используем команду tf([0 420 630 210], [1 15 68 84]) и получаем:  
 

zpk-форма - форма в которой передаточная функция описывается двумя векторами-строками и одним числом, задающим нули, полюсы и коэффициент усиления системы.

Для её записи используем команду zpk(W), где W это наша функция в tf форме записи, и получаем:  


ss-форма представляет передаточную функцию в параметрах пространства состояний.  


Далее составим временные характеристики системы.

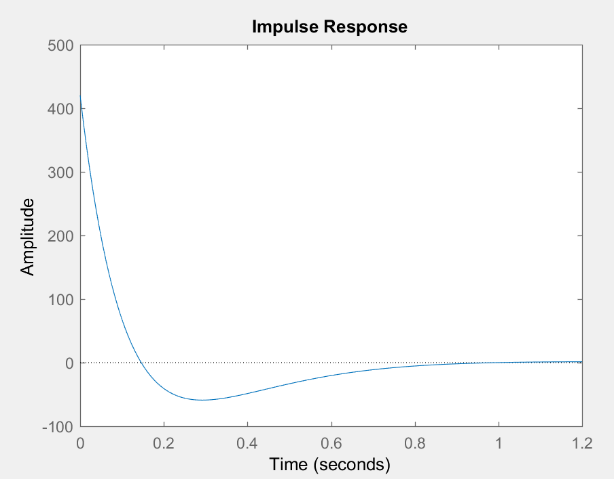
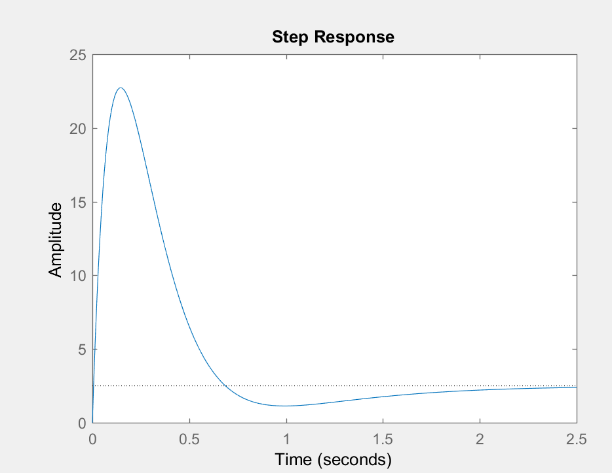
Переходная функция h(t) – это функция, определяющая изменение выходной величины системы (или отдельного элемента) при воздействии на входе единичного ступенчатого сигнала 1(t) при нулевых начальных условиях.

Для составления переходной функции необходимо разложить нашу передаточную функцию на слагаемые и взять отображение по Лаплассу.

Из этой формулы получаем систему уравнений:

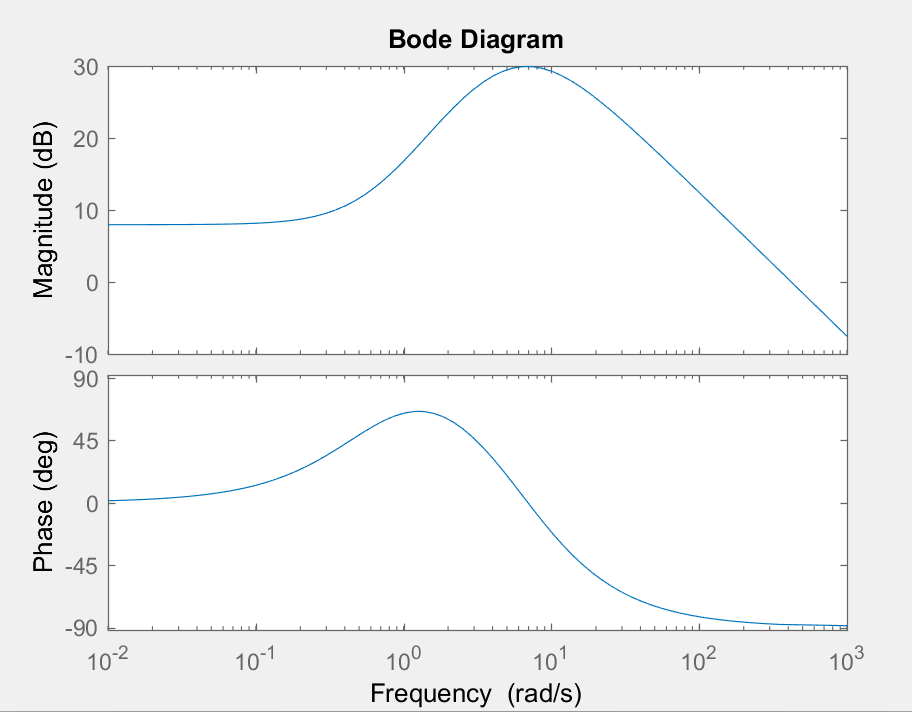
Из этого следует, что:

Далее получаем временную характеристику и весовую функцию:

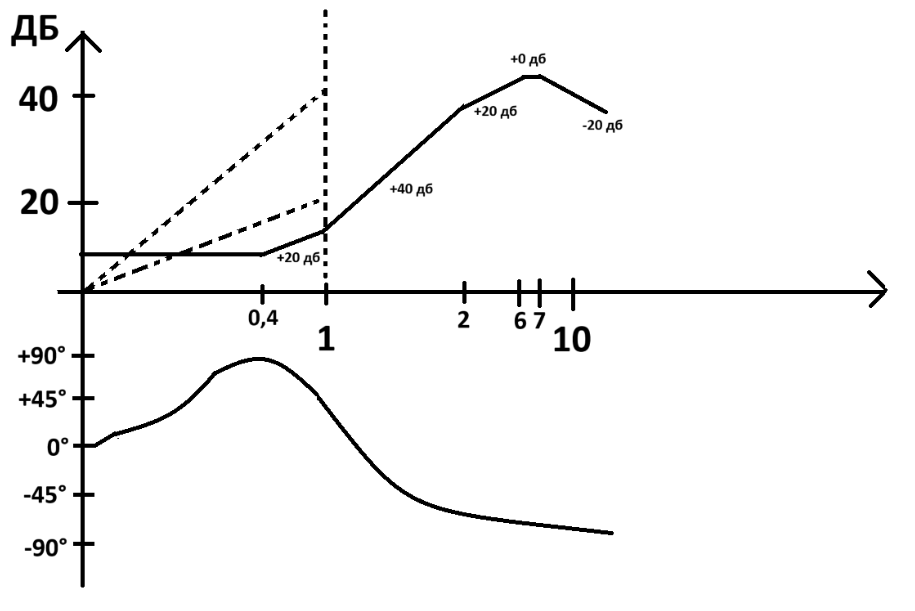
  
Переходная характеристика Импульсная характеристика

## ***Построение в МАТLAB частотных характеристик и вид асимптотических ЛАЧХ и ЛФЧХ***

Построим график ЛАЧХ (Логарифмическая амплитудная частотная характеристика) и ЛФЧХ (Логарифмическая фаза-частотная характеристика) используя функцию bode в MATLAB:



Построенная вручную:



## ***Составление уравнений состояния в нормальной и канонической формах. Результаты моделирования системы.***

Кроме входных и выходных переменных при описании систем выделяют переменные х, связанные с внутренней структурой устройства - переменные состояния. Тогда систему можно описать с помощью уравнений состояния.

Нормальная форма уравнения состояния имеет вид:

 где А – матрица Фробениуса.

Матрица Фробениуса – это квадратная матрица, размер которой определяется порядком дифференциального уравнения. Элементы, стоящие над главной диагональю – единицы; элементы нижней строки — это коэффициенты левой части дифференциального уравнения, взятые с противоположным знаком, все остальные элементы – нулевые.

Дифференциальное уравнение системы имеет вид:

Составим матрицу А =

Для того что бы составить матрицу B необходимо рассчитать коэффициенты b1 b2 b3.

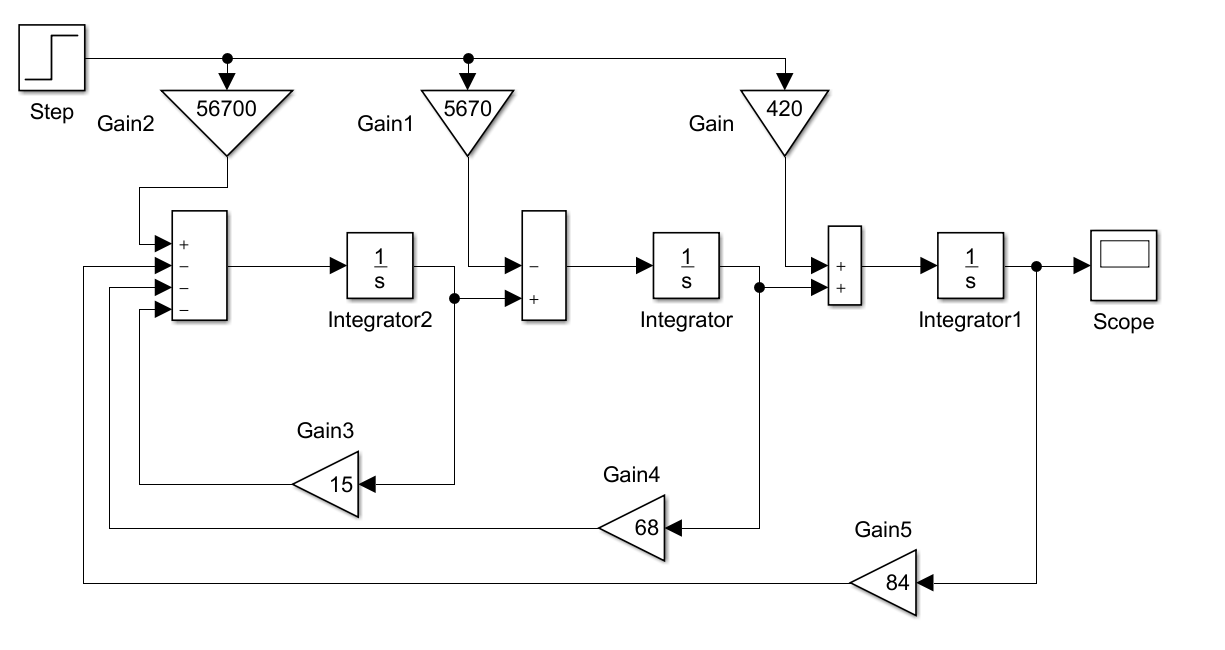
Составим матрицу B =

И запишем матрицу C =

Далее составим матричное уравнение:

Переводим уравнение в линейный вид:

Строим схему стандартного вида в Simulink согласно уравнению:

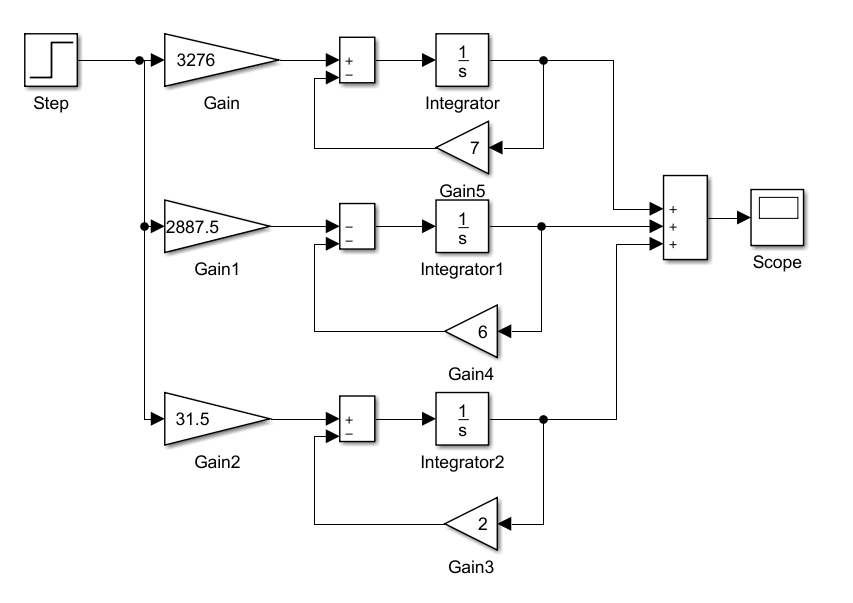


Далее начнем переводить систему в каноническую форму, для этого составим матрицы , и

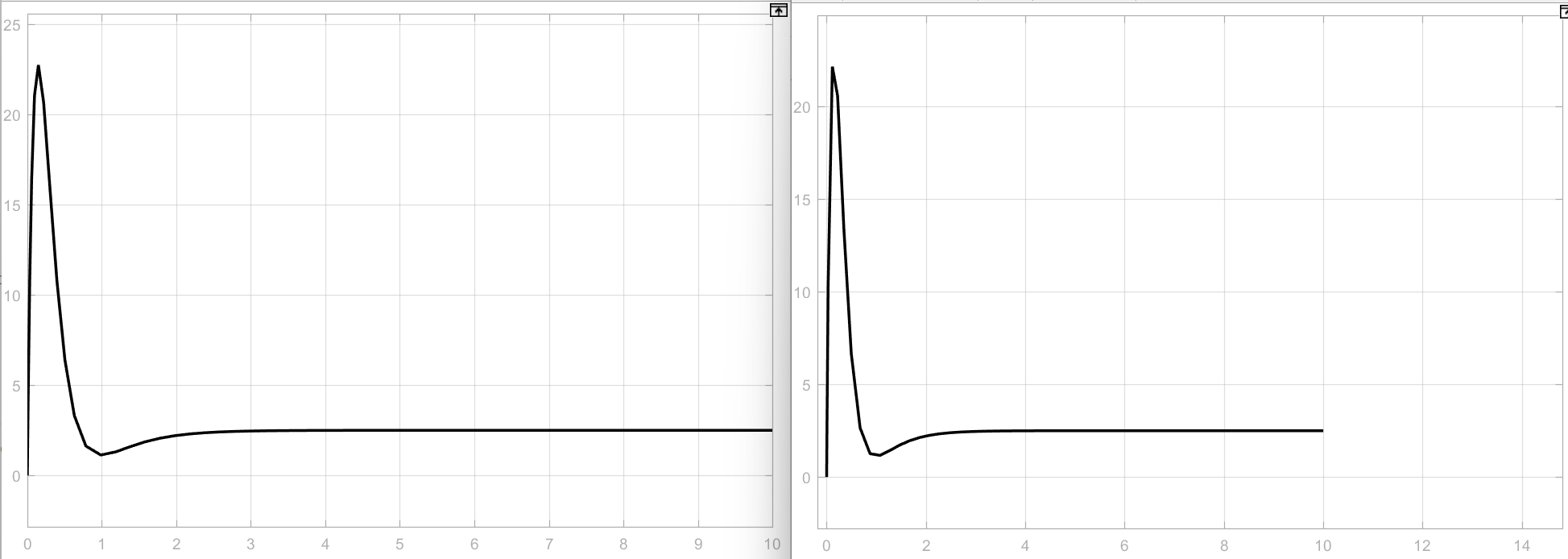
Далее составим матричную систему уравнений:

Переведем в линейный вид:

Далее построим схему в Simulink:

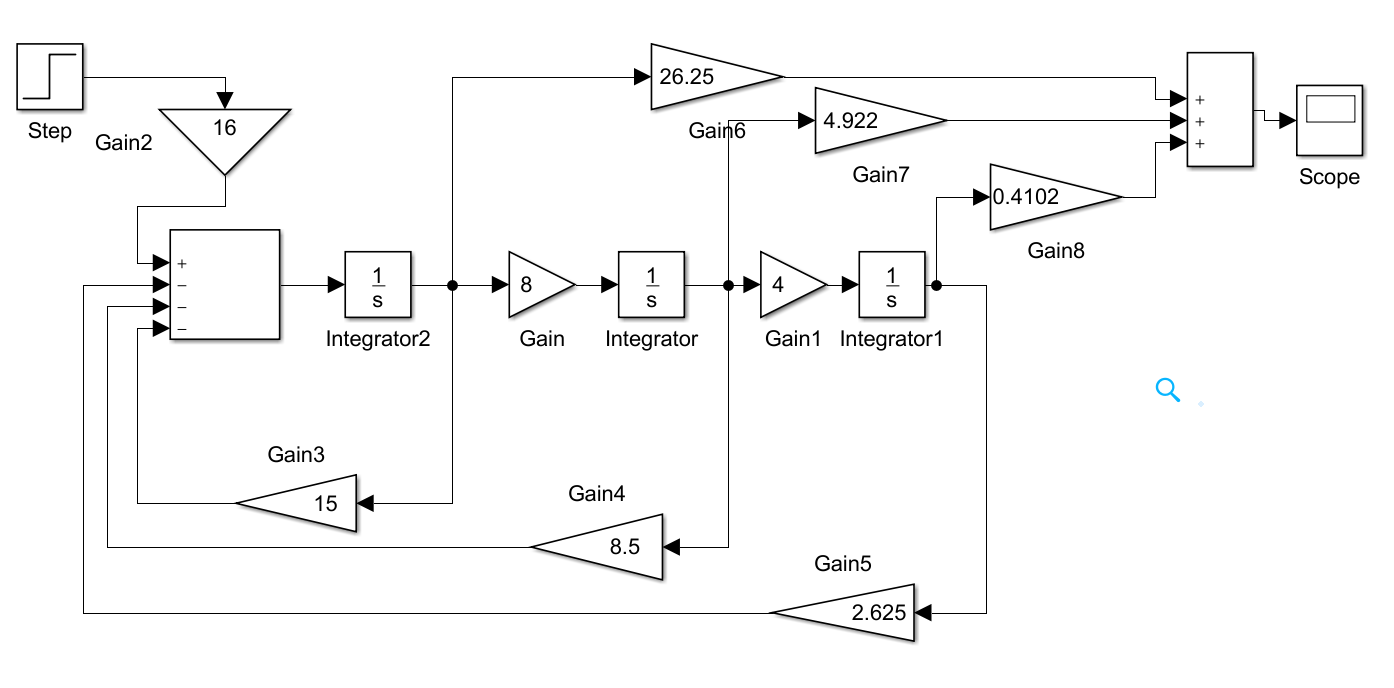


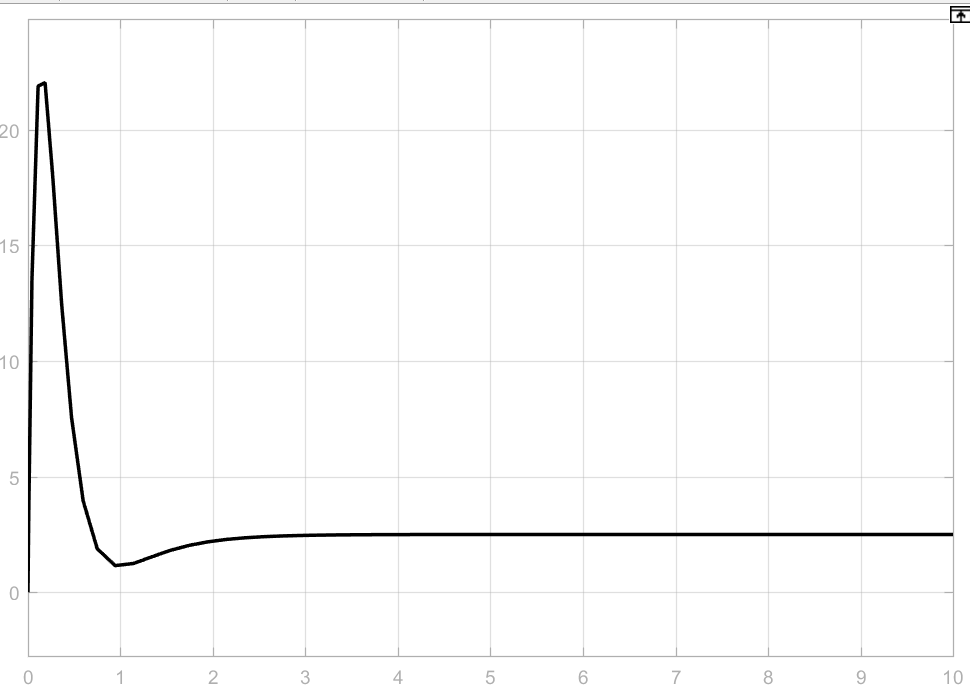
Запускаем симуляцию стандартной и каноничной схем и получаем одинаковые результаты.



Исходя из вида графика, можно сказать, что вид переходного процесса – апериодический.

Далее составим схему на основе ss-формы рассчитанной в MATLAB.





## ***Решение уравнения состояния в канонической форме***

Найдем решение y(t) для системы уравнений в канонической форме, при начальных условиях y(0) =1.5;. Сигнал на входе u(t)=15\*1(t).

Каждое из дифференциальных уравнений первого порядка зависит только от одной переменной, и его решение в общем виде имеет вид:

Определим начальные условия q(0) для вектора q(t).

Т.к. , то

Далее найдем выражения для , .

В результате получим:

Выполним проверку:

Из этого следует что уравнение верно.

# ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

## ***Расчет оптимального плана и экстремального значения функции цели***

Найти минимальное значение функции

При следующих ограничениях:

Домножим второе ограничение на (-1) и введем в ограничения дополнительные переменные *x4, x5*, *x6*, и искусственную переменную *R* следующим образом:

На основании данной системы ограничений составим первую симплекс таблицу (таблица 1):

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| БП | СВЧ | НП | | |  | БП | СВЧ | НП | | |
|  |  |  |  |  |  |  |
| R | 12 | 6 | 0 | -2 |  | R | 12 | 6 | 0 | -2 |
|  | 21 | 4 | -2 | 5 |  |  | 30 |  |  |  |
|  | -24 | -1 | -6 | -1 |  |  | 3 | 1 | -1 | 0 |
|  | -27 | -2 | -6 | -1 |  |  |  |  |  |  |
| F | 0 | -3 | 3 | 3 |  | F |  | -4 |  | 0 |
| M | -12 | -6 | 0 | 2 |  | M | -12 | -6 | 0 | 2 |

Таблица 1 Таблица 2

Поскольку в столбце свободных членов есть отрицательные элементы, то необходимо ее пересчитать, взяв ведущей строкой ту строку, в которой самое большое по модулю отрицательное число, т.е. 27. После этого находим самый большой отрицательный элемент в этом столбце, т.е. (-6). Далее составляем новую симплекс таблицу (таблица 2) на основании ведущего члена, взятого на пересечении строки и столбца.

Составление таблицы происходило по алгоритму:

1. Запишем ведущий член в обратном виде
2. Разделим ведущий столбец на отрицательный ведущий элемент
3. Разделим ведущую строку на ведущий элемент
4. Остальные значения рассчитаем по правилу прямоугольника

После данных действий получим отсутствие присутствия отрицательных членов в столбце свободных членов. Далее пересчитаем симплекс таблицу еще раз для отрицательного коэффициента в переменных M (-6). Определяем ведущую строку при помощи поиска минимального симплекс отношения. И после пересчета получим финальную симплекс таблицу:

|  |  |  |  |
| --- | --- | --- | --- |
| БП | СВЧ | НП | |
|  |  |
|  | 2 | 0 |  |
|  |  |  |  |
|  | 1 | -1 |  |
|  |  |  |  |
| F |  |  |  |
| M | 0 | 0 | 0 |

И получим ответ

Решим эту задачу используя команду linprog в пакете MATLAB:

>> A = [ 4 -2 5; -1 -6 -1; -2 -6 -1; -1 0 0; 0 -1 0; 0 0 -1];

>> B = [21; -24; -27; 0; 0; 0]

>> Aeq = [6 0 -2]

>> Beq = [12]

>> x = linprog(F, A, B, Aeq, Beq)

Получим ответ, который совпадает с нашим:

x =

2.0000

3.8333

0.0000

>> Fmin=F\*x

Fmin = 5.5000

## ***Исследование двойственной задачи линейного программирования***

Исходное условие:

Так как требуется найти минимум целевой функции, то неравенства в системе ограничений должны быть вида ≥. Второе и третье неравенство ограничений прямой задачи умножим на (-1):

После этого составим матрицы A, B, :

Двойственная задача будет иметь 4 переменные, так как прямая содержит 4 ограничения. Запишем двойственную задачу в виде:

Тогда условие пример следующий вид:

Составим симплекс-таблицу:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| БП | СВЧ | НП | | | |  | БП | СВЧ | НП | | | |
|  |  |  |  |  |  |  |  |  |
|  | -3 | 6 | -4 | 1 | 2 |  |  |  |  |  |  |  |
|  | 3 | 0 | 2 | 6 | 6 |  |  |  |  |  |  | 6 |
|  | 3 | -2 | -5 | 1 | 1 |  |  |  |  |  |  |  |
| F | 0 | -12 | 21 | -24 | -27 |  | F |  |  |  |  |  |

Таблица 1 Таблица 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| БП | СВЧ | НП | | | |
|  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| F |  |  |  |  |  |

Таблица 3

Получим решением таблицу 3. Запишем соответствие между переменными прямой и двойственной задач:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ |
|  |  |  |  |  |  |  |

min F(y) = max F(x) =

Таким образом, решая симплексным методом одну из пары двойственных задач, автоматически получаем решение другой.

Решим эту задачу через пакет MATLAB:

>> F = [-12 21 -24 -27]

>> A = [6 -4 1 2; 0 2 6 6; -2 -5 1 1; 0 -1 0 0; 0 0 -1 0; 0 0 0 -1]

>> B = [ -3;3;3;0;0;0]

>> x=linprog(F, A, B)

И получим ответ:

x = [-0.6667, -0.0000, 0.0000, 0.5000]

>> Q = F\*x

Q = -5.5000

## ***Нахождение целочисленного решения задачи***

Для решения целочисленной задачи необходимо ввести дополнительное ограничение по строке, где находится наибольшая дробная часть, в нашем случае, это строка с переменной , получившей нецелочисленное значение в оптимальном плане , составим дополнительное ограничение:

Приведем полученное неравенство в уравнение:

Введем данное ограничение в симплекс таблицу:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| БП | СВЧ | НП | |  | БП | СВЧ | НП | |
|  |  |  |  |  |
|  | 2 | 0 |  |  |  | 2 | 0 |  |
|  |  |  |  |  |  |  |  |  |
|  | 1 | -1 |  |  |  | 2 |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| F |  |  |  |  | F |  |  |  |

Решение получилось целочисленным, вторая итерация не нужна, ответ будет следующим:

# неЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

## ***Построение ОДЗП, выбор начальной точки поиска***

Целевая функция имеет вид:

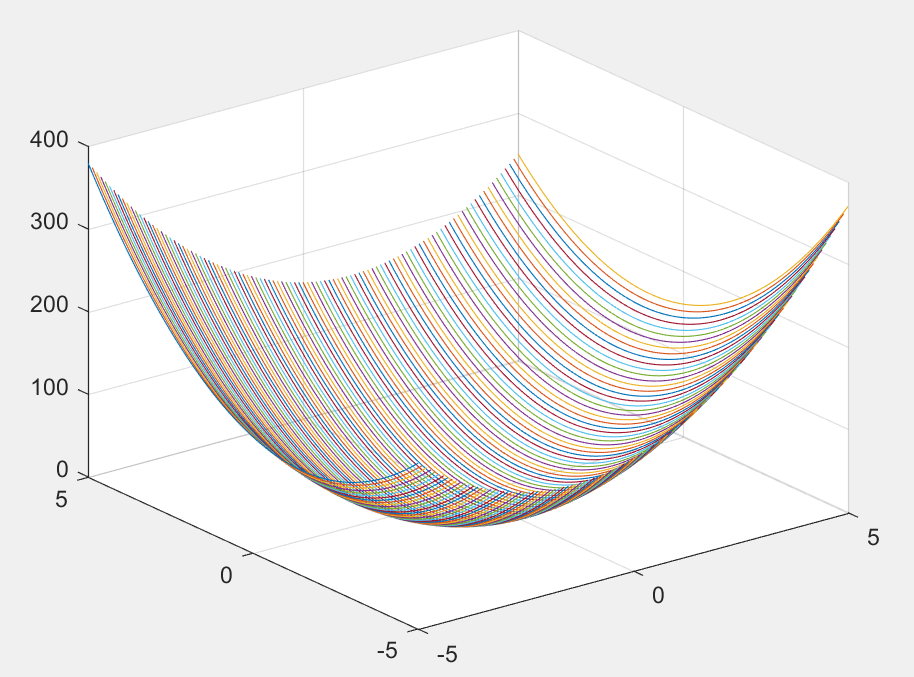
Построим график целевой функции:

>> [x1,x2]=meshgrid([-6:0.1:6]);

>> F=6\*x1.^2+6\*x2.^2-3\*x1.\*x2+2\*x1+3\*x2

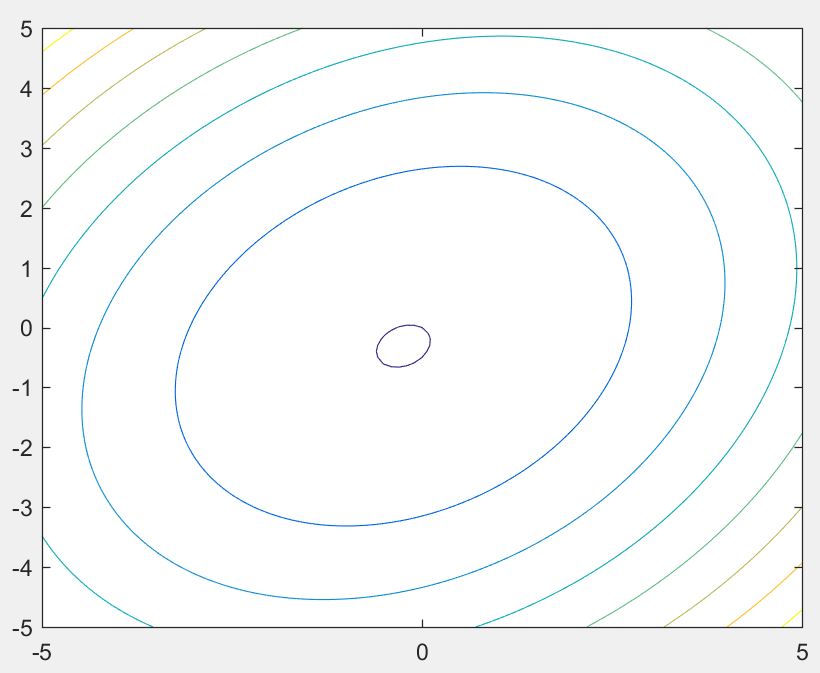
>> plot3(x1,x2,F)

>> grid on



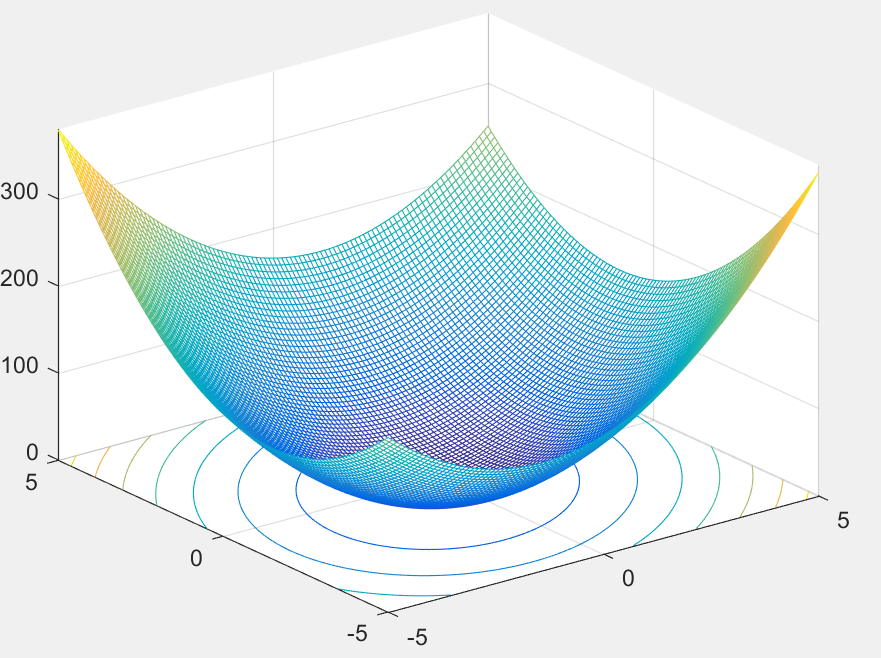
Построим концентрические окружности нашей целевой функции:

>>contour(x1,x2,F)



Совместим данные графики:

>> meshc(x1,x2,F)



Построим ОДЗП:   
>> [x1,x2]=meshgrid([-6:0.1:6]);

>> F=6\*x1.^2+6\*x2.^2-3\*x1.\*x2+2\*x1+3\*x2

>> hold on

>> ax = gca;

>> ax.XAxisLocation = 'origin';

>> ax.YAxisLocation = 'origin';

>> grid on  
>> [c,h] = contour(x1,x2,F)  
>> clabel(c,h)  
>> xx1=[0:0.1:6]

>> yy1=xx1/4

>> xx2=[0:0.1:6]

>> yy2=-xx2+5  
>> plot(xx1,yy1,'r')

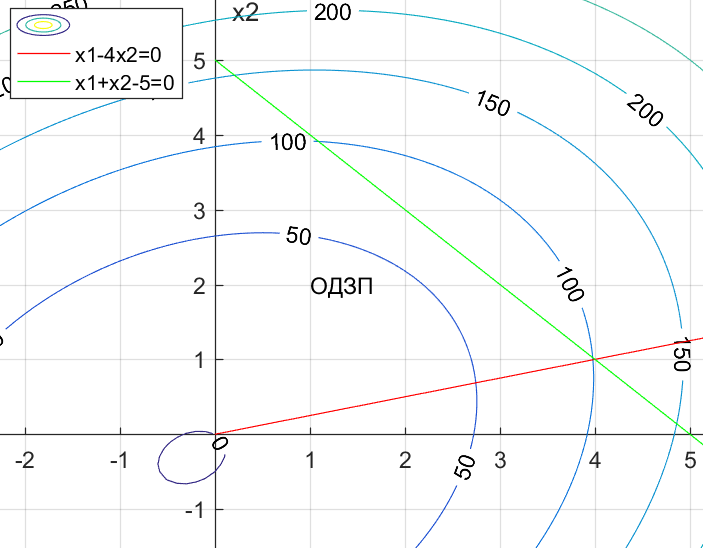
>> plot(xx2,yy2,'g')

>> xlabel('x1')

>> ylabel('x2')

>> legend({'','x1-4x2=0','x1+x2-5=0'},'Location', 'northwest')

>> text(1,2,'ОДЗП')



Выберем начальную точку поиска

## ***Нахождение экстремального значения функции F(x) без учета ограничений на переменные***

### *3.2.1 Метод наискорейшего спуска*

Прежде всего найдем составляющие градиента функции:

Градиент функции в точке будет

***1-й шаг.*** Движение осуществляется из точки вдоль вектора в новую точку :

Величину шага найдем из условия:

Отсюда

После подстановки получим:

Градиент функции в точке будет

***2-й шаг.*** Движение осуществляется из точки вдоль вектора в новую точку :

Величину шага найдем из условия:

Отсюда

После подстановки получим:

Градиент функции в точке будет

***3-й шаг*** Движение осуществляется из точки вдоль вектора в новую точку :

Величину шага найдем из условия:

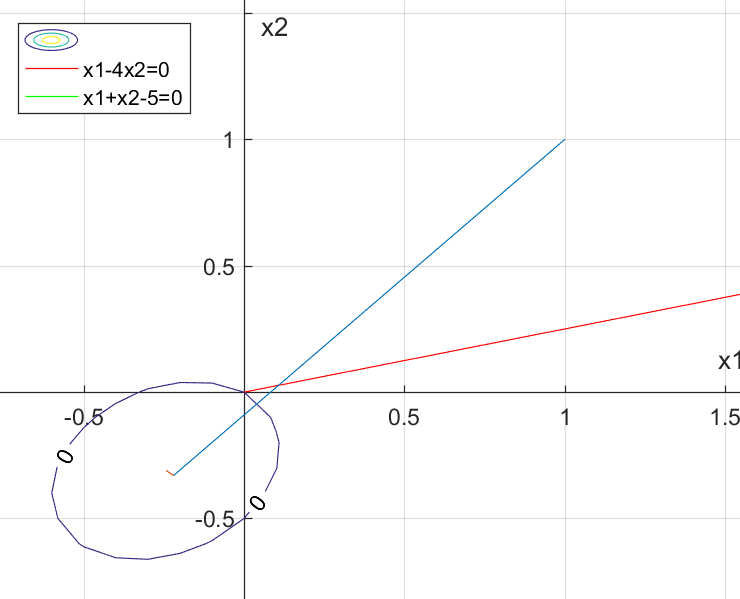
Отсюда

После подстановки получим:

Посчитаем норму вектора градиента :

Поскольку норма вектора крайне мала, а данный метод довольно плохо работает в окрестности точки максимума то примем точку минимума как:

А минимальное значение функции



### *3.2.2 Метод Ньютона-Рафсона*

Данный метод дает решение задачи за 1 шаг. Очередная точка поиска вычисляется в соответствии с выражением:

Где *H(x)* – матрица Гессе функции *F(x)*, – обратная по отношению к *H(x)* матрица.

Градиент *F(x)*:

Градиент функции в точке будет

Теперь подставим в изначальную формулу:

Следовательно, в точке функция достигает максимального значения .

### *3.2.3 Нахождение в среде MATLAB*

Найдем экстремум функции в среде MATLAB при помощи команды fminunc. Для этого пишем следующий код:

>> F=inline('6\*x(1)^2+6\*x(2)^2-3\*x(1)\*x(2)+2\*x(1)+3\*x(2)');

>> x0=[1 1]

>> options=optimset('LargeScale','off');

>> [x,fval,exitflag,output]=fminunc(F,x0,options);

После выполнения данных команд получим следующий выход:



И получаем, что решение в среде MATLAB полностью совпало с решением метода Ньютона-Рафсона и методом экстремального подъема.

## ***Нахождение экстремума функции F(x) с учетом системы ограничений***

### *Метод допустимых направлений Зойтендейка*

Условие задачи:

Градиент функции в точке будет

Движение осуществляется из точки вдоль вектора в новую точку :

Величину шага найдем из условия:

Отсюда

Определим интервал допустимых при котором точка будет находится в ОДЗП, для этого в условие задачи запишем дополнительные ограничения:

Тогда:

Находим величину , которая обеспечит экстремум функции F(x). Воспользуемся уже найденным значением: . Оно не попадает в промежуток. И берем левое значение промежутка т.е. . Значение градиента функции в этой точке:

Градиент функции в точке будет .

Движение выводит за пределы ОДЗП, поэтому очередную точку поиска вычисляет исходя из выражения:

Где новое направление, которое составляет минимальный угол с вектором градиента и направлено либо внутрь, либо по границе ОДЗП. При этом очередная точка должна принадлежать ОДЗП, а функция цели при переходе к очередной точке должна уменьшится максимальным образом.

Направление находим как решение задачи:



Найдем направление очередного шага

Отсюда следует, что

При движении из точки в точку следует двигаться по граничной прямой в направлении

Далее рассчитаем координаты точки :

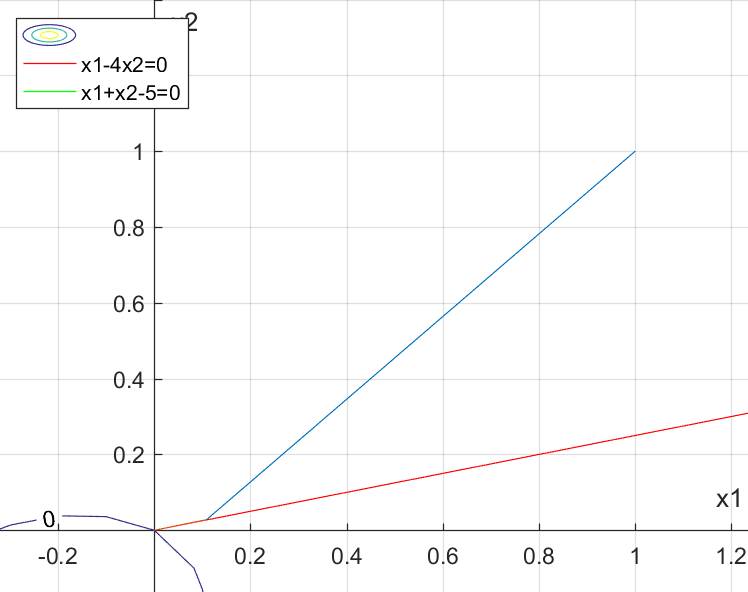
Найдем интервал значения

Тогда

Находим , которое обеспечит максимум функции F(x) в направлении

Значение вне интервала поэтому берем и получаем

Получим точку экстремума



### *Метод линейных комбинаций*

Условие задачи:

Градиент функции в точке будет

Осуществим линеаризацию F(x) относительно точки выражением:

Решается задача ЛП:

|  |  |  |  |
| --- | --- | --- | --- |
| БП | СВЧ | НП | |
|  |  |
|  | 0 | 1 | -4 |
|  | 5 | 1 | 1 |
|  | 0 | 11 | 12 |

Таблица оптимальна. Делаем корректировку найденного решения в соответствии с выражением:

Определим интервал допустимых значений для при котором точка будет принадлежать ОДЗП.

Поскольку то берем правую границу

Поскольку это точка касания одной из линий уровня со стороной ОДЗП, то это точка, при которой достигается минимум функции при наших ограничениях.

### *Теорема Куна-Таккера*

Условие задачи:

Составим функцию Лагранжа:

Точка экстремума является седловой точкой с минимумом по x и максимумом по λ, поэтому ограничения приведены к виду :

Условия теоремы Куна-Таккера записываем следующим образом:



Частные производные функции Лагранжа определяются выражениями:

Для того, чтобы вышеуказанные выражения имели вид равенств, введем в них дополнительные переменные:

Решение этой системы можно найти с помощью симплекс процедуры

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| БП | СВЧ | НП | | | |
|  |  |  |  |
|  | 2 | -12 | 3 | -1 | -1 |
|  | 3 | 3 | -12 | 4 | -1 |
|  | 0 | 1 | -4 | 0 | 0 |
|  | 5 | 1 | 1 | 0 | 0 |

Оптимальное решение найдено, точка минимума является , что совпадает со всеми предыдущими методами.

### *Квадратичное программирование в MATLAB*

>> [x1,x2]=meshgrid([-6:0.1:6]);

>> F=6\*x1.^2+6\*x2.^2-3\*x1.\*x2+2\*x1+3\*x2

>> A=[1 -4;1 1];

>> B=[0 5];

>> x0=[1 1];

>> H=[12 -3;-3 12];

>> f=[2 3];

>> vlb=zeros(2,1);

>> vub=[];

>> x=quadprog(H,f,A,B,[],[],vlb,vub);

>> x

x =

1.0e-07 \*

0.0000

0.1066

>> z

z =

3.1981e-08

# Заключение

В первой части курсовой работы выполнен анализ линейной системы 3-го порядка, заданной в виде передаточной функции. Получены выражения для построения временных характеристик системы. По заданной передаточной функции были построены логарифмические амплитудно-частотная и фазочастотная характеристики. Правильность результатов построения подтверждена моделированием в пакете Matlab/Simulink.

Также на основании заданной передаточной функции были составлены уравнения состояния в нормальной и канонической формах. Получены схемы моделей системы и проведено моделирование в пакете Matlab/Simulink.

Во второй части курсовой работы решена прямая задача линейного программирования с применением симплекс-таблиц, составлена и решена двойственная задача к прямой. Решение прямой задачи и полученное решение при приведении в соответствие переменных двойственной и прямой задачи совпадает. Также решена частично-целочисленная задача.

В третьей части курсовой работы решены задачи нелинейного программирования без ограничений и с ограничениями. В решении задачи без ограничений показано, что методом Ньютона-Рафсона задача решается за один шаг, а метод наискорейшего спуска медленно сходится к решению. В задаче нелинейного программирования с ограничениями показано, что все методы решения задач одинаково сходятся к одному решению, но за разное количество шагов. Приведены графики интерпретации метода наискорейшего спуска, метода допустимых направлений Зойтендейка и метода линейных комбинаций.

# Список использованных источников

[1] Павлова, А. В. Электронный учебно-методический комплекс по учебной дисциплине «Математические основы теории систем» для студентов специальности 1-53 01 07 Информационные технологии и управление в технических системах [Электронный ресурс] / А. В. Павлова, М. К. Хаджинов. – Режим доступа: EUMK\_MOTS\_2013.zip.

[2] Павлова, А. В. Математические основы теории систем : конспект лекций для студентов специальности «Информационные технологии и управление в технических системах». В 2 ч. / А. В. Павлова. – Минск : БГУИР, 2010. – Ч. 2. – 144 с.

[3] Певзнер, Л. Д. Математические основы теории систем / Л. Д. Певзнер, Е. П. Чураков – М. : Высш. шк., 2009.