

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Национальный исследовательский университет «МЭИ»  
Институт ИВТИ Кафедра Управления и информационных технологий

**Отчет по учебной практике**  
студента группы А-01-19 курса 1  
*Черемных Юрия Алексеевича*

**Почта:** CheremnykhYurA@mpei.ru

**Вариант:** 19

**Проверил:** Полотнов М.М.

2020

## **Оглавление**

<b>1) Назначение программы .....</b>	<b>3</b>
<b>2) Структура программы и руководство пользователя .....</b>	<b>3</b>
<b>«Как программа работает?» .....</b>	<b>3</b>
<b>Структура программы со схемой. ....</b>	<b>4</b>
<b>3) Описание процедур и модулей .....</b>	<b>5</b>
<b>4) Пример работы контрольной задачи .....</b>	<b>6</b>
<b>5) Цели дополнительного исследования и методика выполнения .....</b>	<b>6</b>
<b>6) Результаты применения программы в процессе исследования .....</b>	<b>7</b>
<b>7) Выводы по проделанному исследованию .....</b>	<b>9</b>

## **1) Назначение программы**

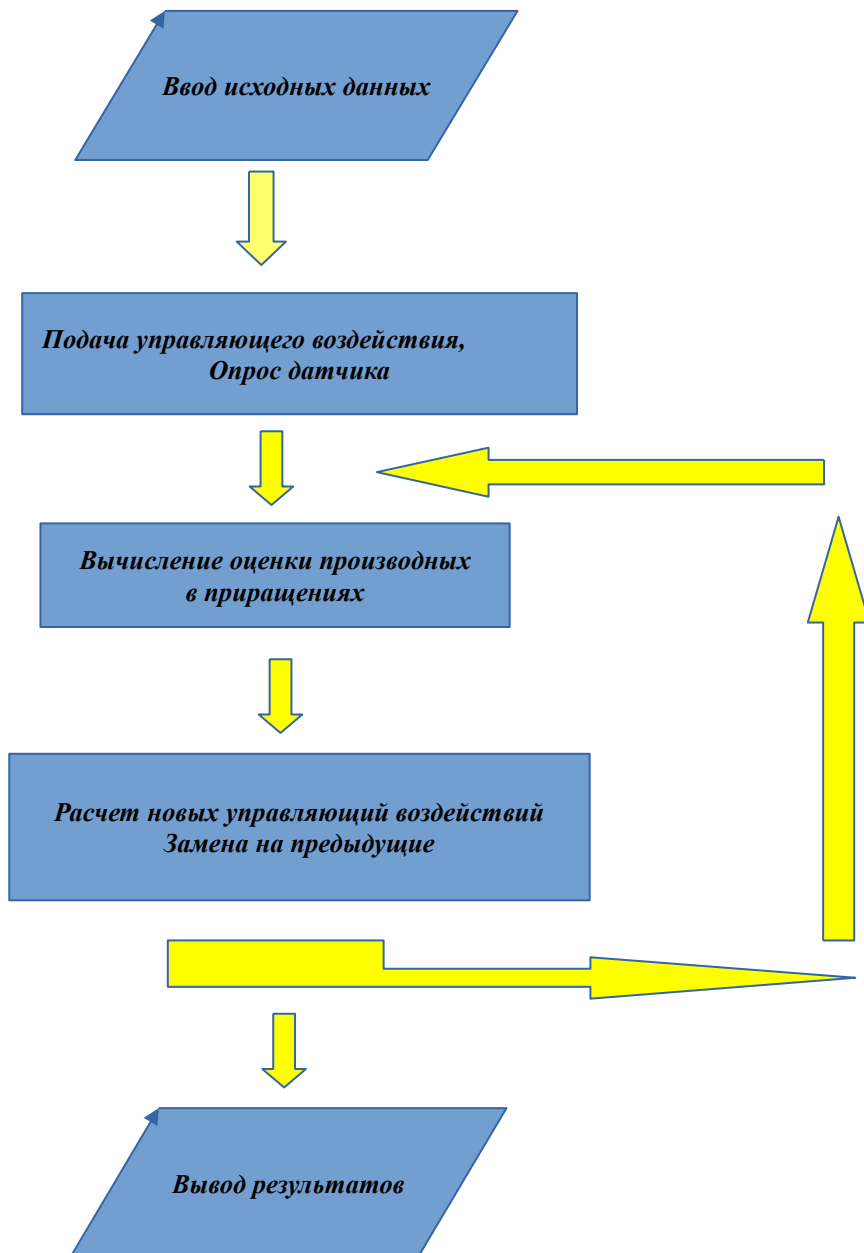
Разработанная программа должна осуществлять опрос датчиков и оптимизировать объект градиентным методом

## **2) Структура программы и руководство пользователя «Как программа работает?»**

1. Ввод пользователем исходных данных, таких как количество шагов поиска , номер датчика, количество каналов управления, начальные значения каналов управления, величина шага поиска.
2. Программа подает управляющее воздействие на объект, используя введенные данные
3. Опрашивает датчик
4. При помощи градиентного метода оптимизации вычисляет оценку производных в приращениях
5. Вычисляет оценку коррекции и заменяет старые управляющие значения воздействий на новые
6. Повторяет N раз
7. Выводит таблицу со всеми промежуточными данными

## Структура программы со схемой.

Программа состоит из основной функции *main*, процедуры ввода данных, расчета оценки производной в приращениях, расчета новых управляющих воздействий (более подробно о каждой функции на стр 5 )



### 3) Описание процедур и модулей

#### 1. Процедура ввода исходных данных

Процедура обеспечивает ввод номера датчика (  $M$  ), число каналов управления (  $K$  ), массив из номеров каналов управления (  $L$  ), шага поиска (  $D$  ), массив из начальных управляющих воздействий (  $U$  ).

```
void input(int &M, int &K, int L[ARR], double &D, double U[])
```

#### 2. Функция, возвращающая оценку производной в приращениях

На вход подаются число каналов управления (  $L$  ), управляющие воздействия, номер текущего канала управления (так как в задании сказано использовать процедуру для расчета одной оценки производной), шаг поиска, номер датчика, измеренные значения с датчика (  $Y$  ), полученное значение с датчика после расчетов (  $Y1$  ) (для вывода промежуточных значений, модель объекта (  $plant$  )

```
double gradient(int L[], double U[], int i, double D, int M, double Y, double &Y1, Plant plant)
```

#### 3. Функция считает $(P_1^2 + P_2^2 + \dots + P_K^2)^{1/2}$

```
double sqrt_sum_squares(double P[], int K)
```

#### 4. Процедура, которая вычисляет новые значения управляющих воздействий

$K$  — количество каналов управления

$Q$  — оценка коррекции, вычисляемая по формуле  $Q = D / (P_1^2 + P_2^2 + \dots + P_K^2)^{1/2}$

$P$  — массив оценок производных, вычисляемый в функции 2

$U$  — начальные управляющие воздействия

```
void new_U(double U2[], int K, double Q, double P[], double U[])
```

## 4) Пример работы контрольной задачи

```
Enter the number of search steps : 10
Enter number of sensor : 35
Enter the number of control channels : 2

*****
NUMBERS OF CONTROL CHANNELS
Enter the number of the 1st control channel : 7
Enter the number of the 2st control channel : 8

*****
INITIALS VALUES
Enter the initial value of the 7st control channel : 23
Enter the initial value of the 8st control channel : 1

*****
Enter search step value : 0.8

*****
RESULT OUTPUTS
N  U1      U2      Y      Y1_1    Y1_2    P1      P2      U'1     U'2
1  23      1      368.555  373.792  458.238  6.54659  112.104  23.0466  1.79864
2  23.0466  1.79864  437.736  456.031  555.787  22.8692  147.564  23.1692  2.5892
3  23.1692  2.5892   564.818  544.426  681.328  -25.4904  145.637  23.0312  3.37722
4  23.0312  3.37722  680.44   672.076  840.872  -10.4546  200.541  22.9896  4.17614
5  22.9896  4.17614  832.659  815.698  1001.64  -21.2018  211.23   22.9097  4.97214
6  22.9097  4.97214  1002.64  1005.75  1007.91  3.88882   6.58519  23.3165  5.66099
7  23.3165  5.66099  1018.02  1011.31  1005.97  -8.38683  -15.0521  22.9271  4.96215
8  22.9271  4.96215  998.155  1007.69  1001.2   11.9222   3.80733  23.6892  5.20552
9  23.6892  5.20552  1011.56  1012.16  1005     0.754334  -8.20058  23.7625  4.40888
10 23.7625  4.40888  865.998  881.662  1015.6   19.5796   187.003  23.8458  5.20453

Q = 0.00425475
Y = 865.998
M = 35
D = 0.8
```

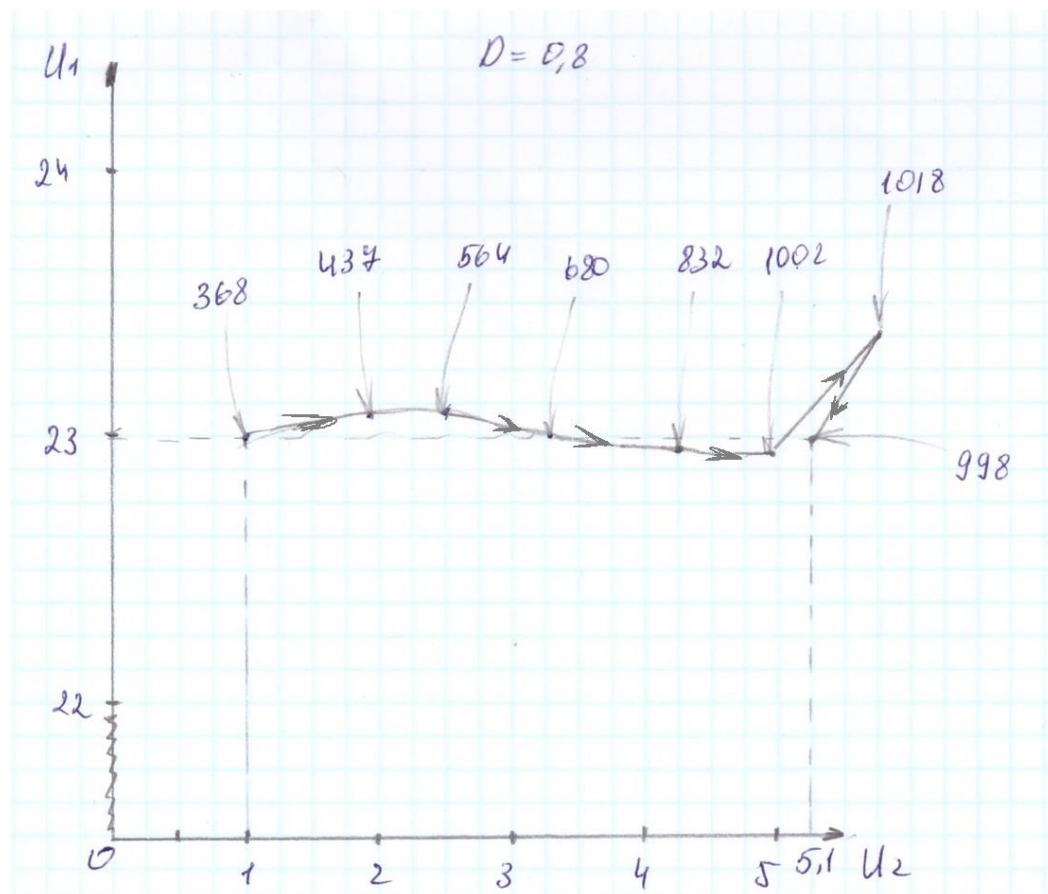
## 5) Цели дополнительного исследования и методика выполнения

Целью исследования является изучение «траектории движения» значений управления при поиске оптимальных значений и ее зависимости от шага поиска.

Для выполнения исследования приведем несколько примеров работы программы при различных значениях шага поиска и построим графики

## 6) Результаты применения программы в процессе исследования

1. Возьмем пример из контрольного задания (пункт 4 D = 0.8 )



Как мы видим траектория движения плавная и точная, но для достижения оптимального значения понадобилось 7 — 9 шагов (циклов)

## 2. Пример при $D = 1.2$

```

Enter the number of the 1st control channel : 7
Enter the number of the 2st control channel : 8

*****
INITIALS VALUES
Enter the initial value of the 7st control channel : 0
Enter the initial value of the 8st control channel : 1

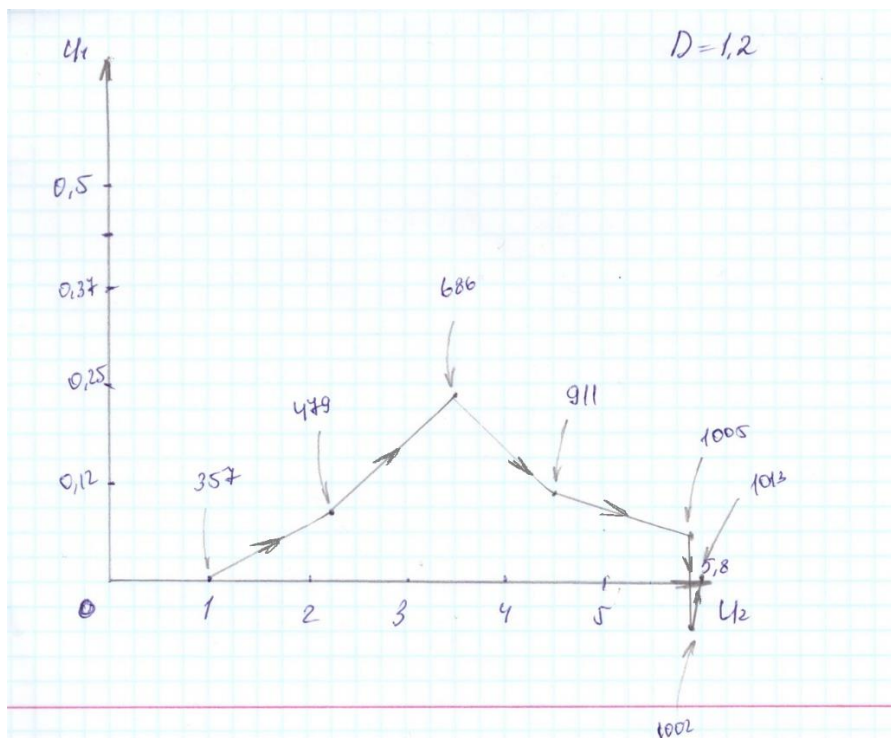
*****
Enter search step value : 1.2

*****
RESULT OUTPUTS
N   U1      U2      Y      Y1_1    Y1_2    P1      P2      U'1      U'2
1   0       1       357.515  367.168  503.153  8.04439  121.365  0.0793649  2.19737
2   0.0793649  2.19737  479.035  500.999  680.013  18.3035  167.482  0.209733  3.39027
3   0.209733  3.39027  686.576  669.108  912.992  -14.5565  188.68  0.117428  4.58671
4   0.117428  4.58671  911.074  904.888  1016     -5.15485  87.4399  0.046807  5.78463
5   0.046807  5.78463  1005.97  990.933  1004.33  -12.5345  -1.37318  -1.14606  5.65395
6   -1.14606  5.65395  1002.41  1007.45  1003.2   4.19254  0.65094  0.0397365  5.83806
7   0.0397365  5.83806  1013.28  1008.49  1003.16  -3.99122  -8.43474  -0.473528  4.75337
8   -0.473528  4.75337  945.074  956.686  997.563  9.67638  43.741  -0.214331  5.92504
9   -0.214331  5.92504  1006.41  1008.94  1001.77  2.10289  -3.86706  0.358944  4.87083
10  0.358944  4.87083  962.257  979.921  1013.29  14.7202  42.5312  0.751427  6.00483

Q = 0.0266628
Y = 962.257
M = 35
D = 1.2

Process returned 0 (0x0)   execution time : 25.878 s
Press ENTER to continue.

```



В этом примере траектория более ломаная, но результат был достигнут с 5 – 6 циклом



## 7) Выводы по проделанному исследованию

Проанализировав результаты исследования и ознакомившись с графиками можно сделать вывод: чем ниже значение шага поиска, тем точнее проводится поиск, но циклов (шагов  $N$ ) требуется больше, следовательно и вычислительной мощности также требуется больше. А при большем значении шага поиска исследование проводится быстрее, но с меньшей точностью.

*С исходным кодом проекта можно ознакомиться по ссылке*

*<https://github.com/yurchest/practice>*