

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”  
КАФЕДРА ИИТ

ОТЧЁТ  
по ССОИ  
по лабораторной работе №3

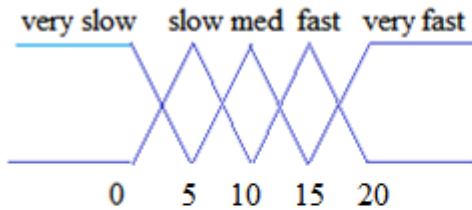
Выполнил:  
студентка группы АС-32  
Предко А.И.

Проверил:  
Акира Имада

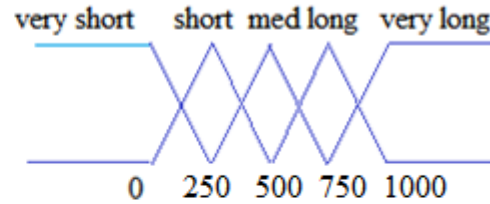
Брест 2014

### Лабораторная работа №3

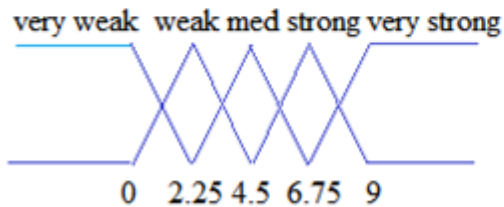
Speed



Distance



Brake



Rules:

IF speed is fast AND distance is short THEN brake is strong;

IF speed is slow AND distance is long THEN brake is weak;

IF speed is medium AND distance is medium THEN brake is medium;

IF speed is very fast AND distance is very short THEN brake is very strong;

IF speed is very slow AND distance is very long THEN brake is very weak;

IF speed is very slow AND distance is medium THEN brake is weak;

IF speed is fast AND distance is medium THEN brake is medium;

IF speed is slow AND distance is medium THEN brake is weak;

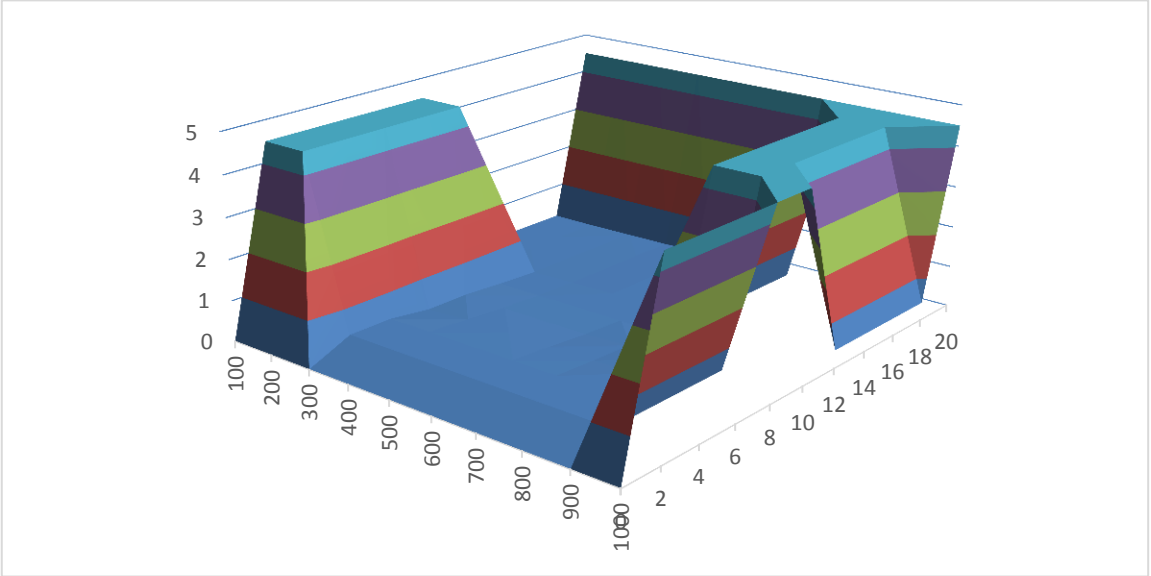
IF speed is medium AND distance is very long THEN brake is weak;

IF speed is medium AND distance is very short THEN brake strong;

Small table

| Speed | Distance | Brake | Mu    |
|-------|----------|-------|-------|
| 10    | 500      | 0     | 0     |
| 10    | 500      | 1     | 0     |
| 10    | 500      | 2     | 0     |
| 10    | 500      | 3     | 0.333 |
| 10    | 500      | 4     | 0.777 |
| 10    | 500      | 5     | 0.779 |
| 10    | 500      | 6     | 0.335 |
| 10    | 500      | 7     | 0     |
| 10    | 500      | 8     | 0     |
| 10    | 500      | 9     | 0     |

Balance Graph



| Speed | Distance | Balance  |
|-------|----------|----------|
| 2     | 100      | 4.5      |
| 2     | 200      | 4.5      |
| 2     | 300      | 0.434375 |
| 2     | 400      | 0.434375 |
| 2     | 500      | 0.434375 |
| 2     | 600      | 0.434375 |
| 2     | 700      | 0.434375 |
| 2     | 800      | 0.434375 |
| 2     | 900      | 0.434375 |
| 2     | 1000     | 4.5      |
| 4     | 100      | 4.5      |
| 4     | 200      | 4.5      |
| 4     | 300      | 0.434375 |
| 4     | 400      | 0.434375 |

|    |      |          |
|----|------|----------|
| 4  | 500  | 0.434375 |
| 4  | 600  | 0.434375 |
| 4  | 700  | 0.434375 |
| 4  | 800  | 0.434375 |
| 4  | 900  | 0.434375 |
| 4  | 1000 | 4.5      |
| 6  | 100  | 4.5      |
| 6  | 200  | 4.5      |
| 6  | 300  | 0.293949 |
| 6  | 400  | 0.363892 |
| 6  | 500  | 0.379382 |
| 6  | 600  | 0.363892 |
| 6  | 700  | 0.379382 |
| 6  | 800  | 0.434375 |
| 6  | 900  | 0.434375 |
| 6  | 1000 | 4.5      |
| 8  | 100  | 4.5      |
| 8  | 200  | 4.5      |
| 8  | 300  | 0.293949 |
| 8  | 400  | 0.272579 |
| 8  | 500  | 0.272579 |
| 8  | 600  | 0.272579 |
| 8  | 700  | 0.33738  |
| 8  | 800  | 0.434375 |
| 8  | 900  | 0.434375 |
| 8  | 1000 | 4.5      |
| 10 | 100  | 4.5      |
| 10 | 200  | 4.5      |
| 10 | 300  | 0.222133 |
| 10 | 400  | 0.222133 |

|    |      |          |
|----|------|----------|
| 10 | 500  | 0.222133 |
| 10 | 600  | 0.222133 |
| 10 | 700  | 0.222133 |
| 10 | 800  | 4.5      |
| 10 | 900  | 4.5      |
| 10 | 1000 | 4.5      |
| 12 | 100  | 0.149176 |
| 12 | 200  | 0.149176 |
| 12 | 300  | 0.165525 |
| 12 | 400  | 0.187391 |
| 12 | 500  | 0.222133 |
| 12 | 600  | 0.222133 |
| 12 | 700  | 0.222133 |
| 12 | 800  | 4.5      |
| 12 | 900  | 4.5      |
| 12 | 1000 | 0.434375 |
| 14 | 100  | 0.149176 |
| 14 | 200  | 0.149176 |
| 14 | 300  | 0.156972 |
| 14 | 400  | 0.187391 |
| 14 | 500  | 0.222133 |
| 14 | 600  | 0.222133 |
| 14 | 700  | 0.222133 |
| 14 | 800  | 4.5      |
| 14 | 900  | 4.5      |
| 14 | 1000 | 0.434375 |
| 16 | 100  | 0.149176 |
| 16 | 200  | 0.149176 |
| 16 | 300  | 0.156972 |
| 16 | 400  | 0.187391 |

|    |      |          |
|----|------|----------|
| 16 | 500  | 0.222133 |
| 16 | 600  | 0.222133 |
| 16 | 700  | 0.222133 |
| 16 | 800  | 4.5      |
| 16 | 900  | 4.5      |
| 16 | 1000 | 0.434375 |
| 18 | 100  | 0.149176 |
| 18 | 200  | 0.149176 |
| 18 | 300  | 0.165525 |
| 18 | 400  | 0.178463 |
| 18 | 500  | 0.222133 |
| 18 | 600  | 0.222133 |
| 18 | 700  | 0.222133 |
| 18 | 800  | 4.5      |
| 18 | 900  | 4.5      |
| 18 | 1000 | 0.434375 |
| 20 | 100  | 4.5      |
| 20 | 200  | 4.5      |
| 20 | 300  | 4.5      |
| 20 | 400  | 4.5      |
| 20 | 500  | 4.5      |
| 20 | 600  | 4.5      |
| 20 | 700  | 4.5      |
| 20 | 800  | 4.5      |
| 20 | 900  | 4.5      |
| 20 | 1000 | 4.5      |

## Program

```
#include <QCoreApplication>
#include <iostream>
#include <fstream>
#include <math.h>
#include <stdio.h>

using namespace std;

ofstream File1("C:\\small1.txt");
ofstream File2("C:\\small2.txt");
ofstream File3("C:\\small3.txt");
ofstream File4("C:\\small4.txt");
ofstream File5("C:\\small5.txt");
ofstream File6("C:\\small6.txt");
ofstream File7("C:\\small7.txt");
ofstream File8("C:\\small8.txt");
ofstream File9("C:\\small9.txt");
ofstream File10("C:\\small10.txt");

double Rule1(int i, int j, int k)
{
    double mu;
    double muS = 0;
    double muD = 0;
    double muB = 0;

    muS = 0.2 * (i-10);
    if (i>15 && i<=20)
    {
        muS = 1- fabs(1-muS);
    }
    if( i<10 || i>20)
    {
        muS = 0;
    }
    muD = 0.004 * (j-0);
    if (j>250 && j<=500)
    {
```



```

        muD = 1- fabs(1-muD);
    }
    if( j<0 || j>500)
    {
        muD = 0;
    }

    muB = 0.444 * (k-4.5);
    if (k>6.75 && k<=9)
    {
        muB = 1- fabs(1-muB);
    }
    if( k<4.5 || k>9)
    {
        muB = 0;
    }
    double minMu = min(muS, muD);
    mu = minMu * muB;
    File1<<i<<'\\t'<<muS<<'\\t'<<j<<'\\t'<<muD<<'\\t'<<k<<'\\t'<<muB<<'\\t'<<mu<<endl;
    return mu;
}

```

```

double Rule2(int i, int j, int k)
{
    double mu;
    double muS =0;
    double muD = 0;
    double muB = 0;

    muS = 0.2 * (i-0);
    if (i>5 && i<=10)
    {
        muS = 1- fabs(1-muS);
    }
    if( i<0 || i>10)
    {
        muS = 0;
    }
    muD = 0.004 * (j-500);
    if (j>750 && j<=1000)
    {

```

```

        muD = 1- fabs(1-muD);
    }
    if( j<500 || j>1000)
    {
        muD = 0;
    }

    muB = 0.444 * (k-0);
    if (k>2.25 && k<=4.5)
    {
        muB = 1- fabs(1-muB);
    }
    if( k<0 || k>4.5)
    {
        muB = 0;
    }
    double minMu = min(muS, muD);
    mu = minMu * muB;
    File2<<i<<'\\t'<<muS<<'\\t'<<j<<'\\t'<<muD<<'\\t'<<k<<'\\t'<<muB<<'\\t'<<mu<<endl;
    return mu;
}

```

```

double Rule3(int i, int j, int k)
{
    double mu;
    double muS =0;
    double muD = 0;
    double muB = 0;

    muS = 0.2 * (i-5);
    if (i>10 && i<=15)
    {
        muS = 1- fabs(1-muS);
    }
    if( i<5 || i>15)
    {
        muS = 0;
    }
    muD = 0.004 * (j-250);
    if (j>500 && j<=750)
    {

```

```

        muD = 1- fabs(1-muD);
    }
    if( j<250 || j>750)
    {
        muD = 0;
    }

    muB = 0.444 * (k-2.25);
    if (k>4.5 && k<=6.75)
    {
        muB = 1- fabs(1-muB);
    }
    if( k<2.25 || k>6.75)
    {
        muB = 0;
    }
    double minMu = min(muS, muD);
    mu = minMu * muB;
    File3<<i<<'\\t'<<muS<<'\\t'<<j<<'\\t'<<muD<<'\\t'<<k<<'\\t'<<muB<<'\\t'<<mu<<endl;
    return mu;
}

```

```

double Rule4(int i, int j, int k)
{
    double mu;
    double muS =0;
    double muD = 0;
    double muB = 0;

    muS = 0.2 * (i-15);
    if(i < 15)
    {
        muS = 0;
    }
    if(i > 20)
    {
        muS = 1;
    }
    if (j < 0)
    {
        muD = 1;
    }
}

```

```

    }
    if ( j>= 0 && j <= 250)
    {
        muD = (250 - j)/(250);
        //muD = 1- fabs(1-muD);
    }

    if(j>250)
    {
        muD = 0;
    }

    muB = 0.444 * (k-0);
    if (k>2.25 && k<=4.5)
    {
        muB = 1- fabs(1-muB);
    }
    if( k<0 || k>4.5)
    {
        muB = 0;
    }
    double minMu = min(muS, muD);
    mu = minMu * muB;
    File4<<i<<'\\t'<<muS<<'\\t'<<j<<'\\t'<<muD<<'\\t'<<k<<'\\t'<<muB<<'\\t'<<mu<<endl;
    return mu;
}

double Rule5(int i, int j, int k)
{
    double mu;
    double muS =0;
    double muD = 0;
    double muB = 0;

    muS = (5-i)/5;
    if(i > 5)
    {
        muS = 0;
    }
    if(i < 0)
    {

```

```

        muS = 1;
    }

    muD = (j - 750) / (1000 - 750);
    if (j < 750)
    {
        muD = 0;
    }

    if (j > 1000)
    {
        muD = 1;
    }

    muB = (2.25 - k) / 2.25;
    if (k > 2.25)
    {
        muB = 0;
    }
    if (k < 0)
    {
        muB = 1;
    }
    double minMu = min(muS, muD);
    mu = minMu * muB;
    File5<<i<<'\\t'<<muS<<'\\t'<<j<<'\\t'<<muD<<'\\t'<<k<<'\\t'<<muB<<'\\t'<<mu<<endl;
    return mu;
}

double Rule6(int i, int j, int k)
{
    double mu;
    double muS = 0;
    double muD = 0;
    double muB = 0;

    muS = (5 - i) / 5;
    if (i > 5)
    {
        muS = 0;
    }

```

```

    }
    if(i < 0)
    {
        muS = 1;
    }

    muD = 0.004 * (j-250);
    if (j>500 && j<=750)
    {
        muD = 1- fabs(1-muD);
    }
    if( j<250 || j>750)
    {
        muD = 0;
    }

    muB = 0.444 * (k-0);
    if (k>2.25 && k<=4.5)
    {
        muB = 1- fabs(1-muB);
    }
    if( k<0 || k>4.5)
    {
        muB = 0;
    }
    double minMu = min(muS, muD);
    mu = minMu * muB;
    File6<<i<<'\\t'<<muS<<'\\t'<<j<<'\\t'<<muD<<'\\t'<<k<<'\\t'<<muB<<'\\t'<<mu<<endl;
    return mu;
}

double Rule7(int i, int j, int k)
{
    double mu;
    double muS =0;
    double muD = 0;
    double muB = 0;

    muS = 0.2 * (i-10);
    if (i>15 && i<=20)
    {

```

```

    muS = 1- fabs(1-muS);
}
if( i<10 || i>20)
{
    muS = 0;
}
muD = 0.004 * (j-250);
if (j>500 && j<=750)
{
    muD = 1- fabs(1-muD);
}
if( j<250 || j>750)
{
    muD = 0;
}

muB = 0.444 * (k-2.25);
if (k>4.5 && k<=6.75)
{
    muB = 1- fabs(1-muB);
}
if( k<2.25 || k>6.75)
{
    muB = 0;
}
double minMu = min(muS, muD);
mu = minMu * muB;
File7<<i<<'\\t'<<muS<<'\\t'<<j<<'\\t'<<muD<<'\\t'<<k<<'\\t'<<muB<<'\\t'<<mu<<endl;
return mu;
}

double Rule8(int i, int j, int k)
{
    double mu;
    double muS =0;
    double muD = 0;
    double muB = 0;

    muS = 0.2 * (i-0);
    if (i>5 && i<=10)
    {

```

```

        muS = 1- fabs(1-muS);
    }
    if( i<0 || i>10)
    {
        muS = 0;
    }

    muD = 0.004 * (j-250);
    if (j>500 && j<=750)
    {
        muD = 1- fabs(1-muD);
    }
    if( j<250 || j>750)
    {
        muD = 0;
    }

    muB = 0.444 * (k-0);
    if (k>2.25 && k<=4.5)
    {
        muB = 1- fabs(1-muB);
    }
    if( k<0 || k>4.5)
    {
        muB = 0;
    }
    double minMu = min(muS, muD);
    mu = minMu * muB;
    File8<<i<<'\\t'<<muS<<'\\t'<<j<<'\\t'<<muD<<'\\t'<<k<<'\\t'<<muB<<'\\t'<<mu<<endl;
    return mu;
}

double Rule9(int i, int j, int k)
{
    double mu;
    double muS =0;
    double muD = 0;
    double muB = 0;

    muS = 0.2 * (i-10);
    if (i>15 && i<=20)

```



```

{
    muS = 1- fabs(1-muS);
}
if( i<10 || i>20)
{
    muS = 0;
}

muD = (j - 750)/(1000-750);
if (j<750 )
{
    muD = 0;
}
if(j>1000)
{
    muD = 1;
}

muB = 0.444 * (k-0);
if (k>2.25 && k<=4.5)
{
    muB = 1- fabs(1-muB);
}
if( k<0 || k>4.5)
{
    muB = 0;
}
double minMu = min(muS, muD);
mu = minMu * muB;
File9<<i<<'\\t'<<muS<<'\\t'<<j<<'\\t'<<muD<<'\\t'<<k<<'\\t'<<muB<<'\\t'<<mu<<endl;
return mu;
}

double Rule10(int i, int j, int k)
{
    double mu;
    double muS =0;
    double muD = 0;
    double muB = 0;

    muS = 0.2 * (i-10);

```

```

    if (i>15 && i<=20)
    {
        muS = 1- fabs(1-muS);
    }
    if( i<10 || i>20)
    {
        muS = 0;
    }
    if (j < 0)
    {
        muD = 1;
    }
    if ( j>= 0 && j <= 250)
    {
        muD = (250 - j)/(250);
    }

    if(j>250)
    {
        muD = 0;
    }

    muB = 0.444 * (k-4.5);
    if (k>6.75 && k<=9)
    {
        muB = 1- fabs(1-muB);
    }
    if( k<4.5 || k>9)
    {
        muB = 0;
    }
    double minMu = min(muS, muD);
    mu = minMu * muB;
    File10<<i<<'\t'<<muS<<'\t'<<j<<'\t'<<muD<<'\t'<<k<<'\t'<<muB<<'\t'<<mu<<endl;
    return mu;
}

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);

```

```

ofstream file("C:\\rules.txt");
ofstream file2("C:\\small.txt");

ofstream Small("C:\\MySmall.txt");
ofstream Ballance("C:\\MyBalance.txt");
file<<"IF speed is fast AND distance is short THEN brake is strong;"<<endl;
file<<"IF speed is slow AND distance is long THEN brake is weak;"<<endl;
file<<"IF speed is medium AND distance is medium THEN brake is medium;"<<endl;
file<<"IF speed is very fast AND distance is very short THEN brake is very strong;"<<endl;
file<<"IF speed is very slow AND distance is very long THEN brake is very weak;"<<endl;
file<<"IF speed is very slow AND distance is medium THEN brake is weak;"<<endl;
file<<"IF speed is fast AND distance is medium THEN brake is medium;"<<endl;
file<<"IF speed is slow AND distance is medium THEN brake is weak;"<<endl;
file<<"IF speed is medium AND distance is very long THEN brake weak;"<<endl;
file<<"IF speed is medium AND distance is very short THEN brake strong;"<<endl;

for(int i=1; i<=20; i++)
{
    for(int j = 1; j<=1000; j++)
    {
        double mass[10];
        for(int k=0; k<=9; k++)
        {
            if(i%2==0 && j%100==0)
            {
                double maxim;
                double mu1 = Rule1(i, j, k);

                double mu2 = Rule2(i, j, k);
                maxim = max(mu1, mu2);
                double mu3 = Rule3(i, j, k);
                maxim = max(maxim, mu3);
                double mu4 = Rule4(i, j, k);
                maxim = max(maxim, mu4);
                double mu5 = Rule5(i, j, k);
                maxim = max(maxim, mu5);
                double mu6 = Rule6(i, j, k);
                maxim = max(maxim, mu6);
                double mu7 = Rule7(i, j, k);
                maxim = max(maxim, mu7);
            }
        }
    }
}

```

```

        double mu8 = Rule8(i, j, k);
        maxim = max(maxim, mu8);
        double mu9 = Rule9(i, j, k);
        maxim = max(maxim, mu9);
        double mu10 = Rule10(i, j, k);
        maxim = max(maxim, mu10);
        file2<<maxim<<endl;
        mass[k] = maxim;
    }
}
if (i == 10 && j == 500)
{
    for(int k = 0; k<=9; k++)
    {
        Small<<i<<'\\t'<<j<<'\\t'<<k<<'\\t'<<mass[k]<<endl;
    }
}
if (i%2==0 && j%100==0)
{
    double Summ = 0;
    double SummX = 0;
    for(int k=0; k<=9; k++)
    {
        Summ += mass[k];
        SummX += k*mass[k];
    }
    double balance;
    if(Summ == 0 || SummX == 0)
    {
        balance = 4.5;
    }
    else
    {
        balance = Summ/SummX;
    }
    Ballance<<balance<<endl;
}
}
}
file.close();

```

```
File1.close();  
File2.close();  
File3.close();  
File4.close();  
File5.close();  
File6.close();  
File7.close();  
File8.close();  
File9.close();  
File10.close();  
file2.close();  
Small.close();  
Ballance.close();  
cout<<"The End";  
return a.exec();  
}
```