

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

ОТЧЁТ
по ССОИ
по лабораторной работе №2

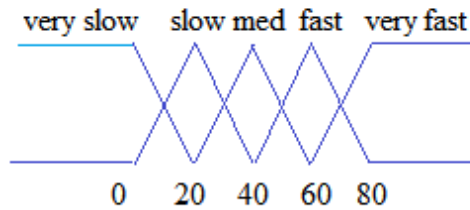
Выполнил:
студентка группы АС-32
Предко А.И.

Проверил:
Акира Имада

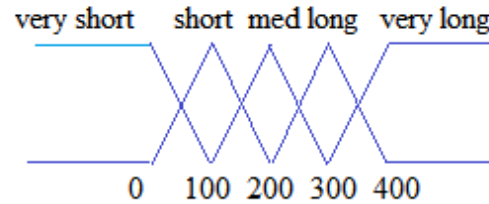
Брест 2014

Лабораторная работа №2

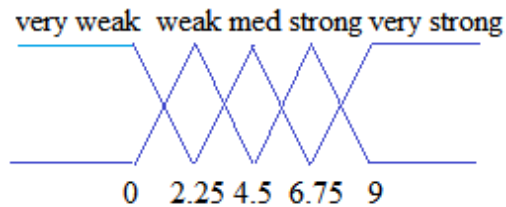
Speed



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;

Big table

54	0.7	40	0.4	4.5	0	0	14	0.7	240	0.4	0	0	0	34	0.7	140	0.4	2.25	0	0	0.040036
54	0.7	40	0.4	4.95	0.199998	0.08	14	0.7	240	0.4	0.45	0.199998	0.079999	34	0.7	140	0.4	2.7	0.19998	0.079992	0.040036
54	0.7	40	0.4	5.4	0.399996	0.16	14	0.7	240	0.4	0.9	0.399996	0.159998	34	0.7	140	0.4	3.15	0.39996	0.159984	0.040036
54	0.7	40	0.4	5.85	0.599994	0.24	14	0.7	240	0.4	1.35	0.599994	0.239998	34	0.7	140	0.4	3.6	0.59994	0.239976	0.040036
54	0.7	40	0.4	6.3	0.799992	0.32	14	0.7	240	0.4	1.8	0.799992	0.319997	34	0.7	140	0.4	4.05	0.79992	0.319968	0.040036

54	0.7	40	0.4	6.75	0.99999	0.4	14	0.7	240	0.4	2.25	0.99999	0.399996	34	0.7	140	0.4	4.5	0.9999	0.39996	0.040036
54	0.7	40	0.4	7.2	0.800012	0.32	14	0.7	240	0.4	2.7	0.800012	0.320005	34	0.7	140	0.4	4.95	0.80012	0.320048	0.320048
54	0.7	40	0.4	7.65	0.600014	0.24	14	0.7	240	0.4	3.15	0.600014	0.240006	34	0.7	140	0.4	5.4	0.60014	0.240056	0.240056
54	0.7	40	0.4	8.1	0.400016	0.16	14	0.7	240	0.4	3.6	0.400016	0.160006	34	0.7	140	0.4	5.85	0.40016	0.160064	0.160064
54	0.7	40	0.4	8.55	0.200018	0.08	14	0.7	240	0.4	4.05	0.200018	0.080007	34	0.7	140	0.4	6.3	0.20018	0.080072	0.080072
54	0.7	60	0.6	4.5	0	0	14	0.7	260	0.6	0	0	0	34	0.7	160	0.6	2.25	0	0	0.080072
54	0.7	60	0.6	4.95	0.199998	0.12	14	0.7	260	0.6	0.45	0.199998	0.119999	34	0.7	160	0.6	2.7	0.19998	0.119988	0.080072
54	0.7	60	0.6	5.4	0.399996	0.24	14	0.7	260	0.6	0.9	0.399996	0.239998	34	0.7	160	0.6	3.15	0.39996	0.239976	0.080072
54	0.7	60	0.6	5.85	0.599994	0.36	14	0.7	260	0.6	1.35	0.599994	0.359996	34	0.7	160	0.6	3.6	0.59994	0.359964	0.080072
54	0.7	60	0.6	6.3	0.799992	0.48	14	0.7	260	0.6	1.8	0.799992	0.479995	34	0.7	160	0.6	4.05	0.79992	0.479952	0.080072
54	0.7	60	0.6	6.75	0.99999	0.6	14	0.7	260	0.6	2.25	0.99999	0.599994	34	0.7	160	0.6	4.5	0.9999	0.59994	0.080072
54	0.7	60	0.6	7.2	0.800012	0.48	14	0.7	260	0.6	2.7	0.800012	0.480007	34	0.7	160	0.6	4.95	0.80012	0.480072	0.480072
54	0.7	60	0.6	7.65	0.600014	0.36	14	0.7	260	0.6	3.15	0.600014	0.360008	34	0.7	160	0.6	5.4	0.60014	0.360084	0.360084
66	0.7	0	0	5.4	0.399996	0	26	0.7	200	0	0.9	0.399996	0	46	0.7	100	0	3.15	0.39996	0	0.040036
66	0.7	0	0	5.4	0.399996	0	26	0.7	200	0	0.9	0.399996	0	46	0.7	100	0	3.15	0.39996	0	0.040036
66	0.7	0	0	5.85	0.599994	0	26	0.7	200	0	1.35	0.599994	0	46	0.7	100	0	3.6	0.59994	0	0.040036
66	0.7	0	0	6.3	0.799992	0	26	0.7	200	0	1.8	0.799992	0	46	0.7	100	0	4.05	0.79992	0	0.040036
66	0.7	0	0	6.75	0.99999	0	26	0.7	200	0	2.25	0.99999	0	46	0.7	100	0	4.5	0.9999	0	0.040036
66	0.7	0	0	7.2	0.800012	0	26	0.7	200	0	2.7	0.800012	0	46	0.7	100	0	4.95	0.80012	0	0.040036
66	0.7	0	0	7.65	0.600014	0	26	0.7	200	0	3.15	0.600014	0	46	0.7	100	0	5.4	0.60014	0	0.040036
66	0.7	0	0	8.1	0.400016	0	26	0.7	200	0	3.6	0.400016	0	46	0.7	100	0	5.85	0.40016	0	0.040036
66	0.7	0	0	8.55	0.200018	0	26	0.7	200	0	4.05	0.200018	0	46	0.7	100	0	6.3	0.20018	0	0.040036
66	0.7	20	0.2	4.5	0	0	26	0.7	220	0.2	0	0	0	46	0.7	120	0.2	2.25	0	0	0.040036

66	0.7	20	0.2	4.95	0.199998	0.04	26	0.7	220	0.2	0.45	0.199998	0.04	46	0.7	120	0.2	2.7	0.19998	0.039996	0.040036
66	0.7	20	0.2	5.4	0.399996	0.08	26	0.7	220	0.2	0.9	0.399996	0.079999	46	0.7	120	0.2	3.15	0.39996	0.079992	0.040036
66	0.7	20	0.2	5.85	0.599994	0.12	26	0.7	220	0.2	1.35	0.599994	0.119999	46	0.7	120	0.2	3.6	0.59994	0.119988	0.040036
66	0.7	20	0.2	6.3	0.799992	0.16	26	0.7	220	0.2	1.8	0.799992	0.159998	46	0.7	120	0.2	4.05	0.79992	0.159984	0.040036
66	0.7	20	0.2	6.75	0.99999	0.2	26	0.7	220	0.2	2.25	0.99999	0.199998	46	0.7	120	0.2	4.5	0.9999	0.19998	0.040036
66	0.7	20	0.2	7.2	0.800012	0.16	26	0.7	220	0.2	2.7	0.800012	0.160002	46	0.7	120	0.2	4.95	0.80012	0.160024	0.160024
66	0.7	20	0.2	7.65	0.600014	0.12	26	0.7	220	0.2	3.15	0.600014	0.120003	46	0.7	120	0.2	5.4	0.60014	0.120028	0.120028
66	0.7	20	0.2	8.1	0.400016	0.08	26	0.7	220	0.2	3.6	0.400016	0.080003	46	0.7	120	0.2	5.85	0.40016	0.080032	0.080032
66	0.7	20	0.2	8.55	0.200018	0.04	26	0.7	220	0.2	4.05	0.200018	0.040004	46	0.7	120	0.2	6.3	0.20018	0.040036	0.040036
78	0.1	140	0.6	7.65	0.600014	0.06	38	0.1	340	0.6	3.15	0.600014	0.060001	58	0.1	240	0.6	5.4	0.60014	0.060014	0.060014
78	0.1	140	0.6	8.1	0.400016	0.04	38	0.1	340	0.6	3.6	0.400016	0.040002	58	0.1	240	0.6	5.85	0.40016	0.040016	0.040016
78	0.1	140	0.6	8.55	0.200018	0.02	38	0.1	340	0.6	4.05	0.200018	0.020002	58	0.1	240	0.6	6.3	0.20018	0.020018	0.020018
78	0.1	160	0.4	4.5	0	0	38	0.1	360	0.4	0	0	0	58	0.1	260	0.4	2.25	0	0	0.020018
78	0.1	160	0.4	4.95	0.199998	0.02	38	0.1	360	0.4	0.45	0.199998	0.02	58	0.1	260	0.4	2.7	0.19998	0.019998	0.020018
78	0.1	160	0.4	5.4	0.399996	0.04	38	0.1	360	0.4	0.9	0.399996	0.04	58	0.1	260	0.4	3.15	0.39996	0.039996	0.020018
78	0.1	160	0.4	5.85	0.599994	0.06	38	0.1	360	0.4	1.35	0.599994	0.059999	58	0.1	260	0.4	3.6	0.59994	0.059994	0.020018
78	0.1	160	0.4	6.3	0.799992	0.08	38	0.1	360	0.4	1.8	0.799992	0.079999	58	0.1	260	0.4	4.05	0.79992	0.079992	0.020018
78	0.1	160	0.4	6.75	0.99999	0.1	38	0.1	360	0.4	2.25	0.99999	0.099999	58	0.1	260	0.4	4.5	0.9999	0.09999	0.020018

78	0.1	160	0.4	7.2	0.800012	0.08	38	0.1	360	0.4	2.7	0.800012	0.080001	58	0.1	260	0.4	4.95	0.80012	0.080012	0.080012
78	0.1	160	0.4	7.65	0.600014	0.06	38	0.1	360	0.4	3.15	0.600014	0.060001	58	0.1	260	0.4	5.4	0.60014	0.060014	0.060014
78	0.1	160	0.4	8.1	0.400016	0.04	38	0.1	360	0.4	3.6	0.400016	0.040002	58	0.1	260	0.4	5.85	0.40016	0.040016	0.040016
78	0.1	160	0.4	8.55	0.200018	0.02	38	0.1	360	0.4	4.05	0.200018	0.020002	58	0.1	260	0.4	6.3	0.20018	0.020018	0.020018
78	0.1	180	0.2	4.5	0	0	38	0.1	380	0.2	0	0	0	58	0.1	280	0.2	2.25	0	0	0.020018
78	0.1	180	0.2	4.95	0.199998	0.02	38	0.1	380	0.2	0.45	0.199998	0.02	58	0.1	280	0.2	2.7	0.19998	0.019998	0.020018
78	0.1	180	0.2	5.4	0.399996	0.04	38	0.1	380	0.2	0.9	0.399996	0.04	58	0.1	280	0.2	3.15	0.39996	0.039996	0.020018
78	0.1	180	0.2	5.85	0.599994	0.06	38	0.1	380	0.2	1.35	0.599994	0.059999	58	0.1	280	0.2	3.6	0.59994	0.059994	0.020018
78	0.1	180	0.2	6.3	0.799992	0.08	38	0.1	380	0.2	1.8	0.799992	0.079999	58	0.1	280	0.2	4.05	0.79992	0.079992	0.020018
78	0.1	180	0.2	6.75	0.99999	0.1	38	0.1	380	0.2	2.25	0.99999	0.099999	58	0.1	280	0.2	4.5	0.9999	0.09999	0.020018
78	0.1	180	0.2	7.2	0.800012	0.08	38	0.1	380	0.2	2.7	0.800012	0.080001	58	0.1	280	0.2	4.95	0.80012	0.080012	0.080012
78	0.1	180	0.2	7.65	0.600014	0.06	38	0.1	380	0.2	3.15	0.600014	0.060001	58	0.1	280	0.2	5.4	0.60014	0.060014	0.060014
78	0.1	180	0.2	8.1	0.400016	0.04	38	0.1	380	0.2	3.6	0.400016	0.040002	58	0.1	280	0.2	5.85	0.40016	0.040016	0.040016
78	0.1	180	0.2	8.55	0.200018	0.02	38	0.1	380	0.2	4.05	0.200018	0.020002	58	0.1	280	0.2	6.3	0.20018	0.020018	0.020018

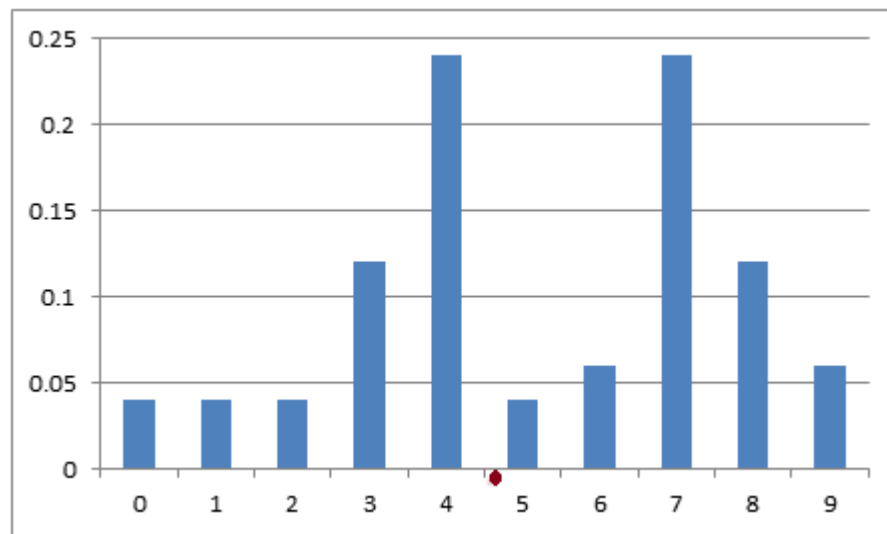
Small table

Speed	Distance	Brake	Mu
46	40	0	0.040036
46	40	1	0.040036

46	40	2	0.040036
46	40	3	0.120048
46	40	4	0.240036
46	40	5	0.040036
46	40	6	0.060054
46	40	7	0.240036
46	40	8	0.120048
46	40	9	0.060054

$$\text{Balance point} = \frac{\sum \text{brake}_i * \text{Mu}_i}{\sum \text{Mu}_i} = \frac{5.182022}{1.00042} = 5.179846$$

Graphic with balance point



Favorite speed = 46; distance = 40; balance = 5.179846;

Program

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

using namespace std;

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

    ofstream file("C:\\rules.txt");
    ofstream file2("C:\\small.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;
    double speed1[20], nu_s1[20], dist1[10], nu_d1[10], brake1[10], nu_b1[10];
    double speed2[20], nu_s2[20], dist2[10], nu_d2[10], brake2[10], nu_b2[10];
    double speed3[20], nu_s3[20], dist3[10], nu_d3[10], brake3[10], nu_b3[10];
    for (int i = 0; i<20; i++)
    {
        speed1[i] = 40+i*2;
        nu_s1[i] = 0.05 * i*2;
        if (speed1[i]>60)
        {
            nu_s1[i] = 1- fabs(1-nu_s1[i]);
        }
        speed2[i] = i*2;
        nu_s2[i] = 0.05 * i*2;
        if (speed2[i]>20)
        {
            nu_s2[i] = 1- fabs(1-nu_s2[i]);
        }
        speed3[i] = 20+i*2;
        nu_s3[i] = 0.05 * i*2;
        if (speed3[i]>40)
        {
```

```

        nu_s3[i] = 1- fabs(1-nu_s3[i]);
    }
}
for (int i = 0; i<10; i++)
{
    dist1[i] = 0+ i*20;
    nu_d1[i] = 0.01 * i*20;
    if (dist1[i]>100)
    {
        nu_d1[i] = 1- fabs(1-nu_d1[i]);
    }
    brake1[i] = 4.5 + i*0.45;
    nu_b1[i] = 0.44444 *i*0.45;
    if (brake1[i]>6.75)
    {
        nu_b1[i] = 1- fabs(1-nu_b1[i]);
    }

    dist2[i] = 200+ i*20;
    nu_d2[i] = 0.01 * i*20;
    if (dist2[i]>300)
    {
        nu_d2[i] = 1- fabs(1-nu_d2[i]);
    }
    brake2[i] = i*0.45;
    nu_b2[i] = 0.44444 * i*0.45;
    if (brake2[i]>2.25)
    {
        nu_b2[i] = 1- fabs(1-nu_b2[i]);
    }
    dist3[i] = 100+ i*20;
    nu_d3[i] = 0.01 * i*20;
    if (dist3[i]>200)
    {
        nu_d3[i] = 1- fabs(1-nu_d3[i]);
    }
    brake3[i] = 2.25 + i*0.45;
    nu_b3[i] = 0.4444 * i*0.45;
    if (brake3[i]>4.5)
    {
        nu_b3[i] = 1- fabs(1-nu_b3[i]);
    }
}

```



```

    }
}

for (int i = 0; i < 20; i++)
{
    double NU1, NU2, NU3, NU;
    for (int j = 0; j < 10; j++)
    {
        if(nu_d1[j]<nu_s1[i])
        {
            NU1 = nu_d1[j];
        }
        else
        {
            NU1 = nu_s1[i];
        }
        if(nu_d2[j]<nu_s2[i])
        {
            NU2 = nu_d2[j];
        }
        else
        {
            NU2 = nu_s2[i];
        }
        if(nu_d3[j]<nu_s3[i])
        {
            NU3 = nu_d3[j];
        }
        else
        {
            NU3 = nu_s3[i];
        }
        for (int k = 0; k < 10; k++)
        {

            file<<speed1[i]<<"\t"<<nu_s1[i]<<"\t";
            file<< dist1[j]<< "\t"<< nu_d1[j]<<"\t";
            file<< brake1[k]<<"\t"<<nu_b1[k]<<"\t";

            file<< NU1 * nu_b1[k]<<"\t";

```

```

file<<speed2[i]<<'\\t'<<nu_s2[i]<<'\\t';
file<< dist2[j]<< '\\t'<< nu_d2[j]<<'\\t';
file<< brake2[k]<<'\\t'<<nu_b2[k]<<'\\t';

file<< NU2 * nu_b2[k]<<'\\t';
file<<speed3[i]<<'\\t'<<nu_s3[i]<<'\\t';
file<< dist3[j]<< '\\t'<< nu_d3[j]<<'\\t';
file<< brake3[k]<<'\\t'<<nu_b3[k]<<'\\t';

file<< NU3 * nu_b3[k]<<'\\t';
if ((NU3 * nu_b3[k])>(NU1*nu_b1[k])) && (( NU3*nu_b3[k])>(NU2*nu_b2[k]))
{
    NU = NU3 * nu_b3[k];
}
if ((NU2*nu_b2[k])>(NU1*nu_b1[k])) && ((NU2*nu_b2[k])>(NU3*nu_b3[k]))
{
    NU = NU2*nu_b2[k];
}
if ((NU1*nu_b1[k])>(NU3*nu_b3[k])) && ((NU1*nu_b1[k])>(NU2*nu_b2[k]))
{
    NU = NU1*nu_b1[k];
}
file<<NU<<endl;

if (brake1[k] == 8.55)
    brake1[k]=9;
file2<<speed1[i]<<'\\t'<<dist1[j]<<'\\t'<<(int)brake3[k]<<'\\t'<<NU<<endl;

file2<<speed1[i]<<'\\t'<<dist1[j]<<'\\t'<<(int)brake1[k]<<'\\t'<<NU<<endl;

file2<<speed1[i]<<'\\t'<<dist1[j]<<'\\t'<<(int)brake2[k]<<'\\t'<<NU<<endl;

    }
}
file.close();
file2.close();
return a.exec();
}

```

