НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп’ютерної та програмної інженерії

Кафедра комп'ютерних інформаційних технологій

**Лабораторна робота № 2.2**

**з навчальної дисципліни**

**"Математичні моделі динамічних систем"**

Тема: Дослідження характеристик стійкості та керованості довгоперіодичного руху літака

в повздовжній площині на ПЕОМ

Виконавець: студент групи УС-213 Дзиговський В.І.

Київ 2020

Дослідження характеристик стійкості та керованості довгоперіодичного руху літака

в повздовжній площині на ПЕОМ

**Мета лабораторної роботи**

1. Вивчити особливості та взаємозв’язок короткоперіодичного та довгоперіодичного руху літака в повздовжній площині.

2. Здійснити моделювання динаміки польоту літака в повздовжній площині на ПЕОМ і дослідити характеристики стійкості та керованості його довгоперіодичного руху.

## **Стислі теоретичні відомості**

Просторовий рух літака описується складною системою нелінійних диференціальних рівнянь 12 порядку як зв’язаний рух тіла, що має шість ступенів свободи, в повздовжній та бічній площинах.

Якщо відомо, що параметри руху літака в бічній площині не змінюються або мало змінюються і практично не впливають на параметри руху літака в повздовжній площині, то можна розглядати ізольований рух літака в повздовжній площині, який описується системою диференціальних рівнянь шостого порядку, визначених відносно . Рух літака в повздовжній площині є основним практично на всіх етапах польоту і займає значну частину всього польотного часу.



У результаті розв’язання характеристичного рівняння руху літака в повздовжній площині отримують чотири попарно спряжені комплексні корені. Перші два з них відповідають так званому короткоперіодичному руху (період коливань – 2-6 с), другі два – довгоперіодичному руху (період коливань – 20-120 с).

Хоча при дії на літак збурень короткоперіодичний і довгоперіодичний рухи розвиваються одночасно, але на початку переважає короткоперіодичний рух, викликаний порушенням рівноваги моментів тангажу, який закінчується протягом декількох секунд після припинення дії збурень. Після цього починають помітно змінюватись кінематичні параметри в довгоперіодичному повільнозгасаючому русі, пов’язаному з порушенням рівноваги сил, які діють на літак.

Найбільш характерними параметрами короткоперіодичного руху є кут атакита вертикальне перевантаження *ny* . Найбільш характерними параметрами довгоперіодичного руху є швидкість польоту *V* та висота *Н* .



Зміни кутів тангажу та нахилу траєкторії відбуваються як при короткоперіодичному, так і при довгоперіодичному рухах.



Умовний поділ повздовжнього збуреного руху на короткоперіодичний та довгоперіодичний є наслідком фізичних особливостей літака, які полягають у тому, що при порушенні вихідного руху кут атаки змінюється швидко, а швидкість польоту *V* літака внаслідок його інерційності – повільно.



Короткоперіодичний рух літака триває до того моменту, поки не настане рівновага моментів, а довгоперіодичний рух – поки не настане рівновага сил, які діють на нього.

Із викладеного вище випливає, що короткоперіодичний та довгоперіодичний рухи літака як би рознесені в часі. Це дозволяє повздовжній збурений рух розбити за часом на два етапи, причому, на першому етапі розглядати тільки короткоперіодичний рух, а на другому – довгоперіодичний.

До довгоперіодичного руху висувається вимога відсутності аперіодичної нестійкості, при якій параметри руху відхиляються від їх початкових значень в одному напрямку з інтенсивністю, що збільшується в часі. Для усунення цих відхилень потрібне постійне втручання пілота в управління літаком, що суттєво ускладнює пілотування.

Коливальна нестійкість у довгоперіодичному русі, на відміну від аперіодичної, розвивається повільно, тому, якщо не висувати жорстких вимог до точного дотримання заданої траєкторії польоту, то вона не дуже ускладнює управління літаком і може бути навіть непомітною для пілота. Це пояснюється тим, що він має достатньо часу для того, щоб помітити та своєчасно усунути відхилення літака від заданого режиму польоту. Однак, якщо потрібно точно дотримуватись траєкторії польоту, то коливальна нестійкість літака у довгоперіодичному русі або недостатня його стійкість, як і аперіодична нестійкість, також вимагатимуть постійного втручання пілота в управління та ускладнює пілотування літака.

Стійкість довгоперіодичного руху літака часто розглядається як стійкість за швидкістю, оскільки вона є визначальним параметром у цьому русі. Якщо довгоперіодичний рух стійкий, то він, як правило, має коливальний характер із слабким затуханням. Причому, суттєвий вплив на характер довгоперіодичного руху має величина частинної похідної , яка



визначає, як відомо, стійкість короткоперіодичного руху.

Період довгоперіодичних коливань літака за швидкістю *V* аналітичним шляхом визначається з виразу:



де: – кругова частота коливань за швидкістю *V*, яка визначається так:



Ступінь повздовжньої статичної стійкості літака за швидкістю та запас стійкості за вертикальним перевантаженням визначаються із співвідношень:

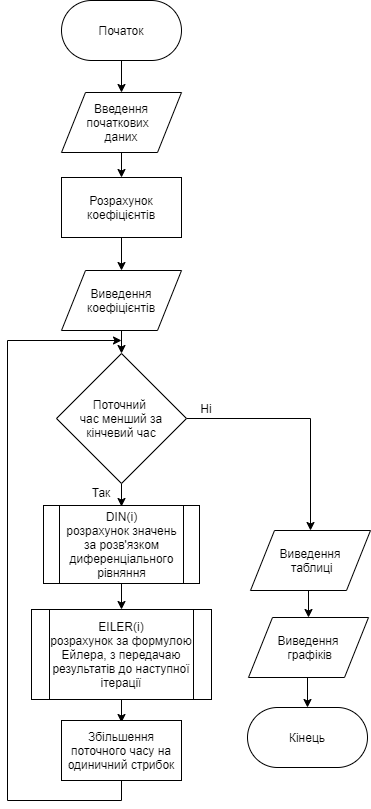


=



Дослідження характеристик довгоперіодичного руху літака можна виконати аналітичним шляхом, а також шляхом моделювання його на ПЕОМ.

**Алгоритм та програмна реалізація на Java**



package programs;

import java.awt.\*;

import java.awt.event.\*;

import java.util.ArrayList;

import javax.swing.\*;

public class MainProgram extends JFrame{

public double S = 201.45, b = 5.285, G = 73000, xt = 0.24, Iz = 660000, ndv = 3, Ydv = 0.5, V = 97.2, H = 500, p = 0.119, a = 338.36, g = 9.81,

Cy0 = -0.255, Cya = 5.78, Cyd = 0.2865, CyM = 0, Cxgp = 0.046, Cxa = 0.286, CxM = 0,

mz0 = 0.2, mzwz = -13, mza\_ = -3.8, mza = -1.83, mzd = -0.96, mzCy = -0.3166, mzM = 0, p1d = 7003, p1V = -13.8;

public double C1, C2, C3, C4, C5, C6, C7, C8, C9, C16, C17, C18, C19, e1, e2, e3, Cygp, agp, dbal, dvny, Tv;

public TextField TC1, TC2, TC3, TC4, TC5, TC6, TC7, TC8, TC9, TC16, TC17, TC18, TC19, Te1, Te2, Te3, TCygp, Tagp, Tdbal, Tdvny, TTv;

public double T = 0, DT = 0.01, TD = 0, TF = 180, DD = 2;

public Button TT, TW, TDV, TDG, Ta, Tteta, Tomega, TNY, TH, TV, TVk;

double WX=0, DG=0, DV = -2;

//double WX=0, DG=5, DV=0;

//double WX=-5, DG=0, DV=0;

TextField Table[][];

double[] min = {0,0,0};

double[] max = {0,0,0};

Scrollbar SB;

Graphics graph;

public class Data{

public double[] Y = new double[8];

public double[] X = new double[8];

public double VV, NY;

public Data(double Y1, double Y2, double Y3, double Y4, double Y5, double Y6) {

Y[0] = Y1;

Y[1] = Y2;

Y[2] = Y3;

Y[3] = Y4;

Y[4] = Y5;

Y[5] = Y6;

}

}

public ArrayList<Data> Database = new ArrayList();

public MainProgram(String s){

super(s);

setLayout(null);

setSize(1545, 830);

setVisible(true);

this.setLocation(0, 0);

this.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

this.setResizable(false);

graph = getGraphics();

TC1 = CreatCTF("", 100, 50);

TC2 = CreatCTF("", 280, 50);

TC3 = CreatCTF("", 460, 50);

TC4 = CreatCTF("", 640, 50);

TC5 = CreatCTF("", 820, 50);

TC6 = CreatCTF("", 1000, 50);

TC7 = CreatCTF("", 1180, 50);

TC8 = CreatCTF("", 100, 80);

TC9 = CreatCTF("", 280, 80);

TC16 = CreatCTF("", 460, 80);

TC17 = CreatCTF("", 640, 80);

TC18 = CreatCTF("", 820, 80);

TC19 = CreatCTF("", 1000, 80);

Te1 = CreatCTF("", 1180, 80);

Te2 = CreatCTF("", 100, 110);

Te3 = CreatCTF("", 280, 110);

TCygp = CreatCTF("", 460, 110);

Tagp = CreatCTF("", 640, 110);

Tdbal = CreatCTF("", 820, 110);

Tdvny = CreatCTF("", 1000, 110);

TTv = CreatCTF("", 1180, 110);

TT = CreateBT("T", 100, 150, 60, 30);

TW = CreateBT("W", 160, 150, 100, 30);

TDV = CreateBT("dв", 260, 150, 100, 30);

TDG = CreateBT("dг", 360, 150, 100, 30);

Ta = CreateBT("a", 460, 150, 100, 30);

Tteta = CreateBT("Teta", 560, 150, 100, 30);

Tomega = CreateBT("Omega", 660, 150, 100, 30);

TNY = CreateBT("ny", 760, 150, 100, 30);

TH = CreateBT("H", 860, 150, 100, 30);

TV = CreateBT("V", 960, 150, 100, 30);

TVk = CreateBT("Vk", 1060, 150, 100, 30);

Table = new TextField[8][11];

for(int i = 0; i < 8; i++) {

for(int j = 0; j < 11; j++) {

Table[i][j] = new TextField("");

Table[i][j].setFont(new Font("TimesRoman", Font.PLAIN, 16));

Table[i][j].setEditable(false);

add(Table[i][j]);

}

Table[i][0].setBounds(102, 180+i\*25, 60, 25);

Table[i][1].setBounds(162, 180+i\*25, 100, 25);

Table[i][2].setBounds(262, 180+i\*25, 100, 25);

Table[i][3].setBounds(362, 180+i\*25, 100, 25);

Table[i][4].setBounds(462, 180+i\*25, 100, 25);

Table[i][5].setBounds(562, 180+i\*25, 100, 25);

Table[i][6].setBounds(662, 180+i\*25, 100, 25);

Table[i][7].setBounds(762, 180+i\*25, 100, 25);

Table[i][8].setBounds(862, 180+i\*25, 100, 25);

Table[i][9].setBounds(962, 180+i\*25, 100, 25);

Table[i][10].setBounds(1062, 180+i\*25, 100, 25);

}

SB = new Scrollbar(Scrollbar.VERTICAL);

SB.setBounds(1162, 150, 20, 230);

SB.setBackground(Color.GRAY);

SB.setMinimum(1);

add(SB);

double m = G/g;

double dx = xt -0.24;

double Mgp = V/a;

Cygp = 2\*G/S/p/V/V;

TCygp.setText(Double.toString(Cygp));

C1 = -(mzwz/Iz)\*S\*b\*b\*p\*(V/2);

TC1.setText(Double.toString(C1));

C2 = -(mza/Iz)\*S\*b\*p\*V\*V/2;

TC2.setText(Double.toString(C2));

C3 = -mzd/Iz\*S\*b\*p\*V\*V/2;

TC3.setText(Double.toString(C3));

C4 = (Cya+Cxgp)/m\*S\*p\*V/2;

TC4.setText(Double.toString(C4));

C5 = -mza\_/Iz\*S\*b\*b\*p\*V/2;

TC5.setText(Double.toString(C5));

C6 = V/57.3;

TC6.setText(Double.toString(C6));

C7 = g/57.3;

TC7.setText(Double.toString(C7));

C8 = ((Cxa-Cygp)\*S\*p\*V\*V/2/57.3/m);

TC8.setText(Double.toString(C8));

C9 = Cyd/m\*S\*p\*V/2;

TC9.setText(Double.toString(C9));

C16 = V/57.3/g;

TC16.setText(Double.toString(C16));

C17 = (-(Cya\*dx)/Iz)\*S\*b\*((p\*V\*V)/2);

TC17.setText(Double.toString(C17));

C18 = (-(Cyd\*dx)/Iz)\*S\*b\*((p\*V\*V)/2);

TC18.setText(Double.toString(C18));

C19 = -((ndv\*p1d)/(57.3\*m));

TC19.setText(Double.toString(C19));

e1 = (Cxgp + (CxM\*Mgp/2)-(ndv\*p1V/p/V/S))\*S\*p\*V/m;

Te1.setText(Double.toString(e1));

e2 = (Cygp+(CyM\*Mgp)/2)\*S\*((57.3\*p)/m);

Te2.setText(Double.toString(e2));

e3 = 57.3\*ndv\*p1V\*Ydv/Iz;

Te3.setText(Double.toString(e3));

agp = 57.3\*(Cygp-Cy0)/Cya;

Tagp.setText(Double.toString(agp));

double abal = 57.3\*(Cygp - Cy0)/Cya;

dbal = -57.3\*(mz0+(mza\*abal/57.3)+Cygp\*dx)/mzd;

Tdbal.setText(Double.toString(dbal));

double sigma = mzCy+p\*S\*b\*mzwz/2/m;

dvny = -57.3\*sigma\*Cygp/mzd;

Tdvny.setText(Double.toString(dvny));

double sigmav = mzCy\*(1+CyM\*Mgp/2/Cygp)-mzM\*Mgp/2/Cygp;

double wv = g\*Math.sqrt(2\*sigmav/sigma)/V;

Tv = 2\*Math.PI/wv;

TTv.setText(Double.toString(Tv));

Database.add(new Data(0, 0, 0, 0, 0, 0));

Database.get(0).Y[0] = V;

Database.get(0).Y[5] = H;

while(T<=TF+DT) {

int i = (int)(Math.round(T/DT));

DIN(i);

EILER(i);

T += DT;

CheckMaxMin(i);

}

for(int i = 0; i < 8; i++) {

Table[i][0].setText(Double.toString((i+1)\*DD));

Table[i][1].setText(Double.toString(WX));

Table[i][2].setText(Double.toString(DV));

Table[i][3].setText(Double.toString(DG));

Table[i][4].setText(Double.toString(Database.get((i+1)\*(int)(DD/DT)).Y[4]));

Table[i][5].setText(Double.toString(Database.get((i+1)\*(int)(DD/DT)).Y[2]));

Table[i][6].setText(Double.toString(Database.get((i+1)\*(int)(DD/DT)).Y[3]));

Table[i][7].setText(Double.toString(Database.get((i+1)\*(int)(DD/DT)).NY));

Table[i][8].setText(Double.toString(Database.get((i+1)\*(int)(DD/DT)).Y[5]));

Table[i][9].setText(Double.toString(Database.get((i+1)\*(int)(DD/DT)).Y[0]));

Table[i][10].setText(Double.toString(Database.get((i+1)\*(int)(DD/DT)).VV));

}

SB.setMaximum((int)(TF/DD)+3);

SB.setValue(1);

SB.addAdjustmentListener(new TableScroll());

CreateGraphicX(max[0], min[0], 55, 440, 450, 350, 1);

CreateGraphicX(max[1], min[1], 555, 440, 450, 350, 2);

CreateGraphicX(max[2], min[2], 1055, 440, 450, 350, 5);

}

public void DIN(int i) {

Database.get(i).VV = Database.get(i).Y[0] - WX;

//Database.get(i).VV = 97.2;

Database.get(i).X[0] = -e1 \* Database.get(i).VV - C8 \* Database.get(i).Y[4] - C7 \* Database.get(i).Y[1] - C19 \* DG;

Database.get(i).X[1] = Database.get(i).Y[2];

Database.get(i).X[3] = C4 \* Database.get(i).Y[4] + e2 \* Database.get(i).VV + C9 \* DV;

Database.get(i).X[4] = Database.get(i).X[1] - Database.get(i).X[3];

Database.get(i).X[2] = -C1 \* Database.get(i).Y[2] - (C2 + C17) \* Database.get(i).Y[4] - C5 \* Database.get(i).X[4] - e3 \* Database.get(i).VV - (C3 + C18) \* DV;

Database.get(i).X[5] = C6 \* Database.get(i).Y[3];

Database.get(i).NY = C16 \* Database.get(i).X[3];

}

public void EILER(int i) {

double[] YN = new double[6];

for(int j = 0; j < 6; j++)

YN[j] = Database.get(i).Y[j] + Database.get(i).X[j]\*DT;

Database.add(new Data(YN[0], YN[1], YN[2], YN[3], YN[4], YN[5]));

}

public void CheckMaxMin(int j) {

if(Database.get(j).NY > max[0])

max[0] = Database.get(j).NY;

if(Database.get(j).NY < min[0])

min[0] = Database.get(j).NY;

if(Database.get(j).VV > max[1])

max[1] = Database.get(j).VV;

if(Database.get(j).VV < min[1])

min[1] = Database.get(j).VV;

if(Database.get(j).Y[5] > max[2])

max[2] = Database.get(j).Y[5];

if(Database.get(j).Y[5] < min[2])

min[2] = Database.get(j).Y[5];

}

public void paint (Graphics gr) {

gr.setColor(Color.lightGray);

gr.fillRect(0, 0, 1545, 830);

gr.setColor(Color.BLACK);

gr.drawLine(0, 430, 1560, 430);

gr.setFont(new Font("TimesRoman", Font.PLAIN, 25));

gr.drawString("Розрахунок коефіцієнтів", 700, 60);

gr.setFont(new Font("TimesRoman", Font.PLAIN, 18));

gr.drawString("C1 =", 60, 100);

gr.drawString("C2 =", 240, 100);

gr.drawString("C3 =", 420, 100);

gr.drawString("C4 =", 600, 100);

gr.drawString("C5 =", 780, 100);

gr.drawString("C6 =", 960, 100);

gr.drawString("C7 =", 1140, 100);

gr.drawString("C8 =", 60, 130);

gr.drawString("C9 =", 240, 130);

gr.drawString("C16 =", 420, 130);

gr.drawString("C17 =", 600, 130);

gr.drawString("C18 =", 780, 130);

gr.drawString("C19 =", 960, 130);

gr.drawString("e1 =", 1140, 130);

gr.drawString("e2 =", 60, 160);

gr.drawString("e3 =", 240, 160);

gr.drawString("Cyгп =", 410, 160);

gr.drawString("aгп =", 600, 160);

gr.drawString("dвбал =", 760, 160);

gr.drawString("dвny =", 950, 160);

gr.drawString("Tv =", 1140, 160);

}

public TextField CreatCTF(String value, int x, int y) {

TextField TF = new TextField(value);

TF.setBounds(x, y, 100, 25);

TF.setEditable(false);

TF.setFont(new Font("TimesRoman", Font.PLAIN, 18));

add(TF);

return TF;

}

public Button CreateBT(String value, int x, int y, int w, int h) {

Button BT = new Button(value);

BT.setBounds(x, y, w, h);

BT.setFont(new Font("TimesRoman", Font.PLAIN, 18));

add(BT);

return BT;

}

public void CreateGraphicX(double max, double min, int x, int y, int w, int h, int mas) {

int bub;

int bx, lx = x;

int by, ly = (int)map(Database.get(0).Y[mas], min, max, y+h, y);

if (mas == 1) {

ly = (int)map(Database.get(0).NY, min, max, y+h, y);

}

if (mas == 2)

ly = (int)map(Database.get(0).VV, min, max, y+h, y);

graph.setColor(Color.white);

graph.fillRect(x, y, w, h);

graph.setFont(new Font("TimesRoman", Font.PLAIN, 20));

double start, end, delta, mod = Math.abs(max - min);

if (mod > 1000) {

delta = 500;

start = Math.ceil(min/100)\*100;

end = Math.floor(max/100)\*100;

}

else if (mod > 200) {

delta = 100;

start = Math.ceil(min/10)\*10;

end = Math.floor(max/10)\*10;

}

else if (mod > 50) {

delta = 20;

start = Math.ceil(min/10)\*10;

end = Math.floor(max/10)\*10;

}

else if (mod > 10) {

delta = 5;

start = Math.ceil(min\*2/10)\*10/2;

end = Math.floor(max);

}

else if (mod > 7) {

delta = 2;

start = Math.ceil(min/2)\*2;

end = Math.floor(max/2)\*2;

}

else if (mod > 3) {

delta = 0.5;

start = Math.ceil(min);

end = Math.floor(max);

}

else if (mod > 1) {

delta = 0.5;

start = Math.ceil(min\*2)/2;

end = Math.floor(max\*2)/2;

}

else if (mod > 0.3) {

delta = 0.1;

start = Math.ceil(min\*10)/10;

end = Math.floor(max\*10)/10;

}

else if (mod > 0.15) {

delta = 0.05;

start = Math.ceil(min\*20)/20;

end = Math.floor(max\*20)/20;

}

else {

delta = 0.02;

start = Math.ceil(min\*500)/500;

end = Math.floor(max\*500)/500;

}

for (double i = start; i <= end; i+=delta) {

bub = (int)map(i, min, max, y+h, y);

if((int)Math.floor(i\*100) != 0) {

graph.setColor(Color.black);

if(i % 1 == 0)

graph.drawString(Integer.toString((int)Math.round(i\*100)/100), x-35, bub+10);

else if (((double)Math.round(i\*100)/100)\*10 % 1 == 0)

graph.drawString(Double.toString((double)Math.round(i\*100)/100), x-35, bub+10);

else

graph.drawString(Double.toString((double)Math.round(i\*100)/100), x-45, bub+10);

graph.setColor(Color.lightGray);

graph.drawLine(x, bub, x+w, bub);

}

}

for (double i = TF/6; i <= TF-TF/6; i += TF/6) {

bub = (int)map(i, 0, TF, x, x+w);

graph.setColor(Color.black);

graph.drawString(Double.toString(i), bub, (int)map(0, min, max, y+h, y));

graph.setColor(Color.lightGray);

graph.drawLine(bub, y, bub, y+h);

}

graph.setColor(Color.black);

graph.fillRect(x, y, 2, h);

bub = (int)map(0, min, max, y+h, y);

graph.fillRect(x, bub, w, 2);

graph.setFont(new Font("TimesRoman", Font.PLAIN, 30));

graph.drawString("^", x-6, y+20);

graph.setFont(new Font("TimesRoman", Font.PLAIN, 30));

graph.drawString(">", x+w-15, bub+11);

graph.setFont(new Font("TimesRoman", Font.PLAIN, 20));

graph.drawString("T", x+w/2, y+h+20);

switch(mas) {

case 1:

graph.drawString("ny", x-30, bub+10);

break;

case 2:

graph.drawString("V", x-30, bub+10);

break;

case 5:

graph.drawString("H", x-20, bub+10);

break;

}

graph.setColor(Color.red);

for(double i = 2\*DT; i < TF-0.1; i+=DT) {

bx = (int)map(i, 0, TF, x, x+w);

if (mas == 1)

by = (int)map((Database.get((int)(i/DT)).NY), min, max, y+h, y);

else if(mas == 2)

by = (int)map((Database.get((int)(i/DT)).VV), min, max, y+h, y);

else

by = (int)map((Database.get((int)(i/DT)).Y[mas]), min, max, y+h, y);

graph.drawLine(lx, ly, bx, by);

graph.drawLine(lx, ly, bx+1, by);

graph.drawLine(lx, ly, bx, by+1);

graph.drawLine(lx+1, ly, bx, by);

graph.drawLine(lx, ly+1, bx, by);

graph.drawLine(lx, ly, bx-1, by);

graph.drawLine(lx, ly, bx, by-1);

graph.drawLine(lx-1, ly, bx, by);

graph.drawLine(lx, ly-1, bx, by);

lx = bx;

ly = by;

}

}

public double map(double input, double INmin, double INmax, double OUTmin, double OUTmax) {

return (input-INmin)\*(OUTmax-OUTmin)/(INmax-INmin)+OUTmin;

}

public class TableScroll implements AdjustmentListener {

public void adjustmentValueChanged(AdjustmentEvent e) {

int bub = SB.getValue();

for(int i = bub; i < bub+8; i++) {

Table[i-bub][0].setText(Double.toString((i)\*DD));

Table[i-bub][1].setText(Double.toString(WX));

Table[i-bub][2].setText(Double.toString(DV));

Table[i-bub][3].setText(Double.toString(DG));

Table[i-bub][4].setText(Double.toString(Database.get((i)\*(int)(DD/DT)).Y[4]));

Table[i-bub][5].setText(Double.toString(Database.get((i)\*(int)(DD/DT)).Y[2]));

Table[i-bub][6].setText(Double.toString(Database.get((i)\*(int)(DD/DT)).Y[3]));

Table[i-bub][7].setText(Double.toString(Database.get((i)\*(int)(DD/DT)).NY));

Table[i-bub][8].setText(Double.toString(Database.get((i)\*(int)(DD/DT)).Y[5]));

Table[i-bub][9].setText(Double.toString(Database.get((i)\*(int)(DD/DT)).Y[0]));

Table[i-bub][10].setText(Double.toString(Database.get((i)\*(int)(DD/DT)).VV));

}

}

}

public static void main(String[] args) {

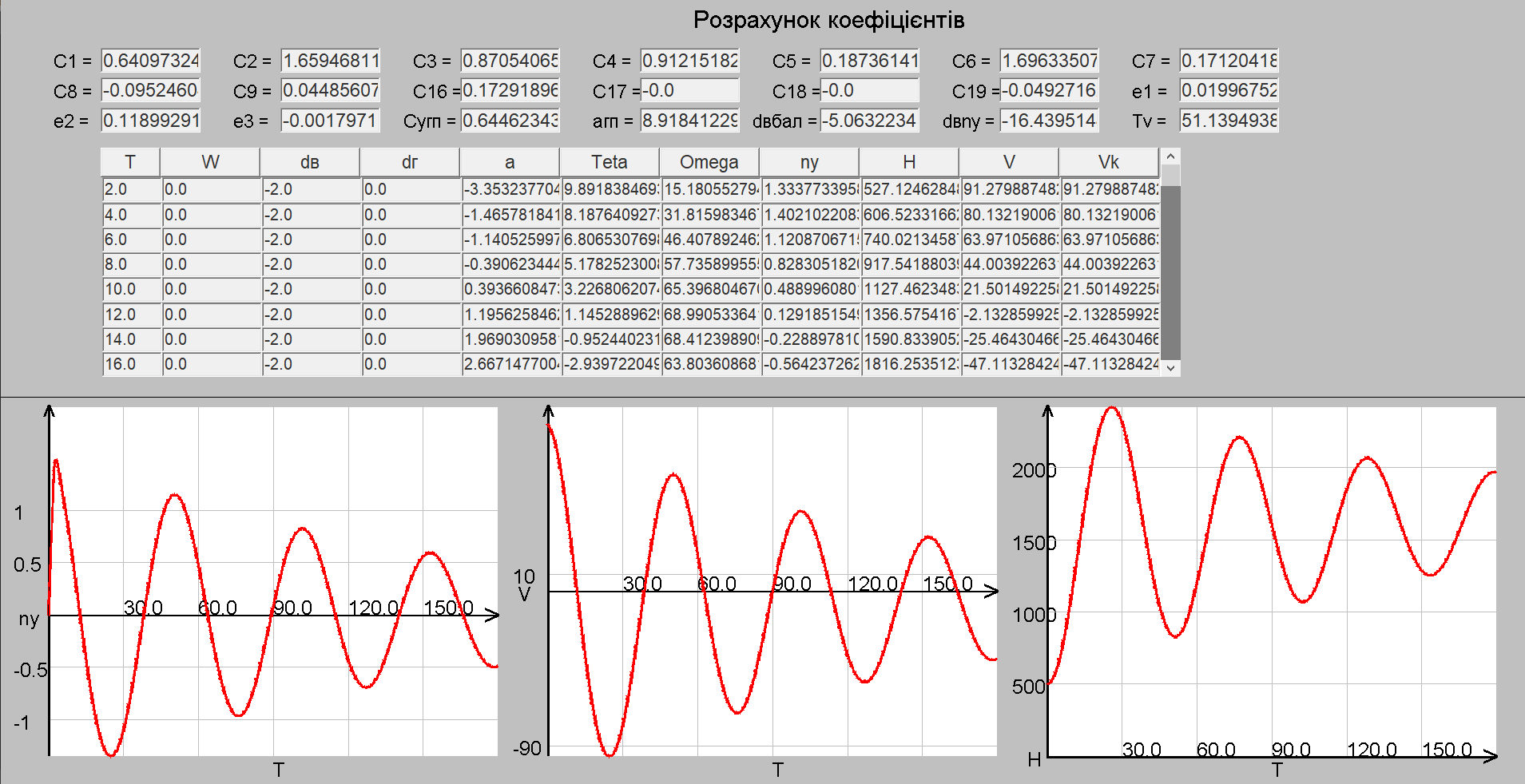
new MainProgram("Lab 2.2");

}

}

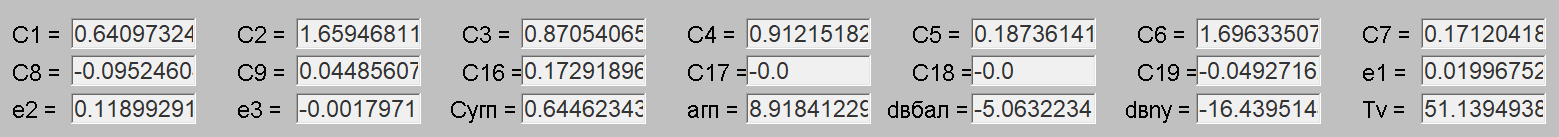
**Результати**

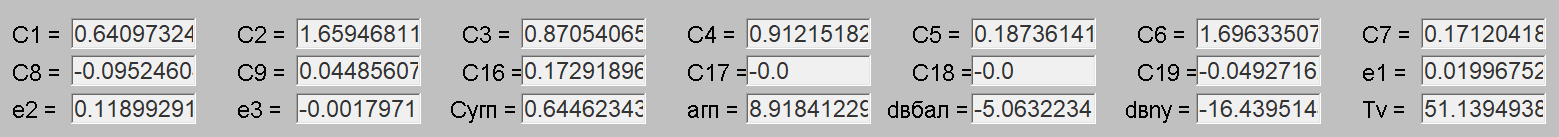
**Загальний вигляд програми**



**Чисельні результати досліджень (при = 24%)**





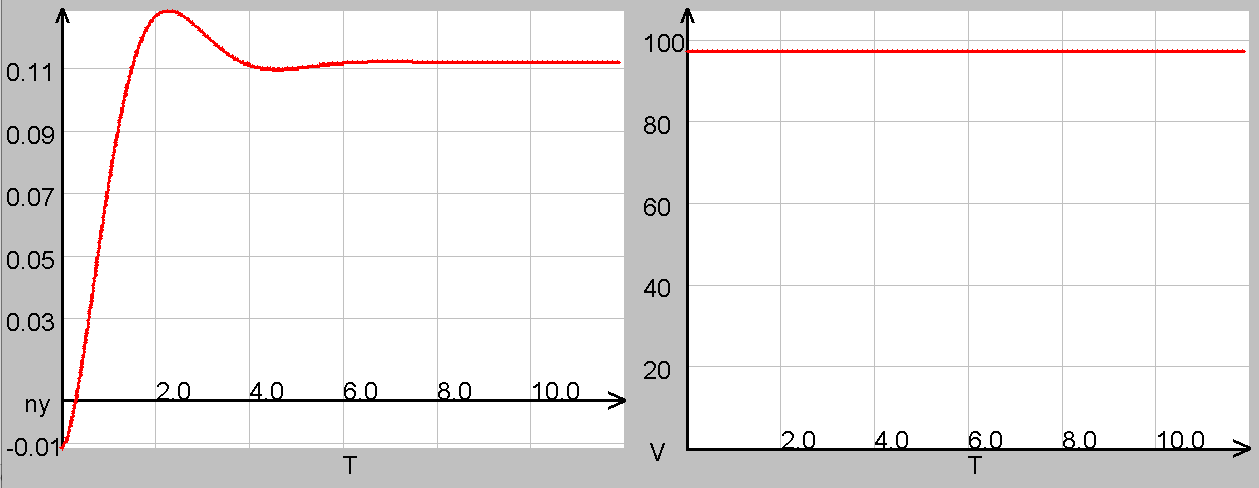


**Результати графічних досліджень**

Вертикальне перевантаження при «дачі» -2 та постійній швидкості:

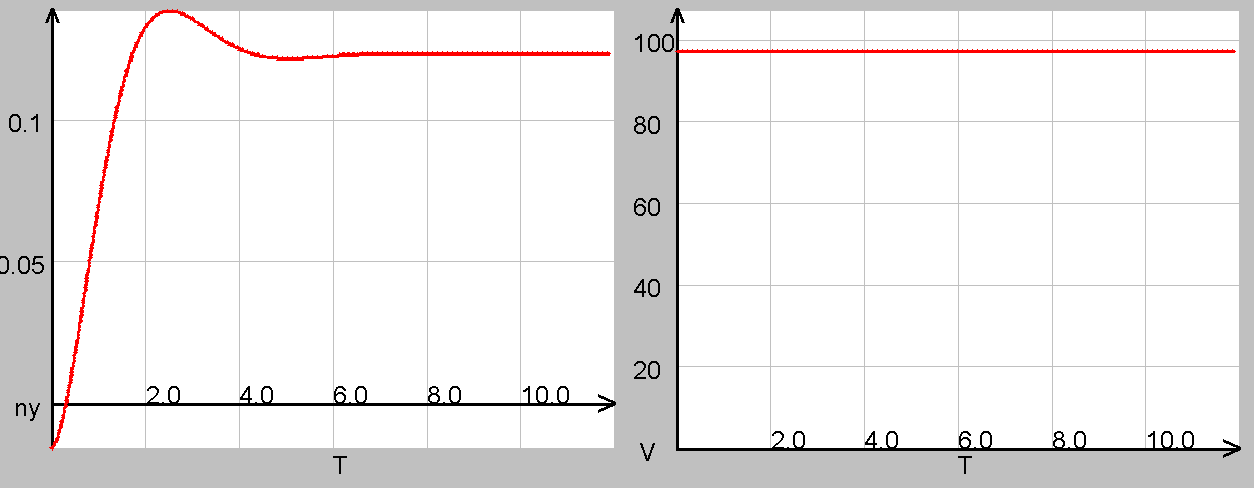
= 18%





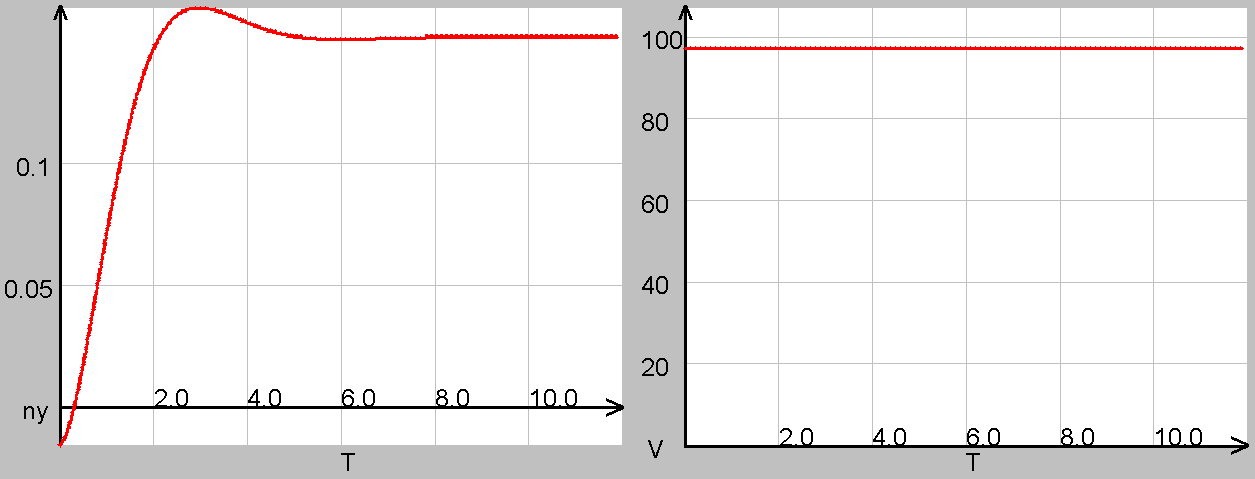
= 24%





= 32%



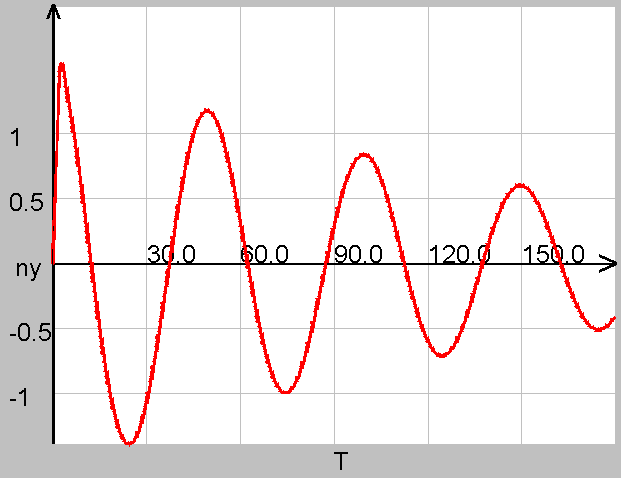


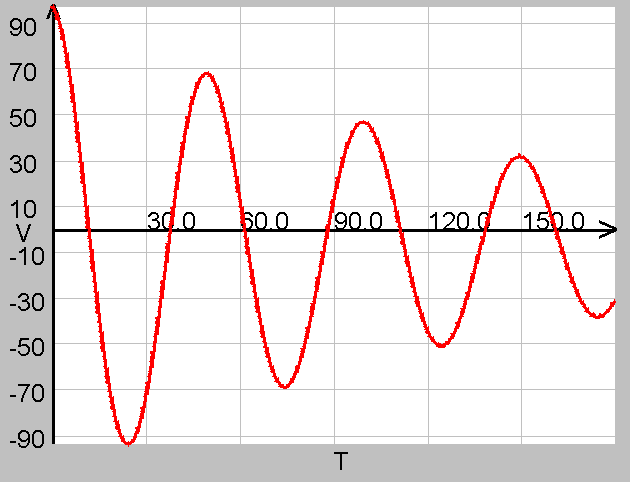
**Статичні та динамічні характеристики повздовжньої керованості літака**

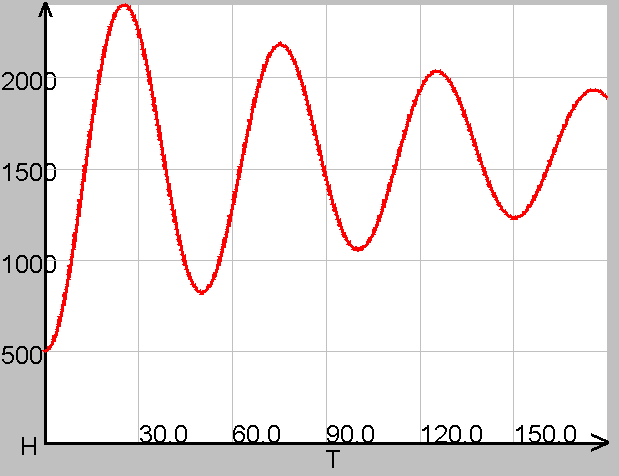
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| , град.  = 24% | | , град/од.пер.  = 24% | | | *Т*v , с | | | |
| Аналіт. розрах. | Реальний  об’єкт | Аналіт. розрах. | Модель | Реальний  об’єкт | Аналіт. розрах.  = 24% | Модель | | |
| = 18% | = 24% | = 32% |
| -5.06 | -6.5 | -16.43 | -16.26 | -18.5 | 51.139 | 50 | 50 | 52 |

1. «Дача» руля висоти на величину –20 при значеннях центрівки:

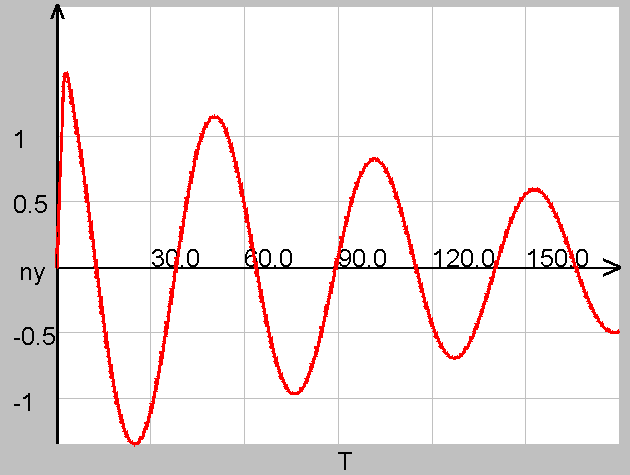
18%

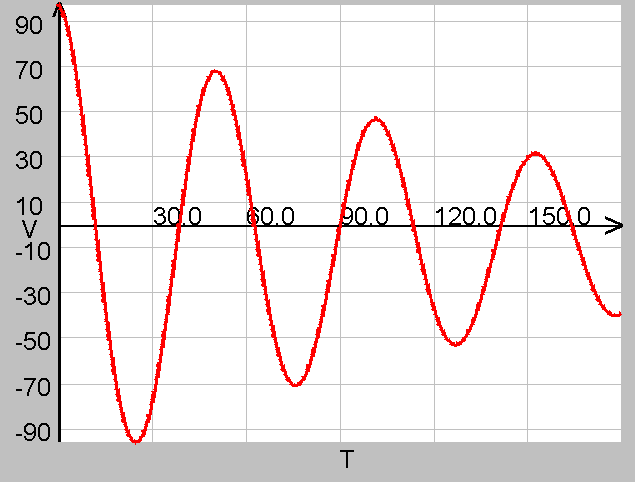


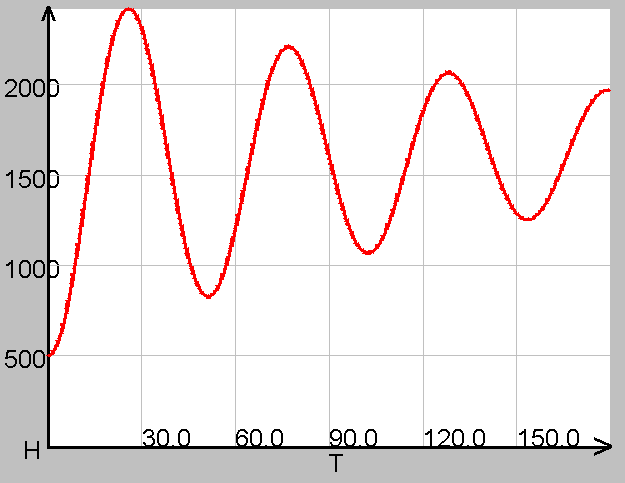




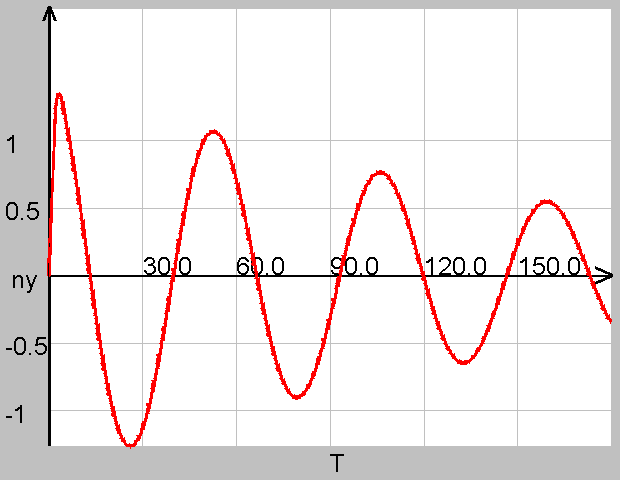
24%

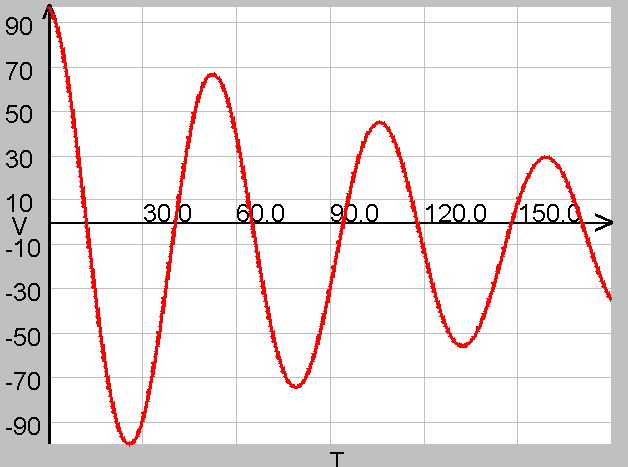


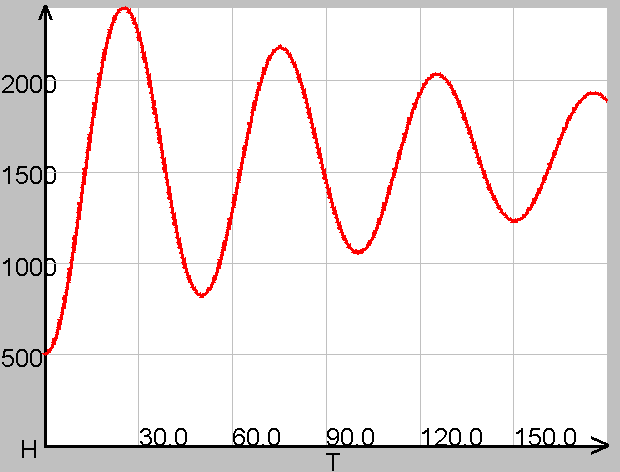




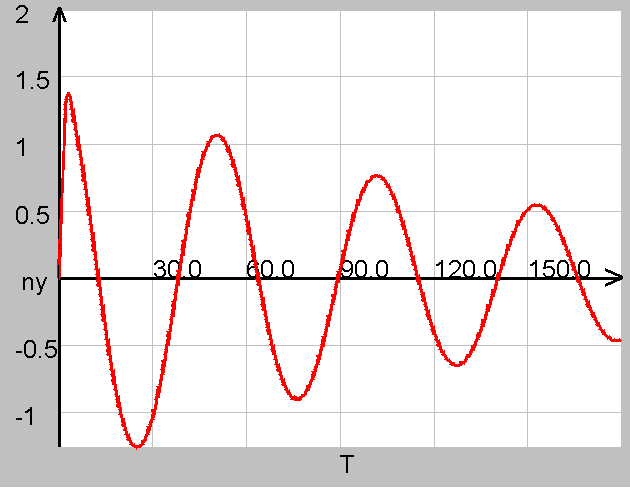
32%

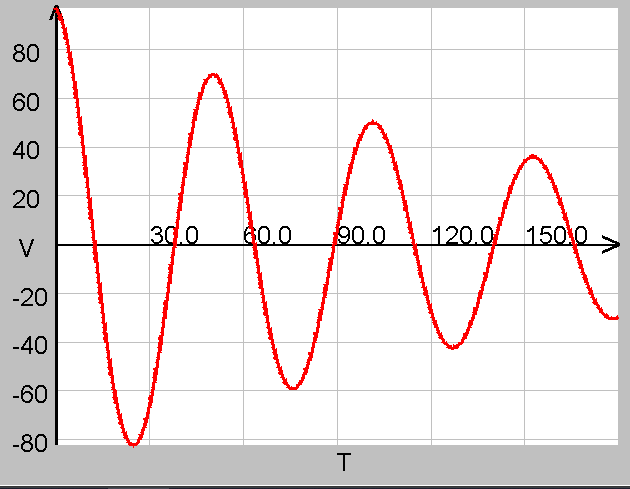


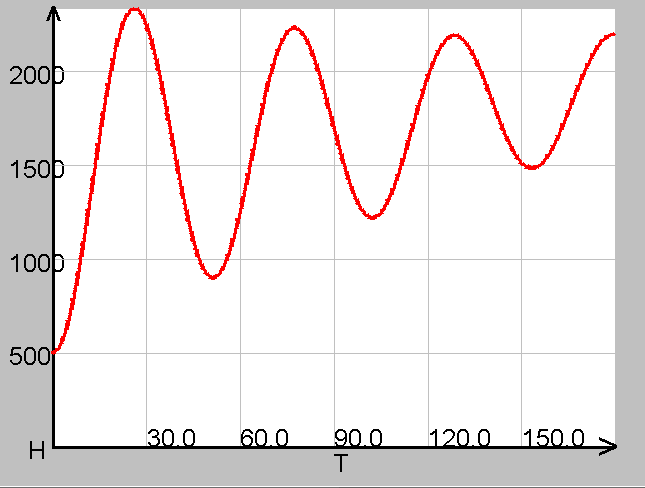




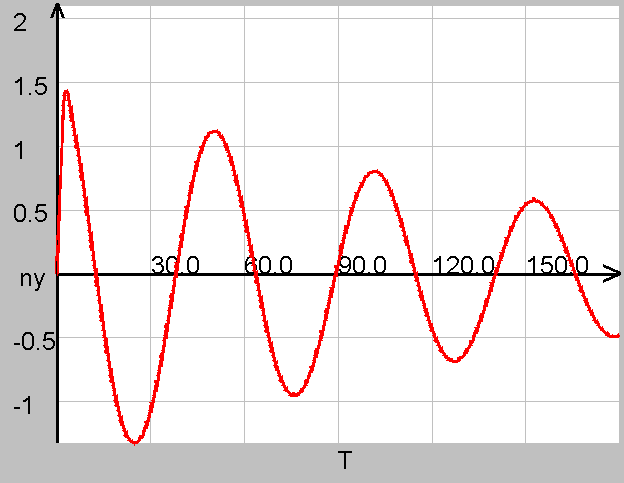
1. «Дача» сектора газу на величину +5 при значенні центрівки 24%:

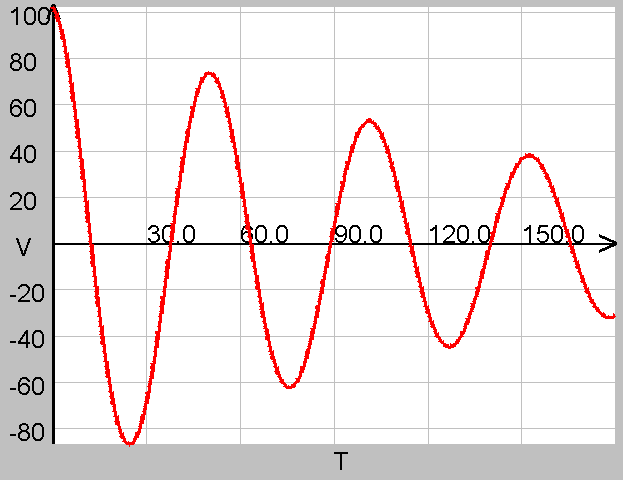


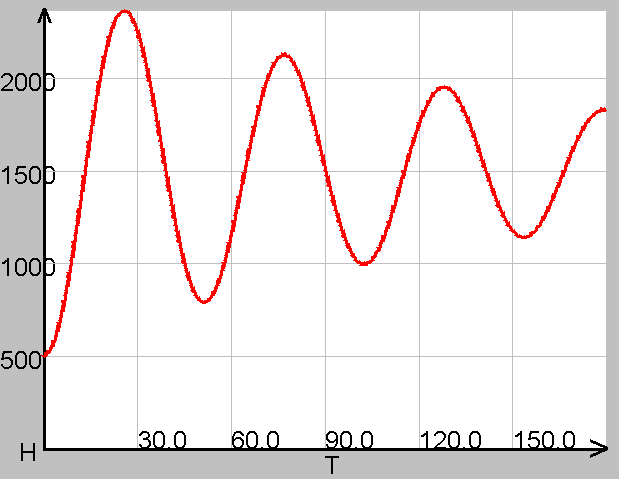




1. Зустрічний порив вітру швидкістю –5 м/с при значенні центрівки 24%:







**Висновок про ступінь динамічної подібності моделі і реального об'єкта.**

За результатами графічних та табличних досліджень можна стверджувати, що система э достатньо подібною, хоча і має певні відхилення. Аналітичні розрахунки мають більші відхилення, аніж графічні. Але час перехідного процесу та періоду коливань є достатньо точними.

**Висновок про стійкість короткоперіодичного руху літака в повздовжній площині.**

За результатами графічних досліджень, а саме графіку перехідного процесу можна стверджувати, що короткоперіодичний рух моделі є стійким, оскільки амплітуда коливань зменшується з кожним періодом, але перехідний процес займає багато часу.

**Висновок про стійкість довгоперіодичного руху літака в повздовжній площині.**

За результатами графічних досліджень, а саме графіку швидкості можна стверджувати, що довгоперіодичний рух моделі є стійким, оскільки амплітуда коливань зменшується з кожним періодом. Перехідний процес займає стільки часу, скільки і перехідний процес короткоперіодичного руху.

**Висновок про вплив центрівки літака на динамічні та статичні характеристики стійкості та повздовжньої керованості літака.**

За результатами графічних досліджень при зменшені центрівки час перехідного процесу та період коливань зменшуються, що може вплинути на стійкість системи, а саме на зменшення стійкості. При збільшені центрівки збільшується час перехідного процесу та період коливань, що робить рух моделі більш плавним.

**Висновок про вплив на короткоперіодичний та довгоперіодичний рух літака зміни тяги двигунів.**

За результатами графічних досліджень при зміні тяги двигуна система має найменший вплив на швидкість, але найбільший на висоту. Вертикальне перевантаження таке саме, як і при «дачі» руля висоти. Стійкість системи майже незмінна.

**Висновок про вплив на короткоперіодичний та довгоперіодичний рух літака вітрових збурень.**

За результатами графічних досліджень при вітрових збурень система має найменший вплив на висоту. Вплив на швидкість менша ніж при «дачі» руля висоти. Вертикальне перевантаження таке саме, як і при «дачі» руля висоти. Стійкість системи майже незмінна.

**Висновки до лабораторної роботи:** Вивчено особливості та взаємозв’язок короткоперіодичного та довгоперіодичного руху літака в повздовжній площині.

Здійснено моделювання динаміки польоту літака в повздовжній площині на ПЕОМ і досліджено характеристики стійкості та керованості його довгоперіодичного руху.