НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп’ютерної та програмної інженерії

Кафедра комп'ютерних інформаційних технологій

**Лабораторна робота № 2.1**

**з навчальної дисципліни**

**"Математичні моделі динамічних систем"**

Тема: Дослідження характеристик стійкості та

керованості короткоперіодичного руху літака

в повздовжній площині на ПЕОМ

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

Київ 2020

**ДОСЛІДЖЕННЯ ХАРАКТЕРИСТИК СТІЙКОСТІ ТА**

**КЕРОВАНОСТІ КОРОТКОПЕРІОДИЧНОГО РУХУ ЛІТАКА**

**В ПОВЗДОВЖНІЙ ПЛОЩИНІ НА ПЕОМ**

**Мета лабораторної роботи**

1. Вивчити динамічні та статичні характеристики стійкості та керованості короткоперіодичного руху літака.

2. Здійснити моделювання короткоперіодичного руху літака на ПЕОМ і дослідити характеристики його стійкості та керованості.

## **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 . Якщо відомо, що параметри руху літака в повздовжній площині не змінюються або змінюються мало і практично не впливають на параметри руху літака в бічній площині, то можна розглядати ізольований рух літака в бічній площині, який описується системою диференціальних рівнянь шостого порядку, визначених відносно .



Якщо відомо, що параметри польоту літака змінюються відносно своїх початкових значень в незначному діапазоні, то з’являється можливість здійснити лінеаризацію як самих нелінійних рівнянь, так нелінійних функцій в їх правих частинах.

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 рознесені в часі. Це дозволяє повздовжній збурений рух літака розбити за часом на два етапи, причому, на першому етапі розглядати тільки короткоперіодичний рух, а на другому – довгоперіодичний.

Роль цих двох рухів з погляду їх оцінки пілотом літака істотно відрізняється. Якщо до довгоперіодичного руху вимоги пілота зводяться тільки до того, щоб він не був аперіодично нестійким, тобто щоб літак під дією збурень не відходив від вихідного режиму з наростаючою за часом амплітудою, то до характеристик короткоперіодичного руху висуваються більш вагомі вимоги (як до його загасання, так і до «закиду» параметрів у періодичному русі).

Така відмінність у ставленні пілота до характеристик цих двох рухів зумовлена тим, що він може втручатися в розвиток довгоперіодичного руху через його тривале протікання і відхиленням органів управління змінювати його характер.

У той же час у розвиток короткоперіодичного руху, з огляду на його швидке протікання, пілот втрутитися може не завжди. Тому характеристики цього руху відіграють важливу роль.

Стійкість збуреного руху літака є однією з найважливіших пілотажних характеристик.

У перший момент після порушення повздовжньої рівноваги моментів найбільш повно виявляється стійкість літака за кутом атаки (вертикальним) перевантаженням – його спроможність без втручання пілота зберігати перевантаження вихідного режиму польоту.

Літак є статично стійким за кутом атаки (вертикальним перевантаженням), якщо аеродинамічний момент, що виникає при відхиленні літака від положення повздовжньої рівноваги, спрямований у бік повернення його до вихідного положення.

Умовою статичної стійкості літака за кутом атаки (вертикальним перевантаженням) є від’ємність за знаком коефіцієнта повздовжньої статичної стійкості .



Умовою динамічної стійкості літака за кутом атаки (вертикальним перевантаженням) є від’ємність дійсних частин коренів характеристичного рівняння короткоперіодичного руху. У цьому випадку коливання по куту атаки (перевантаженню), викликані дією збурень, будуть згасаючими.

Під керованістю літака мають на увазі його здатність реагувати певними змінами параметрів свого руху на відхилення органів управління.

До статичних характеристик повздовжньої керованості літака належать балансоване відхилення руля висоти , витрата руля висоти на створення одиниці вертикального перевантаження та інші.



Балансоване відхилення руля висоти можна визначити з виразу коефіцієнта повздовжнього статичного моменту літака в збалансованому стані, який дорівнює нулю.

Звідси балансоване відхилення руля висоти визначається через аеродинамічні характеристики літака так:

= 57,3



де: = 57,3 ; =



Витрата руля висоти на створення одиниці вертикального перевантаження визначається так:

= = 57,3



де: – величина «дачі» руля висоти відносно балансованого положення;



– величина приросту вертикального перевантаження, що встановлюється після закінчення перехідного процесу;



– запас стійкості літака за вертикальним перевантаженням, що чисельно дорівнює різниці між поточною та нейтральною центрівками, які виражені у відносних одиницях:



=



При дослідженні характеристик стійкості та керованості літака шляхом моделювання динаміки польоту на ПЕОМ визначається як відношення величини «дачі» руля висоти відносно балансованого положення до приросту вертикального перевантаження, що встановлюється після закінчення перехідного процесу :



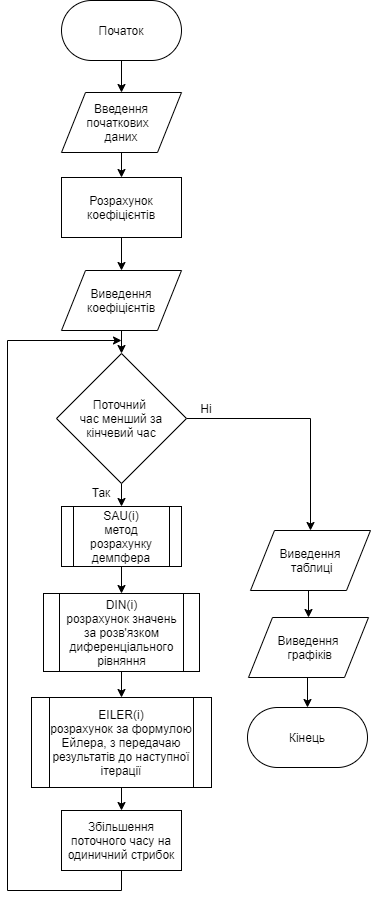
=



Визначення динамічних характеристик керованості, до яких відносяться власна кругова частота (період *Тα* ) короткоперіодичного руху, коефіцієнт його відносного загасання *ξ*α , відносний закид перевантаження, час перехідного процесу *t*пп*α* здійснюється або аналітичним методом, або шляхом моделювання динаміки польоту на ПЕОМ за перехідним процесом за вертикальним перевантаженням (кутом атаки) при збуренні короткоперіодичного руху шляхом «дачі» руля висоти.



**Алгоритм та програмна реалізація на Java**



package prog;

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, x = 0.24, I = 660000, V, H, p, a, Cy0, Cya, Cyd, Cx, mz, mzw, mza\_, mza, mzd, g = 9.81;

public TextField TS, Tb, TG, Tx, TI, TV, TH, Tp, Ta, TCy0, TCya, TCyd, TCx, Tmz, Tmzw, Tmza\_, Tmza, Tmzd;

public double C1, C2, C3, C4, C5, C9, C16, Cy, Ca, Cd;

public TextField TC1, TC2, TC3, TC4, TC5, TC9, TC16, TCy, TCa, TCd;

public double T, DT, TD, TF, DD;

public TextField TT, TDT, TTD, TTF, TDD;

public Button S1, S2, S3;

public double Ks = 0.112, xv = -17.86, kwz = 1.0, Twz = 0.7;

public int switcher = 1;

TextField Table[][];

double min, max;

Scrollbar SB;

Graphics graph;

public class Data{

public double[] Y = new double[5];

public double[] X = new double[5];

public double NY, DV;

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

Y[0] = Y1;

Y[1] = Y2;

Y[2] = Y3;

Y[3] = Y4;

Y[4] = Y5;

}

}

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();

TT = CreatTF("0", 85, 60);

TDT = CreatTF("0.01", 85, 120);

TTD = CreatTF("0", 85, 180);

TTF = CreatTF("20", 85, 240);

TDD = CreatTF("0.5", 85, 300);

TV = CreatTF("97.2", 85, 360);

TH = CreatTF("500", 270, 60);

Tp = CreatTF("0.1190", 270, 120);

Ta = CreatTF("338.36", 270, 180);

TCy0 = CreatTF("-0.255", 270, 240);

TCya = CreatTF("5.78", 270, 300);

TCyd = CreatTF("0.2865", 270, 360);

TCx = CreatTF("0.046", 455, 60);

Tmz = CreatTF("0.20", 455, 120);

Tmzw = CreatTF("-13.0", 455, 180);

Tmza\_ = CreatTF("-3.8", 455, 240);

Tmza = CreatTF("-1.83", 455, 300);

Tmzd = CreatTF("-0.96", 455, 360);

S1 = new Button("1");

S1.setBounds(150, 410, 40, 40);

S1.setFont(new Font("TimesRoman", Font.PLAIN, 25));

S1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

S1.setBackground(Color.green);

S2.setBackground(Color.red);

S3.setBackground(Color.red);

switcher = 1;

}

});

S1.setBackground(Color.green);

add(S1);

S2 = new Button("2");

S2.setBounds(200, 410, 40, 40);

S2.setFont(new Font("TimesRoman", Font.PLAIN, 25));

S2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

S1.setBackground(Color.red);

S2.setBackground(Color.green);

S3.setBackground(Color.red);

switcher = 2;

}

});

S2.setBackground(Color.red);

add(S2);

S3 = new Button("3");

S3.setBounds(250, 410, 40, 40);

S3.setFont(new Font("TimesRoman", Font.PLAIN, 25));

S3.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

S1.setBackground(Color.red);

S2.setBackground(Color.red);

S3.setBackground(Color.green);

switcher = 3;

}

});

S3.setBackground(Color.red);

add(S3);

Button Input = new Button("Розрахувати");

Input.setBounds(350, 410, 200, 40);

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

Input.addActionListener(new Calculate());

add(Input);

TC1 = CreatCTF("", 690, 710);

TC2 = CreatCTF("", 860, 710);

TC3 = CreatCTF("", 1030, 710);

TC4 = CreatCTF("", 1200, 710);

TC5 = CreatCTF("", 1370, 710);

TC9 = CreatCTF("", 690, 750);

TC16 = CreatCTF("", 860, 750);

TCy = CreatCTF("", 1030, 750);

TCa = CreatCTF("", 1200, 750);

TCd = CreatCTF("", 1370, 750);

Button BT= new Button("T");

BT.setBounds(18, 520, 70, 50);

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

add(BT);

Button BDV= new Button("δв");

BDV.setBounds(88, 520, 100, 50);

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

add(BDV);

Button Bny= new Button("ny");

Bny.setBounds(188, 520, 140, 50);

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

add(Bny);

Button BY1= new Button("Кут тангажу");

BY1.setBounds(328, 520, 140, 50);

BY1.setFont(new Font("TimesRoman", Font.PLAIN, 18));

add(BY1);

Button BY3= new Button("Кут атаки");

BY3.setBounds(468, 520, 140, 50);

BY3.setFont(new Font("TimesRoman", Font.PLAIN, 18));

add(BY3);

Table = new TextField[8][5];

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

for(int j = 0; j < 5; 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(18, 570+i\*25, 70, 25);

Table[i][1].setBounds(88, 570+i\*25, 100, 25);

Table[i][2].setBounds(188, 570+i\*25, 140, 25);

Table[i][3].setBounds(328, 570+i\*25, 140, 25);

Table[i][4].setBounds(468, 570+i\*25, 140, 25);

}

SB = new Scrollbar(Scrollbar.VERTICAL);

SB.setBounds(610, 520, 20, 250);

SB.setBackground(Color.GRAY);

SB.setMinimum(1);

add(SB);

}

public void paint (Graphics gr) {

gr.setColor(Color.lightGray);

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

gr.setColor(Color.BLACK);

gr.drawLine(0, 500, 650, 500);

gr.drawLine(650, 700, 1545, 700);

gr.drawLine(650, 0, 650, 830);

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

gr.drawString("Панель введння даних", 170, 60);

gr.drawString("Таблиця даних", 230, 530);

gr.drawString("Графік", 1050, 60);

gr.drawString("Розрахунок констант", 1010, 730);

gr.drawString("Демпфер:", 35, 465);

gr.drawString("T = ", 35, 116);

gr.drawString("DT = ", 35, 176);

gr.drawString("TD = ", 35, 236);

gr.drawString("TF = ", 35, 296);

gr.drawString("DD = ", 35, 356);

gr.drawString("V = ", 35, 416);

gr.drawString("H = ", 222, 116);

gr.drawString("ρ = ", 222, 176);

gr.drawString("a = ", 222, 236);

gr.drawString("C = ", 222, 296);

gr.drawString("C = ", 222, 356);

gr.drawString("C = ", 222, 416);

gr.drawString("C = ", 409, 116);

gr.drawString("m = ", 409, 176);

gr.drawString("m = ", 409, 236);

gr.drawString("m = ", 409, 296);

gr.drawString("m = ", 409, 356);

gr.drawString("m = ", 409, 416);

gr.setFont(new Font("TimesRoman", Font.PLAIN, 15));

gr.drawString("0", 46, 423);

gr.drawString("0", 239, 123);

gr.drawString("н", 236, 243);

gr.drawString("y0", 236, 303);

gr.drawString("y", 236, 363);

gr.drawString("α", 236, 338);

gr.drawString("y", 236, 423);

gr.drawString("δв", 236, 398);

gr.drawString("x", 425, 123);

gr.drawString("z0", 430, 183);

gr.drawString("z", 430, 243);

gr.drawString("ωz", 428, 223);

gr.drawString("\_\_", 430, 210);

gr.drawString("z", 430, 303);

gr.drawString("α", 428, 283);

gr.drawString(".", 430, 273);

gr.drawString("\_", 429, 265);

gr.drawString("z", 430, 363);

gr.drawString("α", 428, 343);

gr.drawString("z", 430, 423);

gr.drawString("δв", 428, 403);

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

gr.drawString("C1 =", 650, 760);

gr.drawString("C2 =", 810, 760);

gr.drawString("C3 =", 980, 760);

gr.drawString("C4 =", 1150, 760);

gr.drawString("C5 =", 1320, 760);

gr.drawString("C9 =", 650, 800);

gr.drawString("C16 =", 810, 800);

gr.drawString("Cуб =", 980, 800);

gr.drawString("αуб =", 1150, 800);

gr.drawString("δуб =", 1320, 800);

gr.setColor(Color.WHITE);

gr.fillRect(25, 550, 600, 250);

gr.fillRect(695, 100, 800, 500);

}

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

TextField TF = new TextField(value);

TF.setBounds(x, y, 120, 30);

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

add(TF);

return TF;

}

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

TextField TF = new TextField(value);

TF.setBounds(x, y, 110, 30);

TF.setEditable(false);

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

add(TF);

return TF;

}

public class Calculate implements ActionListener {

public void actionPerformed(ActionEvent e) {

Database.clear();

min = 0;

max = 0;

T = Double.parseDouble(TT.getText());

DT = Double.parseDouble(TDT.getText());

TD = Double.parseDouble(TTD.getText());

TF = Double.parseDouble(TTF.getText());

DD = Double.parseDouble(TDD.getText());

V = Double.parseDouble(TV.getText());

H = Double.parseDouble(TH.getText());

p = Double.parseDouble(Tp.getText());

a = Double.parseDouble(Ta.getText());

Cy0 = Double.parseDouble(TCy0.getText());

Cya = Double.parseDouble(TCya.getText());

Cyd = Double.parseDouble(TCyd.getText());

Cx = Double.parseDouble(TCx.getText());

mz = Double.parseDouble(Tmz.getText());

mzw = Double.parseDouble(Tmzw.getText());

mza\_ = Double.parseDouble(Tmza\_.getText());

mza = Double.parseDouble(Tmza.getText());

mzd = Double.parseDouble(Tmzd.getText());

C1 = -(mzw/I)\*S\*Math.pow(b, 2)\*(p\*V/2);

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

C2 = -(mza/I)\*S\*b\*(p\*Math.pow(V, 2)/2);

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

C3 = -(mzd/I)\*S\*b\*(p\*Math.pow(V, 2)/2);

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

C4 = ((Cya+Cx)\*g/G)\*S\*(p\*V/2);

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

C5 = -(mza\_/I)\*S\*Math.pow(b, 2)\*(p\*V/2);

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

C9 = (Cyd\*g/G)\*S\*(p\*V/2);

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

C16 = V/57.3/g;

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

Cy = 2\*G/S/p/Math.pow(V, 2);

TCy.setText(Double.toString(Cy));

Ca = 57.3\*(Cy-Cy0)/Cya;

TCa.setText(Double.toString(Ca));

Cd = -57.3\*(mz+(mza\*Ca/57.3)+Cy\*(x-0.24))/mzd;

TCd.setText(Double.toString(Cd));

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

while(T<=TF+0.01) {

int i = (int)(T/DT);

SAU(i);

DIN(i);

EILER(i);

if(T >= TD)

TD += DD;

T += DT;

if (Database.get(i).NY > max)

max = Database.get(i).NY;

else if (Database.get(i).NY < min)

min = Database.get(i).NY;

}

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

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

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

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

Table[i][3].setText(Double.toString(Database.get((i+1)\*(int)(DD/DT)).Y[0]));

Table[i][4].setText(Double.toString(Database.get((i+1)\*(int)(DD/DT)).Y[3]));

}

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

SB.setValue(1);

SB.addAdjustmentListener(new TableScroll());

CreateGraphic();

}

public void DIN(int i) {

Database.get(i).X[0] = Database.get(i).Y[1];

Database.get(i).X[2] = C4\*Database.get(i).Y[3] + C9\*Database.get(i).DV;

Database.get(i).X[3] = Database.get(i).X[0] - Database.get(i).X[2];

Database.get(i).X[1] = -C1\*Database.get(i).Y[1]-C2\*Database.get(i).Y[3]-C5\*Database.get(i).X[3]-C3\*Database.get(i).DV;

Database.get(i).NY = C16\*Database.get(i).X[2];

}

public void SAU(int i) {

double dvs, dvd = 0;

switch (switcher) {

case 1:

dvd = 0;

break;

case 2:

dvd = kwz\*Database.get(i).Y[1];

break;

case 3:

dvd = kwz\*Database.get(i).Y[1] - Database.get(i).Y[4]/Twz;

}

Database.get(i).X[4] = dvd;

dvs = Ks\*xv;

Database.get(i).DV = dvd + dvs;

}

public void EILER(int i) {

double[] YN = new double[5];

for(int j = 0; j < 5; 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]));

}

public void CreateGraphic() {

int bub;

int bX, lX = 695;

int bY, lY = (int)map((Database.get(0).NY), min, max, 600, 100);;

max += 0.005;

graph.setColor(Color.lightGray);

graph.fillRect(651, 70, 50, 530);

graph.setColor(Color.white);

graph.fillRect(695, 100, 800, 500);

graph.setColor(Color.black);

graph.fillRect(695, 100, 2, 500);

bub = (int)map(0, min, max, 600, 100);

graph.fillRect(695, bub, 800, 3);

graph.setFont(new Font("TimesRoman", Font.PLAIN, 40));

graph.drawString("^", 686, 125);

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

graph.drawString(">", 1480, bub+12);

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

for (double i = Math.ceil(min\*100)/100; i < Math.floor(max\*100)/100; i+=0.01) {

bub = (int)map(i, min, max, 600, 100);

graph.drawString(Double.toString(((double)Math.round(i\*100))/100), 650, bub-7);

graph.drawLine(690, bub, 1495, bub);

}

for (double i = DD\*2; i < TF; i += DD\*2) {

bub = (int)map(i, 0, TF, 690, 1495);

graph.drawString(Double.toString(((double)Math.round(i\*10))/10), bub-15, (int)map(0, min, max, 600, 100)+25);

graph.drawLine(bub, 100, bub, 600);

}

graph.setColor(Color.red);

for(double i = 2\*DT; i < TF+0.01; i+=DT) {

bX = (int)map(i, 0, TF, 695, 1495);

bY = (int)map((Database.get((int)(i/DT)).NY), min, max, 600, 100);

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;

}

int j = Database.size();

graph.setColor(Color.blue);

bub = (int)map(Database.get(j-10).NY, min, max, 600, 100);

graph.drawLine(695, bub, 1495, bub);

graph.drawString(Double.toString(((double)Math.round(Database.get(j-10).NY\*1000))/1000), 720, bub-5);

}

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(Database.get((i)\*(int)(DD/DT)).DV));

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

Table[i-bub][3].setText(Double.toString(Database.get((i)\*(int)(DD/DT)).Y[0]));

Table[i-bub][4].setText(Double.toString(Database.get((i)\*(int)(DD/DT)).Y[3]));

}

}

}

public static void main(String[] args) {

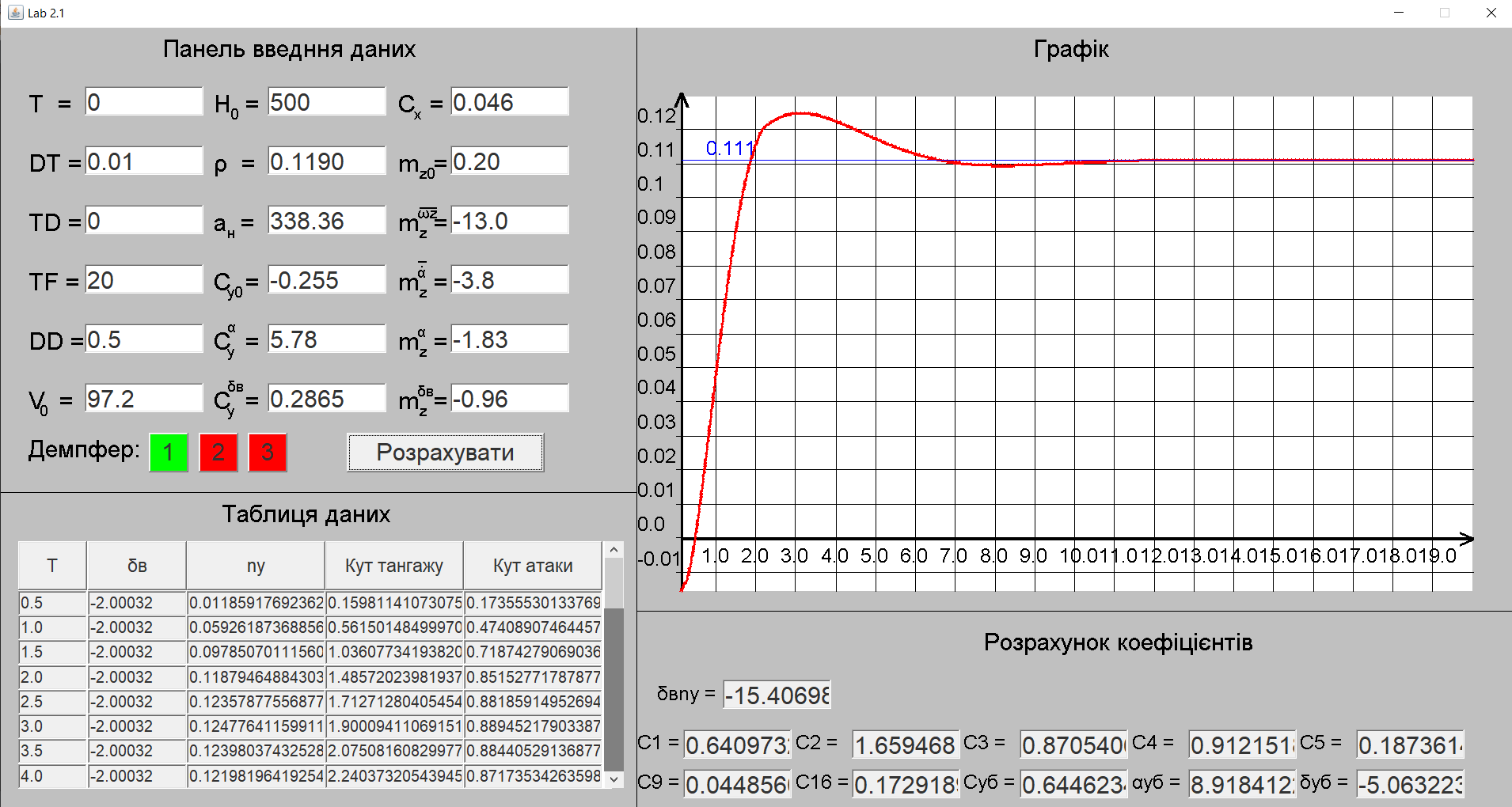
new MainProgram("Lab 2.1");

}

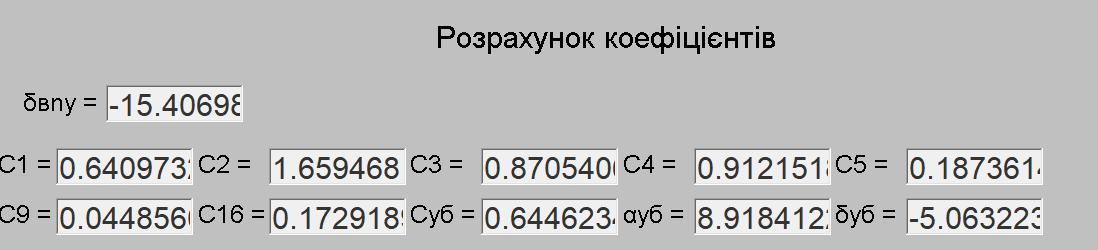
}

**Результати**

**Загальний вигляд програми**



**Чисельні результати досліджень**

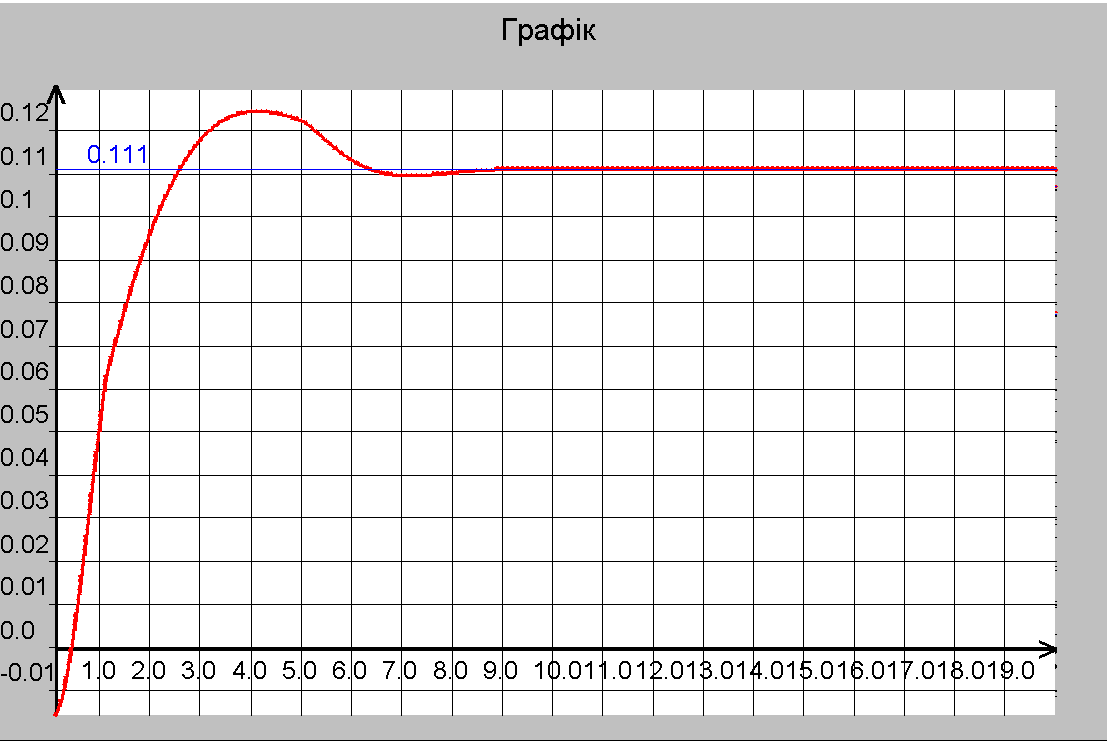


**Результати графічних досліджень**

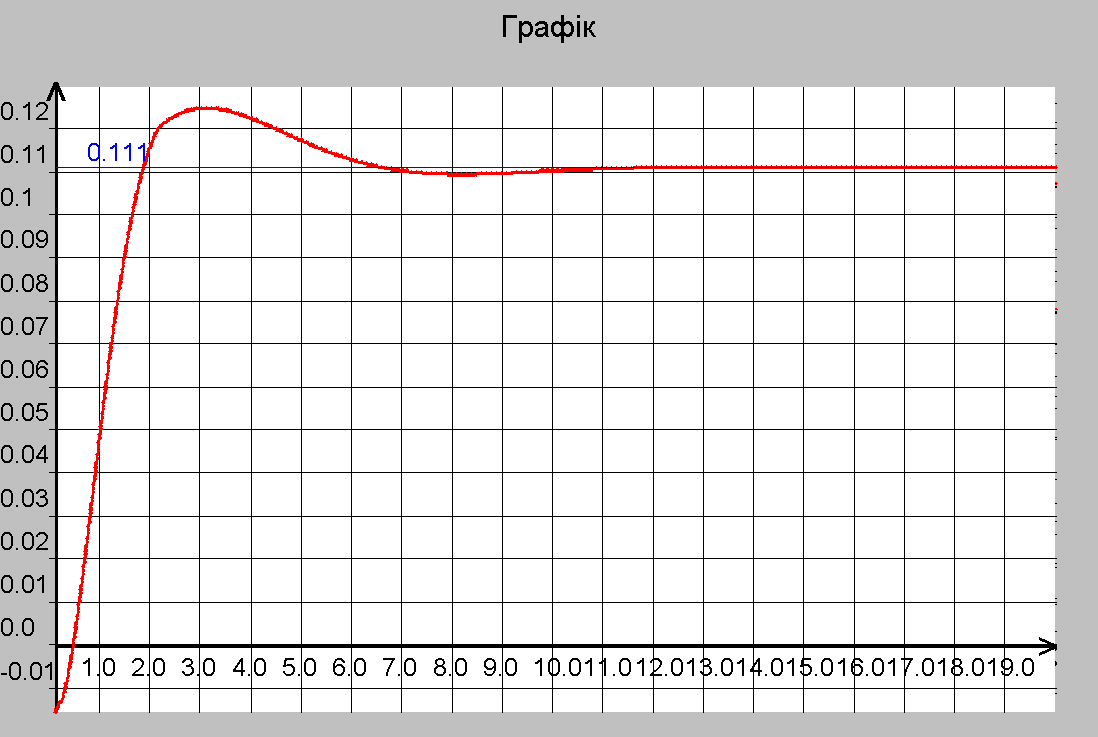
Демпфер тангажу вимкнений:

крок чисельного інтегрування диференціальних рівнянь:

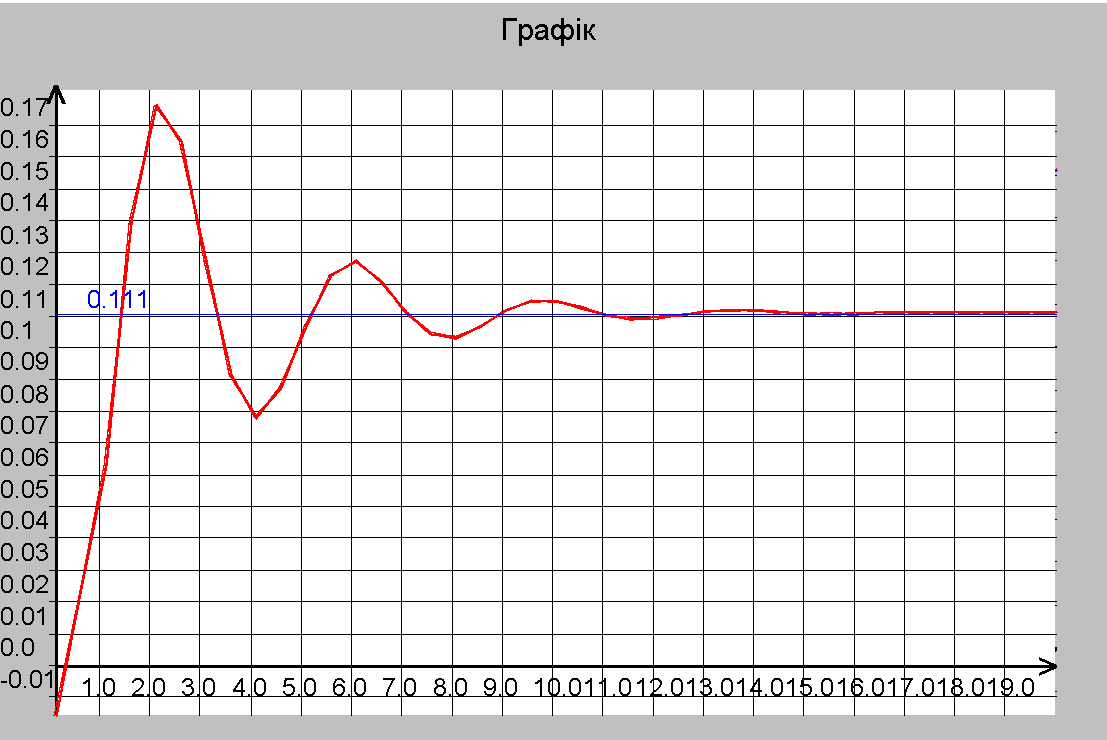
0,001 с,



0,01 с

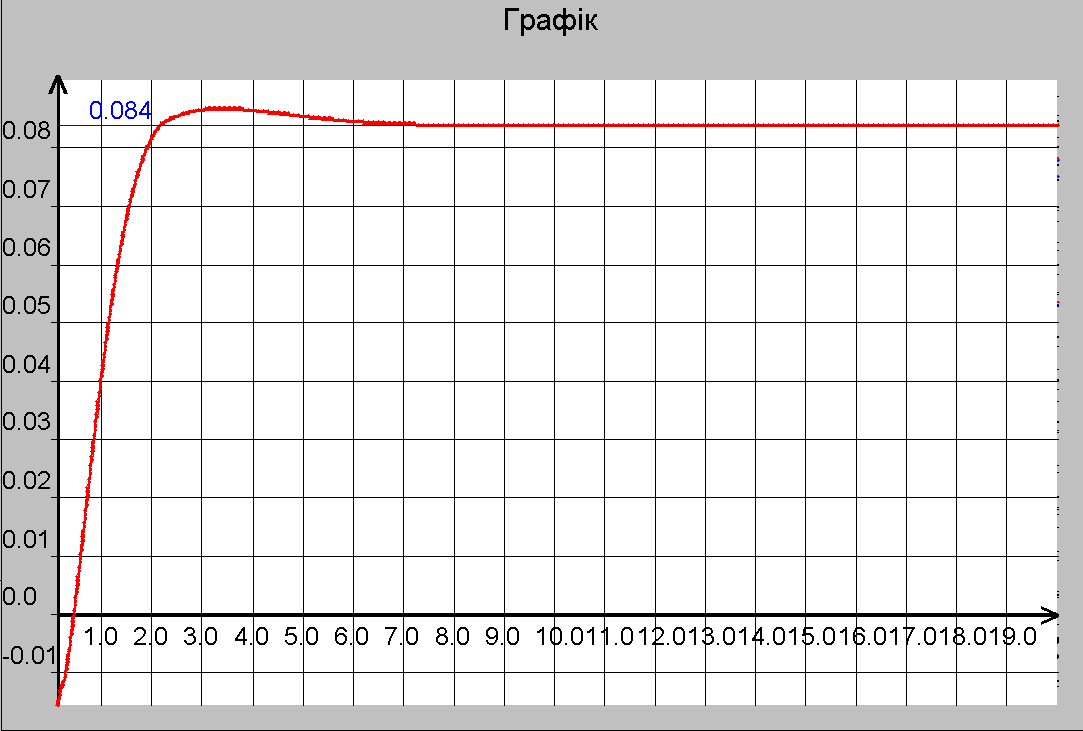


0,5 с



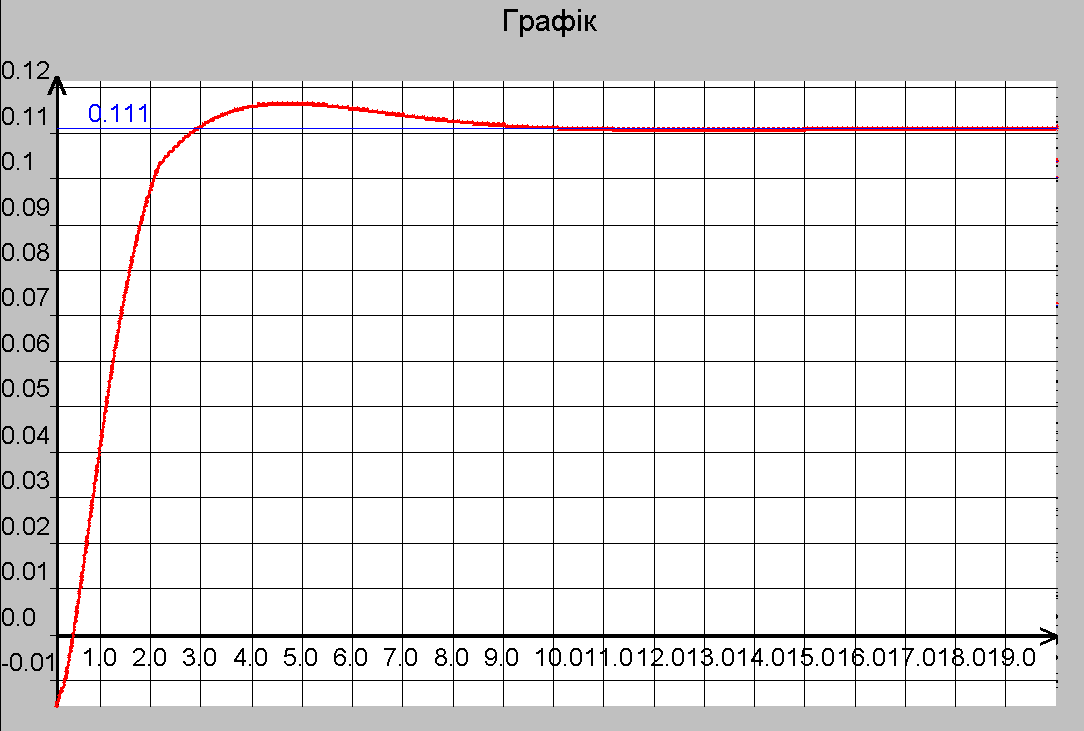
Демпфер тангажу увімкнений і працює із першим законом управління:

Крок інтегрування диференціальних рівнянь – 0.01с



Демпфер тангажу увімкнений і працює із другим законом управління:

Крок інтегрування диференціальних рівнянь – 0.01с



**Статичні та динамічні характеристики бічної керованості літака**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| .  од. | ,  град | , град. | | *t*пп*α* , с | | *Тα* , с | | , град/од.пер. | | |
| Аналіт. розр. | Реальн.  об’єкт | Мо-дель | Реальн. об’єкт | Мо-дель | Реальн. об’єкт | Аналіт розр. | Мо-дель | Реальн. об’єкт |
| 0.64 | 8.91 | -5.06 | -8 | 6.4 | 3-6 | 6.5 | 4-6 | -15.4 | -18 | -17 |

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

За результатами графічних досліджень можна стверджувати, що система є стійка, оскільки графік вертикального перевантаження при наданні імпульсу прямують до певного числа, а коливання згасають. Але при зменшені кількості кроків інтегрування, зменшується і стійкість системи, при збільшені - збільшується.

**Висновок про ступінь динамічної подібності розробленої цифрової математичної моделі та реального об'єкта**

За результатами таблиці статичних та динамічних характеристик бічної керованості літака можна стверджувати, що система достатньо подібна, але має відхилення для більшості значень, які можуть мати значну різницю з реальним об’єктом.

**Висновок про вплив величини кроку інтегрування диференціальних рівнянь на точність та тривалість їх розв'язання на ПЕОМ**

За результатами графічних досліджень при зменшені кількості кроків інтегрування, зменшується стійкість системи, при збільшені – збільшується. Також система стає більш точною, але потребує більше реального часу на аналітичні розрахунки ПЕОМ.

**Висновок про вплив на динамічні та статичні характеристики повздовжньої керованості літака демпфера тангажу з різними законами управління**

За результатами графічних досліджень при виключеному демпфері система має найбільшу максимальну амплітуду коливань. При включеному демпфері першого закону система має найменшу максимальну амплітуду коливань, але руль висоти має менше відхилення. При включеному демпфері другого закону система має малу максимальну амплітуду, а руль висоти підлаштовується під вертикальне навантаження.

**Висновки до лабораторної роботи:** було вивчено динамічні та статичні характеристики стійкості та керованості короткоперіодичного руху літака. Здійснено моделювання короткоперіодичного руху літака на ПЕОМ і досліджено характеристики його стійкості та керованості.