НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп’ютерної та програмної інженерії

Кафедра комп'ютерних інформаційних технологій

**Лабораторна робота № 2.5**

**з навчальної дисципліни**

**"Математичні моделі динамічних систем"**

Тема: Дослідження характеристик стійкості та керованості короткоперіодичного руху літака в повздовжній площині з пілотом у контурі управління на ПЕОМ

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

Київ 2020

Дослідження характеристик стійкості та керованості короткоперіодичного руху літака в повздовжній площині з пілотом у контурі управління на ПЕОМ

**Мета лабораторної роботи**

1. Вивчити особливості функціонування пілота в контурі управління літаком як ланки замкненої системи «літак-пілот» та вплив його власних характеристик на статичні та динамічні характеристики цієї системи.

2. Вивчити особливості детермінованого та стохастичного моделювання дій пілота в контурі системи штурвального управління (СШУ) літаком.

3. Дослідити вплив параметрів моделі дій пілота в контурі СШУ літаком на характеристики стійкості та керованості його короткоперіодичного руху на прикладі режиму стабілізації заданого кута тангажу.

## **Стислі теоретичні відомості**

У режимі штурвального управління літаком пілот забезпечує витримування заданих параметрів польоту на основі інформації, що зберігається в його пам’яті або в пам’яті бортового цифрового обчислювального комплексу (концептуальної моделі польоту), а також поточної інформації, яка збирається за допомоги органів чуття та пілотажно-навігаційних пристроїв (інформаційна модель польоту).

Витримування заданих параметрів польоту літака пілот забезпечує шляхом порівняння поточних значень цих параметрів із заданими значеннями, визначення розбіжності, обробки іншої отриманої інформації, прийняття рішення і відхилення органів управління літаком.

Таким чином, в процесі управління літаком пілот стає ланкою замкненої системи «літак-пілот», для якої його власні динамічні та статичні характеристики мають велике значення.

Чисельні результати досліджень дій пілота в контурі СШУ літаком дали можливість виявити такі особливості та здатності:

1) Пропускати сигнали частотою до 3 Гц.

2) Слідкувати за сигналами певної частоти.

3) Реагувати тільки на сигнали, величина яких вище порогу сприйняття органами чуттів.

4) Дискретно реагувати не неперервні збурення.

5) Логічно фільтрувати інформацію.

6) Реагувати не лише на відхилення керованого параметра польоту від заданого значення, але й на похідну та інтеграл від нього.

7) Пристосовуватися (адаптуватися) через певний час до зміни умов чи ситуації.

Наведене дозволяє зробити висновок, що в загальному випадку пілот в контурі СШУ літаком з точки зору теорії управління становить собою нелінійну, нестаціонарну, адаптивну систему.

У процесі управління літаком пілот прагне «пристосуватись» до характеристик вхідних збурень і динамічних характеристик літака, намагаючись зробити замкнену систему «літак-пілот» стійкою, а помилки управління – мінімальними. Він саме прагне, оскільки при деяких характеристиках вхідних збурень і поганих динамічних характеристиках об’єкта управління можна і не впоратися з поставленим завданням з огляду на те, що динамічні характеристики пілота обмежені та істотно залежать від різноманітних чинників. Зокрема, людина реагує на подразники з деяким запізненням, її здібності з уведення форсування чи запізнювання обмежені.

«Налаштування» параметрів пілота на статичні і динамічні характеристики конкретного літака на різних режимах відбувається в процесі його навчання та тренування на авіаційних тренажерах і «доводиться» під час льотної експлуатації.

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 та довгоперіодичний рухи розвиваються одночасно, але на початку переважає викликаний порушенням рівноваги повздовжніх моментів короткоперіодичний рух, який закінчується протягом декількох секунд після припинення дії збурень. Після цього починають помітно змінюватись параметри довгоперіодичного руху, викликаного порушенням рівноваги сил, які діють на літак.

Роль цих двох рухів з погляду їх оцінки пілотом літака істотно відрізняється. Якщо до довгоперіодичного руху вимоги зводяться лише 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.

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. Оскільки часові та частотні характеристики перехідних процесів за кутом тангажу близькі до характеристик короткоперіодичного руху, то часто в якості керованого параметру короткоперіодичного руху використовується кут тангажу.



Математична модель дій пілота в контурі управління літаком

На практиці при проведенні досліджень використовують моделі дій пілота як у стохастичній (ймовірнісній) постановці, так і в детермінованій.

Стохастичні моделі є більш точними, але й більш складними, тому при проведенні спрощених досліджень частіше використовують моделі в детермінованій постановці в термінах передавальних функцій або диференціальних рівнянь.

Статистична обробка результатів чисельних досліджень при залученні в якості пілота в контурі управління літаком операторів різної кваліфікації та тренованості дозволила встановити, що при вхідному сигналі, який має вигляд розбіжності поточного і заданого значення керованого параметра польоту, в діапазоні частот його зміни до 1,5 Гц, діяльність пілота можна моделювати за допомоги спрощеної детермінованої моделі, яка має таку передавальну функцію:



Зазначена детермінована модель складається з трьох блоків, які мають відповідні передавальні функції:



Блок чистого запізнення характеризує латентний час реакції, необхідний для збудження рецепторів при появі вхідного сигналу , проходження сигналу збудження в мозковий центр, його опрацювання та інтерпретації і має таку передавальну функцію:



де: – сигнал на виході блока чистого запізнення; – латентний час реакції – час, який проходить з моменту появи світлового (акустичного або іншого) подразника до першої реакції людини на нього.



У випадку режиму стабілізації заданого кута тангажу латентний час реакції пілота за візуальним подразником лежить у діапазоні: (0,1÷0,3) с, а вхідний сигнал блока чистого запізнення має вигляд:



де: , – поточне та задане значення кута тангажу.



Блок чистого запізнення зазвичай моделюється як екстраполятор нульового рівня, на виході якого протягом латентного часу підтримується постійним попереднє значення вхідного сигналу.



Коригувальний блок перетворює вихідний сигнал блока чистого запізнення на команду нейро-м’язовій системі пілота щодо переміщення органа управління і характеризує здатність пілота до самоналаштування (адаптації), тобто до пристосування своїх динамічних характеристик до динамічних характеристик основної частини системи «літак-пілот» при зміні параметрів польоту та зовнішнього середовища в широкому діапазоні і має таку передавальну функцію:



де: коефіцієнт передачі пілота; – сталі часу.



Налаштування блока здійснюється шляхом зміни значень , . Коефіцієнт передачі пілота може змінюватися в широкому діапазоні: від 0 до декількох тисяч. Величини сталих часу лежать у діапазонах:



(0,5÷1,5) с, (0,5÷20,0) с.



Нейро-м’язовий блок характеризує інерційні можливості нейро-м’язової системи людини, її нездатність миттєво відпрацьовувати вихідний сигнал коригувального блока , перетворюючи його в переміщення штурвальної колонки , і має таку передавальну функцію:



Величина сталої часу нейро-м’язового блоку змінюється в діапазоні:

= (0,1÷0,3) с



Параметри детермінованої моделі дій пілота в контурі управління кутом тангажу ( передавальної функції ) мають такі номінальні значення:



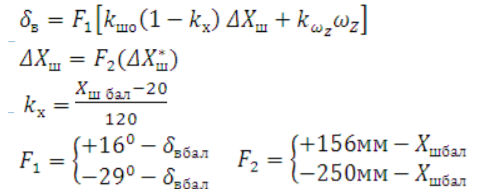
= 8 мм/град; = 0,15 с; 1,1 с; 1,0 с; = 0,15 с



Закон управління СШУ в каналі тангажу

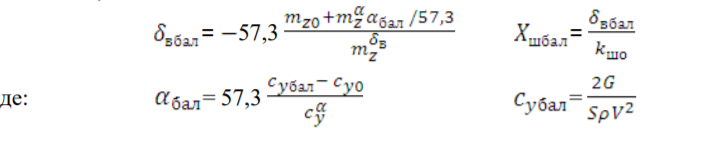
Спрощений закон управління системи штурвального управління (СШУ) в каналі тангажу літака має такий вид:



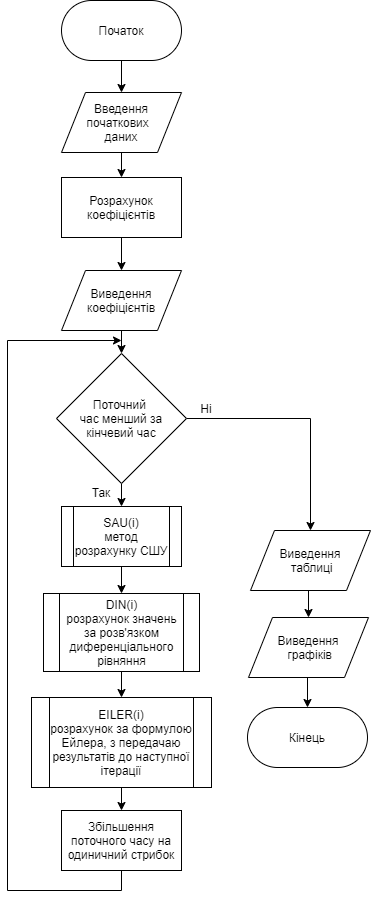


де:  = 0,112 град/мм – коефіцієнт передачі каналу руля висоти СШУ;

| | ≤ 0,4 – передаточне число керованості у повздовжньому каналі СШУ; – відхилення штурвальної колонки відносно балансованого положення з урахуванням його обмеження, мм; = 1,0 с – коефіцієнт передачі демпфера тангажу; F1 , F2 – функції обмеження відхилення, відповідно руля висоти та штурвальної колонки. Повний діапазон відхилення руля висоти відносно нульового положення становить (–290 ÷+160 ) при ході штурвальної колонки відносно нейтрального (нульового) положення (–250мм ÷ +156мм). Балансовані відхилення руля висоти та штурвальної колонки визначаються з таких співвідношень (конфігурація польотна, центрівка – 24% САХ):



**Алгоритм та програмна реалізація на Java**



import java.awt.\*;

import java.awt.event.\*;

import java.util.ArrayList;

import javax.swing.\*;

public class Lab extends JFrame{

public double S = 201.45, b = 5.285, G = 73000, xt = 0.24, Iz = 660000, V = 190, H = 6400, p = 0.0636, a = 314.34, g = 9.81,

Cy0 = -0.28, Cya = 5.9, Cyd = 0.2865, Cx = 0.033, mz0 = 0.22, mzwz = -13.4, mza\_ = -4, mza = -1.95, mzd = -0.92;

public double C1, C2, C3, C4, C5, C6, C9, C16, abal, Cybal, dbal, Xshbal;

public TextField TC1, TC2, TC3, TC4, TC5, TC6, TC9, TC16, Tabal, TCybal, Tdbal, TXshbal;

public double T = 0, DT = 0.01, TD = 0, TF = 20, DD = 0.5;

public Button TT, TXV, TDV, Tteta, TH, TNY;

public double KS = 0.112, KWZ = 1, TWZ = 0.7, dEta=0, Etaz=5, U1,U2, dXsh;

int j = 0;

public double Kp=8, tl=0.15, T1 = 1.1, T2=1, T3=0.15;

//public double Kp=1, tl=0.15, T1=1.1, T2=1, T3=0.15;

//public double Kp=20, tl=0.15, T1=1.1, T2=1, T3=0.15;

//public double Kp=8, tl=0.05, T1=1.1, T2=1, T3=0.15;

//public double Kp=8, tl=0.30, T1=1.1, T2=1, T3=0.15;

//public double Kp=8, tl=0.15, T1=1.1, T2=0.1, T3=0.15;

//public double Kp=8, tl=0.15, T1=1.1, T2=10, T3=0.15;

//public double Kp=8, tl=0.15, T1=1.1, T2=1, T3=0.05;

//public double Kp=8, tl=0.15, T1=1.1, T2=1, T3=0.30;

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 DV, NY;

public Data(double Y1, double Y2, double Y3, double Y4, double Y5, double Y6, double Y7, double Y8) {

Y[0] = Y1;

Y[1] = Y2;

Y[2] = Y3;

Y[3] = Y4;

Y[4] = Y5;

Y[5] = Y6;

Y[6] = Y7;

Y[7] = Y8;

}

}

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

public Lab(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("", 60, 50);

TC2 = CreatCTF("", 240, 50);

TC3 = CreatCTF("", 420, 50);

TC4 = CreatCTF("", 600, 50);

TC5 = CreatCTF("", 60, 80);

TC6 = CreatCTF("", 240, 80);

TC9 = CreatCTF("", 420, 80);

TC16 = CreatCTF("", 600, 80);

Tabal = CreatCTF("", 60, 110);

TCybal = CreatCTF("", 240, 110);

Tdbal = CreatCTF("", 420, 110);

TXshbal = CreatCTF("", 600, 110);

TT = CreateBT("T", 22, 150, 80, 40);

TXV = CreateBT("δХш", 102, 150, 120, 40);

TDV = CreateBT("δв", 222, 150, 120, 40);

Tteta = CreateBT("θ", 342, 150, 120, 40);

TH = CreateBT("H", 462, 150, 120, 40);

TNY = CreateBT("ny", 582, 150, 120, 40);

Table = new TextField[8][6];

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

for(int j = 0; j < 6; 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(22, 190+i\*25, 80, 25);

Table[i][1].setBounds(102, 190+i\*25, 120, 25);

Table[i][2].setBounds(222, 190+i\*25, 120, 25);

Table[i][3].setBounds(342, 190+i\*25, 120, 25);

Table[i][4].setBounds(462, 190+i\*25, 120, 25);

Table[i][5].setBounds(582, 190+i\*25, 120, 25);

}

SB = new Scrollbar(Scrollbar.VERTICAL);

SB.setBounds(700, 150, 20, 240);

SB.setBackground(Color.GRAY);

SB.setMinimum(1);

add(SB);

double m = G/g;

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

TCybal.setText(Double.toString(Cybal));

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

Tabal.setText(Double.toString(abal));

dbal = -57.3\*((mz0 + (mza\*abal/57.3) + Cybal\*(xt - 0.24))/mzd);

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

Xshbal = dbal/KS;

TXshbal.setText(Double.toString(Xshbal));

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+Cx)/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));

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

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

C16 = V/57.3/g;

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

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

while(T<=TF+DT) {

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

SAU(i);

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(Database.get((i+1)\*(int)(DD/DT)).Y[7]));

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

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[4]));

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

}

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

SB.setValue(1);

SB.addAdjustmentListener(new TableScroll());

CreateGraphicX(max[0], min[0], 55, 440, 710, 350, 1);

CreateGraphicX(max[1], min[1], 820, 60, 710, 350, 0);

CreateGraphicX(max[2], min[2], 820, 440, 710, 350, 4);

}

public void SAU(int i) {

double F21 = 156 - Xshbal;

double F22 = -250 - Xshbal;

if (Database.get(i).Y[7] > F21)

Database.get(i).Y[7] = F21;

if (Database.get(i).Y[7] < F22)

Database.get(i).Y[7] = F22;

double delta\_x = Database.get(i).Y[7];

double F11 = 16 - dbal;

double F12 = -29 - dbal;

double k\_x = (Xshbal - 20)/120;

Database.get(i).DV = 0.112 \* (1 - k\_x) \* delta\_x + 1 \* Database.get(i).Y[1];

if (Database.get(i).DV > F11)

Database.get(i).DV = F11;

if (Database.get(i).DV < F12)

Database.get(i).DV = F12;

}

public void DIN(int i) {

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

Database.get(i).X[1] = -C1\*Database.get(i).X[0]-C2\*Database.get(i).Y[3]-C5\*Database.get(i).X[3]-C3\*Database.get(i).DV;

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[4] = C6\*Database.get(i).Y[2];

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

j += 1;

if(j >= (int)(tl/DT){

dEta=Database.get(i).Y[0]-Etaz;

U1=dEta;

j = 0;

}

Database.get(i).X[5] = (Kp\*U1\*T1-Database.get(i).Y[5])/T2;

Database.get(i).X[6] = (U1\*Kp-Database.get(i).Y[6])/T2;

U2 = Database.get(i).X[5]+Database.get(i).Y[6];

Database.get(i).X[7] = (U2-Database.get(i).Y[7])/T3;

}

public void EILER(int i) {

double[] YN = new double[8];

for(int j = 0; j < 8; 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], YN[6], YN[7]));

}

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).Y[0] > max[1])

max[1] = Database.get(j).Y[0];

if(Database.get(j).Y[0] < min[1])

min[1] = Database.get(j).Y[0];

if(Database.get(j).Y[4] > max[2])

max[2] = Database.get(j).Y[4];

if(Database.get(j).Y[4] < min[2])

min[2] = Database.get(j).Y[4];

}

public void paint (Graphics gr) {

gr.setColor(Color.lightGray);

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

gr.setColor(Color.BLACK);

gr.drawLine(0, 430, 765, 430);

gr.drawLine(765, 0, 765, 430);

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

gr.drawString("Розрахунок коефіцієнтів", 260, 60);

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

gr.drawString("C1 =", 20, 100);

gr.drawString("C2 =", 200, 100);

gr.drawString("C3 =", 380, 100);

gr.drawString("C4 =", 560, 100);

gr.drawString("C5 =", 20, 130);

gr.drawString("C6 =", 200, 130);

gr.drawString("C9 =", 380, 130);

gr.drawString("C16 =", 550, 130);

gr.drawString("αбал =", 20, 160);

gr.drawString("Cyбал =", 190, 160);

gr.drawString("δбал =", 370, 160);

gr.drawString("Xшбал =", 540, 160);

gr.setColor(Color.white);

gr.fillRect(30, 180, 680, 240);

}

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);

}

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 > 50) {

delta = 20;

start = Math.ceil(min);

end = Math.floor(max);

}

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 = 1;

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/10; i <= TF-TF/10; i += TF/10) {

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 0:

graph.drawString("θ", x-30, bub+10);

break;

case 4:

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

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(Database.get((i)\*(int)(DD/DT)).Y[7]));

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

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[4]));

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

}

}

}

public static void main(String[] args) {

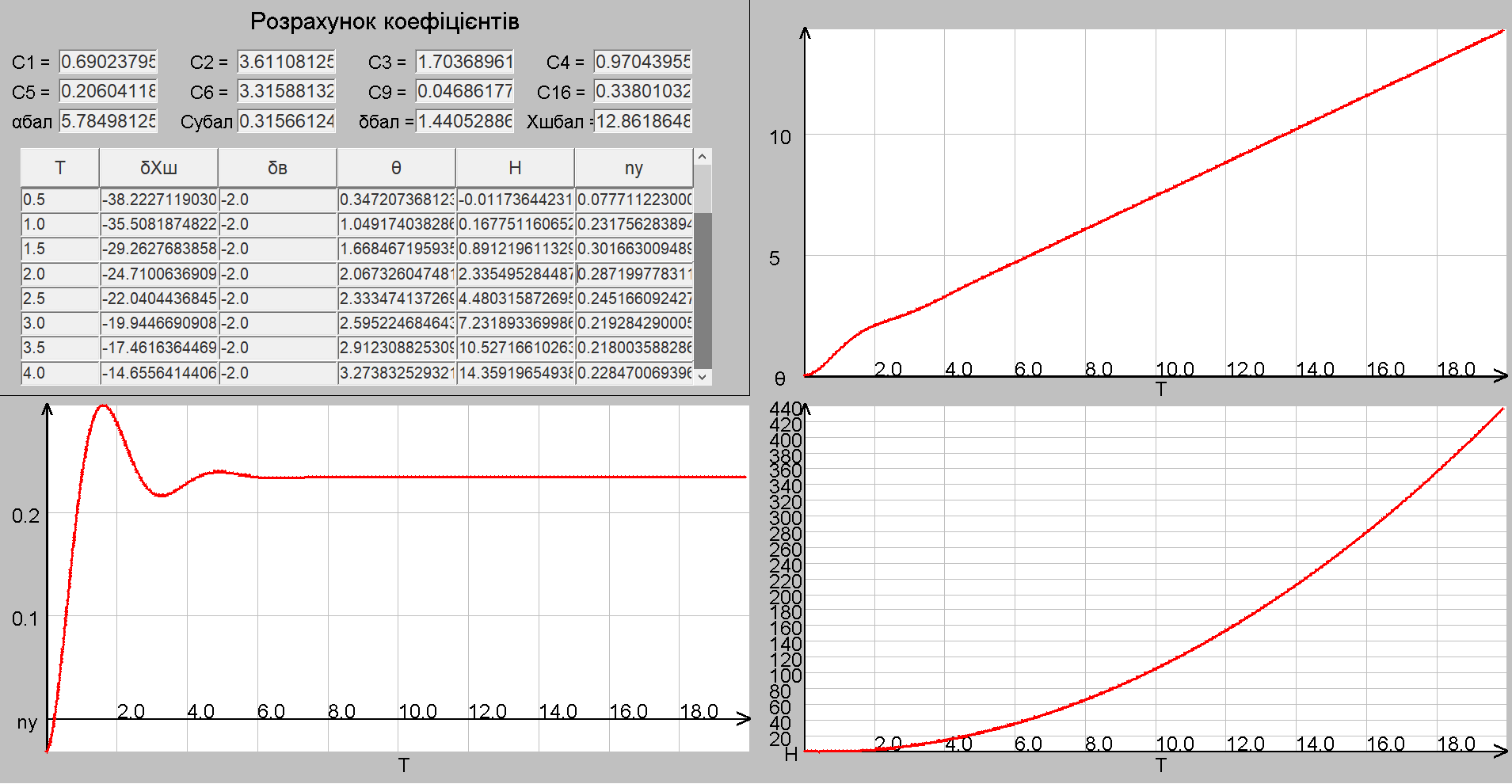
new Lab("Lab 2.3");

}

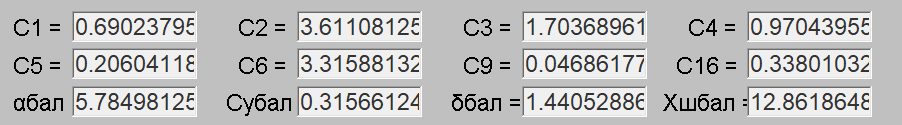
}

**Результати**

**Загальний вигляд програми**



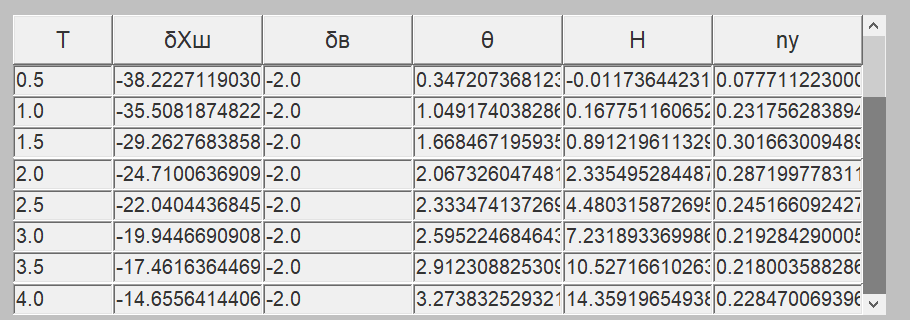
**Чисельні результати досліджень**

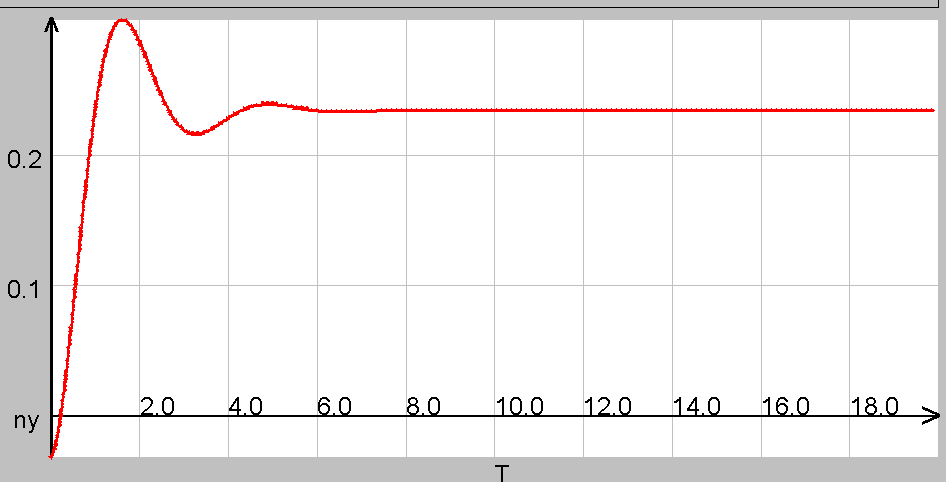


**Результати графічних досліджень**

Вертикальне перевантаження при «дачі» штурвальної колонки на величину = –17,86 мм







**Статичні та динамічні характеристики повздовжньої керованості літака**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ,  град. | ,  од. | , град. | | *Тα* , с | | , град/од.пер. | | |
| Реальн.  об’єкт | Аналіт. розрах. | Реальн. об’єкт | Модель | Реальн. об’єкт | Аналіт. розрах. | Модель |
| 5.78 | 0.31 | 1.75 | 1.44 | 4-6 | 3.5 | -11 | -7.85 | -8.55 |

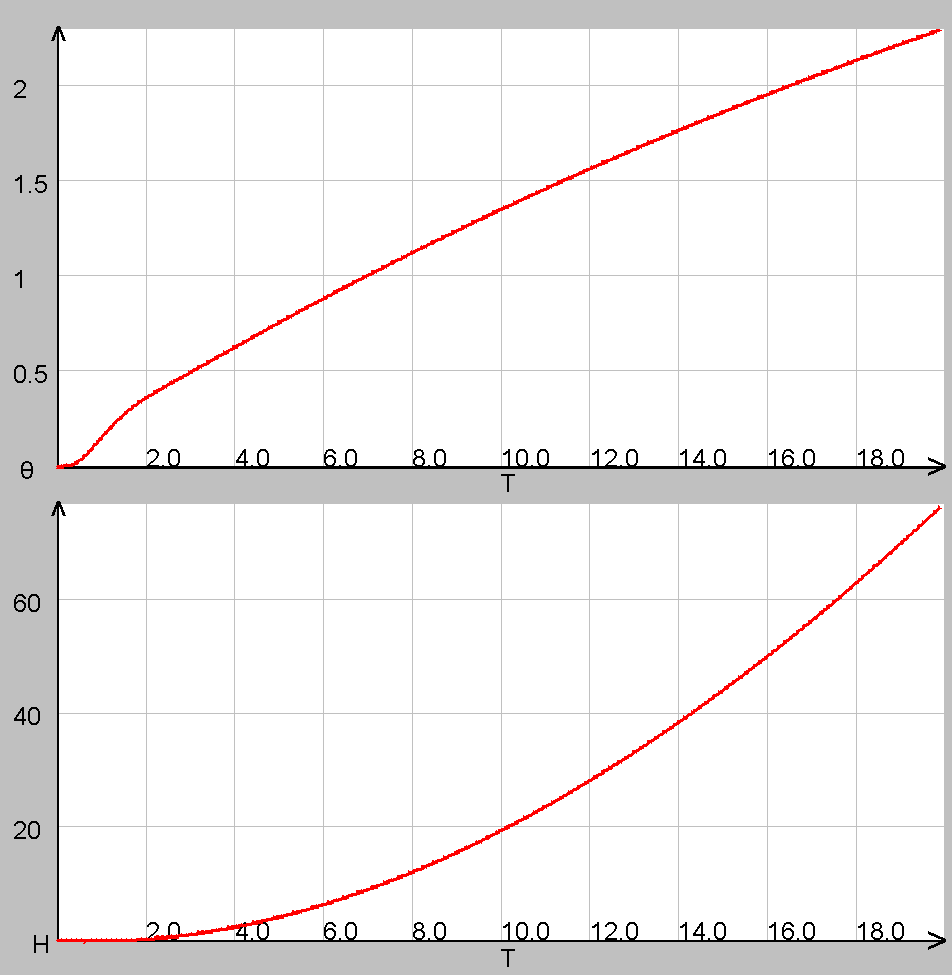
Дослідження впливу на якість стабілізації заданого кута тангажу = 50 параметрів моделі дій пілота:



1. Величини коефіцієнта передачі пілота у коригувальному блоці:

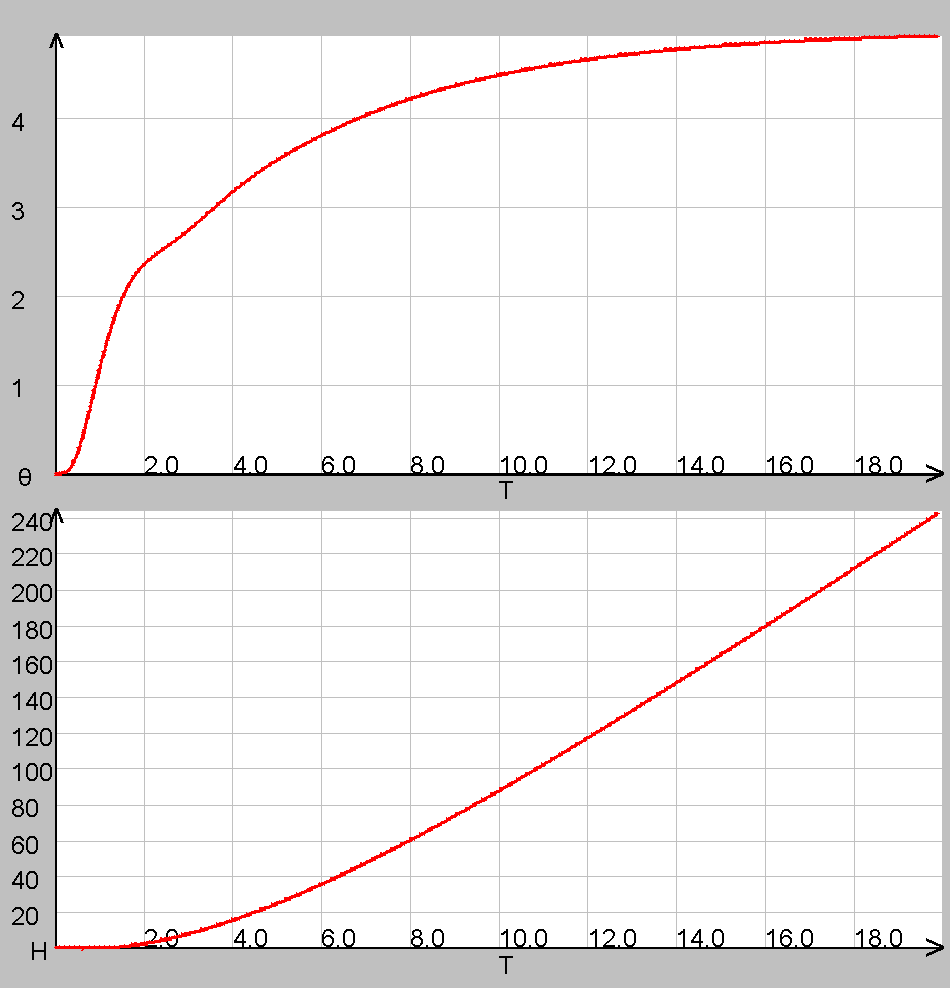
= 1,0 мм/град





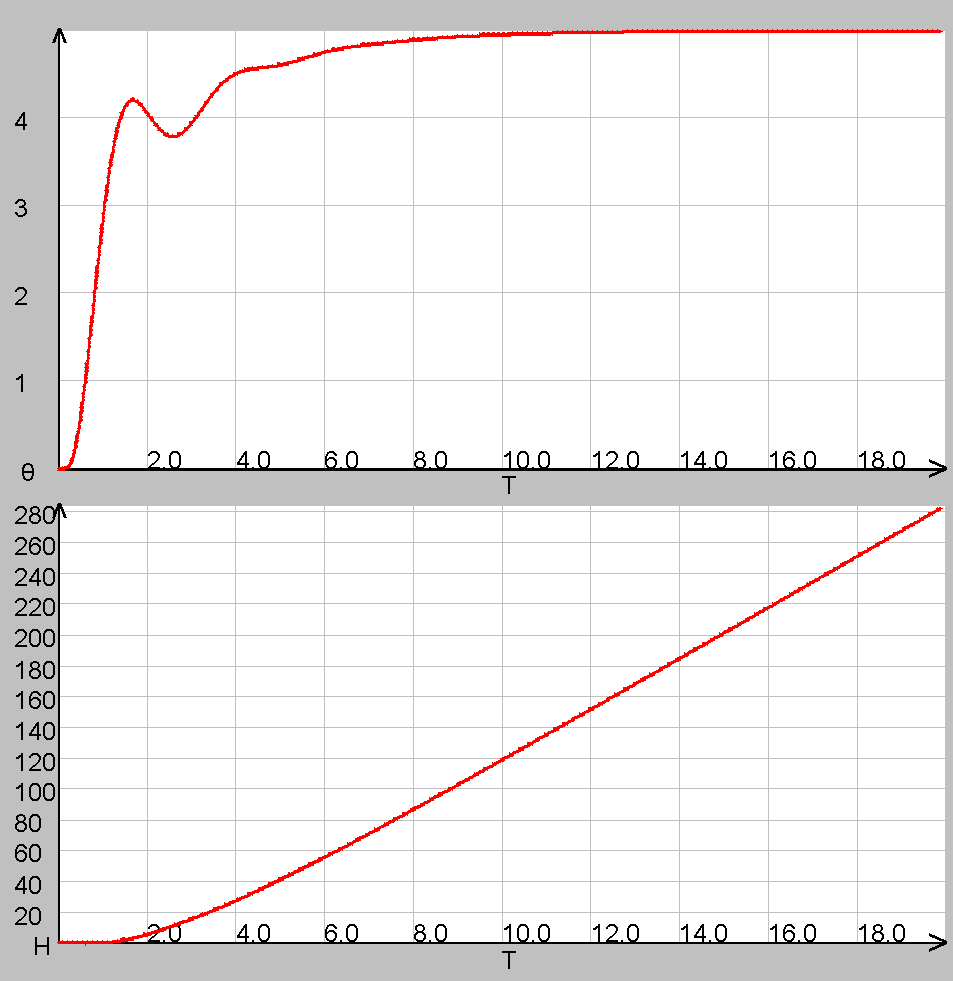
= 8,0 мм/град





= 20,0 мм/град

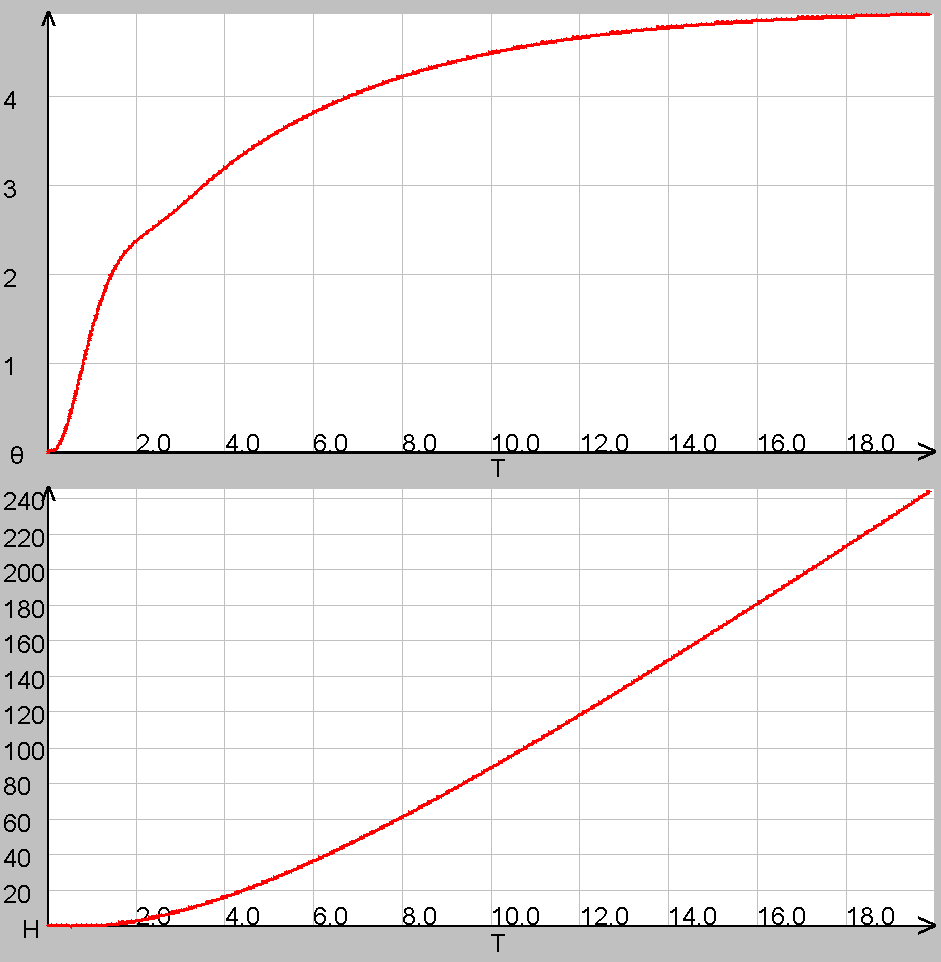




1. Величини латентного часу реакції у блоці чистого запізнення:

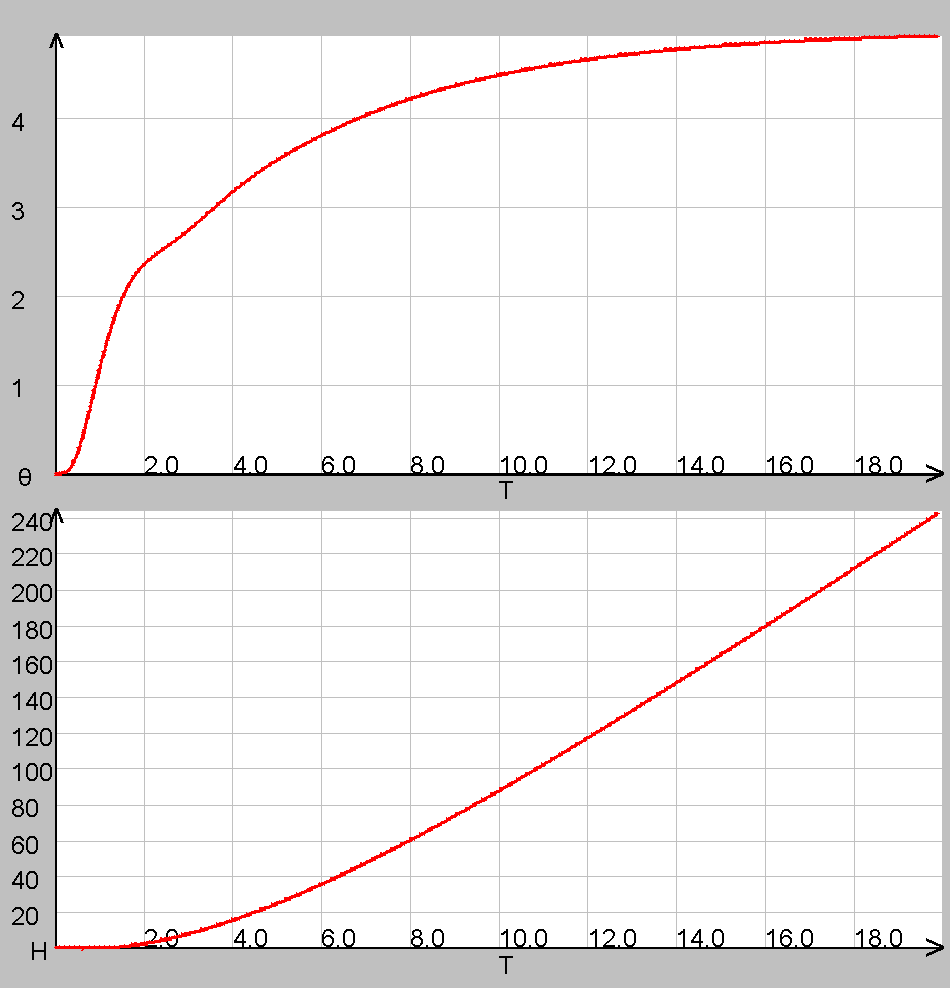
= 0,05 с





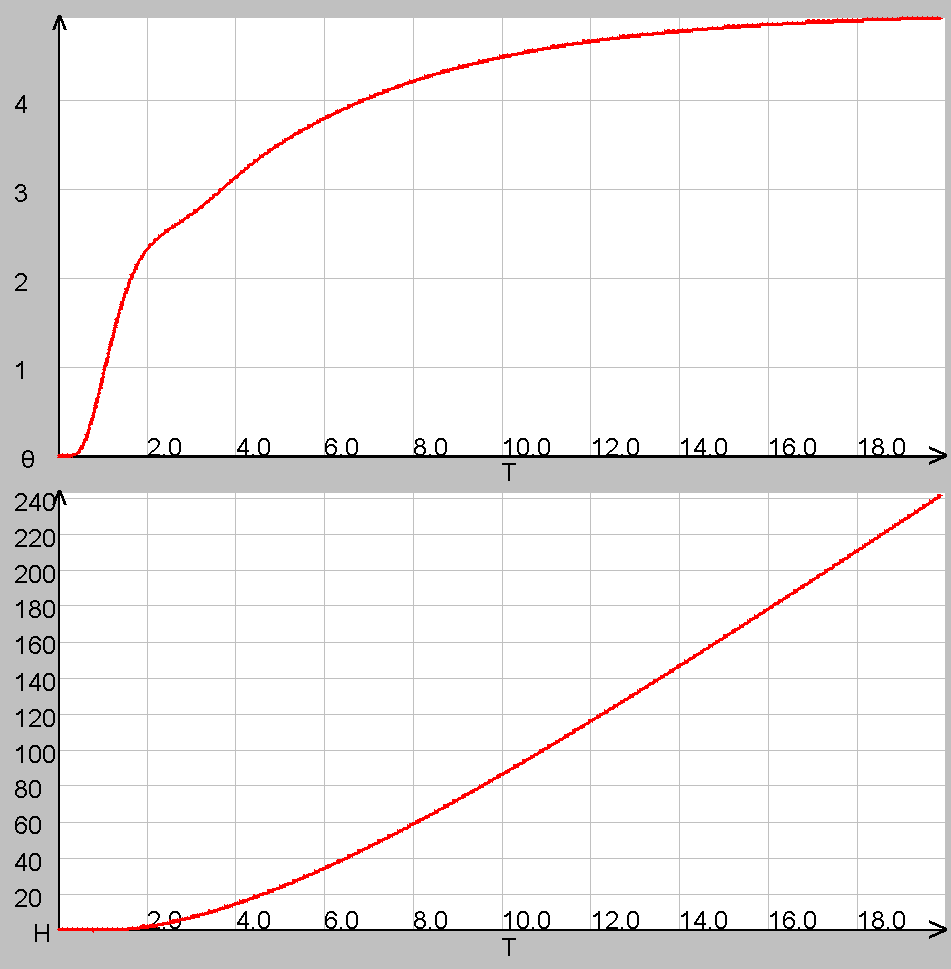
= 0,15 с





= 0,30 с



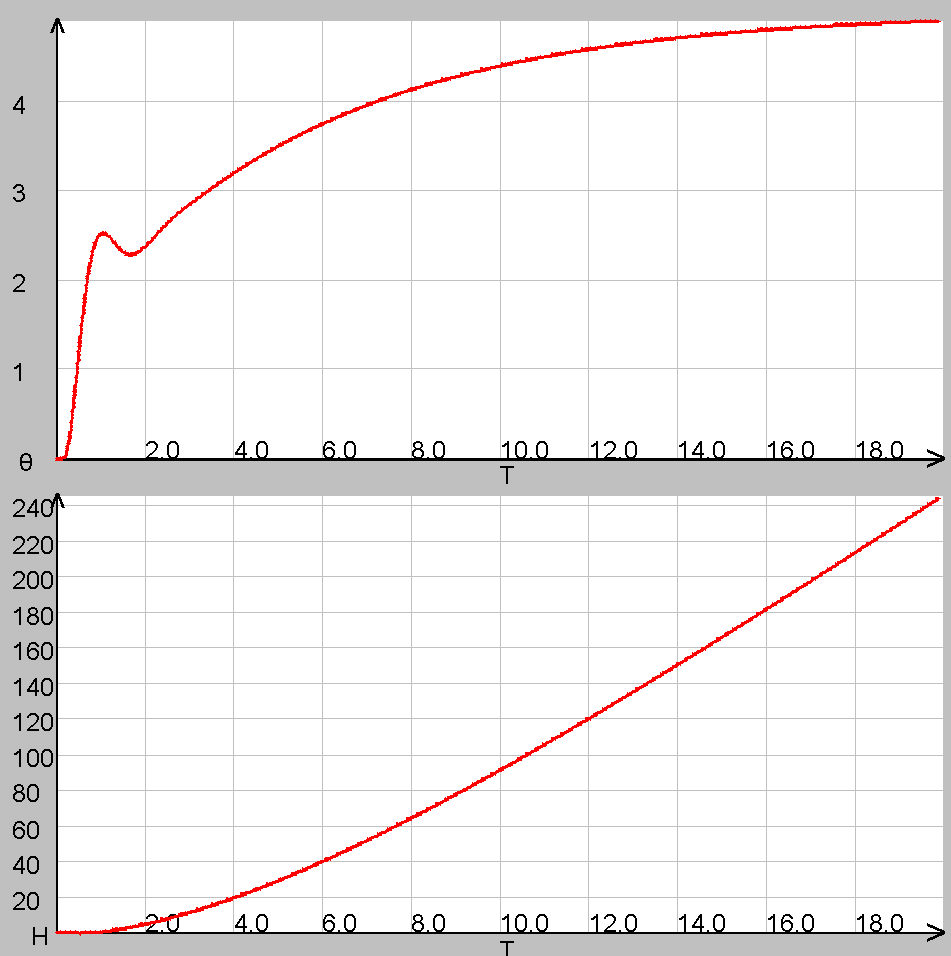


1. Величини сталої часу у коригувальному блоці:



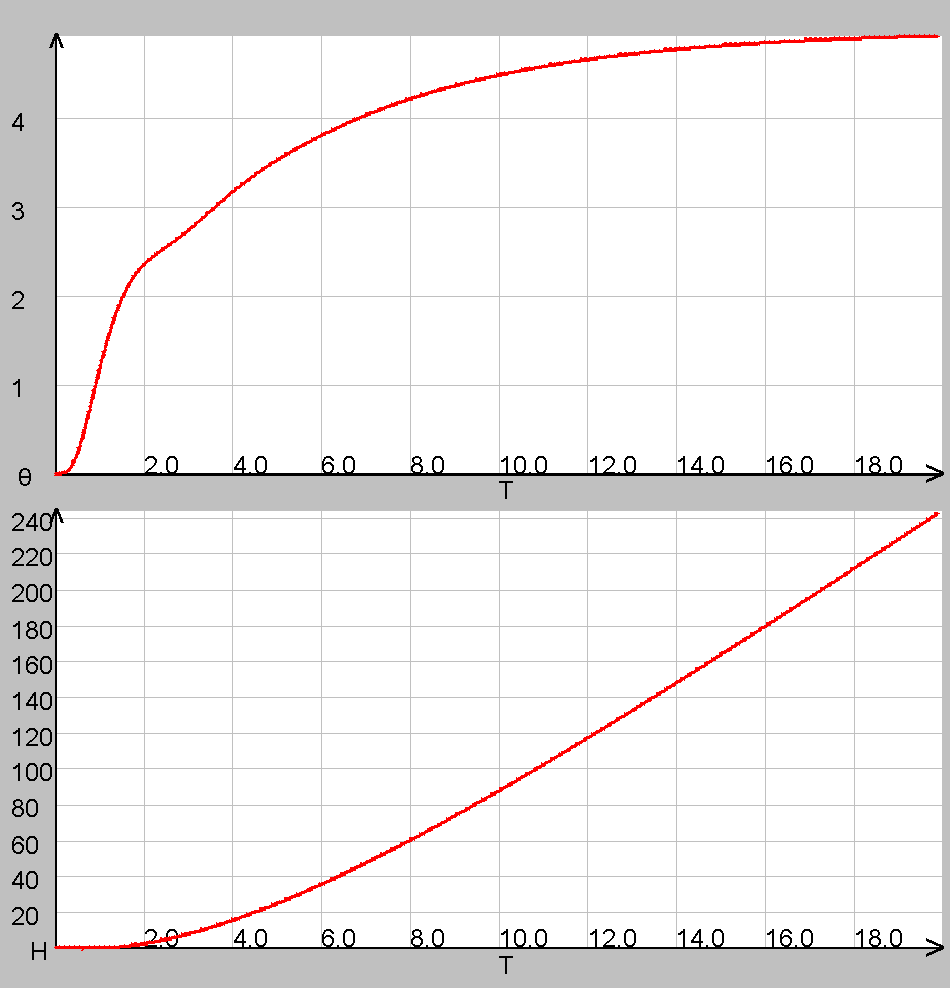
= 0,1 с





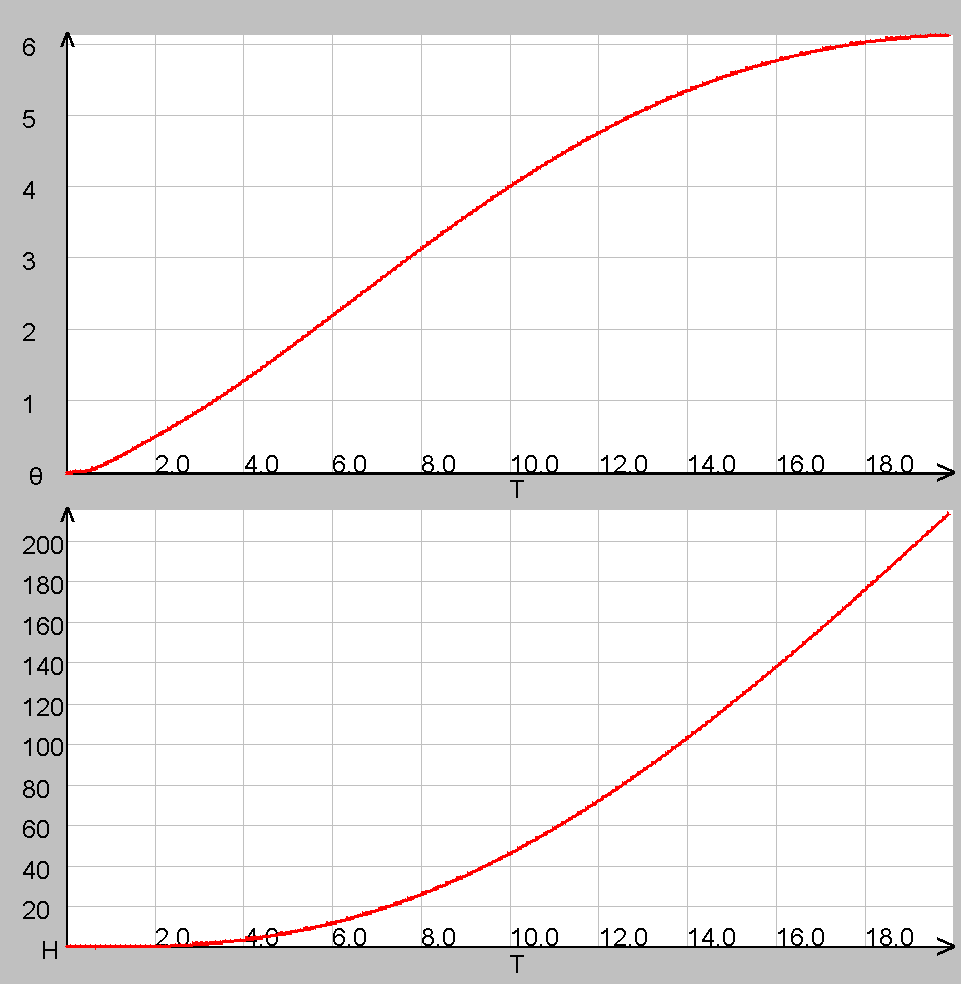
= 1,0 с





= 10,0 с



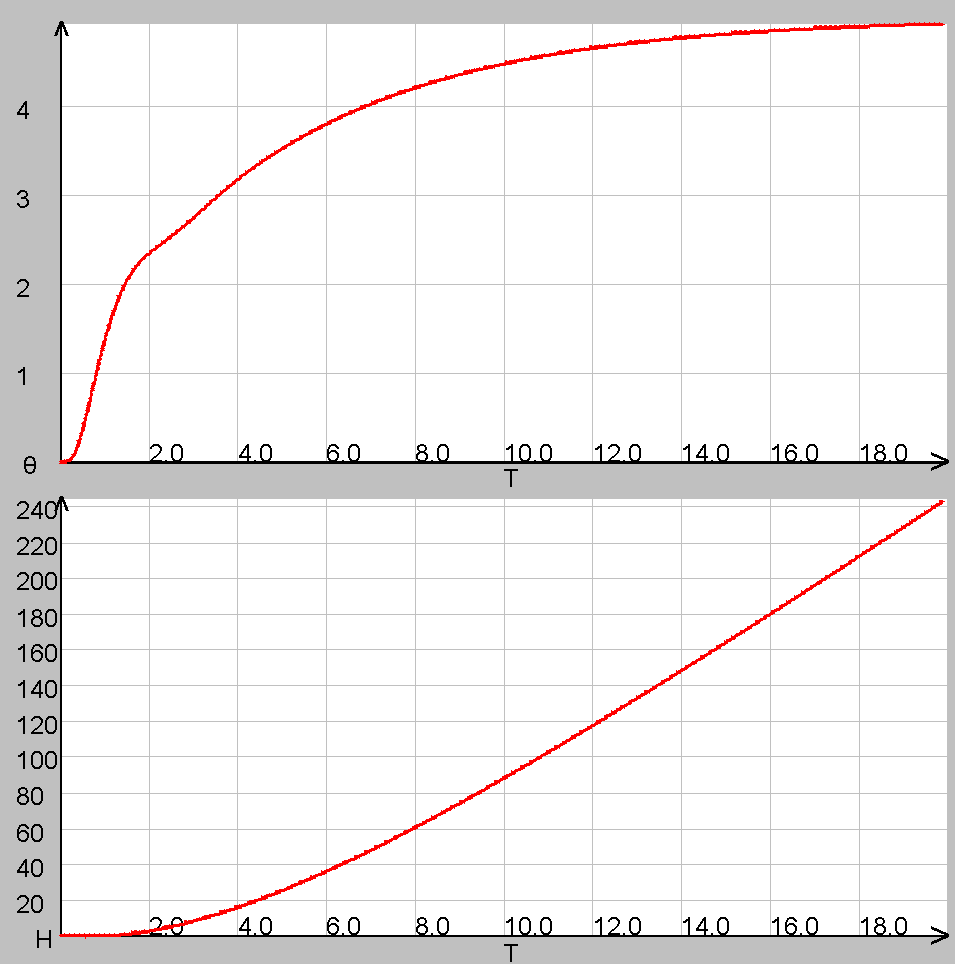


Величини сталої часу у нейро-м’язовому блоці:



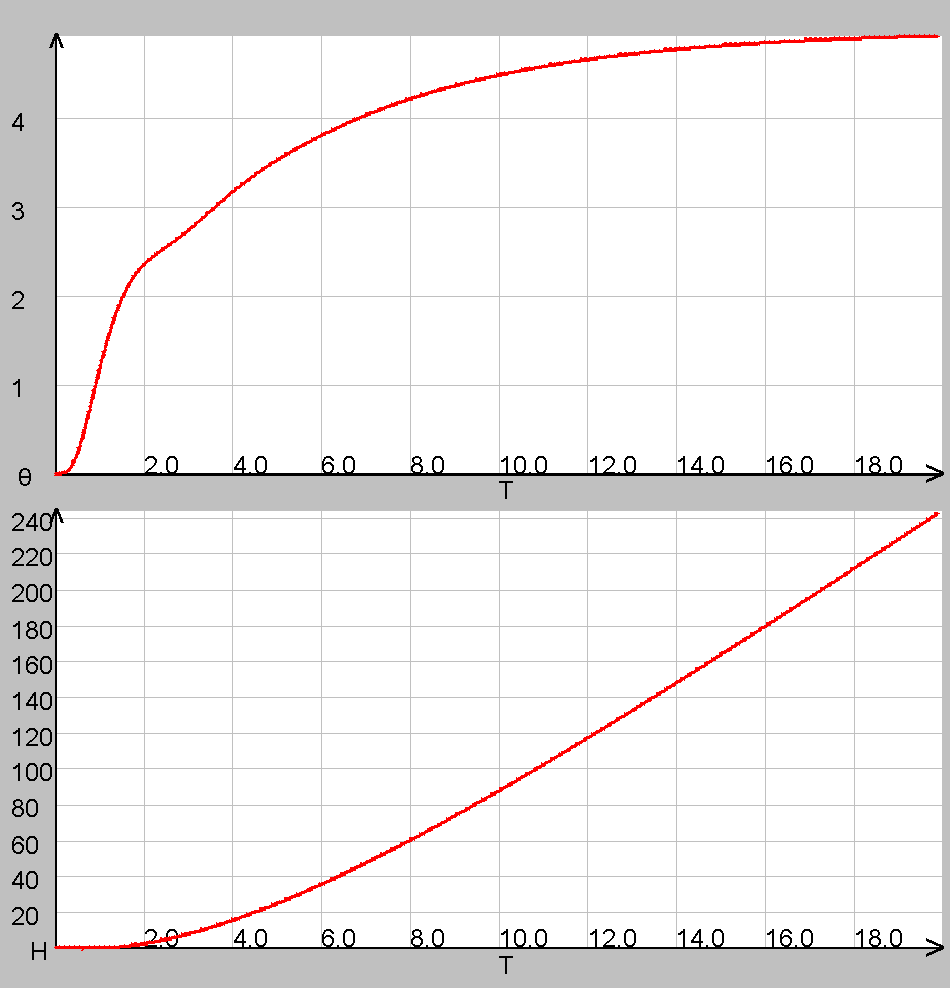
= 0,05 с





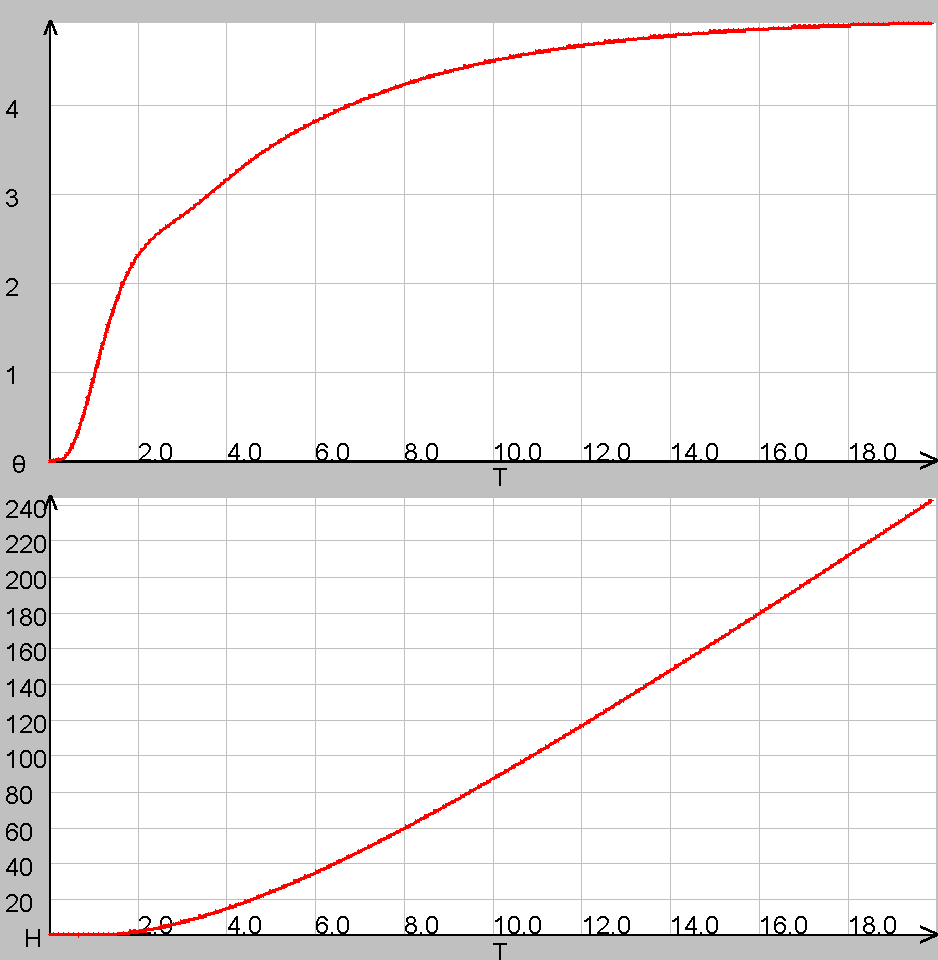
= 0,15 с





= 0,30 с





**Висновок про ступінь динамічної подібності розробленої цифрової математичної моделі літака і реального об'єкта.**

За результатами табличних досліджень можна стверджувати, що система є достатньо подібною, але та мають відхилення, особливо . Також *Тα* має відхилення.



**Висновки про вплив на якість стабілізації заданого кута тангажу параметрів моделі дій пілота**

За результатами графічних досліджень при зменшені система стає більш стійкою, але потребує більше часу на зміну значення, при збільшені – зменшується стійкість, але зменшується час перехідного процесу. При зміні стійкість системи має незначні зміні. При зменшені система стає менш стійкою, при збільшені – більш стійкою. При зміні стійкість системи має незначні зміні.



**Висновки до лабораторної роботи:** вивчено особливості функціонування пілота в контурі управління літаком як ланки замкненої системи «літак-пілот» та вплив його власних характеристик на статичні та динамічні характеристики цієї системи. Вивчено особливості детермінованого та стохастичного моделювання дій пілота в контурі системи штурвального управління (СШУ) літаком. Досліджено вплив параметрів моделі дій пілота в контурі СШУ літаком на характеристики стійкості та керованості його короткоперіодичного руху на прикладі режиму стабілізації заданого кута тангажу.