НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп’ютерної та програмної інженерії

Кафедра комп'ютерних інформаційних технологій

**Лабораторна робота № 4.3**

**з навчальної дисципліни**

**"Математичні моделі динамічних систем"**

Тема: ДОСЛІДЖЕННЯ РЕЖИМУ АВТОМАТИЧНОГО ЗАХОДУ НА ПОСАДКУ ЛІТАКА ЗА СИГНАЛАМИ КУРСОВОГО РАДІОМАЯКА НА ПЕОМ

Виконавець: студент групи УС-311Дзиговський В.І.

Київ 2020

**Мета лабораторної роботи:**

1. Вивчити закони управління та функціонування системи автоматичного управління (САУ) літака, наземних та бортових радіотехнічних систем у режимі автоматичного заходу на посадку (АЗП).

2. Дослідити динаміку польоту літака в режимі АЗП за сигналами курсового радіомаяка (КРМ) на ПЕОМ.

**Стислі теоретичні відомості**

Швидкоплинність етапів заходу на посадку та посадки літака і неминучий дефіцит часу на прийняття рішення підвищує нервово-психологічне навантаження пілота. Тому застосування систем автоматичного управління літаком не тільки забезпечує виконання польотів незалежно від погодних умов, але й підвищує ступінь безпеки польотів, тому що автоматизація зменшує ймовірність виникнення суб’єктивних помилок пілота під час прийняття рішення і виконання тих чи інших еволюцій літака.

Етап заходу на посадку складається з двох ділянок. На першій ділянці здійснюється передпосадковий маневр за прямокутним маршрутом типу «коробочка», в процесі якого літак робить у загальному випадку чотири розвороти. Друга ділянка починається з моменту входу літака в глісаду і закінчується на висоті (15÷20) м.

Траєкторія заходу літака на посадку в бічній площині задається курсовим радіомаяком (рис. 1). Діаграма випромінювання КРМ має вигляд двох пелюсток. Причому, при знаходженні літака в зоні лівої пелюстки, по ходу руху до злітно-посадкової смуги (ЗПС), на виході курсового радіоприймача (КРП) на борту літака виникає постійний електричний струм, від’ємний за знаком. При знаходженні літака в зоні правої пелюстки – електричний струм, додатній за знаком.

В результаті виконання четвертого розвороту літак повинен виходити в площину посадкового курсу – вертикальну площину, яка проходить через вісь злітно-посадкової смуги та її продовження.

Площина посадкового курсу утворюється як вертикальна зона дотику лівої і правої пелюсток діаграми випромінювання КРМ, і називається його рівносигнальною зоною.

Ця зона отримала таку свою назву через те, що при знаходженні в ній літака електричний струм на виході КРП дорівнює нулю, тому що сигнали лівої і правої пелюсток компенсують один одного.

Бортовий курсовий радіоприймач видає в САУ сигнал, пропорційний кутовому відхиленню літака від рівносигнальної зони КРМ.

САУ на підставі отриманої інформації про кутове положення літака відносно КРМ і реалізованих у ній законів управління здійснює автоматичний вихід літака у площину посадкового курсу (рівносигнальної зони КРМ) і стабілізацію літака на ній.

Маневр заходу на посадку літака за сигналами КРМ може починатися з різними початковими значеннями його бічного відхилення від осі ЗПС *Z0* і кута підходу до цієї осі Δ :

Δ = г – ЗПС

де: г – поточний гіроскопічний курс літака, ЗПС  – посадковий курс.

При цьому можливі такі варіанти маневру заходу на посадку:

* маневр заходу з четвертого розвороту, яким закінчується виконання стандартного маневру типу «коробочка», при якому Δ0 = 900;
* маневр заходу з «паралельного» курсу, при якому Δ0 = 00, *Z0* ≠ 0 м;
*  маневр заходу «з прямої», при якому Δ0 = 00, *Z0* = 0 м.

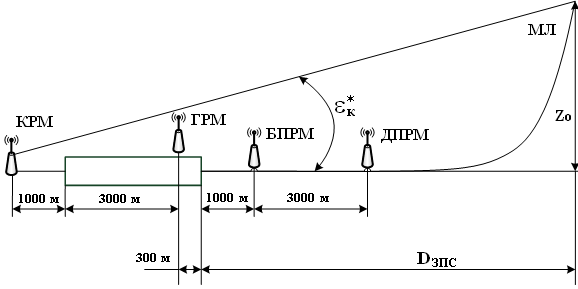


Рис. 1. Типова схема розташування радіотехнічних систем заходу

на посадку та лінії шляху літака в бічній площині

Процес «захвату» зони випромінювання КРМ починається не з моменту перетинання його рівносигнальної зони, а в момент входження літака в цю зону. Тому захід на посадку з виконанням четвертого розвороту можна умовно поділити на три етапи:

* виконання четвертого розвороту до моменту «захвату» зони випромінювання КРМ;
* «захват» зони випромінювання КРМ і вихід в площину посадкового курсу (рівносигнальної зони КРМ);
* стабілізація літака в площині посадкового курсу (рівносигнальної зони КРМ).

Типові вимоги до якості перехідних процесів при виході літака в площину посадкового курсу (рівносигнальної зони КРМ) і точності руху літака по ній такі: процес виходу літака в площину посадкового курсу (рівносигнальної зони КРМ) повинен наближатися до аперіодичного з одним перерегулюванням, що не перевищує по вихідному сигналу КРП ±42 мкА; в усіх випадках він повинен закінчуватися до початку процесу «захвату» глісади, причому процес вважається закінченим, якщо вихідний сигнал КРП увійшов у «трубку», що дорівнює ±35 мкА і продовжує залишатися в межах цієї «трубки» з ймовірністю *Р* = 0,95 аж до висоти прийняття рішення.

Наведені вимоги повинні виконуватися при розумному поєднанні таких умов: зустрічний вітер зі швидкістю до 13 м/с, попутний вітер зі швидкістю до 5 м/с, турбулентність атмосфери викликає нормальне перевантаження, яке не перевищує σ*ny* = 0,5.

Тривалість та характер процесу виходу літака в площину посадкового курсу залежить від точності настроювання параметрів САУ, ваги і швидкості руху літака, крутизни сигналу тракту КРМ–КРП та інших чинників.

Крутизною сигналу тракту КРМ–КРП *Sк* називається параметр, який показує, на скільки зміниться струм на виході КРП при кутовому відхиленні літака відносно КРМ на 10.

У залежності від виду та настроювання параметрів обладнання КРМ та КРП крутизна сигналу тракту КРМ-КРП в різних аеропортах знаходиться в межах:

= (54-280) мкА/град.

Математична модель динаміки польоту літака в режимі

автоматичного заходу на посадку за сигналами КРМ

У режимі заходу на посадку рух літака за сигналами радіотехнічних систем характеризується порівняно малим діапазоном зміни більшості параметрів польоту відносно їх значень у вихідному горизонтальному польоті перед початком режиму, що дозволяє використовувати лінеаризовані диференціальні рівняння.

З метою спрощення будемо вважати, що рух літака здійснюється з постійною швидкістю *V =* *V0* . Тоді лінеаризовані диференціальні рівняння руху літака в бічній площині можна записати в такому вигляді (знак варіації Δ опущено, позначення параметрів загальноприйняті, ЗПС  = 00):

= – *a1* – *b6* – *a2 – a3* – *b5*

= – *a6* – *b1* – *b2 – a5* – *b3*

*= + b7 + b4 γ – a4 – a7*

= *–*

= Sin (

= Cos (

**Закон управління САУ в режимі АЗП за сигналами КРМ**

**(бічний канал)**

Спрощений закон управління САУ в режимі автоматичного заходу на посадку за сигналами КРМ у бічній площині має такий вигляд:

= *F*δe

= *F*δн

= *F*2



= = 00



= 2,0 = 1,5 = 2,5 = 2,0

= 1,3 = 1,3 = 8,0 = 1,0

=

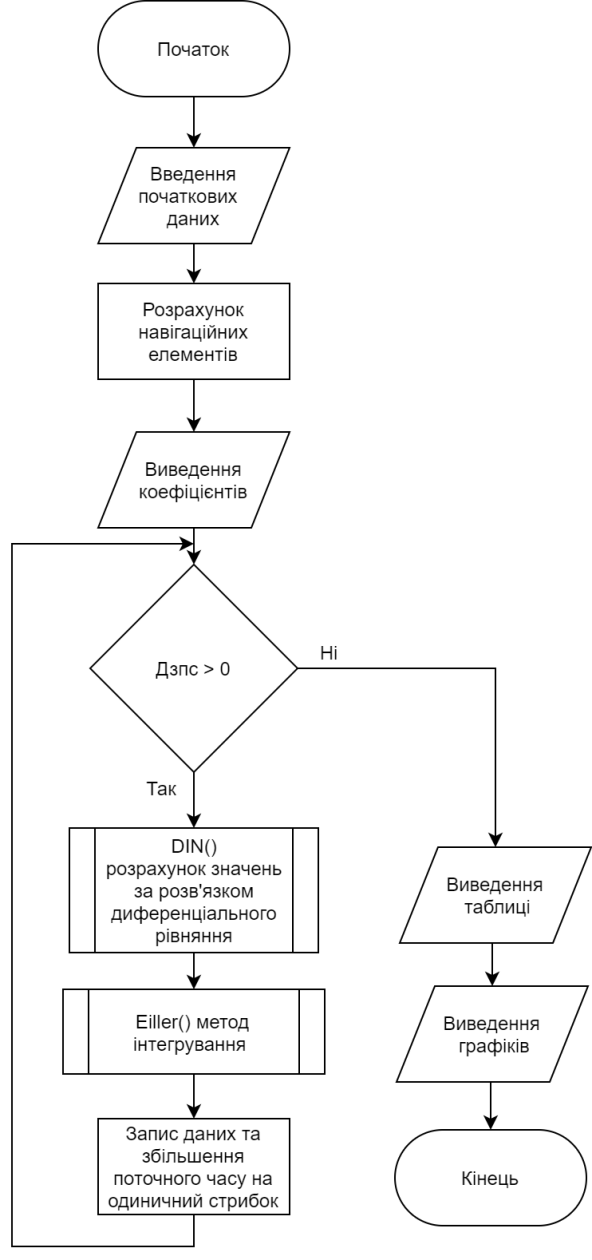
=

= 1,6 c = 2,5 c = 2,3 с

= 0,85 c = 2,3 c

*F*δe = ±120 *F*δн = ±100 *F*1 = ±250 *F*2 = ±200

**Алгоритм роботи**



**Код програми**

public class Rozrakhunok

{

private double T = 0, DT = 0.01, TD = 1, DD = 1;

private double S = 201.45f, l = 37.55f, G0 = 80000f, Ix = 250000f, Iy = 900000f;

private double V = 78f, H = 500f, p = 0.119f, g = 9.81f, abal = 7.1f, Teta0 = 0f;

private double mywy = -0.21f, myb = -0.2f, mydh = 0.0716f, mxdh = -0.0206f, mxwy = -0.31f, mxwx = -0.583f, mxb = -0.186f, mxde = -0.0688f, myde = 0f, mywx = -0.006f;

private double Czb = -1.0715f, Czdh = -0.183f;

private double rad = 53.7f;

private double a1, a2, a3, a4, a5, a6, a7, b1, b2, b3, b4, b5, b6, b7;

private double k5 = 2f, k17 = 170f;

private double[] Y = new double[13], X = new double[13];

private double Sk, Dg, psig, Ekz, Ikz, dIk = 0, Ik = 0, Fk, Ek, f1, F1, f2, F2, fde, de, fdh, dh;

public int state = 1;

public List<double> Time = new List<double>();

public List<double> massFi = new List<double>();

public List<double> massPsi = new List<double>();

public List<double> massX = new List<double>();

public List<double> massZ = new List<double>();

public List<double> massGp = new List<double>();

public List<double> graphTime = new List<double>();

public List<double> graphZ = new List<double>();

public List<double> graphDzz = new List<double>();

public List<double> graphPsi = new List<double>();

public Rozrakhunok()

{

double m = G0 / g;

a1 = -mywy \* S \* l \* l \* p \* V / Iy / 4;

a2 = -myb \* S \* l \* p \* V \* V / Iy / 2;

a3 = -mydh \* S \* l \* p \* V \* V / Iy / 2;

a4 = -Czb \* S \* p \* V / m / 2;

a5 = -mxdh \* S \* l \* p \* V \* V / Ix / 2;

a6 = -mxwy \* S \* l \* l \* p \* V / Ix / 4;

a7 = -Czdh \* S \* p \* V / m / 2;

b1 = -mxwx \* S \* l \* l \* p \* V / Ix / 4;

b2 = -mxb \* S \* l \* p \* V \* V / Ix / 2;

b3 = -mxde \* S \* l \* p \* V \* V / Ix / 2;

b4 = g \* Math.Cos(abal / rad) / V;

b5 = -myde \* S \* l \* p \* V \* V / Iy / 2;

b6 = -mywy \* S \* l \* l \* p \* V / Iy / 4;

b7 = Math.Sin(abal / rad);

switch (state)

{

case 1:

{

Sk = 167f;

Y[0] = 90f;

Y[5] = 5000f;

break;

}

case 2:

{

Sk = 167f;

Y[0] = 90f;

Y[5] = 3000f;

break;

}

case 3:

{

Sk = 167f;

Y[0] = 90f;

Y[5] = 1000f;

break;

}

case 4:

{

Sk = 54f;

Y[0] = 90f;

Y[5] = 3000f;

break;

}

case 5:

{

Sk = 167f;

Y[0] = 90f;

Y[5] = 3000f;

break;

}

case 6:

{

Sk = 280f;

Y[0] = 90f;

Y[5] = 3000f;

break;

}

case 7:

{

Sk = 167f;

Y[0] = 0f;

Y[5] = -300f;

break;

}

case 8:

{

Sk = 167f;

Y[0] = 0f;

Y[5] = -2000f;

break;

}

}

Y[6] = -18000f;

Dg = (500 - 300 \* Math.Tan(3f / rad)) / Math.Tan(2.67f / rad);

while (Y[6] < 0)

{

if (Math.Abs(Y[6]) < Dg)

{

k5 = 3f;

k17 = 120f;

}

psig = -Y[0];

Ekz = rad \* Math.Atan(Y[5] / (Math.Abs(Y[6]) + 4000f));

Ikz = Sk \* Ekz + dIk;

Ek = Y[12] / 167f;

X[12] = (Fk - Y[12]) / 0.2f;

if (Ikz >= 250f)

{

Fk = 250f;

}

else if (Ikz <= -250f)

{

Fk = -250f;

}

else

{

Fk = Ikz;

}

f1 = -1.3f \* psig + 8f \* Ek;

if (f1 >= 25f)

{

F1 = 25f;

}

else if (f1 <= -25f)

{

F1 = -25f;

}

else

{

F1 = f1;

}

X[7] = (k5 \* psig - Y[7]) / 2.3f;

X[8] = (k17 \* Ek - Y[8]) / 2.3f;

f2 = 1.3f \* psig + X[7] + X[8] + F1;

if (f2 >= 20f)

{

F2 = 20f;

}

else if (f2 <= -20f)

{

F2 = -20f;

}

else

{

F2 = f2;

}

X[9] = -(F2 + Y[9]) / 0.85f;

X[10] = 1.5f \* X[3] - Y[10] / 1.6f;

fde = 2f \* (Y[1] - Y[9]) + Y[10];

if (fde >= 12f)

{

de = 12f;

}

else if (fde <= -12f)

{

de = -12f;

}

else

{

de = fde;

}

X[11] = 2.5f \* X[2] - Y[11] / 2.5f;

fdh = 2 \* Y[2] - Y[11];

if (fdh >= 10f)

{

dh = 10f;

}

else if (fdh <= -10f)

{

dh = -10f;

}

else

{

dh = fdh;

}

dh = 0;

X[0] = Y[2];

X[1] = Y[3];

X[2] = -a1 \* X[0] - b6 \* X[1] - a2 \* Y[4] - a3 \* dh - b5 \* de;

X[3] = -b1 \* X[1] - a6 \* X[0] - b2 \* Y[4] - a5 \* dh - b3 \* de;

X[4] = X[0] + b7 \* X[1] + b4 \* Y[1] - a4 \* Y[4] - a7 \* dh;

X[5] = V \* Math.Sin((psig + Y[4]) / rad);

X[6] = V \* Math.Cos((psig + Y[4]) / rad);

for (int i = 0; i < 13; i++)

{

Y[i] = Y[i] + X[i] \* DT;

}

if (T >= TD)

{

Time.Add(TD);

massFi.Add(Y[3]);

massPsi.Add(psig);

massX.Add(Math.Abs(Y[6]));

massZ.Add(Math.Abs(Y[5]));

massGp.Add(Y[7]);

TD += DD;

}

graphTime.Add(T);

graphZ.Add(-Y[5]);

graphPsi.Add(-psig);

graphDzz.Add(-Y[6]);

T = T + DT;

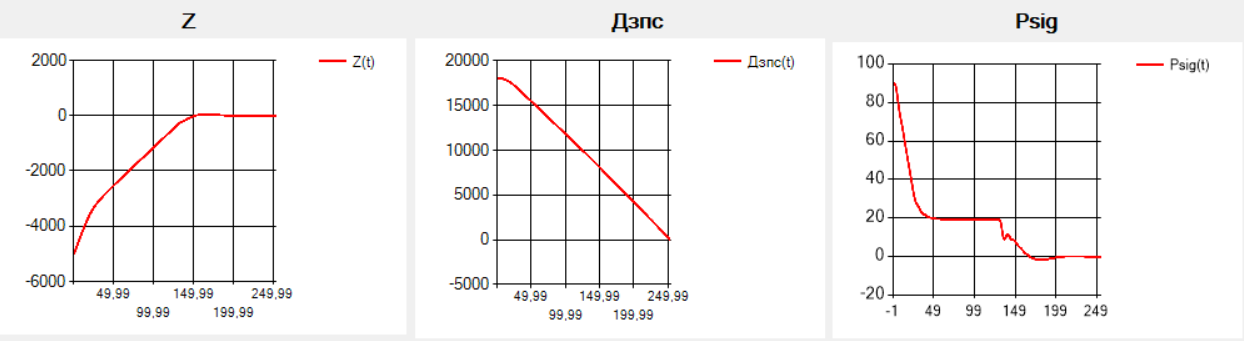
}

}

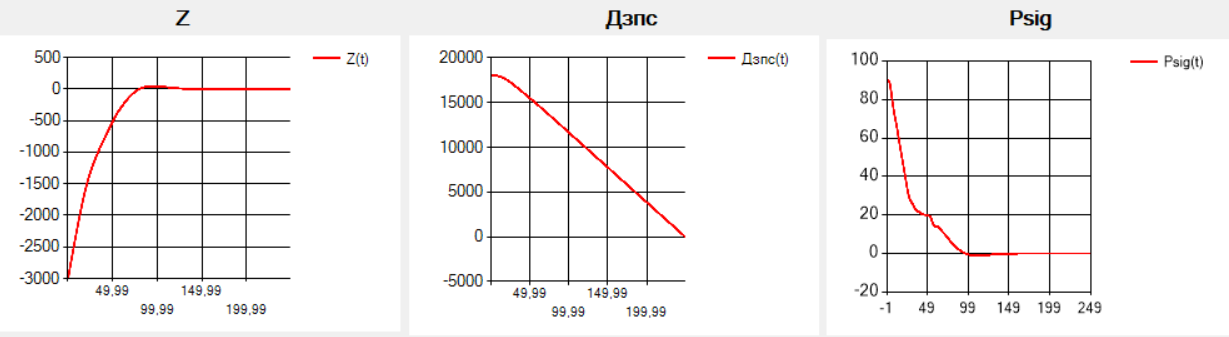
**Дослідження динаміки заходу на посадку**

**літака в режимі АЗП до моменту досягнення ним торця ЗПС.**

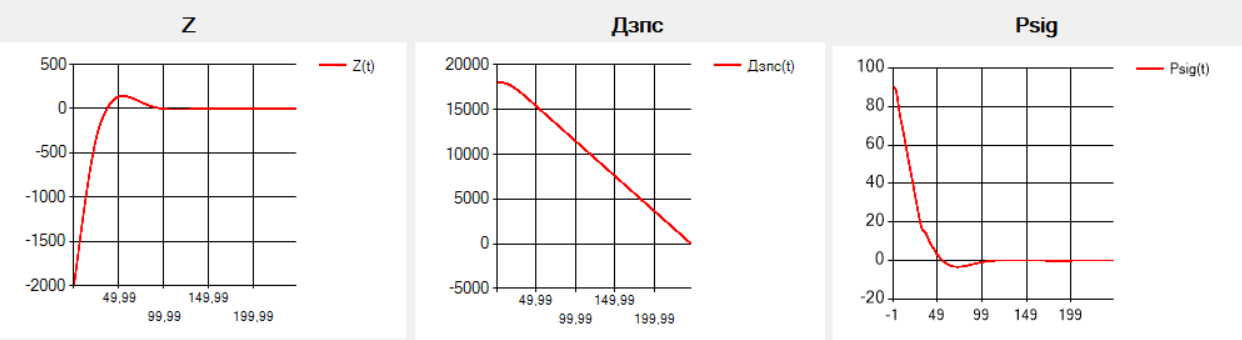
3.1.1 Z0 = -5000 м, ψг0 = +900



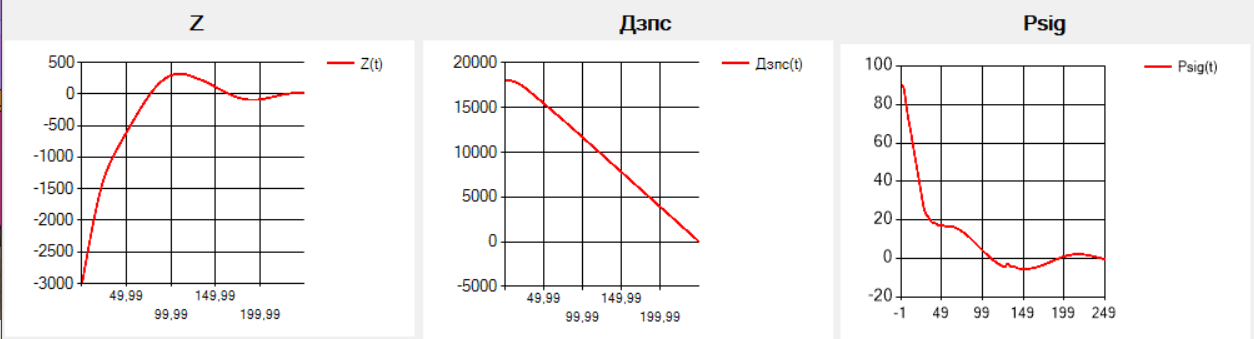
3.1.2 Z0 = -3000 м, ψг0 = +900



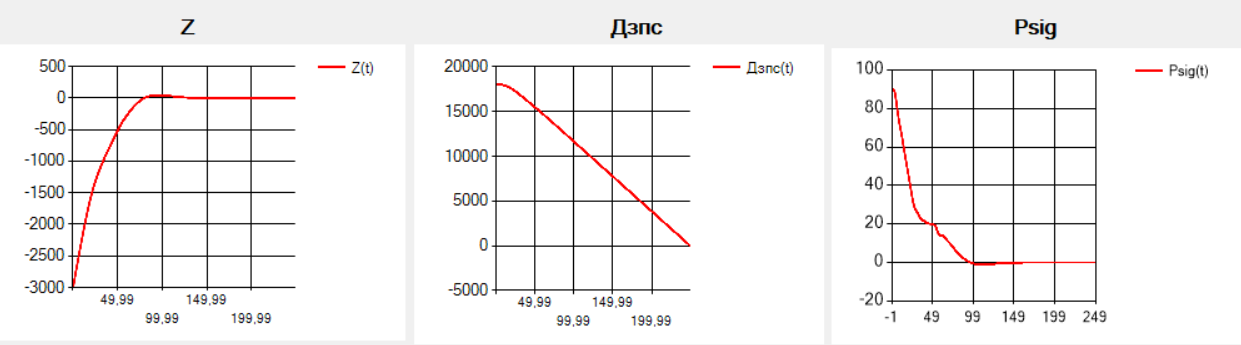
3.1.3 Z0 = -2000 м, ψг0 = +900



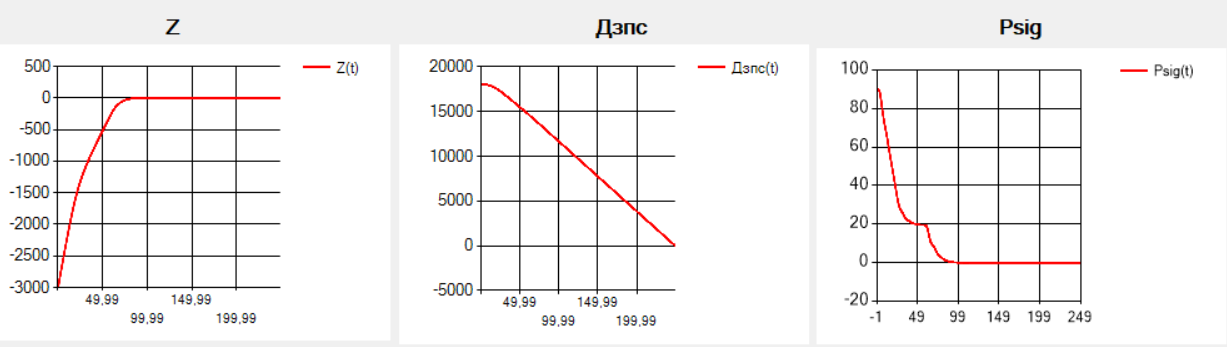
3.2.1 Z0 = -2000 м, ψг0 = +900, Sk = 54;



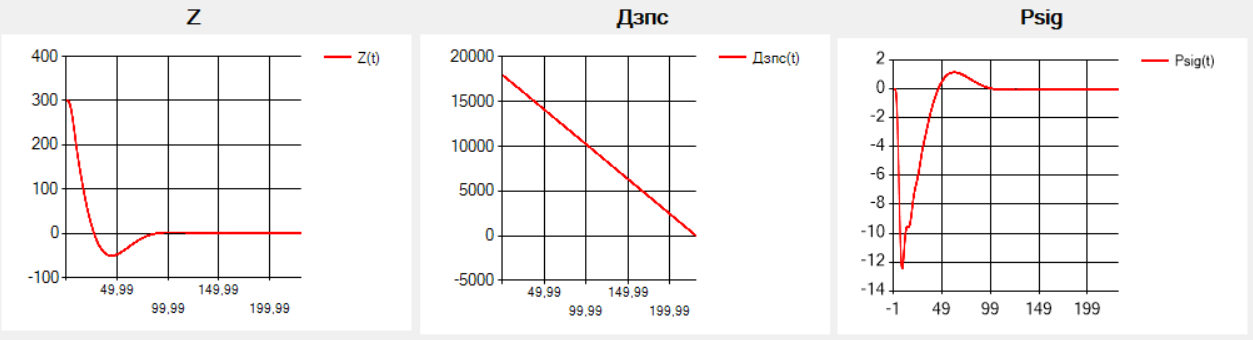
3.2.2 Z0 = -2000 м, ψг0 = +900, Sk = 167;



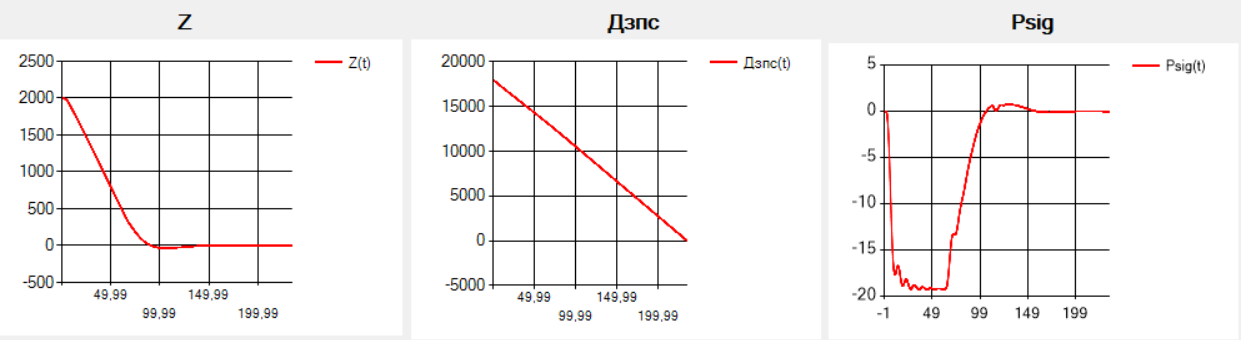
3.2.3 Z0 = -2000 м, ψг0 = +900, Sk = 280;



3.3.1 Z0 = 300 м, ψг0 = 00;



3.3.2 Z0 = 2000 м, ψг0 = 00



**Висновок про особливості, переваги та недоліки досліджених способів виводу літака в ЗТ.**

За результатами досліджень можна побачити, що чи більше відстань по Z, тим більший час перехідного процесу. При Sk = 54 система має найменшу стійкість та найбільший час перехідного процесу. При Sk = 280 система має найбільшу стійкість та найменший час перехідного процесу. Система працює правильно при різних ψг0 та при Z0 як від’ємних, так і додатних.