

**Державний вищий навчальний заклад
«Український державний хіміко-технологічний університет»**

Факультет комп'ютерних наук та інженерії

(повна назва факультету)

Кафедра комп'ютерно-інтегрованих технологій та метрології

(повна назва кафедри)

Пояснювальна записка

до курсової роботи з дисципліни

«Сучасні проблеми автоматизованого управління»

Виконав: студент ___ курсу, групи _____

напряму підготовки (спеціальності) 151 –

Автоматизація та комп'ютерно-інтегровані технології
(шифр і назва напряму підготовки (професійна спрямованість), спеціальності)

(прізвище та ініціали)

Керівник _____ Манко Г.І.

(підпис)

(прізвище та ініціали)

Дніпро – 2020 року

Державний вищий навчальний заклад
«Український державний хіміко-технологічний університет»
(повне найменування вищого навчального закладу)

Факультет, відділення КНтаІ
Кафедра КІТіМ
Освітньо-кваліфікаційний рівень бакалавр
Напрямок підготовки _____
(шифр і назва)
Спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології
(шифр і назва)

З А В Д А Н Н Я
НА КУРСОВИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТУ

_____ (прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Параметричний синтез регулятора і аналіз отриманої замкненої системи управління антеною
- керівник проекту (роботи) Манко Г.І., к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
2. Строк подання студентом проекту (роботи) 09.06.2020
3. Вихідні дані до проекту (роботи) згідно Методичних вказівок № 4850
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1 Модель системи у формі передатної функції
2 Графічний інтерфейс користувача для інтерактивного синтезу регулятора
3 Синтез моделі спостерігача
4 Перевірка результатів моделюванням у SIMULINK
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) –

6. Дата видачі завдання 20.04.2020

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів курсового проекту (роботи) | Строк виконання етапів проекту (роботи) | При- мітка |
|----------|---|---|---------------|
| 1 | <i>Отримання передатних функції ланок системи</i> | <i>30.04.2020</i> | |
| 2 | <i>Ознайомлення з теоретичними відомостями</i> | <i>08.05.2020</i> | |
| 3 | <i>Створення макету графічного інтерфейсу користувача</i> | <i>15.05.2020</i> | |
| 4 | <i>Розробка функцій графічного інтерфейсу користувача</i> | <i>23.05.2020</i> | |
| 5 | <i>Наладка графічного інтерфейсу користувача</i> | <i>29.05.2020</i> | |
| 6 | <i>Побудова Simulink–моделі та аналіз системи</i> | <i>08.06.2020</i> | |
| 7 | <i>Оформлення і захист курсової роботи</i> | <i>15.06.2020</i> | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Студент

_____ (підпис)

_____ (прізвище та ініціали)

Керівник проекту (роботи)

_____ (підпис)

Манко Г.І.

_____ (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до курсової роботи: 32 с., 11 рис., 1 табл., 6 літературних джерел.

Об'єкт дослідження – система управління антеною.

Мета роботи полягає в параметричному синтезі регуляторів і аналізі отриманої замкнутої системи управління з точки зору її відповідності заданим вимогам з точності і якості.

У курсовому проекті виконано:

- отримання моделі динамічного об'єкта у формі системи диференціальних рівнянь;
- розрахунок передатної функції об'єкта;
- перехід до лінійної моделі у формі рівнянь стану;
- вибір структури управління об'єктом;
- параметричний синтез модального регулятора по повному вектору стану;
- проектування спостерігача стану об'єкта;
- побудова графічного інтерфейсу користувача (GUI) для автоматизації розрахунків;
- аналіз точнісних і динамічних характеристик спроектованої системи.

SIMULINK, ГРАФІЧНИЙ ІНТЕРФЕЙС КОРИСТУВАЧА,
МАТЕМАТИЧНА МОДЕЛЬ, РЕГУЛЯТОР, СИСТЕМА АВТОМАТИЧНОГО
РЕГУЛЮВАННЯ, СПОСТЕРІГАЧ, ЯКІСТЬ ПЕРЕХІДНИХ ПРОЦЕСІВ

ЗМІСТ

| | |
|---|----|
| Перелік умовних скорочень | 4 |
| Вступ | 5 |
| 1 Модель системи у формі передатної функції | 6 |
| 2 Графічний інтерфейс користувача для інтерактивного синтезу регулятора | 9 |
| 3 Синтез моделі спостерігача | 18 |
| 4 Перевірка результатів моделювання у SIMULINK | 24 |
| Висновки | 29 |
| Література | 30 |

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

GUI – Graphical User Interface

ПФ – передатна функція

РЛС – радіолокаційна станція

САР – система автоматичного регулювання

ВСТУП

Розвиток систем, що функціонують в умовах невизначеності під впливом зовнішніх збурень, обумовлює підвищені вимоги до створення адекватних математичних моделей, використання більш досконалих методів синтезу регуляторів, більш прискіпливого аналізу результатів. Ефективне вирішення науково-прикладної задачі синтезу високоякісної системи регулювання може бути здійснене шляхом комплексного використання як класичних, так і новітніх методів і підходів.

Радіолокаційне спостереження повітряних цілей забезпечує їх своєчасне виявлення, спостереження за переміщенням, розпізнавання, визначення координат цілі, безпеку польотів, запобігання аварійних і катастрофічних ситуацій. До відповідних систем управління пред'являються вимоги до точності, надійності і здатності зберігати свої властивості в умовах невизначеностей і завад. Тому задача синтезу високоточної стежачої системи управління радіолокаційною станцією (РЛС) є актуальною задачею.

У курсовій роботі виконаний синтез стежачої системи управління антеною і визначення оптимальних значень регуляторів для покращення динамічних характеристик антенного пристрою. З метою автоматизації розрахунків і отримання можливості інтерактивного втручання у процес синтезу розроблений графічний інтерфейс користувача (англ. Graphical User Interface, GUI). Для аналізу якості синтезованої системи побудована SIMULINK-модель системи.

1 МОДЕЛЬ СИСТЕМИ У ФОРМІ ПЕРЕДАТНОЇ ФУНКЦІЇ

Структурна схема системи управління антеною РЛС, що відстежує рух цілі в горизонтальній площині [1–3], показана на рис. 1.1.



Рис. 1.1– Структура системи управління антеною

Завдання управління полягає в такій дії на електричний двигун, при якому $\theta(t) \approx \theta_r(t)$, $t > t_0$, де $\theta(t)$ – кутове положення антени, $\theta_r(t)$ – кутове положення цілі.

Об'єкт управління складається з антени і двигуна. Завданням є кут $\theta_r(t)$. Збуренням – момент вітрового навантаження, прикладений до антени. Спостережувані змінні визначаються складом датчиків. Такою змінною може служити вихідний сигнал потенціометра на валу антени, визначуваний вираженням $\eta(t) = \theta(t) + v(t)$, де $v(t)$ – шум вимірів. Кутом $\theta(t)$ необхідно управляти, отже, він є керованою змінною.

Спрощену модель двигуна приймемо у вигляді

$$T_d \frac{d\mu(t)}{dt} + \mu(t) = k_1 u(t), \quad (5.10)$$

де T_d – стала часу двигуна, $\mu(t)$ – момент двигуна

Рух антени може бути описаний диференціальним рівнянням

$$J \frac{d^2\theta(t)}{dt^2} + \beta \frac{d\theta(t)}{dt} = k\mu(t) + \mu_f(t). \quad (5.9)$$

Тут J – приведений момент інерції частин, що обертаються, β – коефіцієнт в'язкого тертя, $\mu_f(t)$ – момент вітрового навантаження.

Для отримання передатної функції необхідно перетворити кожне з отриманих диференціальних рівнянь в операторну форму за нульових початкових умов.

Для рівняння двигуна отримаємо:

$$T_d s \mu(s) + \mu(s) = k_1 u(s);$$

$$W_d = \frac{\mu(s)}{u(s)} = \frac{k_1}{T_d s + 1}.$$

Для антени:

$$J s^2 \theta(s) + \beta s \theta(s) = k_2 \mu(s);$$

$$W_a = \frac{k_2}{J s^2 + \beta s}.$$

Передатна функція об'єкта управління являє собою послідовне з'єднання двох ланок (рис. 1.2).

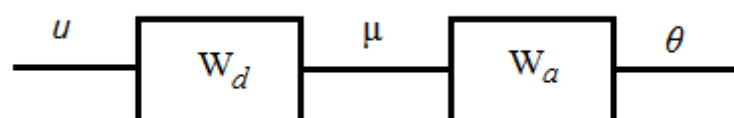


Рис. 1.2 – Структурна схема об'єкта управління

Згідно варіанту n завдання, параметри передатних функцій мають значення, наведені у табл. 1.1.

Таблиця 1.1

| T_d, c | k_1 | $J, кгм^2$ | $\beta, Нм/с$ | k_2 |
|----------|-------|------------|---------------|-------|
| 0,5 | 0,2 | 40 | 3,8 | 8 |

ПФ об'єкта в цілому отримаємо, використовуючи команди Matlab [4]:

```
k1=0.2; Td=0.5; Wd=tf(k1, [Td, 1])
```

```
k2=8; J=40; Beta=3.8; Wa=tf(k2, [J, Beta, 0])
```

```
Wo=Wd*Wa
```

Одержуємо результат:

$$W_o = \frac{1.6}{20 s^3 + 41.9 s^2 + 3.8 s}$$

Таким чином, маємо динамічну систему третього порядку. Її полюси:

$$P = \text{pole}(W_o)'$$

$$P = \begin{matrix} 0 & -2.0000 & -0.0950 \end{matrix}$$

Процедура модального синтезу системи регулювання включає два основні етапи:

- вибір бажаного розташування полюсів замкнутої системи;
- розрахунок коефіцієнтів зворотного зв'язку і регуляторів, які забезпечують задане розташування полюсів.

Для вирішення першого завдання можна вибрати таке розташування, виходячи з міри стійкості і коливальності замкнутої системи. Нехай треба забезпечити якість регулювання, задану параметрами:

- а) перерегулювання $\sigma < 20\%$;
- б) час перехідного процесу $t_n < 30$ с.

2 ГРАФІЧНИЙ ІНТЕРФЕЙС КОРИСТУВАЧА ДЛЯ ІНТЕРАКТИВНОГО СИНТЕЗУ РЕГУЛЯТОРА

Інтерфейс GUI [5] складається з форми, на якій розміщуються елементи управління – контролі (controls), такі, як кнопки, текстові поля і графіки. Кожен тип контролю має деякі властивості, які впливають на його поведінку. Деякі властивості є загальними для усіх типів, наприклад, тег (tag), що означає ім'я контролю, тоді як інші властивості можуть бути унікальними для цього типу контролів. Кожен з контролів має внутрішній ідентифікатор, або хендл (handle). За допомогою хендлу можна отримати доступ до властивостей контролю безпосередньо з програмного коду.

Одним із засадничих понять є також callback-функція. Ці функції є механізмом, за допомогою якого в Matlab відбувається обробка подій, згенерованих контролами. Кожна подія, наприклад, натиснення кнопки, має свою callback-функцію, яка викликається автоматично при настанні цієї події.

У кожному callback-функцію передаються три параметри:

а) hObject – хендл об'єкта, який викликав цю подію, маючи хендл, ми можемо легко отримати доступ до властивостей цього контролю;

б) eventdata – зарезервована змінна;

в) Handles – список хендлів усіх контролів, розташованих на формі. Цю змінну можна використати в конструкціях виду handles.{tag потрібного контролю} для отримання хендлів інших контролів.

Іноді може бути потрібно використати одну загальну змінну в декількох функціях. У такому разі, в кожній функції, що використовує цю змінну, вона має бути оголошена із специфікатором доступу global, наприклад:

```
global Wo P Wzpk
```

Для створення макету інтерфейсу користувача треба ввести команду **guide**. На екрані з'явиться вікно, в якому можна створити новий або відкрити

вже існуючий інтерфейс. Створюватимемо інтерфейс з нуля, тому в списку шаблонів вибираємо "Blank GUI".

Після цього з'являється вікно з редактором інтерфейсу. У лівій частині вікна розташована панель контролів, з яких може складатися інтерфейс. Щоб помістити елемент на форму, досить вибрати його зі списку і клацнути в тому місці форми, де треба помістити контрол. Двічі клацнувши на елементі, ви можете викликати редактор властивостей.

Створюємо макет нашої майбутньої програми. Помістимо на форму елементи і задамо їх властивості так, як показано на рис. 2.1.

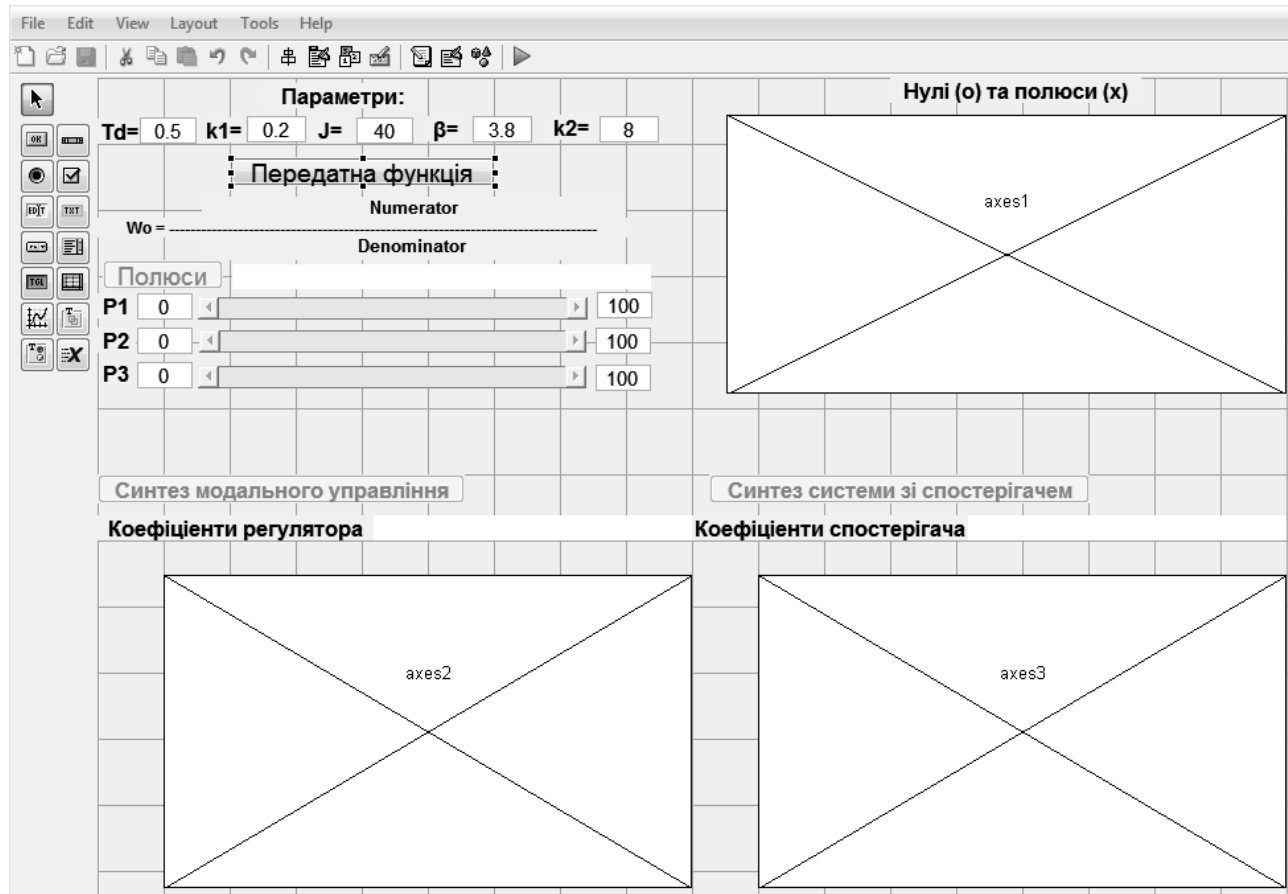





Рис. 2.1 – Макет графічного інтерфейсу користувача

На панелі контролів виберемо  (Static Text) і розміщуємо його вгорі макету. Двічі клацаємо по створеному контролу і у вікні властивостей у полі String пишемо текст «Параметри:». У відповідних полях встановлюємо бажані параметри: шрифт (FontName), кегль (FontSize), жирність (FontWeight). Тег контролу можна залишити таким, що встановлюється автоматично. Для

пришвидшення роботи інші тексти GUI можна копіювати зі вже створеного описаним способом, замінюючи поля String на потрібні написи.

Далі створюємо віконця для введення і редагування числових значень. На панелі контролів виберемо  (Edit Text) і розміщуємо у потрібних місцях.


Для полегшення орієнтації у програмі міняємо теги на зручні для запам'ятовування: перші три літери будуть скороченим записом типу тегу, далі з великої літери – ім'я змінної або виконувана операція. Наприклад, edtTd, edtJ, edtBeta.

На панелі контролів виберемо  (Push Button) і розміщуємо кнопку управління «Передатна функція». Задаємо для неї тег btnTF.


Створюємо контроли для виведення вигляду передатної функції: Numerator (чисельник), Denominator (знаменник) і напис «Wo= -----...- » для розділюючої лінії. Для чисельника і знаменника розміщуємо теги txtNum і txtDen.

Розміщуємо кнопки управління «Полюси», «Синтез модального управління», «Синтез системи зі спостерігачем) з тегами btnPole, btnSint, btnEst. Їх можна скопіювати з кнопки «Передатна функція», помінявши вміст полей String і Tag. Крім того, для останніх кнопок треба установити значення off у полях Enable, щоб унеможливити їх передчасне використання.

Для виведення розрахованих значень створюємо текстові поля з тегами txtPole, (спочатку для розрахованих, а потім для бажаних полюсів системи), txtK (коефіцієнти регулятора) txtL (коефіцієнти спостерігача). Для цих текстів встановлюємо білий фон (властивість BackgroundColor).

Щоб мати можливість переміщення полюсів у бажане положення, вводимо контроли слайдерів. На панелі контролів виберемо  (Slider). Задаємо для них теги sldP1, sldP2, sldP3 та встановлюємо значення off у полях Enable. Кількість слайдерів дорівнює порядку системи.

З лівого і правого боків кожного слайдеру розташовуємо контроли Edit Text для можливості зміни границь діапазонів положень полюсів. Задаємо теги edtMinP1 (P2, P3) та edtMaxP1 (P2, P3).

Нарешті розміщуємо вікна виведення графіків. На панелі контролів виберемо  (Axes). Теги axes1, axes2, axes3 не міняємо.

Далі необхідно створити функції, що викликаються при задіюванні контролів.

Правою клавішею миші клацаємо по кнопці «Передатна функція». У контекстному меню вибираємо View Callbacks та Callback. У вікні редактора програм Edit виділяється рядок:

```
function btnTF_Callback(hObject, eventdata, handles)
```

Під цим рядком дописуємо текст функції:

```
% Функція визначення передатної функції
% Глобальні змінні:
% Wo - передатна функція (ПФ) об'єкта управління
% P - бажане розміщення полюсів системи
% Wzpk - ПФ з бажаним розміщенням полюсів у формі zpк
global Wo P Wzpk
% Зчитування параметрів ПФ об'єкта
Td= str2double(get(handles.edtTd, 'String'));
k1= str2double(get(handles.edtk1, 'String'));
J= str2double(get(handles.edtJ, 'String'));
Beta= str2double(get(handles.edtBeta, 'String'));
k2= str2double(get(handles.edtk2, 'String'));
Wd=tf(k1, [Td, 1]); % ПФ двигуна
Wa=tf(k2, [J, Beta, 0]); % ПФ антени
Wo= Wd*Wa; % ПФ об'єкта
TFout(hObject, eventdata, handles) % Виведення формули ПФ
set(handles.btnPole, 'Enable', 'on')
```

Останнім рядком цієї функції задіюється кнопка «Полюси».

Для виведення формули передатної функції треба додати функцію TFout:

```
% Виведення формули ПФ
function TFout(hObject, eventdata, handles)
global Wo P Wzpk
```

```

[num,den]=tfdata(Wo,'v')
m=length(num) % розмір чисельника ПФ
n=length(den) % розмір знаменника ПФ
strnum = ''; % символний рядок для чисельника
for i=1:m - 1
    if not(num(i)==0)
        strnum=[strnum,num2str(num(i)), '*s^', num2str(m-i), '+'];
    end
end
if not(num(m)==0)
    strnum = [strnum,num2str(num(m))]
end
strden = ''; % символний рядок для знаменника
for i=1:n - 1
    strden=[strden,num2str(den(i)), '·s^', num2str(n-i), ' +
']
end
if den(n)~=0
    strden = [strden,num2str(den(n))]
else
    strden = strden(1:length(strden) - 3)
end
set(handles.txtNum, 'String', strnum) ;
set(handles.txtDen, 'String', strden) ;

```

Функція, що викликається при натисненні кнопки «Полюси», створюється так, як і для кнопки «Передатна функція». Після рядків

```

% - - - Executes on button press in btnPole.
function btnPole_Callback(hObject, eventdata, handles)

```

додаємо текст:

```

% Функція визначення полюсів системи
global Wo P Wzpk
% Задіявання необхідних слайдерів
set(handles.sldP1, 'Enable', 'on')
set(handles.sldP2, 'Enable', 'on')
set(handles.sldP3, 'Enable', 'on')
P=pole(Wo)'; % вектор полюсів системи
set(handles.txtPole, 'String', num2str(P)) ;
Wzpk=zpk(Wo) % переведення ПФ у форму zero-pole-gain
DrawPoles(hObject, eventdata, handles)% Виведення графіка
% Установлення границь для полюсів
set(handles.edtMinP1, 'String', num2str(P(1)*2 - 1)) ;
set(handles.edtMaxP1, 'String', num2str(P(1)*0.5)) ;
set(handles.edtMinP2, 'String', num2str(P(2)*2 - 1)) ;
set(handles.edtMaxP2, 'String', num2str(P(2)*0.5)) ;
set(handles.edtMinP3, 'String', num2str(P(3)*2 - 1)) ;
set(handles.edtMaxP3, 'String', num2str(P(3)*0.5)) ;
% Задіяється кнопка синтезу
set(handles.btnSint, 'Enable', 'on');

```

Тут викликається функція DrawPoles побудови графіка нулів і полюсів:

```

% Функція виведення графіка полюсів
function DrawPoles(hObject, eventdata, handles)
global Wo P Wzpk
axes(handles.axes1); % вибір вікна графіка
cla; % очистка вікна
pzmap(Wzpk) % Побудова графіка нулів і полюсів

```

Для переміщення полюсів у бажане положення використовуються функції обробки переміщення повзунків слайдерів. Як приклад наведемо функцію для слайдера sldP1:


```

% - - - Executes on slider movement.
function sldP1_Callback(hObject, eventdata, handles)
global Wo P Wzpk
max1=str2double(get(handles.edtMaxP1,'String'));
%Максимум
min1=str2double(get(handles.edtMinP1,'String')); %
Мінімум c1=get(handles.sldP1,'Value'); % Зчитування
положення слайдера
curl=0.5*(max1+min1); % середнє значення
curl=curl+(max1 - curl)*c1; % нове значення
P(1)=curl; % привласнення нового значення полюсу P1
Wzpk.p=P; % занесення полюсів у передатну функцію
% Виведення нової матриці полюсів
set(handles.txtPole, 'String', num2str(P));
DrawPoles(hObject, eventdata, handles) % Виведення
графіка

```

Для інших слайдерів можна копіюємо цей текст, замінивши всюди P1 на P2, P3, ... і P(1) на P(2), P(3), ...


При натисненні на кнопку «Синтез модального управління» виконується наступна функція:

```

% - - - Executes on button press in btnSint.
function btnSint_Callback(hObject, eventdata, handles)
% Функція синтезу системи модального управління
global Wo P
% Модель у просторі станів
Wss=ss(Wo);
% Вилучення матриць коефіцієнтів
A=Wss.a; B=Wss.b; C=Wss.c; D=Wss.d;
% Будуємо матрицю керованості
Q = ctrb(A, B);

```

```
% Визначаємо ранг матриці Q:
if rank(Q)~=length(B)
    errordlg('система не є керованою, модальне управління
неможливе','Проблема!');
    set(handles.btnSint, 'Enable', 'off');
    return;
end
% значення коефіцієнтів зворотних зв'язків
K=acker(A,B,P);
set(handles.txtK, 'String', num2str(K)) ;
% Матриця A коефіцієнтів замкненої системи:
As=A - B*K;
% Модель замкненої системи:
sys=ss(As,B,C,D);
% ПФ замкненої системи:
Wcls=tf(sys);
% Розрахунок перехідного процесу
[out,t]=step(Wcls);
% обчислення коефіцієнта масштабування
k0=1/out(length(out));
% Будуємо графік
axes(handles.axes2); % вибір вікна
cla; % очистка графіку
plot(t, k0*out)
xlabel('Час')
ylabel('Вихід')
grid on % нанесення сітки
set(handles.btnEst, 'Enable', 'on'); % задіюється кнопка
«Синтез системи зі спостерігачем»
```

Побудований GUI зберігаємо іменем Anten.fig. Опробування проводиться натисненням кнопки , комбінацією клавіш Ctrl+T або через меню Tools→Run.

Вводимо значення параметрів і тиснемо «Передатна функція». На екран виводиться вигляд передатної функції і стає діючою кнопка «Полюси». Тиснемо її і чекаємо, поки не побудується графік розміщення полюсів. Тоді повзунками слайдерів переміщуємо полюси у бажані положення і тиснемо кнопку «Синтез модального управління». Якщо система не є керованою, формується відповідне повідомлення і треба міняти положення полюсів. Якщо система керована, формується матриця коефіцієнтів регулятора і будується графік перехідного процесу.

3 СИНТЕЗ МОДЕЛІ СПОСТЕРІГАЧА

Нехай стаціонарний об'єкт описується традиційною системою рівнянь в просторі станів:

$$\begin{cases} \dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{U}; \\ \mathbf{Y} = \mathbf{C}\mathbf{X}. \end{cases} \quad (3.1)$$

Припустимо, що матриці \mathbf{A} , \mathbf{B} , \mathbf{C} відомі, тоді вектор \mathbf{X} можна замінити вектором $\hat{\mathbf{X}}$ моделі, яка має той же вхід, що і об'єкт (3.1).

$$\dot{\hat{\mathbf{X}}} = \mathbf{A}\hat{\mathbf{X}} + \mathbf{B}\mathbf{U}. \quad (3.2)$$

При відновленні за допомогою моделі (3.2) не використовується вимірюваний вихід. Якість відновлення покращується, якщо ввести в модель різницю вимірюваного виходу і його оцінки $\mathbf{Y} - \mathbf{C}\hat{\mathbf{X}}$ у вигляді зворотного зв'язку:

$$\dot{\hat{\mathbf{X}}} = \mathbf{A}\hat{\mathbf{X}} + \mathbf{B}\mathbf{U} + \mathbf{L}(\mathbf{Y} - \mathbf{C}\hat{\mathbf{X}}). \quad (3.3)$$

Тут \mathbf{L} - деяка матриця, що забезпечує необхідний вигляд перехідних процесів оцінки вектору стану.

Введемо помилку відновлення $\tilde{\mathbf{X}} = \mathbf{X} - \hat{\mathbf{X}}$. Віднімаючи (3.3) з диференціального рівняння (3.1), отримаємо

$$\dot{\tilde{\mathbf{X}}} = \mathbf{A}\tilde{\mathbf{X}} - \mathbf{L}(\mathbf{Y} - \mathbf{C}\hat{\mathbf{X}}) = \mathbf{A}\tilde{\mathbf{X}} - \mathbf{L}\mathbf{C}\mathbf{X} + \mathbf{L}\mathbf{C}\hat{\mathbf{X}} = \mathbf{A}\tilde{\mathbf{X}} - \mathbf{L}\mathbf{C}(\mathbf{X} - \hat{\mathbf{X}}) = (\mathbf{A} - \mathbf{L}\mathbf{C})\tilde{\mathbf{X}}. \quad (3.4)$$

Очевидно, для того, щоб помилка відновлення прагнула до нуля, необхідно вибрати матрицю \mathbf{L} так, щоб система (3.4) була асимптотично стійка.

Таким чином, вводячи зворотний зв'язок в модель відновлення, можна нівелювати помилку, навіть якщо система (3.1) нестійка.

Модель (3.3) відновлює усі складові вектору стану, тому вона називається спостерігачем стану повного порядку Калмана [6].

Модель спостерігача Калмана звичайно представляють у вигляді:

$$\dot{\mathbf{X}}_c = \mathbf{A}\mathbf{X}_c + \mathbf{B}\mathbf{U} + \mathbf{L}(\mathbf{Y} - \mathbf{Y}_c).$$

Схематично це зображається так, як показано на рис. 3.1.

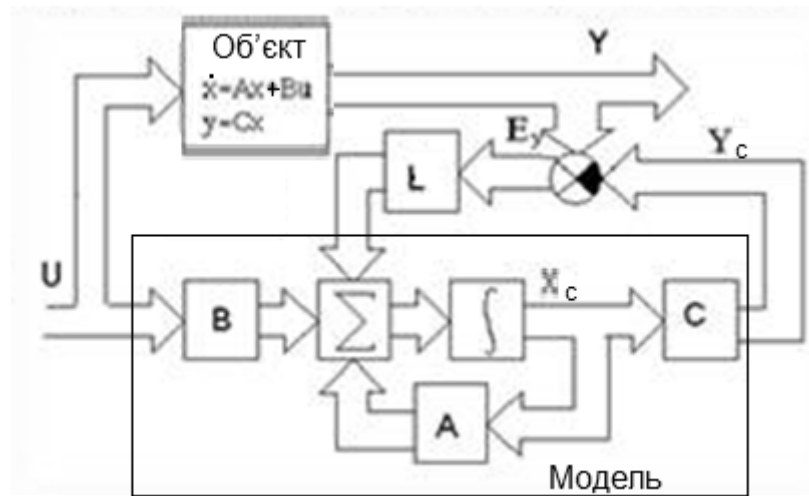


Рис. 3.1 – Структура спостерігача Калмана

Один і той же вхід u подається і на об'єкт, і на модель. Похибка спостерігача, що є різницею виходів об'єкта і моделі $E_y = Y - Y_c$, обробляється матрицею коефіцієнтів підсилення фільтра L і як зворотний зв'язок подається на модель, щоб звести похибку спостерігання до мінімуму.

Така схема забезпечує оцінку змінних стану об'єкту управління при наявності випадкових зовнішніх збурень і похибок вимірів.

Позначимо через w збурення за входом, через v – збурення за спостеріганням внаслідок похибок вимірів.

Модель об'єкта у просторі станів з урахуванням збурень, має вигляд:

$$\begin{cases} \dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{U} + \mathbf{G}\mathbf{w} & \text{(рівняння станів);} \\ \mathbf{Y} = \mathbf{C}\mathbf{X} + \mathbf{D}\mathbf{U} + \mathbf{H}\mathbf{w} + \mathbf{v} & \text{(рівняння вимірів).} \end{cases}$$

Треба виконати синтез спостерігача, який мінімізує помилку оцінювання, яка для одновимірної системи задається в наступному вигляді:

$$P = \lim_{t \rightarrow \infty} M \{ (x - \hat{x})^2 \}$$

У загальному випадку у матричній формі:

$$\mathbf{P} = \lim_{t \rightarrow \infty} M \{ (\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T \}.$$

Калман показав, що оптимальним рішенням є система, що описується рівняннями:

$$\begin{cases} \dot{\hat{\mathbf{X}}} = \mathbf{A}\hat{\mathbf{X}} + \mathbf{B}\mathbf{U} + \mathbf{L}(\mathbf{Y} - \mathbf{C}\hat{\mathbf{X}} - \mathbf{D}\mathbf{U}); \\ \begin{bmatrix} \mathbf{Y} \\ \mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{C} \\ \mathbf{I} \end{bmatrix} \hat{\mathbf{X}} + \begin{bmatrix} \mathbf{D} \\ \mathbf{0} \end{bmatrix} \mathbf{U} + \mathbf{H}\mathbf{w} + \mathbf{v}. \end{cases}$$

Оскільки така система забезпечує подавлення збурень \mathbf{w} і \mathbf{v} , її називають фільтром Калмана.

Пакет Control System має функцію для формування спостерігача Калмана у вигляді LTI-об'єкта:

```
[kest, L, P] = kalman(sys, Qn, Rn, Nn).
```

Ця функція повертає модель спостерігача Калмана kest в просторі станів для

- а) моделі sys спостерігача;
- б) матриці випадкових збурень Qn;
- в) матриці завад вимірів Rn;
- г) матриці кореляційного взаємозв'язку Nn.

Синтез моделі спостерігача відбувається автоматизовано за допомогою GUI Anten.fig, у який додаємо відповідні функції.

Для кнопки «Синтез системи зі спостерігачем» створюємо таку функцію:

```
% - - - Executes on button press in btnEst.
function btnEst_Callback(hObject, eventdata, handles)
% Функція синтезу системи зі спостерігачем
global Wo P
% Модель у просторі станів
Wss=ss(Wo);
% Вилучення матриць коефіцієнтів
A=Wss.a; B=Wss.b; C=Wss.c; D=Wss.d;
% Будуємо матрицю спостережуванності
N = obsv(A, C);
% Перевірка спостережуванності
if det(N)==0
    error('Система не є спостережуваною', 'Проблема!');
```

```

        set(handles.btnEst, 'Enable', 'off');
        return;
end
% Бажане розміщення полюсів спостерігача:
Pe=1.2*P; % лівіше полюсів системи
% Матриця коефіцієнтів спостерігача L:
Lt=acker(A', C', Pe); L=Lt'
set(handles.txtL, 'String', num2str(Lt)) ;
% Матриця A коефіцієнтів замкненої системи:
K=acker(A,B,P);
As=A - B*K;
% Модель замкненої системи:
sys=ss(As,B,C,D);
% Формуємо спостерігаючий пристрій
rsys=reg(sys,K,L)
% Передатна функція спостерігача
We=tf(rsys);
% перехідний процес спостерігача
axes(handles.axes3); % вибір вікна
cla; % очистка графіку
step(We)
xlabel('Час')
ylabel('Вихід')
grid on % нанесення сітки
% Експорт результатів у робочий простір Matlab
checkLabels = {'Зберегти матрицю A під іменем:' ...
               'Зберегти матрицю B під іменем:' ...
               'Зберегти матрицю C під іменем:' ...
               'Зберегти матрицю D під іменем:' ...
               'Зберегти матрицю K під іменем:' ...
               'Зберегти матрицю L під іменем:'};

```

```

varNames = {'A','B','C','D','K','L'};
items = {A,B,C,D,K,L};
export2wsdlg(checkLabels,varNames,items,...
            'Зберегти?');

```

Наприкінці цієї функції розміщені команди створення вікна експорту розрахованих матриць у робочий простір MATLAB для подальшого використання.

Тиснемо кнопку «Синтез системи зі спостерігачем». Якщо система не є спостережуваною, видається повідомлення про це, і робота припиняється. В іншому випадку формується матриця коефіцієнтів спостерігача і будується графік перехідного процесу спостерігача. Під кінець виводиться вікно експорту результатів (рис. 3.2). Ці результати необхідні для використання в Simulink-моделі синтезованої системи.

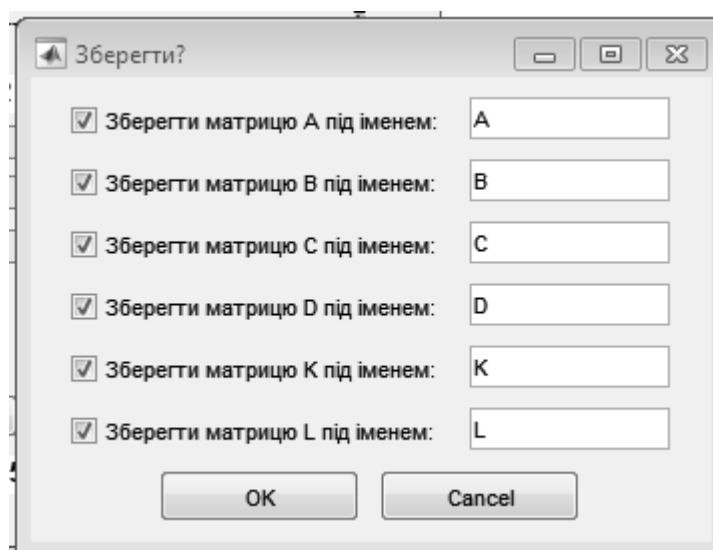


Рис. 2.2 – Вигляд вікна експорту результатів

Кінцевий вигляд вікна GUI показаний на рис. 3.3. Як можна побачити, час встановлення виходу спостерігача на порядок менший часу встановлення системи модального управління.

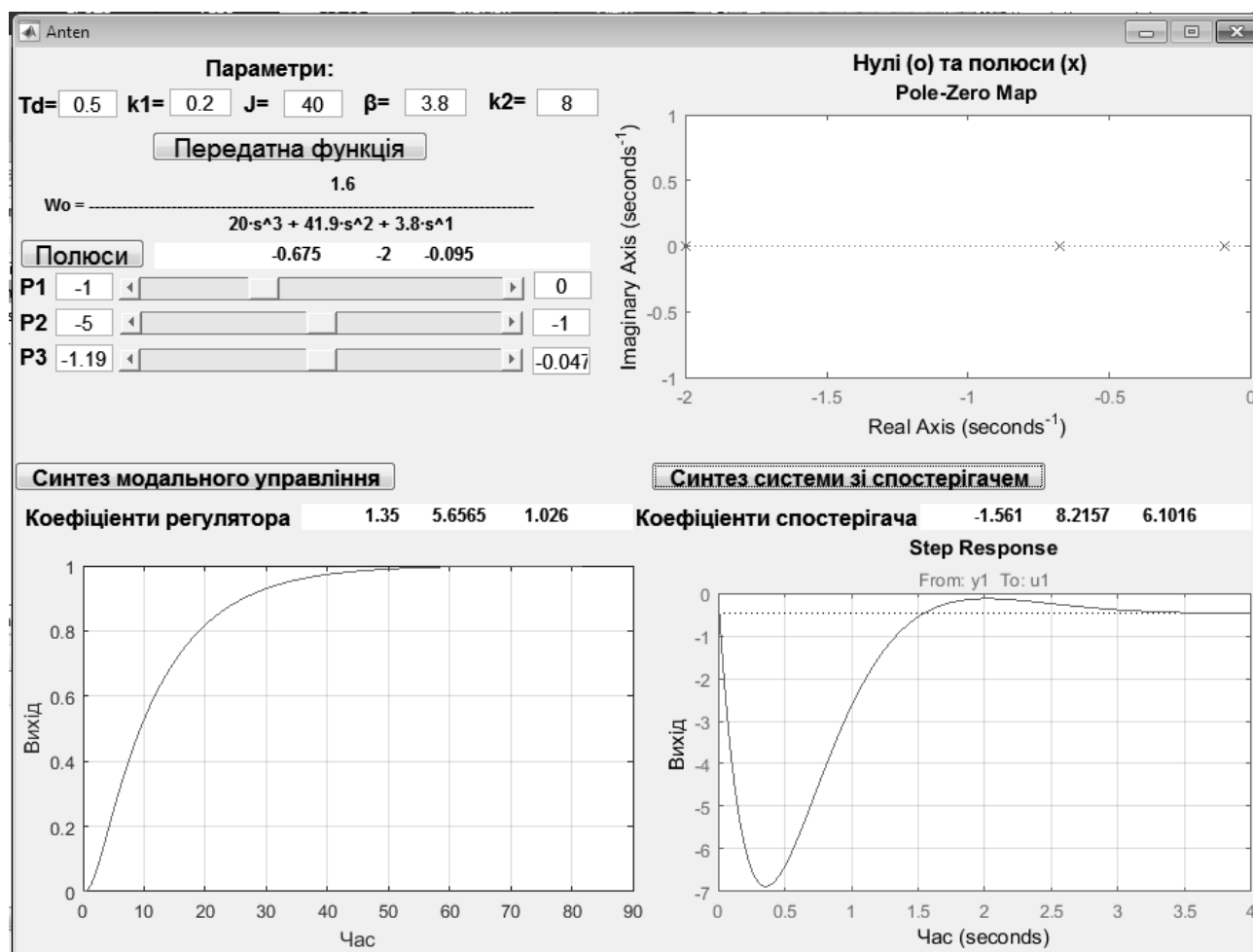


Рис. 3.3 – Результати розрахунків за допомогою GUI

4 ПЕРЕВІРКА РЕЗУЛЬТАТІВ МОДЕЛЮВАННЯМ У SIMULINK

Для аналізу отриманих результатів необхідно побудувати перехідні характеристики синтезованої системи, використовуючи пакет Simulink.

Модель спостерігача будується у відповідності з рівнянням:

$$\dot{\hat{X}} = A\hat{X} + BU + L(Y - C\hat{X}).$$

Для моделі об'єкта управління використовується блок State-Space з бібліотеки Continuous. Вигляд Simulink – моделі показаний на рис. 4.1.

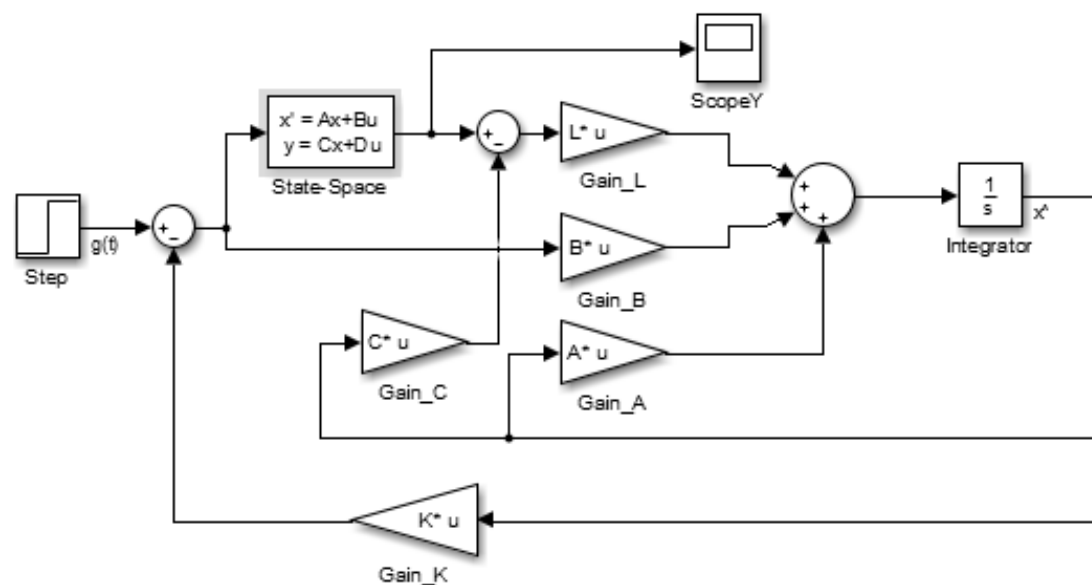


Рис. 4.1 – Simulink – модель синтезованої системи зі спостерігачем

Усі підсилювачі **Gain** налаштовані на матричне множення установкою параметра Multiplication: Matrix ($K*u$). Кожен з них виконує множення відповідно до рівнянь моделі: блок **Gain_A** – множення $A\hat{X}$, блок **Gain_B** – множення BU , блок **Gain_L** – множення $L(Y - C\hat{X}) = L(Y - \hat{Y})$, блок **Gain_C** – множення $\hat{Y} = C\hat{X}L(Y - C\hat{X}) = L(Y - \hat{Y})$, блок **Gain_K** – множення $K\hat{X}$. Двічі клацнувши по кожному блоку, треба занести у поле *Gain* ім'я відповідної матриці **A**, **B**, **C**, **L**, **K**.

Параметри блоку State-Space установлюємо відповідно рис. 4.2.

State Space

State-space model:
 $\frac{dx}{dt} = Ax + Bu$
 $y = Cx + Du$

Parameters

A:
A

B:
B

C:
C

D:
D

Initial conditions:
[0 0 0]

Absolute tolerance:
auto

State Name: (e.g., 'position')
"

? OK Cancel Help Apply

Рис. 4.2 – Налаштування блоку State-Space

Для блоку Step встановлюємо кінцеве значення 0.2.

Час моделювання вибираємо такий, щоб отримати на осцилографі значення вихідної величини, що встановилось. Для системи управління антеною це 50 с.

Перед початком моделювання виконуємо синтез системи і експорт матриць у робочий простір MATLAB згідно п. 2 і 3. Результат моделювання показаний на рис. 4.3.

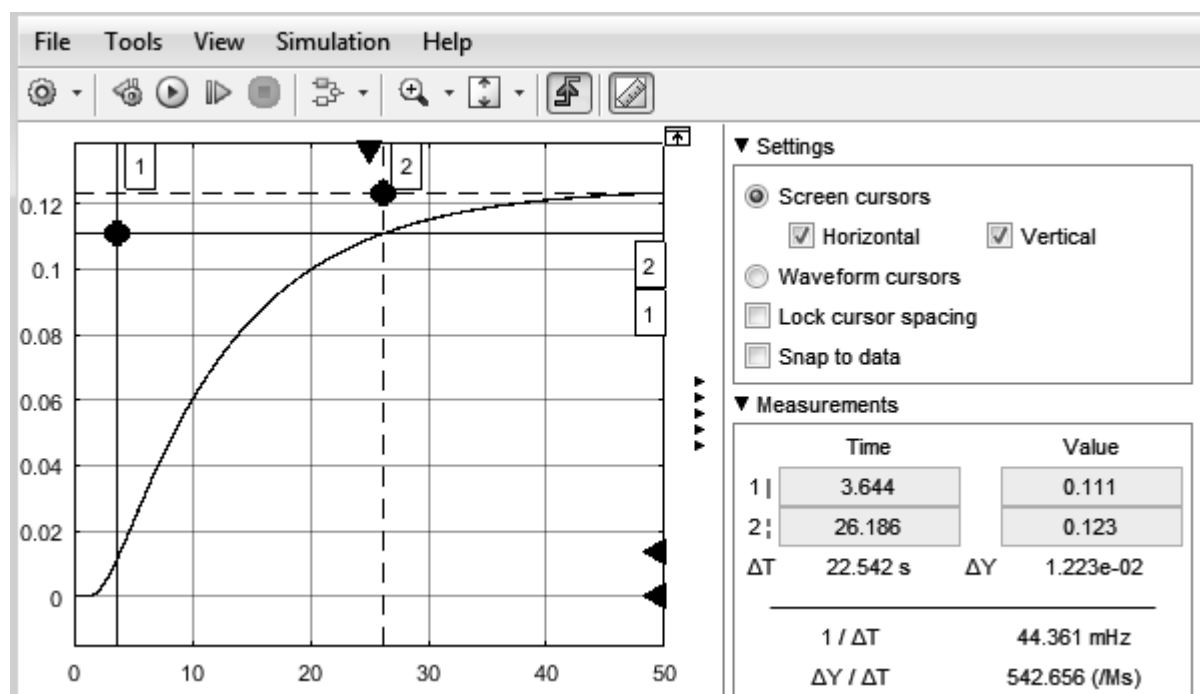



Рис. 4.3 – Результат моделювання з відкритим вікном Cursor Measurements

Вікно Cursor Measurement відкривається натисненням кнопки  або через меню Tools→Measurements →Cursor Measurements. Відкриваємо підвікно Settings і вибираємо варіанти Screen cursors, Horizontal, Vertical. Встановлюємо горизонтальне положення курсору 2 на кінцеве значення вихідної величини, а курсору 1 на значення мінус 10% від положення курсору 2 (можна контролювати по значенню Value для курсору 1). Вертикальне положення курсору 1 ставимо на перетин кривої виходу при її значенні 10% від кінцевого значення, а курсору 2 на перетин кривої з горизонтальною лінією курсору 1.

Далі оцінюємо параметри якості регулювання. Як бачимо, перерегулювання σ відсутнє. Час зростання читаємо як $\Delta T=22,542$ с, час встановлення як значення Time для курсору 2, тобто 26,186 с. Це менше заданого значення 30 с.

Щоб оцінити запас стійкості, треба побудувати діаграму Боде. У бібліотеці Simulink Control Design / Linear Analysis Plots вибираємо блок Bode Plot і переносимо його у модель. Клацнувши правою клавiшею миші по лініях зв'язку для вхідного і вихідного сигналів, вибираємо у контекстному меню Linear Analysis Points опції Input Perturbation і Output Measurement відповідно. Двічі клацаємо по доданому блоку і у вікні властивостей блоку тиснемо кнопки Get the linearization I/Os (показана стрілкою на рис. 4.4) і Show Plot. Виводиться вікно Bode Plot (рис. 4.5). На графіку тиснемо кружечок і читаємо значення запасу стійкості Gain Margin 33.7 dB.

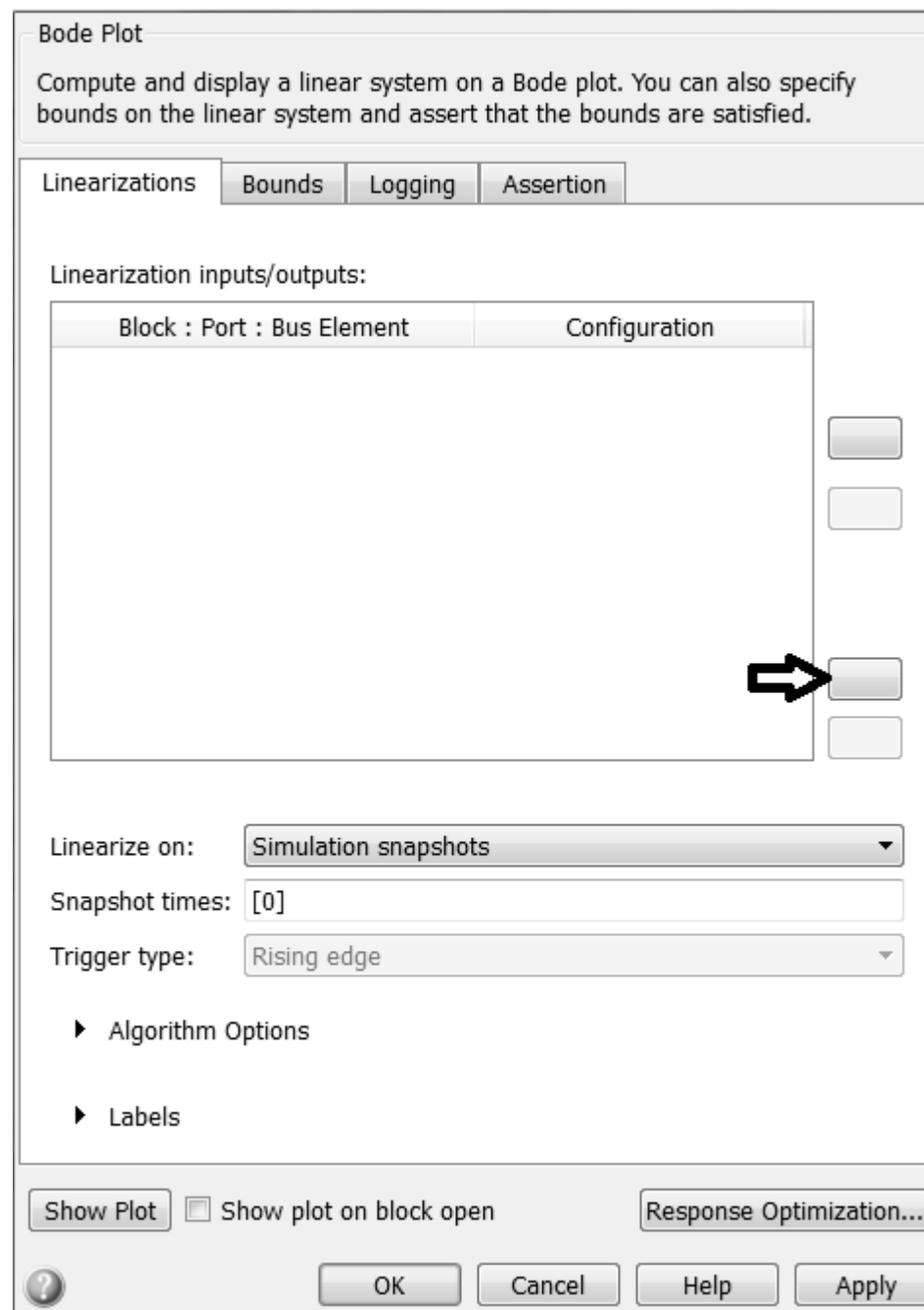


Рис. 4.4 – Вікно властивостей блоку Bode Plot

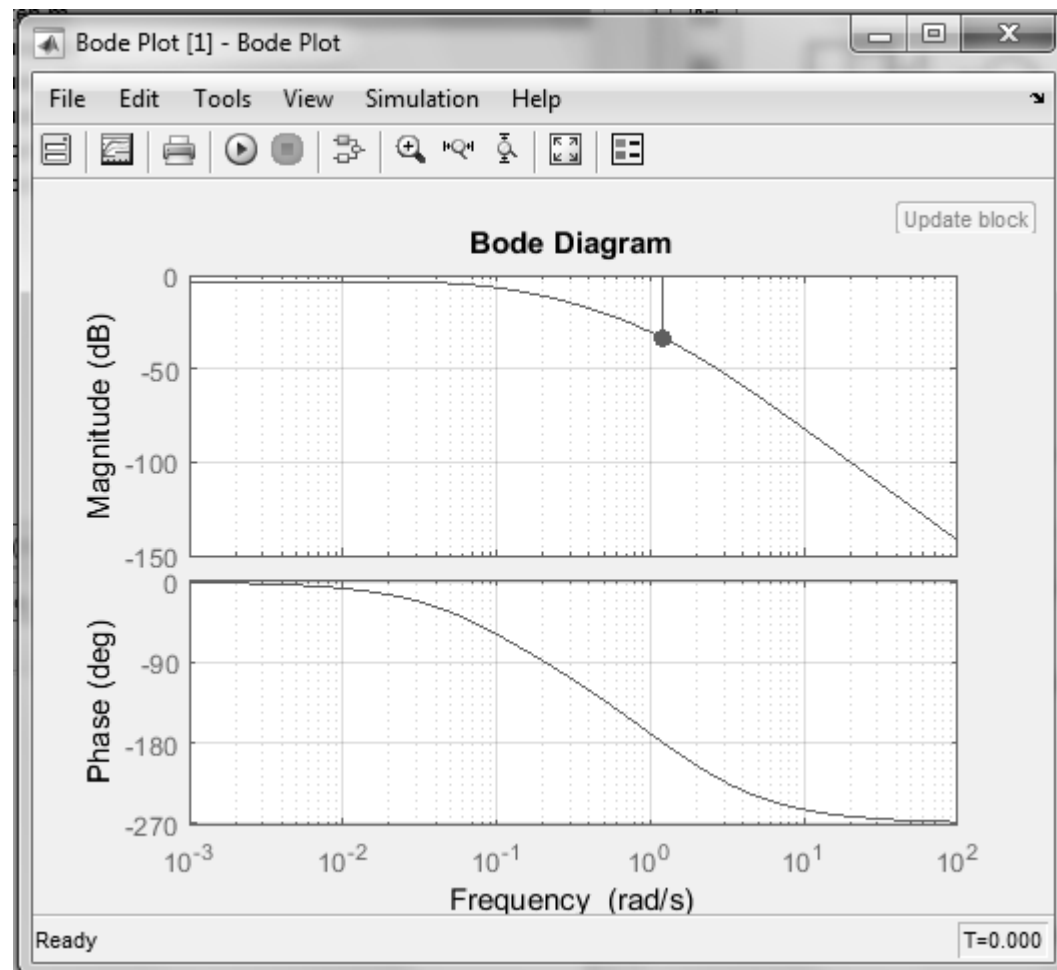


Рис. 4.5 – Діаграма Бодє

Таким чином, синтезована система управління антеною є стійкою з достатнім запасом стійкості.

ВИСНОВКИ

У курсовій роботі виконаний синтез системи автоматичного управління антеною РЛС, що відстежує рух цілі в горизонтальній площині. Для автоматизації розрахунків побудований графічний інтерфейс користувача, який дозволяє в інтерактивному режимі отримати параметри модального регулятора і спостерігача стану об'єкта управління.

Для оцінки якості синтезованої системи виконане SIMULINK-моделювання синтезованої системи. Це дозволило встановити, що синтезована система управління антеною є стійкою з достатнім запасом стійкості.

ЛІТЕРАТУРА

1. Энциклопедия Кругосвет. Радиолокация [Электронный ресурс] – Режим доступа: http://www.krugosvet.ru/enc/nauka_i_tehnika/voennaya_tehnika/RADIOLOKATSIYA.html.
2. Кондратенко, Ю.П. Синтез слідкуючих систем управління кутом повороту антени РЛС : навчальний посібник / Ю.П. Кондратенко, Г.В. Кондратенко, О.В. Коробко, О.В. Козлов. – Миколаїв : НУК ім. адм. Макарова, 2014. – 230 с.
3. Tsyukh, A.M. Accuracy refinement determination of objects angular coordinates in radio wave range / A. M. Tsyukh // Extension and Connection of Reference Frames Using Ground Based Technique. – Nikolaev, ATOLL. – 2001. – Pp. 208–214.
4. Гурко, О.Г. Аналіз та синтез систем автоматичного керування в MATLAB : Навчальний посібник / О.Г. Гурко, І.Ф. Єрьоменко. – Харків: ХНАДУ, 2011. – 286 с.
5. Дьяконов, В. П. MATLAB R2006/2007/2008 + Simulink 5/6/7. Основы применения. – М. : СОЛОН-ПРЕСС, 2010. – 800 с.
6. Никульчев, Е.В. Практикум по теории управления в среде MATLAB : Учебное пособие. – М. : МГАПИ, 2002. – 88 с.