МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УКРАЇНСЬКИЙ ДЕРЖАВНИЙ ХІМІКО-ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ»

МЕТОДИЧНІ ВКАЗІВКИ ДО ПРАКТИЧНИХ ЗАНЯТЬ З ДИСЦИПЛІНИ “СУЧАСНІ ПРОБЛЕМИ АВТОМАТИЗОВАНОГО УПРАВЛІННЯ” ДЛЯ СТУДЕНТІВ СПЕЦІАЛЬНОСТІ 151 – АВТОМАТИЗАЦІЯ ТА КОМП’ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ

Дніпропетровськ УДХТУ 2017

ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УКРАЇНСЬКИЙ ДЕРЖАВНИЙ ХІМІКО-ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ»

МЕТОДИЧНІ ВКАЗІВКИ
ДО ПРАКТИЧНИХ ЗАНЯТЬ З ДИСЦИПЛІНИ
“ СУЧАСНІ ПРОБЛЕМИ АВТОМАТИЗОВАНОГО УПРАВЛІННЯ ”
ДЛЯ СТУДЕНТІВ СПЕЦІАЛЬНОСТІ151 – АВТОМАТИЗАЦІЯ ТА КОМП’ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ

Затверджено на засіданні кафедри
комп’ютерно-інтегрованих
технологій і метрології.
Протокол № 6 від 21 грудня 2015 р.

Дніпропетровськ УДХТУ 2017

Методичні вказівки до практичних занять з дисципліни «Сучасні проблеми автоматизованого управління» для студентів спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології / Укл.: Г. І.Манко, Я. О. Довгополий. – Дніпропетровськ: УДХТУ, 2017. – 74 с.

УкладачіГ.І.Манко, канд. техн. наук,
Я. О. Довгополий

Відповідальний за випуск Ю. К. Тараненко, д. техн. наук

Навчальне видання

Методичні вказівки
до практичних занять
з дисципліни «Сучасні проблеми автоматизованого управління»
для студентів спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології

Укладачі: МАНКО Геннадій Іванович,
ДОВГОПОЛИЙ Ярослав Олександрович

Редактор Л.М. Тонкошкур
Коректор Л.Я. Гоцуцова

Підп. до друку ______________. Формат 60х84 1/16. Папір ксерокс. Друк різограф. Умовн.-друк. арк. ______Облік. – вид. арк. _______ Тираж _________прим. Зам. № ______ Свідотство ДК №303 від 27.12.2000.

УДХТУ, 49005, Дніпропетровськ-5, пр-т Гагаріна, 8

Видавничо-поліграфічний комплекс ІнКомЦентру

1 КОРОТКИЙ ОПИС РОБОТИ В MATLAB

1.1 Робота в командному режимі

Робота MATLAB можлива в двох режимах: командному режимі і режимі виконання m-файлу.

Робота системи в командному режимі нагадує роботу калькулятора. У командному рядку MATLAB після маркера>> можна відразу вводити початкові дані для обчислення за допомогою вбудованого текстового редактора. Введення даних завершується натисненням клавіші ENTER.

Результати обчислень виводяться з нового рядка (без маркера >>).

Для блокування виведення результатів обчислень деякого вираження його потрібно закінчити знаком «;» (крапка з комою). Це буває зручним для приховання результатів проміжних обчислень.

Як знак привласнення використовується знак рівності «=». Наприклад, введення рядка x = 1 означає оголошення і ініціалізацію змінної x.

Ім'я змінної може містити до 30 символів і повинне не співпадати з іменами функцій, процедур системи і системних змінних.

При цьому система розрізняє великі і маленькі букви в змінних, тобто змінні **unit** і **Unit**– це різні змінні.

Система MATLAB має декілька імен змінних, які використовуються самою системою і входять до складу зарезервованих:

i, j – уявна одиниця (корінь квадратний з мінус одиниці);

pi– число π (зберігається у вигляді 3.141592653589793);

inf– позначення машинної нескінченності;

NaN–«не число», позначення невизначеного результату (наприклад, типу0/0 чи inf/inf);

eps– погрішність операцій над числами з плаваючою комою;

ans– результат останньої операції без знаку привласнення;

realmax і **realmin** - максимально і мінімально можливі величини числа, які можуть бути використані.

Ці змінні можна використати в математичних виразах.

Вбудовані функції записуються малими літерами, а їх аргументи наводяться в круглих дужках, наприклад, sin(x).

У одному сеансі роботи системи MATLAB можна визначити декілька змінних і/або обчислити декілька виразів, наприклад:

Вводиться з клавіатури	`>> a=1, b=5, c=a+b`

Відповідь системи	`a=`
	`1`
	`b=`
	`5`
	`c=`
	`6`

Для виведення значення конкретної змінної досить ввести їїім'я, наприклад:

Вводиться з клавіатури	`>>a`
Відповідь системи	`a=` `1`

Якщо виведення результату обчислення тільки що введеного виразження не заблоковане і не вказана змінна, значення якої необхідно вивести, то система MATLAB сама призначає таку змінну з ім'ям ans, привласнює їй значення останнього виразження і виводить її як результат обчислень, наприклад:

Вводиться з клавіатури	`>>sin(pi/2)`
Відповідь системи	`ans=` `1`

1.1.1 Числа, матриці і вектори

Система MATLAB спочатку створювалася як «матричний обчислювач» і оптимізована для проведення обчислень з векторами, матрицями і масивами. Більше того, в системі за умовчанням передбачається, що кожна змінна - це вектор або матриця. Наприклад, якщо задане X = 1, то система сприймає це як завдання вектору з єдиним елементом, значення якого дорівнює 1. У свою чергу, вектор – це матриця, число рядків або стовпців якої дорівнює 1.

Для завдання вектору з великим числом елементів, їх значення потрібно перерахувати в прямокутних дужках, використовуючи як роздільник пропуск або кому. Завдання матриць здійснюється аналогічно, для розділення рядків використовується знак «;» (крапка з комою) Наприклад:

Завдання вектора з 4 елементів	`>>X =[1 2 3 4]`
Відповідь системи	`X =` `1 2 3 4`
Завдання матриці 2×2	`>>Y =[1 2; 4 5]`

Відповідь системи	`Y =`
	`1 2`
	`3 4`

У MATLAB передбачена наявність великої кількості операцій, які можна здійснювати з матрицями і векторами. Серед них:

- складання, віднімання (+,-);
- множення (*);
-обернення (**inv**);
- ділення зліва направо (/) і справа наліво (\);
- піднесення до степеня (^);
- транспонування (');
-створення нижньої трикутної матриці A: **tril(A)**;
-створення верхньої трикутної матриці A: **triu(A)**;
-формування матриці одиниць заданого розміру $n \times m$: **ones(n, m)**. Для створення квадратної матриці: **ones(n)**;

-формування матриці нулів заданого розміру $n \times m$: **zeros(n, m)**. Для створення квадратної матриці: **zeros(n)**;

-формування одиничної матриці заданого розміру n: **eye(n)**;

- формування матриціз одиницями на діагоналі і нулями поза діагоналлю:**eye(m,n)**;

- формування матриці з випадковими елементами, рівномірно розподіленими на відрізку [0,1]: **rand(m,n);**

-вилучення діагоналі заданої матриці A: **daig(A)**;

-власні числа матриці A: **eig(A)**;

- визначник матриці: **det(A)**.

1.1.2. Математичні функції

У MATLAB передбачена наявність великого числа вбудованих математичних функцій [1]. Серед них:

а)елементарні функції:

· **sin(Z), cos(Z), tan(Z), cot(Z), sec(Z), csc(Z)** - синус, косинус, тангенс, котангенс числа Z;

· **sinh(Z), cosh(Z), tanh(Z), coth(Z), sech(Z), csch(Z)** - гіперболічні синус, косинус, тангенс, котангенс, секанс, косеканс числа Z;

· **asin(Z), acos(Z), atan(Z), acot(Z), asec(Z), acsc(Z)** - арксинус (у радіанах в діапазоні від $-\pi/2$ до $+\pi/2$), арккосинус (у діапазоні від 0 до π), арктангенс (у діапазоні від $-\pi/2$ до $+\pi/2$), арккотангенс, арксеканс, арккосеканс;

· **asinh(Z), acosh(Z), atanh(Z), acoth(Z), asech(Z), acsch(Z)** - зворотні гіперболічні синус, косинус, тангенс, котангенс, секанс, косеканс;

· **exp(Z)** - експонента числа Z;

· **log(Z)** - натуральний логарифм;

· **log10(Z)** - десятковий логарифм;

· **sqrt(Z)** - квадратний корінь з числа Z;

· **abs(Z)** - модуль числа Z.

· **round(Z)** - звичайне округлення числа Z до найближчого цілого;

· **mod(X, Y)** - цілочислене ділення X на Y;

· **rem(X, Y)** - обчислення залишку від ділення X на Y;

· **sign(Z)** - обчислення сигнум-функції числа Z;

б)функції комплексного аргументу:

– практично усі вищеперелічені елементарні математичні функції обчислюються при комплексних значеннях аргументу і набувають в результаті цього комплексних значень результату. Наприклад, функція **sqrt** обчислює, на відміну від інших мов програмування, квадратний корінь з негативного аргументу, а функція **abs** при комплексному значенні аргументу обчислює модуль комплексного числа.

У MATLAB є декілька додаткових функцій, розрахованих тільки на комплексний аргумент :

real(Z) - виділяє дійсну частину комплексного аргументу Z;

imag(Z) - виділяє уявну частину комплексного аргументу;

angle(Z) - обчислює значення аргументу комплексного числа Z (у радіанах в діапазоні від $-\pi$ до $+\pi$);

conj(Z) - видає число, комплексно зв'язане відносно Z.

1.1.3 Рішення системи лінійних рівнянь алгебри

Розглянемо систему лінійних рівнянь алгебри :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1; \\ a_{21}x_1 + a_{22}x_2 = b_2. \end{cases}$$

У матричній формі ця система може бути записана у вигляді **AX = B**, де матриці **A, B, X** мають вигляд:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}; \ \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}; \ \mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

Рішення системи рівнянь в матричній формі запису представляється вираженням:

$\mathbf{X} = \mathbf{A}^{-1}\mathbf{B}$,

де $\mathbf{A}^{-1}$– матриця, зворотна матриці **A**.

У MATLAB це рішення може бути знайдене при виконанні команд:
`>>X=inv(A)*B`або`>>X=A\B`.

1.1.4 Оператори «:» (двокрапка) і «.» (крапка)

Оператор «:» (двокрапка) використовується для створення впорядкованої послідовності чисел з рівновіддаленими значеннями. Формат запису цього оператора:

[Початкове_значення]:[Крок]:[Кінцеве_значення]

Якщо крок не заданий, то він приймається рівним 1 або –1 залежно від того, більше кінцеве значення початкового значення або менше.

При його використанні створюється вектор:

Задається послідовність **k** з кроком 2	`>>k = 2:2:10`
Відповідь системи	`k =` `2 4 6 8 10`
Задається послідовність **m** з кроком 1	`>>m = 0:2`
Відповідь системи	`m =` `0 1 2`
Використання послідовності **m**	`>>s=sin(m)`
Відповідь системи	`s =` `0 0.8415 0.9093`

При необхідності здійснення почленного множення (чи ділення) елементів одного масиву на елементи іншого масиву (тієї же розмірності) використовуються відповідно оператори (.*) і (./). Наприклад:

Складання векторів **m, s**	`>>m+s`
Відповідь системи	`ans =` `0 1.8415 2.9093`
Почленне множення елементів **m, s**	`>>m.*s`

Відповідь системи	`j =` `0 0.8415 1.8186`

Для здійснення операції піднесення елементів масиву до степеня використовується операція «.^».

1.1.5 Побудова графіків

Графіки в MatLAB завжди виводяться в окреме графічне вікно, котре називають фігурою (figure). Перше вікно формується автоматично і має номер 1. Для створення додаткових вікон використовується команда **figure(n)**, де n – номер вікна.

У одному графічному вікні, але на окремих графічних полях можна побудувати декілька графіків, використовуючи процедуру

subplot(m,n,p).

Тут m вказує, на скільки частин розділяється графічне вікно по вертикалі, n – по горизонталі, а p – номер підвікна, в якому будуватиметься графік. При цьому підвікна нумеруються зліва направо по-рядково зверху вниз.

Для побудови графіків функцій може використовуватися функція **plot**.

Ця функція має різні форми запису залежно від набору вхідних аргументів. Якщо **y**– вектор, функція **plot(y)** робить побудову шматково-лінійного графіку елементів y залежно від індексу елементів **y**. Якщо функція **plot(x,y)** має два векторні аргументи однакової розмірності, то буде побудований графік **y** залежно від **x**.

Функція **plot** може мати декілька пар аргументів x–y. В цьому випадку вона зображує декілька функцій y(x) кривими різного кольору, використовуючи зумовлений (але змінюваний користувачем) список кольорів. Наприклад, **plot(x1, y1, x2, y2, x3, y3)**– вибудовуються три графіки функції : y1(x1), y2(x2), y3(x3).

Також є можливість визначати колір, стиль лінії і маркери за допомогою ще одного аргументу функції **plot**:

```
plot(x, y[,'color_style_marker']),
```

де color_style_marker– це символьний рядок (обмежений поодинокими лапками), що характеризує колір, стиль лінії і тип маркера:

-символи визначення кольору: 'c', 'm', 'y', 'r', 'g', 'b', 'w', 'k'. Вони відповідають кольорам cyan, magenta, yellow, red, green, blue, white, black;

- позначення стилю лінії: '**-**' суцільна, '**--**' пунктирна, '**:**' штрихова, '**-.**'штрихпунктирна;

- типи маркера: '+', 'o', '*', 'x'.

Наприклад, вираження: plot(x, y, 'r--o') креслить червону пунктирну криву і вставляє маркери у вигляді символу «o» в кожній точці даних.

Будь-яка з трьох перерахованих складових color-style-marker може бути пропущена. Якщо визначається тип маркера, а стиль лінії опущений, система MATLAB будує тільки маркери для цієї кривої.

На графік додатково можна нанести сітку з координатних ліній, інформацію про криву (заголовок графіку), а також позначити координатні осі.

Нанесення координатної сітки на графік здійснюється за допомогою функції **grid**, до якої слід звернутися відразу після звернення до функції **plot**:

```
>>plot(x, y), grid
```

Заголовок графіку виводиться за допомогою процедури **title**. Якщо після звернення до процедури **plot** викликати **title** таким чином:

```
title('<текст>')
```

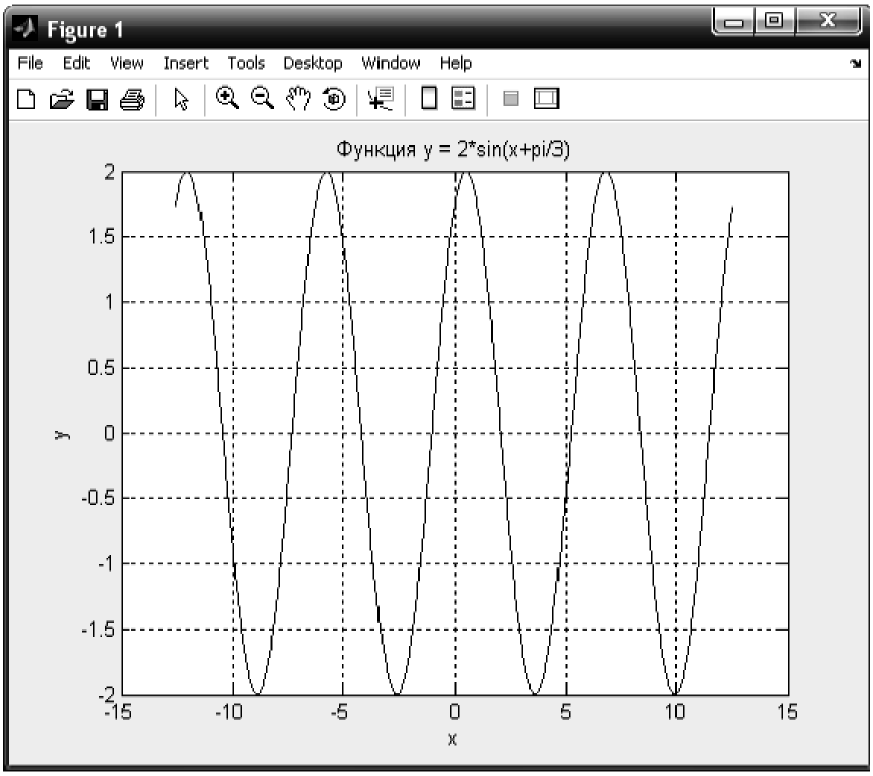
то над графіком з'явиться текст, записаний між апострофами в дужках. При цьому слід пам'ятати, що текст завжди повинен поміщатися в апострофи.

Аналогічно можна вивести пояснення до графіку, які розміщуються уздовж горизонтальної осі (функція **xlabel**) і уздовж вертикальної осі (функція **ylabel**).

Наприклад, сукупність операторів

```
>> x = - 4*pi: pi/100 : 4*pi;
>> y = 2* sin(x+pi/3);
>>plot(x, y), grid;
>>title('Функція y = 2*sin(x+pi/3)');
>>xlabel('x'); ylabel('y');
```

приведе до оформлення поля фігури у вигляді:



Команда **legend** служить для виведення легенди графіка. Легенда потрібна, якщо на графіку є декілька ліній і треба показати, що означає кожна з них. Параметрами команди **legend**є символьні рядки, їх повинно бути стільки, скільки побудовано ліній.

У написах можна використовувати грецькі букви, які записуються у виді «**\alpha**», «**\beta**» тощо. Верхній індекс (міра) позначається знаком «**^**». Наприклад, a^2 запишеться як «**a^2**». Для позначення нижнього індексу використовують нижнє підкреслення, наприклад, a_{22} кодується як «**a_{22}**».

Якщо до виведеного графіка треба додати новий у тих же осях, використовують команду **holdon**. Інакше новий графік заміщає старий.

1.2 Програмування в середовищі MATLAB

Програми мовою MATLAB мають два різновиди – так звані Script-файли (файли-сценарії, або програми, що управляють) і файли-функції (процедури). Обидва різновиди повинні мати розширення імені файлу «.m». За допомогою Script-файлів оформлюють основні програми, які керують від початку до кінця організацією усього обчислювального процесу, і окремі частини основних програм (вони можуть бути записані у вигляді окремих Script-файлів). Як файл-функції оформлюються окремі процедури і функції (тобто такі частини програми, які розраховані на неодноразове використання Script-файлами або іншими процедурами при змінених значеннях початкових параметрів і не можуть бути виконані без попереднього завдання значень змінних, які називають вхідними).

Надалі під М-файлом будемо розуміти будь-який файл (файл-функцію або Script-файл), записаний мовою системи MATLAB.

Основні особливості запису тексту програми (М-файлу) :

а) зазвичай кожен оператор записується в окремому рядку тексту програми. Закінчення введення рядка здійснюється натисненням клавіші <Enter>;

б) можна розміщувати декілька операторів в одному рядку, які відділяються один від одного символом «;» чи пробілом. Можна довгий оператор записувати в декілька рядків. При цьому попередній рядок оператора повинен закінчуватися трьома крапками«...»;

в) якщо черговий оператор не закінчується символом «;», результат його дії при виконанні програми буде виведений в командне вікно. Щоб запобігти виводу на екран результатів дії оператора програми, запис цього оператора в тексті програми повинен закінчуватися символом «;»;

г) рядок програми, що розпочинається з символу «%», не виконується,цей рядок сприймається системою MATLAB як коментар;

д) у програмах MATLAB відсутній символ закінчення тексту програми;

е) у мові MATLAB змінні не описуються і не оголошуються.Будь-яке нове ім'я, що з'являється в тексті програми при її виконанні, сприймається системою MATLAB як ім'я матриці. Розмір цієї матриці встановлюється при попередньому введенні значень її елементівабо визначається діями зі встановлення значень її елементів, описаними в попередніх операторах або процедурі;

ж) у мові MATLAB неможливе використання матриці або змінної, в якій заздалегідь не введені або не вичислені значення її елементів (при виконанні програми MATLAB з'явиться повідомлення про помилку – "Змінна не визначена").

1.2.1. Створення простих файл-функцій

Файл-функція (процедура) повинна розпочинатися з рядка заголовка :

function[*<OutVar>*] = *<ім'я_процедури>(<InVar>),*

де <OutVar>, <InVar> - відповідно, переліки вихідних і вхіднихвеличин файл-функції. Якщо перелік вихідних величин (OutVar) отримає тільки один об'єкт (у загальному випадку – матрицю), то файл-функціяє звичайною функцією (однієї або декількох змінних). Перший рядок в цьому випадку має вигляд:

function<ім'я змінної> = <ім'я процедури>(<InVar>).

Якщо ж в результаті виконання файл-функції мають бути визначені (вичислені) декілька об'єктів (матриць), така файл-функціяє вже складніший об'єкт. Загальний вигляд першого рядкав цьому випадку стає таким:

function[y1, y2, ..., y] = <ім'я процедури>(<InVar>),

тобто перелік вихідних величин *y1, y2, ...,* y має бути представленийяк вектор-рядок з елементами y1, y2, ..., y (усі вони можуть бути матрицями).

У простому випадку функції однієї змінної заголовок наберевигляду:

```
Functiony = func(x),
```

де func ім'я функції (М-файлу).

Приклад*:* скласти m-файл для обчислення функції

$$y = \ln(x + y) + \frac{x}{x + y}\ .$$

Процес створення m-файлупередбачає наступне.

1. Викликати меню **File** командного вікна MATLAB і вибрати в ньомуспочатку опцію**New**, а потім опцію **M-file**.

2. У вікні текстового редактора, що з'явилося, набрати текст:

```
function y = func(x, z)
y=log(x+z)+x./(x+z);
```

3. Зберегти цей текст у файлі під ім'ям func.m. Необхідний М-файл створений.

Тепер можна користуватися цією функцією при розрахунках. Наприклад:

```
>> y = func(1, 1)
y =
1.1931
>> x1=0:1:5;
>> x2=1:1:6;
>>y=func(x1, x2)
y =
0 1.4319 2.0094 2.3745 2.6417 2.8524
```

При створенні файл-функцій необхідно пам'ятати деякі особливості:

а) з точки зору зони видимості змінні у файл-функції є *локальними*, тому імена змінних, використовувані у файл-функції, можуть не співпадати з іменами відповідних змінних при зверненні до цієї файл-функції;

б) з метою використання у файл-функції глобальної змінної (деякої змінної робочого простору Workspace) необхідно в діалоговому режимі (чи в Script-файлі) і файл-функції оголосити цю змінну з атрибутом **global**. Якщо в одному рядку оголошуються декілька змінних як глобальні, вони повинні відділятися одна від одної пропусками;

в) можливість використання файл-функції як для окремих чисел, так і для векторів і матриць обумовлена застосуванням в записі відповідного М-файлу замість звичайних знаків арифметичних дій їх аналогів з попередньою крапкою (« .* », « ./», « .^» та ін.).

1.2.2 Створення Script-файлів

При створенні Script-файлів враховуються наступні особливості:

а) Script-файли є незалежно (самостійно) виконуваними блоками операторів і команд;

б) усі використовувані змінні утворюють так званий робочий простір (Workspace), який є загальним для усіх виконуваних Script-файлів; з цього виходить, що при виконанні декількох Script-файлів імена змінних в них мають бути погоджені, оскільки одне ім'я означає в кожному з них один і той же об'єкт обчислень;

в) у них відсутній заголовок, тобто перший рядок певного виду і призначення (на відміну від файл-функцій, які розпочинаються із заголовка «function»);

г) звернення до них не вимагає вказівки ніяких імен змінних: усі змінні формуються в результаті виконання програми або сформовані раніше і існують в робочому просторі.

1.2.3Введення і виведення інформації в діалоговому режимі

Для забезпечення взаємодії з користувачем в процесі виконання m-файлу використовуються такі команди:

disp, sprintf, fprintf, input, menu, keyboard, pause.

Команда **disp** здійснює виведення значень вказаної змінної або вказаного тексту в командне вікно. Звернення до неї має вигляд:

disp(<змінна або текст в апострофах>)

Наприклад:

`disp(x1)` – в командне вікно виводиться значення змінної $x1$.

`disp('value')` – в командне вікно виводиться текст `value`.

Щоб вивести значення декількох змінних в один рядок (наприклад, при створенні таблиць даних), треба створити єдиний об'єкт, який містив би усі ці значення. Це можна зробити, об'єднавши відповідні змінні у вектор, користуючись операцією створення вектора-рядка:

```
x =[x1 x2 ... x].
```

Тоді виведення значень декількох змінних в один рядок будемати вигляд:

```
disp([x1 x2 ... x]).
```

Наприклад:

```
>> x1=-3.14; x2=-2.5; x3=5.6; x4=-9.33;
>>disp([x1 x2 x3 x4])
-3.1400 -2.5000 5.6000 -9.3300
```

Для одночасного виведення символьної і цифрової інформації в командне вікно зручно використати функцію sprintf, наприклад:

```
>>disp(sprintf('Параметр 1= %g Параметр2= %g', x1, x2))
Параметр1 = - 3.14 параметр2= - 2.5
```

Аналогічно працює функція **fprintf**, де як перший параметр для виведення інформації в командне вікно використовується 1, наприклад:

```
>>fprintf(1,'Параметр1 = %g Параметр2= %g ', x1, x2)
Параметр1 = - 3.14 параметр2= - 2.5
```

Введення інформації з клавіатури в діалоговому режимі можна здійснити за допомогою функції **input**:

x = input('<пояснюючий_текст>').

При виконанні цієї функції програма чекає введення інформації з клавіатури. Після закінчення введення, яке визначається натисненням клавіші <Enter>, введена інформація запам'ятовується в програмі під ім'ям «x», і виконання програми триває.

Команда **pause** тимчасово припиняє виконання програми до тих пір, поки користувач не натисне будь-яку клавішу клавіатури.

1.2.4 Галуження і цикли

У мові MATLAB для організації галуження використовуються команди **if**та**switch**.

Загальна форма запису оператора **if** має вигляд

if<умова 1>
<група операторів>;
[elseif<умова 2>]
[<група операторів>;]
else
<група операторів>;
end

Приклад.СтворитиМ-файл-функцію, що здійснює визначення знаку числа:

$$\operatorname{sgn}(x) = \begin{cases} 1, & x > 0; \\ 0, & x = 0; \\ -1, & x < 0. \end{cases}$$

Функція розташовуватиметься в М-файлі*signum.m*.

```
function y = signum(x)
if x > 0
y = 1;
elseif x == 0
y = 0;
else
y = - 1;
end
```

Команда **switch** дозволяє здійснювати галуження за декількома умовами, причому умови розглядаються на рівність. Нижче представлений простий приклад з розділенням на три умови формування параметра введення.

```
Functionz = count(x)
switch x
case 1
z = 'one';
case 2
z = 'two';
otherwise
```

```
z = 'many';
end
```

Тут вираження **switch** обчислює **x**, а потім виконання програми у файлі переходить туди, де вираження **case** має те ж саме значення, що й **x**. Таким чином, якщо параметр введення **x** має значення, рівне 1, тоді параметр виводу **z** визначається у вигляді рядка 'one', якщо **x** дорівнює 2, тоді **z** визначається у вигляді рядка 'two'.

Після виконання усіх команд, що йдуть за вибраною умовою, програма MATLAB зустрічає або інше вираження **case**, або вираження **otherwise**, що призводить до переходу виконання програми до вираження **end**.

Для організації циклів використовуються команди **for** і **while**.

Організація циклу за допомогою команди **for** виглядає таким чином:

for n=n_0: [step:] n_k
[тіло циклу]
end

Цикл розпочинається з вираження **for** і закінчується вираженням **end**.

Величина n_0 – початкове значення змінної циклу, step– крок, n_k – кінцеве значення змінної циклу.

Фрагмент коду для визначення факторіалу 10! представлений нижче. Він може бути виконаний безпосередньо в командному режимі:

```
f=1;
for n=2:1:10
f=f*n;
end
f
```

Відповідь системи:

```
f=
3628800
```

Організація циклу за допомогою команди **while** виглядає таким чином:

while<умова>
[тіло циклу]
end

Цикл **while** триває до тих пір, поки <умова> істинно.

Приклад: вичислити ряд $\sum \frac{1}{n^2}$ з точністю 10^{-9}.

```
n = 1;
oldsum = - 1;
newsum = 0;
while (newsum - oldsum)>1e-9
oldsum = newsum;
newsum = newsum + 1/(n^2);
n = n +1;
end
newsum
```

Відповідь системи:

```
newsum =
```

```
1.6449
```

Іноді буває необхідно, щоб програма MATLAB передчасно вийшла з циклу, наприклад, при виникненні певної умови. Для цієї мети використовується команда **break**.

1.3Задачі для самостійного розв'язання

Задача 1.1. Задайте матрицю **A**:

$$\begin{bmatrix} 3.15 & -2.08 & 6.11 \\ 10.10 & 0.25 & -4.78 \\ 7.03 & -45.2 & 8.46 \end{bmatrix}.$$

Згенеруйте масив **B** розміром 3×3 з випадковими елементами, рівномірно розподіленими на інтервалі від 0 до 1. Виконайте дії:

а) $\mathbf{A}\cdot n\cdot\mathbf{B}$; $\mathbf{A}\cdot\mathbf{B}^{T}$,де n – номер варіанта;

б) почленне множення **A** на **B**;

в) визначити максимальний і мінімальний елементи матриці **B**;

г) обчислити визначник матриці **B**.

Задача 1.2. Задайте вектор-стовпчик **C**, трьома елементами якого є числа 0, n, – n,де n – номер варіанта. Вирішити систему лінійних алгебраїчних рівнянь:

$$\mathbf{A}\cdot\mathbf{X} = \mathbf{C}.$$

Виконайте перевірку правильності рішення підстановкою **X** у рівняння.

Задача 1.3. Побудуйте два графіки у рамках одних осей координат:

$$y = e^{-x^2};$$

$$z = \mathrm{arctg}(x^{1/2});$$

$$x \in [0, 4\pi].$$

Зробіть написи на осях, заголовок для графіку, напис пояснення на рисунку. Задайте самостійно тип ліній і колір.

Задача 1.4. Побудуйте графіки функцій y(x) і z(x) задачі 1.3 в різних підобластях одного графічного вікна. Інтервали зміни для x визначте самостійно.

Задача 1.5. Створіть Script-файл для розв’язання задач 1.1, 1.2, 1.3. Для виведення результатів використайте команди **disp, sprintf** та **fprintf.** У зошиті з практичних робіт наведіть текст скріпту, графіки і скріншот вікна команд MATLAB з результатами обчислень.

2 ДОСЛІДЖЕННЯ ЛІНІЙНОЇ СТАЦІОНАРНОЇ ДИНАМІЧНОЇ СИСТЕМИ В СЕРЕДОВИЩІ MATLAB

2.1 Короткі теоретичні положення

2.1.1 Моделі лінійних систем

Для опису лінійних стаціонарних систем можуть застосовуватися декілька способів:

а) диференціальні рівняння;

б) моделі в просторі станів;

в) передатні функції (ПФ);

г) моделі виду «нулі-полюси-коефіцієнти передачі»;

Перші два способи описують поведінку системи в часовій області і відображають внутрішні зв'язки між сигналами. Передатні функції і моделі виду «нулі-полюси» відносяться до частотних способів опису, оскільки безпосередньо пов'язані з частотними характеристиками системи і відображають тільки вхід-вихідні властивості.

Лінеаризована модель об’єкта може бути описана диференціальним рівнянням виду:

$$a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \ldots + a_1 \frac{dy}{dt} + a_0 y = b_m \frac{d^m u}{dt^m} + b_{m-1} \frac{d^{m-1} y}{dt^{m-1}} + \ldots + b_1 \frac{du}{dt} + b_0 u$$

за нульових початкових умов. Причому $m \leq n$.

Модель в просторі станів пов'язана із записом диференціальних рівнянь у формі Коши (у вигляді системи рівнянь першого порядку):

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{By};$$

$$\mathbf{y} = \mathbf{Cx} + \mathbf{Du},$$

де **x** - вектор змінних стану розміру n; **u** - вектор вхідних сигналів (вектор управління) розміру m і **y** - вектор вихідних сигналів розміру p. Крім того, **A, B, C** і **D** - постійні матриці.

Згідно з правилами матричних обчислень, матриця **A** має бути квадратною розміру n×n, матриця **B** має розмір n×m, матриця **C** - p×n і матриця **D** - p×m . Для систем з одним входом і одним виходом матриця **D** – скалярна величина.

Передатна функція $W(s)$ лінійної стаціонарної системи як функція від комплексної змінної s визначається відношенням перетворення Лапласа виходу до перетворення Лапласа входу за нульових початкових умов:

$$W(s) = \frac{Y(s)}{U(s)} .$$

Для об'єкта, що описується представленим вище диференціальним рівнянням, передатна функція має вигляд:

$$W(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0} .$$

З цієї передатної функції можна побудувати модель у формі «нулі-полюси». Нулями називаються корені чисельника, полюсами – корені знаменника ПФ.

2.1.2 Динамічні характеристики лінійних систем

Динамічні властивості систем характеризують реакції на вхідні дії спеціального виду. Динамічні характеристики систем підрозділяються на часові і частотні.

Часові характеристики. Імпульсною характеристикою (ваговою функцією) $w(t)$ називається реакція системи на одиничний нескінченний імпульс (дельта-функція або функція Дирака) за нульових початкових умов. Дельта-функція $\delta(t)$ визначається наступним чином:

$$\delta(t)=\begin{cases}0, & t\neq 0,\\ \infty, & t=0,\end{cases}\qquad \int_{-\infty}^{\infty}\delta(t)\,dt=1.$$

Це узагальнена функція – математичний об'єкт, що є ідеальним сигналом, який можна розглядати як межу прямокутного імпульсу одиничної площі з центром в точці $t = 0$ при прагненні ширини імпульсу до нуля.

Перехідною характеристикою (перехідною функцією) $h(t)$ називається реакція системи (за нульових початкових умов) на одиничний ступінчастий сигнал (одиничний стрибок) :

$$1(t)=\begin{cases}0, & t<0;\\ 1, & t\geq 0.\end{cases}$$

Імпульсна $w(t)$ і перехідна $h(t)$ функції пов'язані виразами

$$w(t)=\frac{dh(t)}{dt};\;\; h(t)=\int_0^t w(\tau)\,d\tau.$$

Для систем без інтеграторів перехідна характеристика прагне до постійного значення. Перехідна характеристика системи з диференційною ланкою (чисельник передатної функції має нуль в точці $s = 0$) прагне до нуля. Якщо система містить інтегруючі ланки, перехідна характеристика асимптотично прагне до прямої, параболи і так далі, залежно від кількості інтеграторів.

За перехідною характеристикою можна знайти найважливіші показники якості системи – перерегулювання (*overshoot*) і час перехідного процесу (*settling time*).

Перерегулювання визначається як

$$\sigma=\frac{h_{\max}-h_{\infty}}{h_{\infty}}\times 100\%,$$

де $h_{\max}$– максимальне значення функції $h(t)$, а h_{∞}– значення виходу, що встановилося.

Час перехідного процесу – це час, після якого сигнал виходу відрізняється від значення, що встановилося, не більше, ніж на задану малу величину (у середовищі MATLAB за умовчанням використовується точність 2%).

Частотні характеристики. При дослідженні стійкості динамічних систем і проектуванні регуляторів широке застосування отримали частотні характеристики.

При поданні на вхід лінійної системи гармонійного (синусоїдального) сигналу $u(t) = \sin(\omega t)$ з частотою ω (вона вимірюється в радіанах за секунду), на виході буде також гармонійний сигнал тієї ж частоти, але іншої амплітуди і фази $y(t) = A\sin(\omega t + \varphi)$, де A– амплітуда і φ – зсув фази.

Для побудови частотної характеристики потрібно використати підстановку $s = j\omega$ у ПФ $W(s)$. Вираження $W(j\omega)$ називається частотною передатною функцією або амплітудно-фазовою частотною характеристикою системи (АФЧХ).

Залежність модуля величини $W(j\omega)$ від частоти називається амплітудною частотною характеристикою (АЧХ), а залежність аргументу комплексноїфункції $W(j\omega)$ (фази) від частоти – фазовою частотною характеристикою (ФЧХ):

$$A(\omega) = \left| W(j\omega) \right|, \quad \varphi(\omega) = \arg W(j\omega) = \mathrm{arctg}\frac{\mathrm{Im}\, W(j\omega)}{\mathrm{Re}\, W(j\omega)}.$$

АЧХ показує, наскільки посилюється амплітуда сигналів різних частот після проходження через систему, а ФЧХ характеризує зсув фаз сигналів різних частот.

На практиці широке застосування отримала діаграма Боде (логарифмічна амплітудна характеристика – ЛАХ), яка визначається як $L(\omega) = 20\lg A(\omega)$. Вимірюється ЛАХ в децибелах і будується як функція від $\lg(\omega)$.

2.1.3 Пакет Control System Toolbox

До складу системи MATLAB входить пакет програм Control System Toolbox, який призначений для роботи з LTI-моделями (Linear Time Invariant Models – лінійні інваріантні в часі моделі) систем управління.

Для побудови ПФ використовується функція **tf**. Вона має вигляд: `tf(B, A)`, де **B** – вектор коефіцієнтів чисельника передатної функції, записаних по убуванню степеня s; **A** – вектор коефіцієнтів знаменника передатної функції.

З передатної функції можна створити модель у формі «нулі-полюси-коефіцієнт передачі» :

```
>>f_zpk = zpk(f)
Zero/pole/gain:
          2 (s+2)
------------------------------
(s+1) (s^2 - 0.2s + 2.26)
```

Нулями називаються корені чисельника, полюсами – корені знаменника ПФ. Ця функція має нуль в точці $s = -2$ і три полюси в точках $s = -1$ і $s = -0{,}1 \pm 1{,}5i$. Парі комплексних полюсів відповідає квадратний тричлен $s^2 - 0{,}2s + 2{,}26$.

Для визначення нулів і полюсів передатної функції використовуються відповідно функції **zero** і **pole** :

```
>>zero(f)
ans =
-2
>>pole(f)
ans =
0.1000 + 1.5000i
0.1000 - 1.5000i
-1.0000
```

Для перетворення ПФ у модель у просторі станів використовується команда:

```
>> f_ss = ss(f)
a =
x1        x2        x3
x1   - 0.8          - 1.03  - 1.13
x2   2        0        0
```

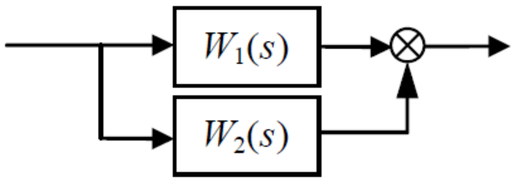
```
x3   0         1         0
b =
u1
x1   2
x2   0
x3   0
c =
x1   x2   x3
y1   0    0.5  1
d =
u1
y1   0
Continuous - time model.
```

Це означає, що матриці моделі мають вигляд:

$$A=\begin{bmatrix}-0,8 & -1,03 & -1,13\\ 2 & 0 & 0\\ 0 & 1 & 0\end{bmatrix},\ B=\begin{bmatrix}2\\0\\0\end{bmatrix},\ C=\begin{bmatrix}0 & 0,5 & 1\end{bmatrix},\ D=0\,.$$

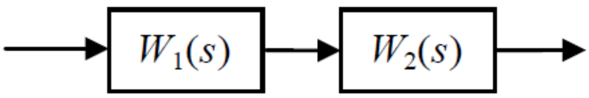
На практиці досить часто реальні системи автоматичного управління (САУ) представляються у вигляді сполучених між собою окремих динамічних ланок.У MATLAB передбачена можливість програмно будувати схему САУ шляхом попереднього введення моделей ланок і подальшого з'єднання цих ланок в єдину структуру. До процедур, що здійснюють розрахунок характеристик з'єднань окремих ланок, відносяться:

а) **plus (minus)** або **«+» («–»)** – виконує паралельне з'єднання вказаних при зверненні ланок, тобто визначає характеристики моделі системи, що складається з паралельно сполучених ланок, наприклад W_1 і W_2 :



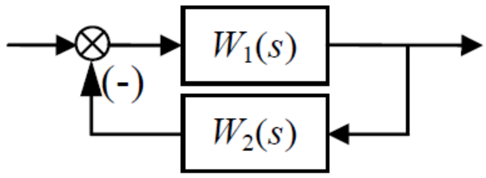
б) **parallel -** здійснює ту ж процедуру паралельного з'єднання ланок; на відміну від попередньої процедури може використовуватися для багатовимірних систем і для здійснення паралельного з'єднання лише по деяких входах і виходах;

в) **mtimes** або «*»– здійснює послідовне з'єднання ланок, імена яких вказані (застосовується для одновимірних систем) :



г) **series** – послідовне з'єднання (може використовуватися для багатовимірних систем);

д) **feedback –** з'єднання із зворотним зв'язком, тобто таке з'єднання двох ланок, коли друга вказана ланка складає коло негативного зворотного зв'язку для першої ланки:



Пакет Control System Toolbox надає можливості побудови часових і частотних характеристик систем з використанням функцій, приведених в таблиці 2.1.

Таблиця 2.1 – Функції побудови характеристик систем

Синтаксис	Опис
step(<LTI - об'єкт >)	Побудова графіка перехідної функції
impulse(<LTI - об'єкт>)	Побудова графіка імпульсної перехідної функції (вагової функції)
bode(<LTI - об'єкт >)	Побудова логарифмічних частотних характеристик (діаграми Боде)
nyquist(<LTI - об'єкт >)	Побудова частотного годографа Найквіста

Іншим варіантом отримання графіків динамічних характеристик системи є використання графічного вікна – LTI Viewer. Виклик його здійснюється командою **ltiview,** у якій, як параметр, можна вказати ім'я змінної, що містить LTI-об'єкт, наприклад:

```
>>ltiview(f)
```

Завантаження досліджуваного об’єкта можна здійснити також безпосередньо з вікна LTI Viewer, викликавши меню **File,** а потім команду **Import.**

Використовуючи команду **Plot Configuration,** з меню **Edit** можна вибрати конфігурацію області відображення графіків (кількість і розташування вікон побудови графіків), а також види вибудуваних характеристик системи (**Step**, **Impulse, Bode** та ін.) – розділ *Responsetype* (рис. 2.1).

Щоб побудувати частотну характеристику можна використати функцію **freqresp(f, w),** де **f -** модель лінійної системи у вигляді передатної функції, в просторі станів або у формі «нулі-полюси-коефіцієнт посилення», **w -** вектор частот**.**Заздалегідь до використання цієї функції необхідно створити масив частот в потрібному діапазоні за допомогою функцій **linspace** і **logspace.** Перша функція здійснює рівномірний розподіл точок за лінійною шкалою, друга –за логарифмічною.

Процедура побудови АЧХ системи на ділянці частот [0, 10] має вигляд:

```
>> w= linspace (0, 10, 100);
>>freq = freqresp(f, w);
>>freq = freq(:);
>>plot(w, abs(freq));
```

Для обчислення фази (у градусах) і побудови її на графіку можуть використовуватися команди:

```
>>phi = angle(freq)*180/pi;
>>plot (w, phi);
```

2.2 Приклади і задачі для самостійного розв'язання

Задача 2.1. Необхідно сформувати передатну функцію

$$F(s) = \frac{2s+4}{s^3 + 0{,}8s^2 + 2{,}06s + 2{,}26}\ .$$

Розв'язування

В даному випадку вектори коефіцієнтів чисельника і знаменника передатної функції мають вигляд: B=[2 4], A=[1 0.8 2.06 2.26].

Процедура утворення передатної функції має вигляд:

```
>>B=[2 4];
>>A=[1 0.8 2.06 2.26];
>>f=tf(B, A)
```

У пам'яті створюється об'єкт класу **tf,** що описує передатну функцію, і на екрані з'явиться передатна функція у вигляді:

```
Transfer function:
           2 s + 4
-----------------------------
s^3 + 0.8 s^2 + 2.06 s + 2.26
```

Визначення передатної функції можна також здійснити без попередньої побудови чисельника і знаменника :

```
>>f=tf([2 4], [1 0.8 2.06 2.26])
```

Задача 2.2. Розробити модель замкнутої системи автоматичного управління, в якій об'єкт управління представляється у вигляді з'єднання трьох аперіодичних ланок першого порядку з передатними функціями

$$W_1 = \frac{2}{3s+1}\ ,\ W_2 = \frac{1}{2s+1}\ ,\ W_3 = \frac{1}{4s+1}\ ,$$

а в зворотний зв'язок включений регулятор з передатною функцією

$$R = 0{,}5 + \frac{0{,}2}{s}\ .$$

Розв'язування

```
>> w1=tf([2],[3 1]) %формування ланки w1
>> w2=tf([1],[2 1]) %формування ланки w2
>> w3=tf([1],[4 1]) %формування ланки w2
>> w=w1*w2*w3 %формування об’єкта
>>R=tf([0.5],[1])+tf([0.2],[1 0]) %формування регулятора
>>W_zs= feedback(w, R) %формування замкнутої системи
>>step(W_zs) %побудова відгуку системи
```

Задача 2.3. Виконати аналіз системи автоматичного управління, передатна функція якої наведена у таблиці. 2.2.

Таблиця 2.2 – Вихідні дані до задачі 2.3

Варіант	Передатна функція	b_2	b_1	b_0	a_4	a_3	a_2	a_1	a_0
1	$W(s)=\dfrac{b_2s^2+b_1s+b_0}{a_4s^4+a_3s^3+a_2s^2+a_1s+a_0}$	1	2	2	1	3	4	5	3
2		1	3	1	2	2	4	2	1
3		1	2	1	1	3	5	4	7
4		1	0	1	1	2	5	4	3
5		1	2	3	1	3	4	5	2
6		1	1	2	2	3	3	4	4
7	$W(s)=\dfrac{b_2s^2+b_1s+b_0}{a_3s^3+a_2s^2+a_1s+a_0}$	1	2	1	-	2	5	4	3
8		2	1	1	-	2	4	5	1
9		1	3	2	-	1	5	2	3
10		2	1	2	-	1	5	5	2
11		1	5	1	-	1	5	4	2
12		2	3	1	-	2	3	4	1
13	$W(s)=\dfrac{b_2s^2+b_1s}{a_3s^3+a_2s^2+a_1s+a_0}$	1	1	-	-	2	5	4	3
14		1	3	-	-	2	4	5	1
15		3	1	-	-	1	5	2	3
16		1	4	-	-	1	5	5	2
17		1	2	-	-	1	5	4	9
18		3	2	-	-	2	4	3	2
19	$W(s)=\dfrac{b_2s^2+b_1s+b_0}{a_3s^3+a_2s^2+a_1s}$	1	1	1	-	1	1	5	-
20		2	0	1	-	2	3	2	-
21		0	2	1	-	1	3	3	-
22		0	3	2	-	2	2	1	-
23		1	0	1	-	2	1	2	-
24		1	2	1	-	1	2	2	-
25		2	3	0	-	2	0	1	-

Порядок проведення аналізу наступний.

1. Запустіть середовище MATLAB і в командному режимі очистіть робочий простір і вікно:

```
>>clear all;clc
```

2. Введіть передатну функцію як об'єкт **tf** за зразком:

```
>> A =[a4 a3 a2 a1 a0]
>> B =[b2 b1 b0]
>> f = tf(B, A)
```

3. Знайдіть нулі, полюси передатної функції і коефіцієнт посилення:

```
>> z = zero(f)
>> p = pole(f)
>> k = dcgain(f)
```

Показати полюси на комплексній площині.

4. Побудуйте модель початкової системи у формі «нулі-полюси»:

```
>>f_zpk = zpk(f)
```

5. Побудуйте модель системи в просторі стану :

```
>> f_ss = ss(f)
```

Напишіть у зошиті відповідне матричне рівняння.

6. У командному режимі побудуйте перехідну характеристику:

```
>>step(f)
```

7. У командному режимі побудуйте імпульсну характеристику:

```
>>impulse(f)
```

8. Побудуйте логарифмічні частотні характеристики (діаграми Боде) :

```
>>bode(f)
```

9. Побудуйте амплітудно-фазову частотну характеристику (частотний годограф Найквіста) :

```
>>nyquist(f)
```

10. Запустіть графічне вікно LTI Viewer :

```
>>ltiview
```

11. Завантажте об'єкт **f.** Для цього в меню **File** необхідно вибрати пункт **Import**, а далі вибрати об'єкт **f**.

12. Побудуйте вищеперелічені динамічні характеристики, використовуючи інтерфейс LTI Viewer. Для цього у меню **Edit** вибрати Plot Configurations. У вікні, що відкрилося, на панелі «Select…» вибрати варіант з п’ятьма графіками, а на панелі «Response type» задати step, impulse, bode, nyquist та pole-zero.

Після закінчення роботи з LTI Viewer закрийте усі вікна за винятком командного вікна MATLAB.

13. Побудуйте сигнал, що імітує прямокутні імпульси одиничної амплітуди з періодом 10 секунд:

```
>> [u, t] = gensig('square', 10);
```

14. Виконайте моделювання і побудуйте на графіку сигнал виходу системи **f** при сформованому вхідному сигналі **u(t)** :

```
>>lsim (f, u, t)
```

15. У зошиті для розв’язання задач наведіть зміст командного вікна і всі отримані графіки.

3 ПОБУДОВА МАТЕМАТИЧНОГО ОПИСУ ОБ'ЄКТА УПРАВЛІННЯ ЕКСПЕРИМЕНТАЛЬНИМИ МЕТОДАМИ

3.1 Короткі теоретичні положення

3.1.1 Статичні і динамічні характеристики

Математичні моделі об'єктів управління, процедура їх побудови залежать від їх цільового призначення, властивостей об’єкта, а також функцій і завдань системи управління.

При автоматизації виробничих процесів визначають статичні і динамічні характеристики регульованих об'єктів.

Статичні характеристики системи або окремих її ланок є функціональною залежністю вихідних величин від вхідних при статичному (стаціонарному, такому, що встановився) режимі роботи, тобто при значеннях вхідних і вихідних величин, що не змінюються в часі. Статичні характеристики дозволяють визначити значення регульованих параметрів, положення регулюючих органів, витрати речовин або енергії і інші дані для будь-якого стану досліджуваного об’єкта, що встановився.

Динамічні характеристики є залежностями між змінами вхідних і вихідних величин в динамічному режимі (у часі). Динамічні характеристики дають інформацію про інерційні властивості регульованих об'єктів (систем, елементів систем) і таким чином є початковими даними при синтезі автоматичних систем регулювання. Вони дозволяють виконати цю роботу в повному об'ємі і завершити її розрахунком параметрів налаштування на цьому регульованому об'єкті з метою отримання бажаних форм графіків процесів регулювання.

Статичні і динамічні характеристики можуть бути отримані аналітичним, експериментальним або експериментально-аналітичним методом.

При побудові систем автоматичної стабілізації окремих технологічних параметрів досить мати в розпорядженні динамічні і статичні характеристики об’єкта управління у вузькому діапазоні зміни вхідних і вихідних координат. Для отримання цих характеристик застосовуються численні експериментальні методи. Ці методи базуються на припущенні про лінійність і зосередженість параметрів об'єкта, незмінність в часі його статичних і динамічних характеристик. Прийняття цих допущень дозволяє досить просто описати зміну спостережуваних вихідних координат об’єкта лінійними диференціальними рівняннями з постійними коефіцієнтами.

При побудові експериментальних математичних моделей використовує пасивний або активний експерименти.

При активних методах на вході досліджуваного об’єкта створюються штучні випробувальні дії (ступінчасті, імпульсні, гармонійні та ін.). Пасивними методами називають такі, при яких не допускаються штучні обурення на вході досліджуваного об'єкта, а як вхідні випробувальні дії використовуються природні дії, що виникають в процесі нормальної експлуатації.

3.1.2 Апроксимація статичних характеристик

При експериментальному дослідженні статичних характеристик деякого одноканального об’єкта управління отримано m значень вхідної x_i (i = 1, ..., m) і вихідної y_i координат. Треба знайти функціональну залежність $\hat{\mathbf{y}}$ = f($\mathbf{x}$, $\mathbf{a}$), що якнайкраще описує експериментальні дані, де **a** - вектор параметрів функції. Для цієї мети досить часто використовується метод найменших квадратів. Згідно з цим методом вводиться функція, що характеризує міру близькості експериментальних і розрахункових даних:

$$S = \sum_i (y_i - \hat{y}_i)^2 .$$

При використанні методу найменших квадратів задаються видом функції і вектором параметрів цієї функції. Часто як апроксимуючу функцію $\hat{y} = f(x,\mathbf{a})$ використовують поліном виду:

$$\hat{y} = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots$$

Завдання пошуку апроксимуючої функції (вектору параметрів) ставиться і вирішується як завдання безумовної багатовимірної оптимізації: вимагається знайти вектор параметрів **a**, що забезпечує мінімум функції S.

$$S = \sum_i (y_i - \hat{y}_i)^2 \to \min_{\mathbf{a}} . \tag{3.1}$$

При використанні пакету MATLAB поліноміальна апроксимація цих вимірів, які сформовані як деякий вектор **Y**,при деяких значеннях аргументу, які утворюють вектор **X** такої ж довжини, що і вектор **Y**, здійснюється процедурою **polyfit(X, Y,** n**)**. Тут n - порядок апроксимуючого полінома. Результатом дії цієї процедури є вектор завдовжки (n + 1) з коефіцієнтів апроксимуючого полінома.

3.1.3 Апроксимація динамічних характеристик

У інженерній практиці при налаштуванні систем управління математичний опис об'єкта досить часто представляється у вигляді передатних функцій.

Передатна функція об'єкта регулювання може бути визначена в результаті апроксимації експериментальної динамічної характеристики, отриманої в результаті проведення активного експерименту. Як випробувальна дія найчастіше використовується ступінчаста дія.

Алгоритм визначення апроксимуючої передатної функції об’єкта регулювання має вигляд:

а) при поданні на вхід об’єкта ступінчастої дії на виході знімається перехідна характеристика $h(t)$;

б) по виду кривої визначається структура передатної функції. Для великого числа реальних об'єктів регулювання модель об'єкта може бути представлена у вигляді:

$$W(s) = \frac{ke^{-s\tau}}{(T_1 s + 1)(T_2 s + 1)} . \tag{3.2}$$

3.1.4 Апроксимація перехідної характеристики об’єкта передатною функцією

Апроксимація полягає в підборі таких параметрів передатної функції, які забезпечують максимальну близькість експериментальної кривої розгону $h(t)$ і розрахованій кривій $h_{\text{р}}(t)$, що відповідає цій передатній функції.

Параметри налаштування k, T_1, T_2 τпередатної функції (3.2) можуть бути отримані при мінімізації функції

$$I = \int (h(t) - h_{\delta}(t))^2 dt \to \min_{k, T_1, T_2, \tau} . \qquad (3.3)$$

3.2 Приклади і задачі для самостійного розв'язання

Задача 3.1. Визначити передатну функцію динамічного об’єкта за кривою розгону, яка представлена даними табл. 3.1.

Таблиця 3.1 – Вихідні дані до задачі 3.1

T	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
h	0	0	0	0.07	0.23	0.43	0.64	0.84	1.03	1.19	1.33	1.45	1.55	1.64	1.71	1.77

Розв'язування

1. Створюємо Script-файл для вирішення задачі. Для цього викликаємо меню **File** командного вікна MATLAB і вибираємо в нім спочатку команду **New**, а потім команду **M-file**.

2. У вікні текстового редактора, що з'явилося, набираємо текст:

```
% t - Масив моментів часу
% h - масив значень експериментальної перехідної
характеристики
% hp - масив значень розрахованої перехідної
характеристики
% a - вектор параметрів передатної функції об’єкта
% a(1)=k, a(2)=t1, a(3)=t2, a(4)=tau
global t h hp
t=0:1:15;
h=[0 0 0 0.07 0.23 0.43 0.64 0.84 1.03 1.19 1.33 1.45 1.55
1.64 1.71 1.77];
[a, fval] = fminunc(@func,[1 1 2 1])
plot(t,h,'o',t,hp,'-')
```

Процедура **fminunc** здійснює пошук мінімуму функції декількох змінних. Ім'я функції, що мінімізується, задається в першому аргументі процедури **fminunc**, а в другому – передається початкова точка для пошуку.

3. Зберігаємо цей текст у файлі під ім'ям *din_app.m*.

4. Створюємо файл-функцію, яка вираховує функцію (3.3), що мінімізується. Текст файл-функції має вигляд:

```
function f=func(a)
global t h hp
y=series(tf([a(1)],[a(2) 1]), tf([1],[a(3) 1]));
y.inputd = a(4);
[hp, t]=step(y, t);
f=sum((h - hp').^2);
```

У цій функції рядки

```
y=series(tf([a(1)],[a(2) 1]), tf([1],[a(3) 1]));
y.inputd = a(4);
```

визначають передатну функцію об’єкта відповідно до (3.2). Тут a(1), a(2), a(3), a(4) відповідно k, T_1, T_2, τ.

Рядок [*hp, t*]=*step*(*y, t*)*;* будує відгук на ступінчасту дію.

Останній рядок визначає суму квадратів нев'язок між експериментальною і розрахованою перехідними характеристиками.

5. Зберігаємо введений текст у файлі під ім'ям *func.m*.

6. У командному режимі запускаємо на виконання Script-файл:

```
>>din_app
```

При завершенні виконання цього Script-файлу в командному вікні MATLAB буде відображається результат рішення задачі мінімізації функції (3.3) у вигляді:

```
a =
2.0382 2.4265 4.8308 2.0181
fval =
3.9104e-005
```

Тобто k=2.0382, T_1=2.4265, T_2=4.8308, τ=2.0181. ПФ має вигляд:

$$W(s)=\frac{2.0382}{(2.4265\,s+1)(4.8308\,s+1)}\exp(-2.0181\,s).$$

На графік виводяться експериментально зняті точки перехідної характеристики і розрахована апроксимуюча перехідна характеристика.

Задача 3.2. Виконати апроксимацію статичної характеристики одноканального об'єкта управління за експериментальними даними, наведеними у таблиці. 3.2.

Таблиця 3.2 – Вихідні дані до задачі 3.2

Варіант	Експериментальні дані										
0	x_i	0	1	2	3	4	5	6	7	8	9
	y_i	3,0	5,1	7,4	9,9	12,6	15,5	18,6	21,9	25,4	29,1
1	x_i	0	1	2	3	4	5	6	7	8	9
	y_i	6,0	7,9	9,6	11,1	12,4	13,5	14,4	15,1	15,6	15,9
2	x_i	0	1	2	3	4	5	6	7	8	9
	y_i	4,0	6,8	9,2	11,2	12,8	14,0	14,8	15,2	15,2	14,8
3	x_i	0	1	2	3	4	5	6	7	8	9
	y_i	5,0	10,1	14,4	17,9	20,6	22,5	23,6	23,9	23,4	22,1
4	x_i	0	1	2	3	4	5	6	7	8	9
	y_i	35,0	34,3	33,0	31,1	28,6	25,5	21,8	17,5	12,6	7,1
5	x_i	0	2	4	6	8	10	12	14	16	18
	y_i	3,0	5,1	7,4	9,9	12,6	15,5	18,6	21,9	25,4	29,1
6	x_i	0	2	4	6	8	10	12	14	16	18
	y_i	6,0	7,9	9,6	11,1	12,4	13,5	14,4	15,1	15,6	15,9
7	x_i	0	2	4	6	8	10	12	14	16	18
	y_i	4,0	6,8	9,2	11,2	12,8	14,0	14,8	15,2	15,2	14,8
8	x_i	0	2	4	6	8	10	12	14	16	18
	y_i	5,0	10,1	14,4	17,9	20,6	22,5	23,6	23,9	23,4	22,1
9	x_i	0	2	4	6	8	10	12	14	16	18
	y_i	35,0	34,3	33,0	31,1	28,6	25,5	21,8	17,5	12,6	7,1

Порядок проведення аналізу наступний.

1. Запустити середовище MATLAB і в командному режимі очистити робочий простір і вікно:

```
>>clear all
>>clc
```

2. Сформуйте масиви експериментальних точок – вектори **X** і **Y** (відповідно до варіанту, див. табл. 3.2):

```
>> X = [x1 x2 x3 x4 x5 x6 x7 x8 x9 x10];
>> Y = [y1 y2 y3 y4 y5 y6 y7 y8 y9 y10];
```

3. Виконайте апроксимацію експериментальних точок поліномом першого і другого степеня і визначте вектори коефіцієнтів поліномів P1 і P2 відповідно:

```
>> P1 = polyfit(X,Y,1)
>> P2 = polyfit(X,Y,2)
```

4. Використовуючи функцію polyval, побудуйте апроксимуючі поліноми:

```
>> F1 = polyval(P1,X)
>> F2 = polyval(P2,X)
```

5. В одних координатних осях побудуйте графіки експериментальних точок, а також апроксимуючих поліномів першого і другого ступеня:

```
>>plot(X,Y,'o',X,F1,'-.',X,F2,'-')
```

Задача 3.3. Виконати апроксимацію динамічної характеристики об'єкта управління передатною функцією за кривою розгону, яка представлена даними табл. 3.3.

Таблиця 3.3– Варіанти завдань до задачі 3.3

0		1		2		3		4		5		6		7		8		9	
T	*h*	*T*	*h*	*T*	*h*	*T*	*h*	*T*	*h*	*T*	*h*	*T*	*h*	*T*	*h*	*T*	*h*	*T*	*h*
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1	0	1	0	1	0	2	0	2	0	2	0	2	0	2	0
2	0	2	0	2	0	2	0,14	2	0,098	4	0	4	0	4	0	4	0,14	4	0,1
3	0,07	3	0,09	3	0,13	3	0,46	3	0,31	6	0,07	6	0,09	6	0,13	6	0,46	6	0,32
4	0,23	4	0,29	4	0,43	4	0,84	4	0,56	8	0,23	8	0,29	8	0,43	8	0,84	8	0,57
5	0,43	5	0,53	5	0,79	5	1,19	5	0,8	10	0,43	10	0,53	10	0,79	10	1,19	10	0,88
6	0,64	6	0,77	6	1,15	6	1,52	6	1,02	12	0,64	12	0,77	12	1,15	12	1,52	12	1,1
7	0,84	7	0,99	7	1,49	7	1,81	7	1,21	14	0,84	14	0,99	14	1,49	14	1,81	14	1,23
8	1,03	8	1,19	8	1,78	8	2,05	8	1,36	16	1,03	16	1,19	16	1,78	16	2,05	16	1,4
9	1,19	9	1,35	9	2,03	9	2,24	9	1,49	18	1,19	18	1,35	18	2,03	18	2,24	18	1,5
10	1,33	10	1,49	10	2,23	10	2,4	10	1,6	20	1,33	20	1,49	20	2,23	20	2,4	20	1,62
11	1,45	11	1,61	11	2,4	11	2,52	11	1,69	22	1,45	22	1,61	22	2,4	22	2,52	22	1,71
12	1,55	12	1,69	12	2,54	12	2,63	12	1,75	24	1,55	24	1,69	24	2,54	24	2,63	24	1,78
13	1,64	13	1,76	13	2,64	13	2,71	13	1,81	26	1,64	26	1,76	26	2,64	26	2,71	26	1,83
14	1,71	14	1,82	14	2,72	14	2,79	14	1,85	28	1,71	28	1,82	28	3,72	28	2,79	28	1,88
15	1,77	15	1,86	15	2,79	15	2,82	15	1,88	30	1,77	30	1,86	30	2,79	30	2,82	30	1,92

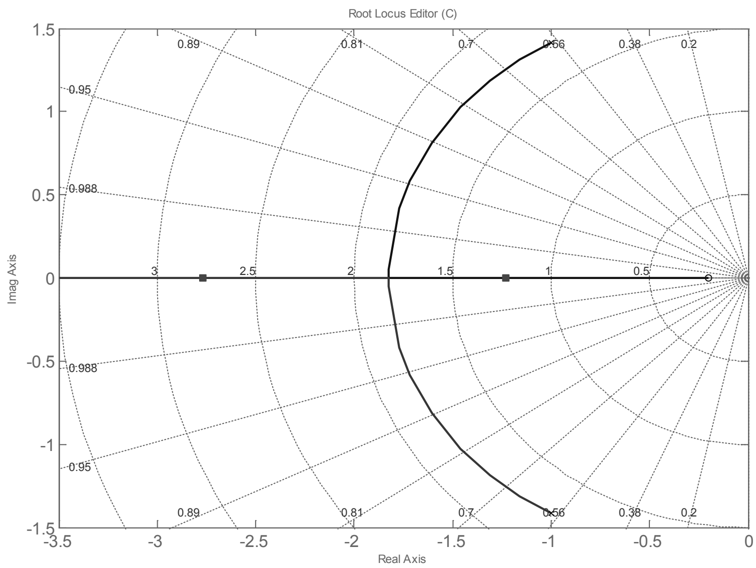
4 АНАЛІЗ СТІЙКОСТІ ЛІНІЙНОЇ СИСТЕМИ

4.1 Короткі теоретичні положення

4.1.1 Кореневий годограф

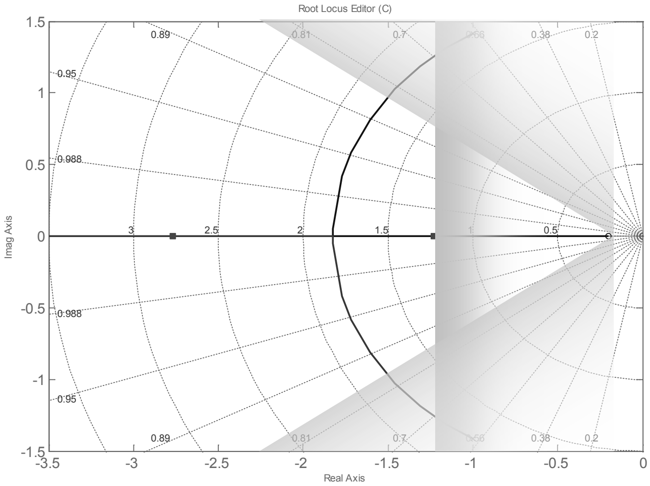
Багато важливих властивостей системи (наприклад, швидкодія, перерегулювання) визначаються розташуванням коренів характеристичного рівняння на комплексній площині. Простий спосіб корекції системи – застосувати П-регулятор (підсилювач з коефіцієнтом k), який змінює коефіцієнт посилення розімкненої системи і розташування цих коренів. При зміні k від 0 до ∞ корені описують криві, які називаються *кореневим годографом*.

За допомогою модуля **SISOTool** (скорочення *SISO=Single Input Single Output* означає систему з одним входом і одним виходом) можна вибирати потрібне розташування коренів (і відповідний коефіцієнт посилення), «перетягуючи» їх мишкою. Помітимо, що при переміщенні одного кореня зміщуються і усі інші, оскільки система має один ступінь свободи – коефіцієнт посилення контуру, що змінюється.



Корені при вибраному коефіцієнті посилення k показуються фіолетовими квадратиками. Кінці годографа для кожного кореня поміщені хрестиком (k=0) і кружечком (k=∞). Сітка (для її виводу потрібно натиснути ПКМ на графіці і вибрати пункт **Grid)** показує лінії рівних показників коливальності (коефіцієнта демпфування ζ, *damping factor*) – це прямі, що виходять з початку координат, а також лінії рівних власних частот (*natural frequency*) – це кола з центром на початку координат.

У контекстному меню (ПКМ) можна встановити обмеження на розташування полюсів так, щоб перерегулювання і час перехідного процесу не перевищували заданих. Для цього потрібно вибрати пункт **Design Constraints – New** і вибрати у випадному списку **Percent Overshoot** (перерегулювання у відсотках) або **Settling Time** (час перехідного процесу з 2%-ою точністю). Обмеження показуються у вигляді меж заборонених зон.



Час перехідного процесу $T_{пп}$ оцінюється за *мірою стійкості* η замкнутої системи. Так називається відстань від найправішого кореня характеристичного рівняння до уявної осі. Зазвичай приймається (як для аперіодичної ланки)

$$T_{пп} \approx -\frac{\ln \Delta}{\eta} \approx \frac{3,912}{\eta},$$

де Δ - величина допустимої помилки (у MATLAB вона приймається рівною 2% або 0,02). Таким чином, при обмеженні тільки на $T_{пп}$ область допустимого розташування коренів p_i є напівплощина $\operatorname{Re} p_i < -\frac{3,912}{T}$.

Вимоги до коефіцієнта демпфування ζдодають обмеження у вигляді сектора

$$\beta = \max_i \left| \frac{\operatorname{Im} p_i}{\operatorname{Re} p_i} \right| < \frac{\sqrt{1-\zeta^2}}{\zeta^2}.$$

Число β називають *коливальністю* або *мірою коливальності* замкнутої системи. Кожному заданому ζ відповідає деяке значення β.

Перерегулювання (у відсотках) оцінюється по формулі

$$\sigma \approx e^{-\pi \frac{\zeta}{\sqrt{1-\zeta^2}}} \times 100\%.$$

Кожному перерегулюванню відповідає своє значення ζ і свій сектор, що обмежує розташування коренів.

Таким чином, при використанні двох обмежень (перше – на $T_{пп}$, друге – на σ або ζ) область допустимого розташування коренів є усіченим сектором в лівій частині малюнка. Якщо перетяганням коренів (тобто, зміною посилення контуру) не вдається розмістити полюси в цій області, потрібно ускладнювати регулятор, додаючи його нулі і полюси (ПКМ **– Add Pole/Zero** або ПКМ **–Edit Compensator).**

4.1.2 Синтез за допомогою ЛАФЧХ

У вітчизняній літературі класичним став метод синтезу пристроїв, що коригують, за допомогою логарифмічних амплітудно-фазових частотних характеристик (ЛАФЧХ) розімкненої системи (діаграм Боде за зарубіжною термінологією).

Нехай розімкнена система має передатну функцію $W(s)$. ЛАФЧХ включає дві криві – амплітудну частотну характеристику (ЛАЧХ)

$$L_m(\omega) = 20\lg\left|W(j\omega)\right|$$

і фазову (ЛФЧХ)

$$\varphi(\omega) = \arg W(j\omega).$$

Кутова частота (у рад/с) на осі ординат відкладається в логарифмічному масштабі. При цьому так звані *асимптотичні* ЛАЧХ є відрізками прямих, це значно полегшує ручну побудову.

Розклавши чисельник і знаменник передатної функції $W(s)$ на співмножники першого і другого порядків, можна представити ЛАЧХ системи як суму ЛАЧХ елементарних ланок (аперіодичних, коливальних, інтегруючих, диференційних і так далі) Для

$$W(j\omega) = \frac{N_1(j\omega)...N_n(j\omega)}{D_1(j\omega)...D_q(j\omega)}$$

отримуємо, використовуючи властивості логарифма

$$\begin{aligned} 20\lg\left|W(j\omega)\right| &= 20\lg\left|N_1(j\omega)\right| + ... + 20\lg\left|N_n(j\omega)\right| \\ &- 20\lg\left|D_1(j\omega)\right| - ... - 20\lg\left|D_n(j\omega)\right| \end{aligned}.$$

У середовищі MATLAB існують засоби, що дозволяють автоматизувати побудову точних (не асимптотичних) ЛАФЧХ.

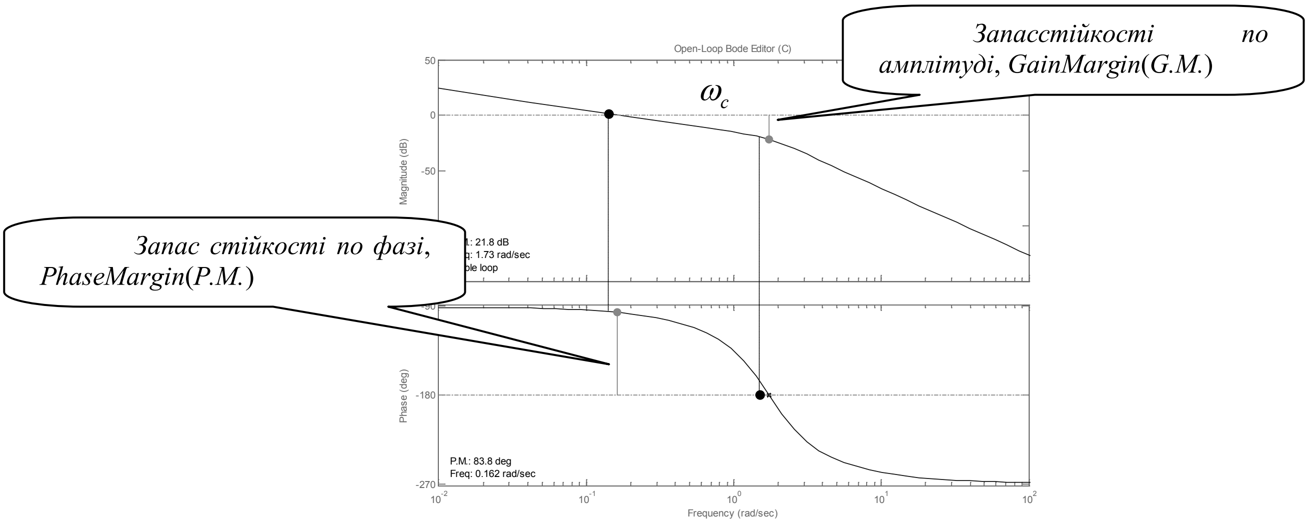
Низькочастотна частина ЛАЧХ визначає точність системи, середньочастотна (поблизу частоти зрізу ω_c) – стійкість і якість перехідного процесу, високочастотна – чутливість до завад. Якщо система містить інтегратор, низькочастотна частина має ненульовий нахил (20 дБ на декаду для одного інтегратора), постійний сигнал відстежується без помилки, що встановилася. Для системи з двома інтеграторами ЛАФЧХ має в області низьких частот нахил 40 дБ на декаду, без помилки, що встановилася, відстежується не лише постійний, але і лінійно зростаючий сигнал. Складніші вимоги до точності призводять до того, що ЛАЧХ не повинна заходити в деякі заборонені області.

Запас стійкості за амплітудою g_m (у дБ) – це відстань від ЛАЧХ до горизонтальної прямої L_m=0 дБ на частоті, на якій фазова характеристика перетинає пряму φ=–180°. На цій частоті система повинна мати коефіцієнт посилення менше 1 (чи L_m=(ω)<0).

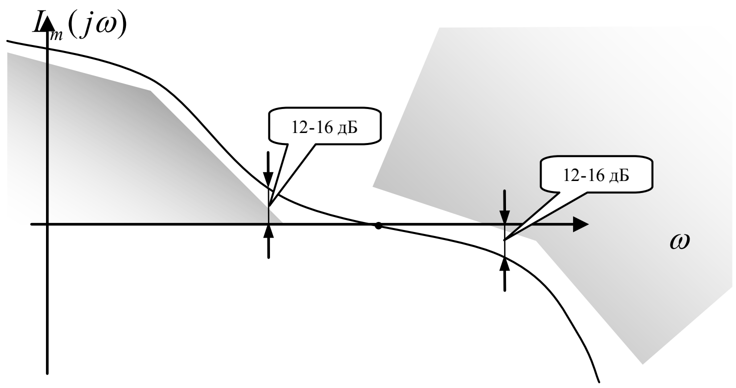
Запас стійкості за фазою φ_m (у градусах) – це відстань від частотної характеристики до горизонтальної прямої φ=–180° на частоті зрізу ω_c. На цій частоті фазова характеристика повинна мати значення *більше*, ніж –180°.

Допустимим вважається запас за амплітудою не менше 6 дБ і запас за фазою не менше 30 градусів, інакше не гарантується стійкість системи.

«Підйом» ЛАЧХ означає збільшення коефіцієнта посилення контуру, фазова характеристика не змінюється. Точність системи (при відробітку низькочастотних сигналів) підвищується, проте збільшується і вплив високочастотних завад. Оскільки частота зрізу збільшується, підвищується швидкодія системи. При цьому перехідні процеси набувають вираженого коливального характеру, запаси стійкості зменшуються, при подальшому збільшенні коефіцієнта посилення втрачається стійкість.



Зазвичай вимагається, щоб система мала високу точність (великий коефіцієнт посилення по контуру) для низьких частот і пригнічувала високочастотні завади (мала низьке посилення в області високих частот). Частота зрізу вибирається виходячи з вимог до швидкодії. Таким чином, типова ЛАЧХ має вигляд, показаний на малюнку. Сірим кольором показані заборонені області, які визначаються вимогами до точності і пригнічення завад.



Для забезпечення хорошої якості перехідних процесів рекомендується, щоб ЛАЧХ перетинала вісь L=0 з нахилом 20 дБ/дек. Це пояснюється тим, що нахил 20 дБ/дек, що відповідає аперіодичній ланці, призводить до найменшої коливальності перехідного процесу. Точки переходу (зламу асимптотичної ЛАЧХ) від низькочастотної частини до середньочастотної і далі до високочастотної повинні знаходитися від осі L=0 на 12…16 дБ.

У загальному випадку будується бажана ЛАЧХ $L_б$(jω), що задовольняє вимогам до системи, потім ЛАЧХ послідовного пристрою, що коригує, визначається як різниця між і $L_б$(jω) і ЛАЧХ існуючої розімкненої системи.

4.1.3 Критерії стійкості САУ

Критерій Гурвиця є алгебраїчним критерієм і застосовується до коефіцієнтів характеристичного рівняння замкнутої системи.

Нехай є характеристичне рівняння замкнутої системи :

$$a_n s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0 = 0.$$

З коефіцієнтів характеристичного рівняння складають матрицю за правилом:

1. По діагоналі записуються коефіцієнти від a_{n-1} до a_0.

2. Кожен рядок доповнюється коефіцієнтами із зростаючими індексами зліва направо так, щоб чергувалися рядки з непарними і парними індексами.

3. У разі відсутності індексу, а також, якщо він менше 0 або більше n, на його місце пишеться 0.

Таким чином, матриця Гурвиця набуває наступного вигляду:

$$\begin{bmatrix} a_{n-1} & a_{n-3} & a_{n-5} & \ldots & 0 & 0 \\ a_n & a_{n-2} & a_{n-4} & \ldots & 0 & 0 \\ 0 & a_{n-1} & a_{n-3} & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & a_1 & 0 \\ 0 & 0 & 0 & \ldots & a_2 & a_0 \end{bmatrix}.$$

Критерій стійкості формулюється так:

Щоб система була стійкою, необхідно і достатньо, щоб при $a_n > 0$ були позитивними усе n діагональних визначників, що отримуються з матриці Гурвиця.

Перші три визначники матриці Гурвиця мають наступний вигляд:

$$\Delta_1 = a_{n-1};\ \Delta_2 = \begin{vmatrix} a_{n-1} & a_{n-3} \\ a_n & a_{n-2} \end{vmatrix};\ \Delta_3 = \begin{vmatrix} a_{n-1} & a_{n-3} & a_{n-5} \\ a_n & a_{n-2} & a_{n-4} \\ 0 & a_{n-1} & a_{n-3} \end{vmatrix}.$$

Таким чином, критерій Гурвиця дозволяє судити про абсолютну стійкість, але він не дає можливості оцінювати відносну стійкість по коренях характеристичного рівняння.

Частотний критерій стійкості Найквіста аналізує АФЧХ розімкненої системи.

Нехай є ПФ розімкненої системи $W(j\omega)$.

Для знаходження дійсної і уявної частини частотної ПФ треба звільнитися від уявності в знаменнику шляхом множення чисельника і знаменника на величину, комплексноспряжену знаменнику, а потім виконати розділення на дійсну і уявну частині. Передатна функція набуває вигляду:

$$W(j\omega) = P(\omega) + Q(\omega).$$

Основні властивості АФЧХ розімкненої системи наступні.

1. Якщо розімкнена система не має інтегруючих ланок, то при $\omega = 0$ її АФЧХ починається на дійсній осі в точці $P(\omega)=K$ (де K – коефіцієнт посилення розімкненої системи). Закінчується АФЧХ на початку координат при $\omega \to \infty$ (рис. 4.1, а).

2. Якщо розімкнена система має одну інтегруючу ланку, то її АФЧХ починається при ω = 0 в нескінченності на негативній уявній півосі, а закінчується на початку координат при ω → ∞ (рис. 4.1, б).

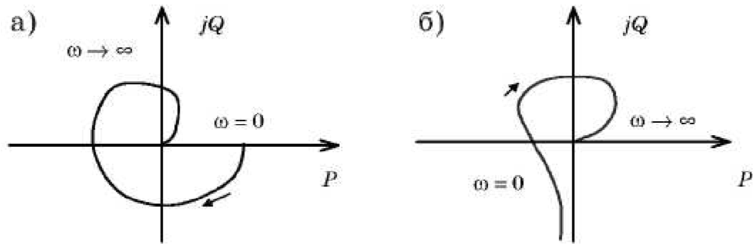


Рис. 4.1 – АФЧХ розімкненої системи

Критерій стійкості Найквіста формулюється так:

а) якщо розімкнена система стійка або знаходиться на межі стійкості, то для того, щоб замкнута система була стійка, необхідно і достатньо, щоб АФЧХ розімкненої системи при зміні частоти ω від 0 до ∞ не охоплювала точку з координатами (–1, j0);

б) якщо розімкнена система нестійка, а її передатна функція має m полюсів праворуч від уявної осі на комплексній площині, то для стійкості замкнутої системи необхідно і достатньо, щоб АФЧХ розімкненої системи при зміні частоти ω від –∞ до +∞ охоплювала m разів точку з координатами (–1, j0).

При використанні цього критерію треба враховувати дві особливості:

а) якщо розімкнена система знаходиться на межі стійкості, то її АФЧХ йде в нескінченність. Для перевірки критерію Найквіста треба подумки з'єднати кінець АФЧХ дугою нескінченно великого радіусу з позитивною дійсною піввіссю;

б) на практиці АФЧХ може будуватися тільки для позитивних частот ($0<\omega<+\infty$). При застосуванні критерію Найквіста вважається, що гілка АФЧХ для негативних частот симетрична відносно дійсної осі.

Фізичний сенс критерію стійкості Найквіста полягає в тому, що система буде нестійка, якщо фаза вихідного сигналу протилежна до фази вхідного сигналу, а коефіцієнт посилення більше 1.

4.1.4 Точність в режимі, що встановився

Нехай передатну функцію розімкненої системи можна представити у виді

$$W(s) = KW_1(s),$$

де передатна функція $W_1(s)$ має властивість $\lim_{s\to 0} W_1(s) = 1$. Тоді передатна функція замкнутої системи за помилкою

$$\Phi_\varepsilon(s) = \frac{1}{1 + KW_1(s)}.$$

Значення помилки, що встановилося, при постійному вхідному сигналі $x(t) = x_0$, що має зображення за Лапласом $X(s) = \frac{x_0}{s}$, може бути обчислене за теоремою про кінцеве значення:

$$\varepsilon_\infty = \lim_{t\to\infty} \varepsilon(t) = \lim_{s\to 0} s\,\Phi_\varepsilon(s)\,X(s) = \frac{x_0}{1+K}.$$

Таким чином, при збільшенні коефіцієнта посилення *K*помилка зменшується (проте запас стійкості також зменшується і система може стати нестійкою). Величина *K* називається *добротністю* системи. При будь-якому кінцевому *K* в такій системі помилка буде кінцевою. Для лінійно зростаючого сигналу помилка лінійно зростатиме.

Тепер нехай

$$W(s)=\frac{K}{s^{\nu}}W_1(s),$$

де ν - ціле число і $\lim_{s\to 0}W_1(s)=1$. Тоді для усіх вхідних сигналів виду

$$x(t)=x_0+x_1t+x_2t^2+\ldots+x_{\nu-1}t^{\nu-1}$$

система забезпечуватиме нульову помилку, що встановилася, при будь-яких значеннях коефіцієнтів $x_0,\ldots x_{\nu-1}$. Таким чином, при ν>0 система відстежує постійний сигнал без помилки, що встановилася. Такі системи називають *астатичними.*

Число ν називається *порядком астатизму.* Для сигналу

$$x(t)=x_{\nu}t^{\nu},\quad X(s)=\frac{\nu!x_{\nu}}{s^{\nu+1}}$$

помилка, що встановилася, є рівною

$$\varepsilon_{\infty}=\frac{\nu!x_{\nu}}{K}.$$

Вище розглянутим є випадок астатизму по відношенню до задаючої дії. Аналогічно може йтися про астатизм по відношення до збурюючої дії.

4.2 Приклади і задачі для самостійного розв'язання

Задача 4.1.За допомогою критеріюГурвиця визначити стійкість системи, що має передатну функцію

$$\mathrm{W}(s)=\frac{Y(s)}{U(s)}=\frac{5\cdot s+4}{s^5+16\cdot s^4+95\cdot s^3+260\cdot s^2+324\cdot s+144}$$

Розв'язування

Для перевірки стійкості САУ по Гурвицю треба побудувати матрицю Гурвиця:

```
C=
    16   260   144      0      0
     1    95     324     0      0
     0    16     260   144      0
     0     1      95   324      0
     0     0      16   260    144
```

і знайти її детермінант (функція **det**).

```
>>det(C)
ans =
  9.1446e+009
```

Потім, послідовно зменшуючи розмір матриці, треба знайти значення усіх діагональних детермінантів:

```
>> C1=C(1:3, 1:3)
C1 =
    16   260   144
     1    95   324
     0    16   260
>>det(C1)
ans =
      246960
>> C2=C(1:2, 1:2)
C2 =
    16   260
     1    95
>>det(C2)
ans =
        1260
```

Система є стійкою, оскільки всі визначники позитивні.

Задача 4.2. За допомогою критерію Гурвиця визначити стійкість системи, що має передатну функцію, наведену у табл. 4.1.

Таблиця 4.1 – Вихідні дані до задачі 4.2

Варі-ант	Передатна функція розімкненої системи
0	$W = \dfrac{2}{s^4 + 5s^3 + 5s^2 + 3s + 1}$
1	$W = \dfrac{1}{0,05s^4 + 0,1s^3 + s^2 + s + 1}$
2	$W = \dfrac{1}{0.1s^3 + 0.1s^2 + s + 1}$
3	$W = \dfrac{100}{5s^4 + 0.1s^3 + 2s^2 + 2s + 1}$
4	$W = \dfrac{1}{8s^3 + 4s^2 + 2s + 1}$
5	$W = \dfrac{10}{s^5 + 3s^4 + 2s^3 + 2s^2 + s + 1}$
6	$W = \dfrac{3}{0,1s^3 + 0,01s^2 + 0,1s + 1}$
7	$W = \dfrac{10}{2s^3 + 2s^2 + s + 1}$
8	$W = \dfrac{1}{s^3 + 0,1s^2 + 0,1s + 1}$
9	$W = \dfrac{10}{2s^5 + 3s^4 + 3s^3 + 0.5s^2 + 0.5s + 1}$

Задача 4.3. Дана передатна функція розімкнутої системи

$$W = \frac{2s+1}{2s^4 + 3s^3 + 2s^2 + 3s + 1}.$$

Визначити, чи буде стійкою замкнута система.

Розв'язування

Для перевірки стійкості САУ за Найквістом спочатку треба з'ясувати, чи є стійкою розімкнена система.

Сформуємо передатну функцію:

```
>>w=tf([2 1],[2 3 2 3 1])
```

Розглянемо реакцію на стрибок:

```
>>step(w)
```

Будується графік перехідного процесу, показаний на рис. 4.2.

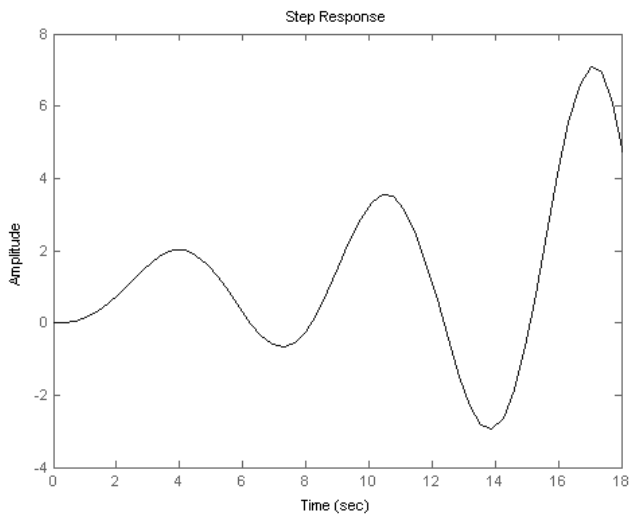


Рис. 4.2 – Перехідна реакція

Як видно з графіка, розімкнена система нестійка, і, згідно з критерієм Найквіста, потрібно, щоб АФЧХ розімкнутої системи охоплювала точку (–1, j0) стільки разів, скільки полюсів є праворуч від уявної осі.Для побудови АФЧХ досить викликати команду **nyquist:**

```
>>nyquist(w)
```

Діаграма Найквіста показана на рис. 4.3. Як бачимо, АФЧХ жодного разу не охоплює точку(–1, j0),тому замкнута система буде нестійкою.

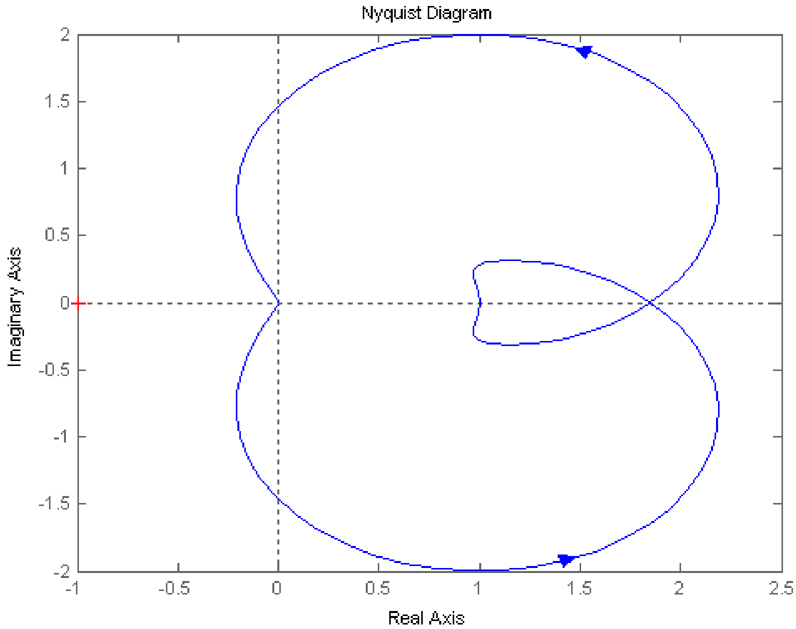


Рис. 4.3 – Діаграма Найквіста для задачі 4.3

Задача 4.4. САУ має перехідну характеристику

$$W = \frac{10}{2s^3 + 2s^2 + 3s + 1}.$$

Визначити запаси стійкостісистеми по амплітуді і по фазі.

Розв'язування

Використовуючи ЛАЧХ і ЛФЧХ, можна оцінити запаси стійкостісистеми по амплітуді і по фазі за допомогою команди **margin**.

```
>>w=tf([10],[2 2 3 1]);
>>margin(w)
```

Відповідний графік показаний на рис. 4.4. Запас стійкості системи по амплітуді –14 дБ, по фазі –52°.

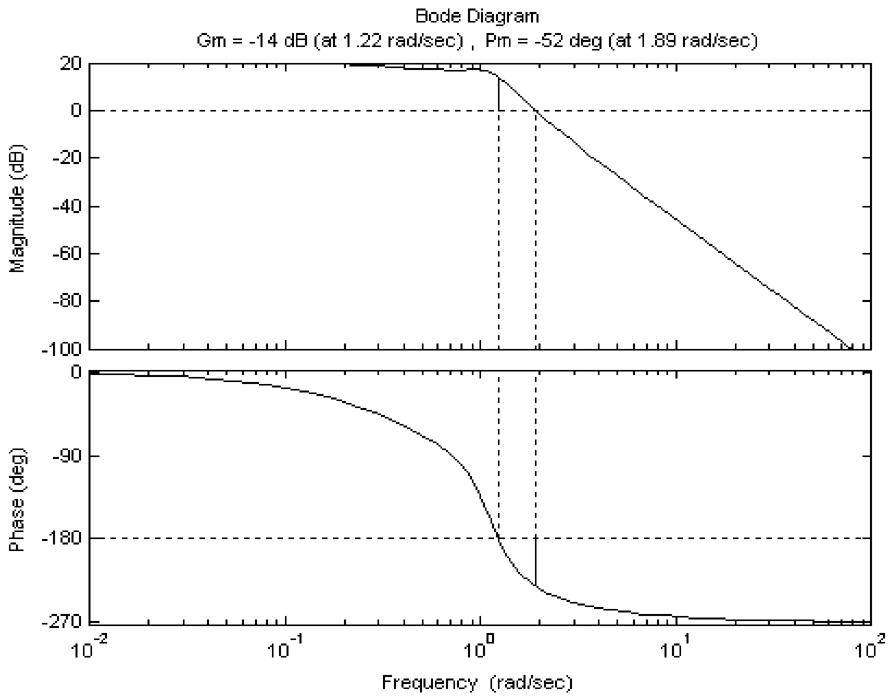
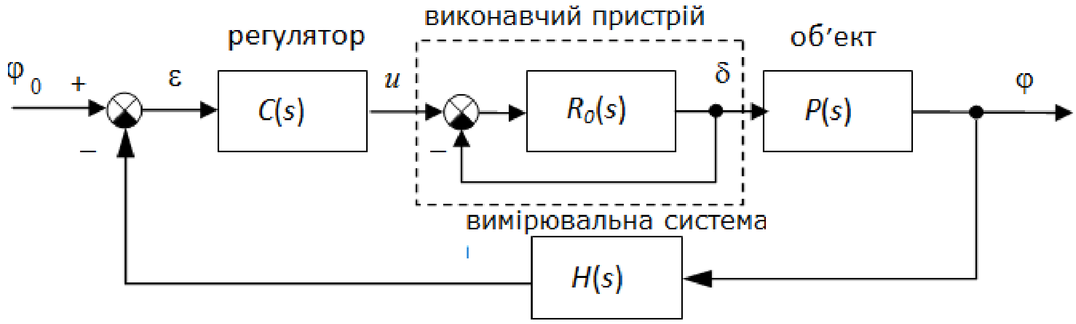


Рис. 4.4 – Визначення запасів стійкості по амплітуді і по фазі

Задача 4.5. Виконати дослідження стійкості замкнутої САУ по заданій передатній функції розімкненої системи. Варіанти завдань приведені в табл. 4.1.

Задача 4.6. САУ має наступну структурну схему



Передатна функція об'єкта

$$P(s)=\frac{K}{s\,(T_s s+1)},\text{ де } K=0.0694\text{ рад/с}, T_s=18.2\text{ с.}$$

Виконавчий механізм являє собою інтегруючу ланку

$$R_0(s)=\frac{1}{T_R\,s},\qquad T_R=2\text{ с},$$

охоплену одиничним негативним зворотним зв'язком. Модель вимірювального пристрою є аперіодичною ланкою з передатною функцією

$$H(s)=\frac{1}{T_{oc}s+1},\qquad T_{oc}=6\text{ с.}$$

Дослідити описану САУ.

Розв'язування

1. Передатна функція об'єкта

```
>> K = 0.0694;Ts=18.2;
>> P = tf(K, [Ts 1 0])
Transfer function:
   0.0694
------------
18.2 s^2 + s
```

2. Передатна функція інтегруючої ланки

```
>> TR=2; R0 = tf(1, [TR 0])
Transfer function:
 1
---
2 s
```

3. Передатна функція виконавчого пристрою

```
>> R = feedback(R0, 1)
Transferfunction:
   1
-------
2 s + 1
```

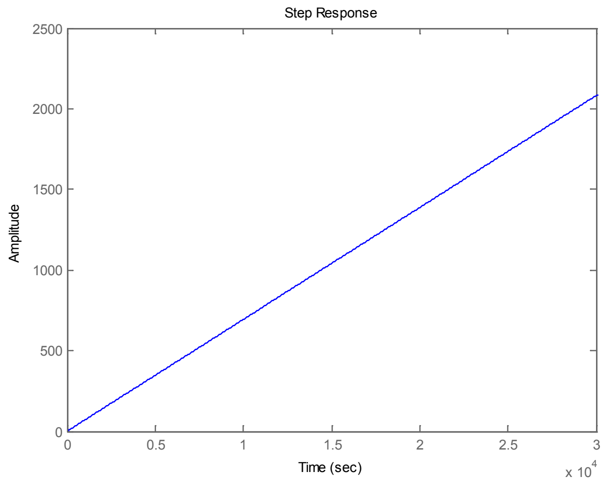
4. Передатна функція послідовного з'єднання об'єкта з виконавчим пристроєм

```
>> G = P*R

Transfer function:
        0.0694
-----------------------
36.4 s^3 + 20.2 s^2 + s
```

4. Перехідна характеристика для отриманої моделі

```
>>step(G)
```



5. Передатна функція вимірювального пристрою (датчика)

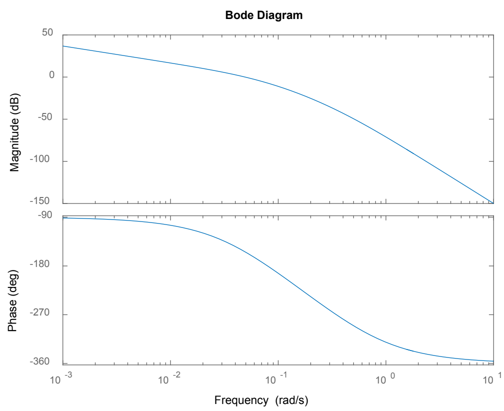
```
>> Toc=6; H = tf(1, [Toc 1])
 Transfer function:
1
-------
6 s + 1
```

6. Передатна функція розімкнутого контуру

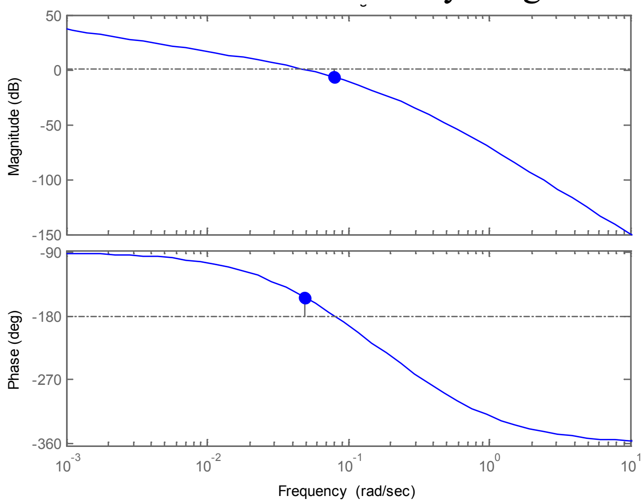
```
>> L = G*H
Transfer function:
               0.0694
------------------------------------
218.4 s^4 + 157.6 s^3 + 26.2 s^2 + s
```

7. Побудуємо диаграмуБоде в новому вікні

```
>>figure(2)
>>bodeplot(L)
```



8. Клікнемо правою клавішею миші по побудованому графіку і виберемо з контексного меню Characteristics - MinimumStabilityMargin.



9. Підведемо мишею курсор до кружечків і отримаємо запаси стійкості по амплітуді (*Gainmargin*) 7.12 дБ і по фазі (*Phasemargin*) 26°. Замкнута система є стійкою, про що говорить відповідь «Yes» на питання «Closed Loop Stable?»

10. Клікнемо правою клавішею миші по побудованому графіку і виберемо з контексного меню Characteristics – Peak Response (максимальний коефіцієнт посилення розімкнутої системи). Підведемо мишею курсор до кружечку, що з'явився на осі ординат і прочитаємо значення 377 дБ.

Задача 4.7. Дослідити САУ із задачі 4.6, згідно варіантів, наведених у табл. 4.2. Збережіть змінні робочого середовища для подальшого використання.Для цього використайте команду *save<ім'я файла>*.

Таблиця 4.2 – Вихідні дані для задачі 4.7

Варіант	T_s, с	K, рад/с	T_R, с	T_{oc}, с
1	16.0	0.06	1	1
2	16.2	0.07	2	2
3	16.4	0.08	1	3
4	16.6	0.07	2	4
5	16.8	0.06	1	5
6	17.0	0.07	2	6
7	17.2	0.08	1	1
8	17.4	0.07	2	2
9	17.6	0.06	1	3
10	17.8	0.07	2	4
11	18.0	0.08	1	5
12	18.2	0.09	2	6
13	18.4	0.10	1	1
14	18.6	0.09	2	2
15	18.8	0.08	1	3
16	16.0	0.07	2	4
17	17.2	0.08	2	5
18	18.4	0.09	2	6
19	19.6	0.10	1	1
20	18.2	0.07	2	6

21	18.5	0.12	1	4
22	19.1	0.08	2	3
23	19.4	0.09	2	6
24	17.6	0.10	1	2
25	18.0	0.0694	3	3

Задача 4.8. Для САУ із задачі 4.7 виконати дії, перераховані у табл. 4.3. Основна частина команд вводиться в командному вікні середовища MATLAB. Команди, які потрібно застосовувати в інших вікнах, позначені у табл. 4.3 іконками відповідних програм.

Таблиця 4.3 – Порядок виконання задачі 4.8

Етап виконання задачі	Команди MATLAB
Відновіть змінні робочого середовища, створені при рішенні задачи4.6	`load<ім’я файла>`
Закрийте вікно з ЛАФЧХ і запустіть модуль SISOTool	`sisotool`
Імпортуйте передатну функцію G як модель об'єкта (G) і H як модель датчика (H). Блоки *F* (передфільтр) і *C* (регулятор) залиште без змін (рівними 1)	SISO Design Tool `File-Import,` Кнопка `Browse,` Виділити потрібну модель – `Import`
Відключіть зображення кореневого годографа так, щоб у вікні залишилася тільки ЛАФЧХ	SISO Design Tool `View-Design Plot Configurat` Для`Plot 1` замінити`RootLocus` на `None`

Для того, щоб відразу бачити зміни перехідних процесів, запустите LTIViewerз верхнього меню вікна SISOTool. Розташуйте два вікна поруч, щоб вони не перекривали один одного	SISO Design Tool
	`Analysis -`
	`ResponsetoStepCommand`

Залиште тільки графік перехідного процесу на виході, відключивши виведення сигналу управління	LTI Viewer
	`ПКМ - Systems -`
	`Closedloop r to u`

Визначте пере регулюванняσ і час перехідного процесу T_n. Скопіюйте графік в звіт.	LTI Viewer
	`ПКМ - Characteristics -`
	`PeakResponse,`
	`SettlingTime`

Перейдіть у вікно SISOTool. Визначте коефіцієнт посилення, при якому перерегулювання приблизно рівне 10%. Як змінився час перехідного процесу? Які запаси стійкості в цьому випадку? Скопіюйте графік в звіт.	SISO Design Tool
	`перетягання мишею ЛАЧХ`
	`редагування в`
	`полі`*`CurrentCompensator`*

Перейдіть у вікно середовища MATLABі введіть передатну функцію пропорційно-диференціального (ПД) регулятора $C_{pd}(s)=1+\frac{T_s s}{T_v s+1}$, де T_v=1 с, а T_s - постійна часу об’єкта	`Cpd = 1 + tf([Ts 0], [Tv 1])`

Продовження табл. 4.3

Етап виконання задачі	Команди MATLAB
Перейдіть у вікно SISOTool. Імпортуйте регулятор C_{pd} як базову модель для блоку *C*.	SISO Design Tool `File - Import, Cpd -> C`

Визначте додатковий коефіцієнт посилення, при якому перерегулювання приблизно рівне 10%. Знайдіть час перехідного процесу і запаси стійкості. Порівняйте пропорційний і ПД-регулятори. Скопіюйте в звіт графік перехідного процесу.	SISO Design Tool
	`перетягання мишею ЛАЧХ`
	`редагування в`
	`полі`*`CurrentCompensator`*

Визначте додатковий коефіцієнт посилення, при якому час перехідного процесу мінімальний. Скопіюйте в звіт графік перехідного процесу.	SISO Design Tool
	`перетягання мишею ЛАЧХ`
	`редагування в`
	`полі`*`CurrentCompensator`*

Експортуйте отриманий регулятор в робочу область MATLAB.	SISO Design Tool
	`File - Export`
	`у стовпці`*`Exportas`*`змінити ім'я Cpd на C`
	`кнопка`*`Exporttoworkspace`*

Побудуйте передавальну функцію отриманої замкнутої системи	`W = C*G / (1 + C*G*H)`
Побудуйте мінімальну реалізацію передатної функції *W*. Функція minrealвикористовуєтьсядля скороченняоднакових полюсів и нулівПФ	`W = minreal(W)`
Визначте полюси передатної функції замкнутої системи	`pole(W)`
Знайдіть коефіцієнт посилення системи в режимі, що встановився	`dcgain(W)`
Побудуйте мінімальну реалізацію передатної функції замкнутої системи від входу до сигналу управління (виходу регулятора).	`Wu = minreal(C/(1 + C*G*H))`
Побудуйте зміну сигналу управління при одиничному ступінчастому вхідному сигналі і скопіюйте графік в звіт	`step(Wu)`

5 СИНТЕЗ РЕГУЛЯТОРІВ ЗАСОБАМИ CONTROL SYSTEM TOOLBOX

5.1 Прості типи регуляторів

П-регулятор. Простий пропорційний регулятор (П-регулятор) є звичайним підсилювачем з передатною функцією

$$C(s) = K_c.$$

ПІ-регулятор. Якщо остаточна помилка регулювання є небажаною в регулятор вводять інтегруючу складову:

$$C(s) = K_c\left(1 + \frac{1}{T_I s}\right),$$

де T_I - постійна часу інтегруючої ланки.

ПД-регулятор. Для поліпшення якості регулювання і підвищення швидкодії в закон управління вводять похідну від сигналу помилки:

$$C(s) = K_c(1 + T_D s),$$

де T_D - постійна часу диференційної ланки. На практиці реалізувати ідеальне диференціювання неможливо, оскільки частотна характеристика ланки нескінченно збільшується на високих частотах. Тому диференційну ланку використовують з додатковим фільтром

$$C(s) = K_c\left(1 + \frac{T_D s}{T_f s + 1}\right).$$

Такий регулятор має назву ПДФ-регулятора (pdf-controller).

Тут постійна часу фільтру T_f зазвичай в 3–10 разів менше, ніж T_D. Надмірне збільшення T_D може привести до нестійкості системи, зменшення цієї величини затягує перехідний процес.

ПІД-регулятор. На відміну від ПД-регулятора, він містить інтегратор і система стає астатичною як по задаючій дії, так і по збуренню (тобто, постійне збурення повністю компенсується). Його передатна функція має вигляд

$$C(s) = K_c\left(1 + T_D s + \frac{1}{T_I s}\right).$$

При збільшенні T_I перехідний процес затягується, при зменшенні – зменшується запас стійкості, перехідний процес набуває вираженого коливального характеру, при подальшому зменшенні T_I втрачається стійкість.

В зашумлених системах диференційну ланку використовують з додатковим фільтром:

$$C(s) = K_c\left(1 + \frac{T_D s}{T_f s + 1} + \frac{1}{T_I s}\right).$$

Такий регулятор має назву ПІДФ-регулятора (pidf-controller).

За допомогою правильно налагодженого ПІД- або ПІДФ-регулятора у більшості випадків вдається забезпечити виконання усіх вимог до системи. В силу своєї простоти, вони отримали найширше поширення. За статистикою більше 90% усіх промислових регуляторів є саме ПІД-регуляторами.

5.2 Конструктор PID Tuner

PID Tunerпризначений для проектування в інтерактивному режимі PID регулювальника в прямому ланцюзі замкнутої одновимірної (SISO) системи регулювання з головним одиничним негативним зворотним зв'язком (рис. 5.1)

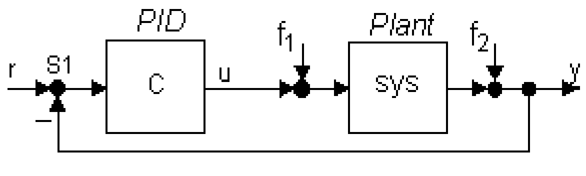


Рис. 5.1 – Схема, що використовується програмою PID Tuner: *PID* – регулятор; *Plant*– об’єкт управління

PID Tuner дозволяє сконструювати дві форми реалізації регулятора С:

а) паралельну форму з ПФ у вигляді:

$$C(s) = K_p + \frac{K_i}{s} + K_d s; \tag{5.1}$$

б) стандартну форму:

$$C(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right); \tag{5.2}$$

PID Tuner запускається командою *pidtool*, яка вводиться у командному вікні MATLAB і спричиняє появу діалогового вікна, показаного на рис. 5.2.

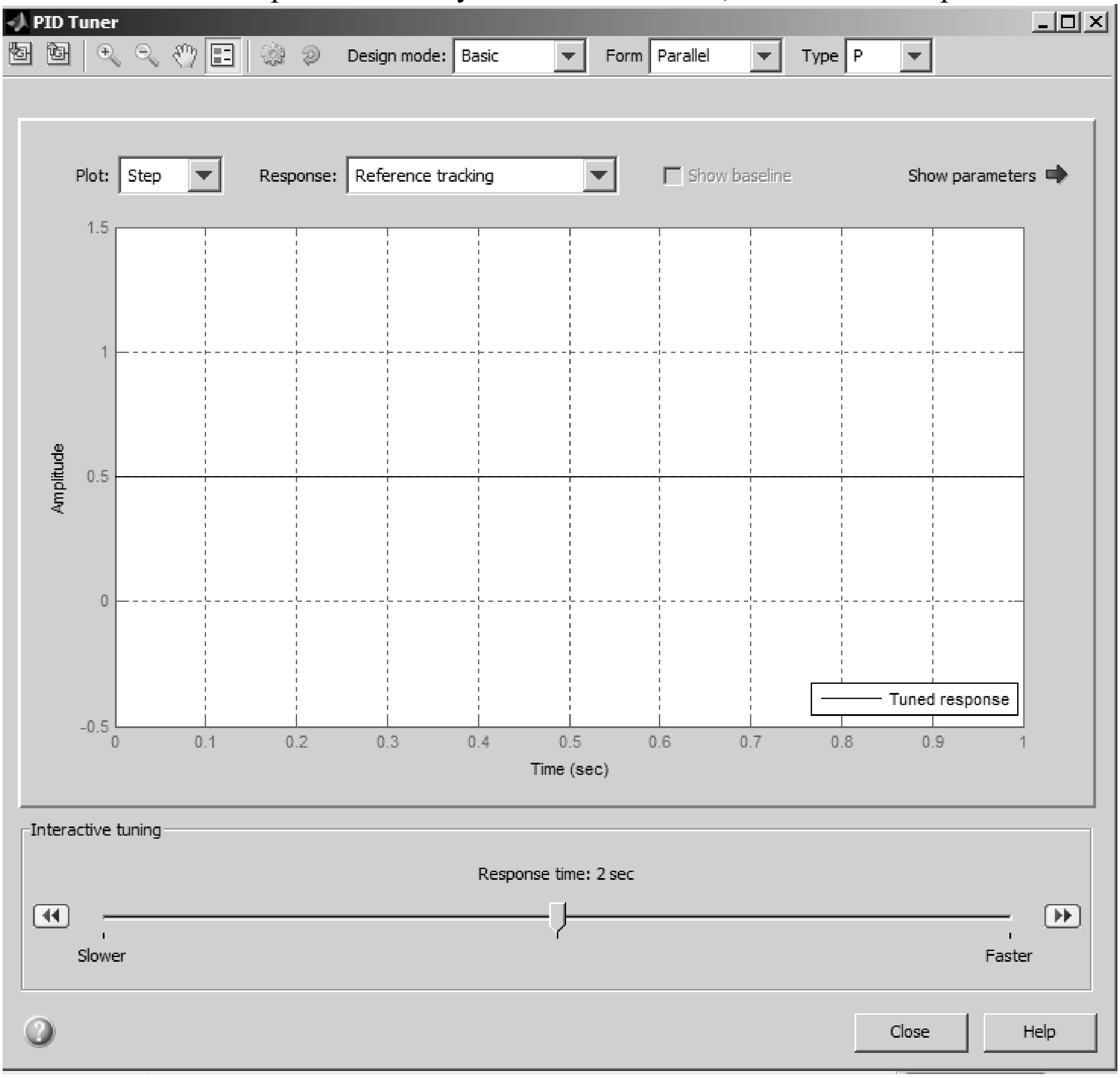
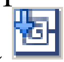


Рис. 5.2 – Діалогове вікно PID Tuner

Діалогове вікно дозволяє вибирати спосіб синтезу *DesignMode* (Basic, Extended), форму опису *Form* (Parallel, Standard), тип регулятора *Type* (P, I, PI, PD, PDF, PID, PIDF), тип графіка *Plot* (Step, Bode) і вид аналізованої реакції в меню Response:

- *Referencetracking* – вплив завдання на керовану величину замкнутої системи (від r до y);
- *Controllereffort* – вплив завдання на дію замкнутої системи (, що управляє, від r до u);
- *InputdisturbanceRejection* – вплив збурення на вході об’єкта на керовану величину замкнутої системи (від f_1 до y)
- *OutputdisturbanceRejection* – вплив збурення на виході об’єкта (наприклад, шумів виміру) на керовану величину замкнутої системи (відf_2доy);
- *Open-loop* – характеристики розімкненої системи (потрібні при частотних методах дослідження стійкості, наприклад, за ЛЧХ);
- *Plant* – характеристики тільки об'єкта управління.

У рядку меню діалогового вікна є кнопка  імпорту об’єкта регулювання *PlantModel* або зразкового регулювальника *BaselineController* з робочого простору Workspace, кнопка експорту отриманого регулювальника як об'єкта в робочий простір Workspace. PID Tuner створює за умовчанням перехідну характеристику замкнутої системи з регулювальником, параметри якого вибрані збалансовано між вимогами найменшого часу регулювання і задовільної робасності (прийнятних запасів стійкості).

Усі стандартні показники якості (стале значення, час наростання, час регулювання, перерегулювання, час максимуму) можна викликати клацанням правої кнопки миші на графіці. Якщо значення показників не задовольняють технічним вимогам, що пред'являються, до системи, їх можна відкоригувати за допомогою повзунка *Responsetime*: у бік *Slower* (повільніше) або *Faster* (швидше), безпосередньо спостерігаючи зміни в табличках біля відповідних маркерів. Час *Responsetime* не пов'язаний із стандартними поняттями «час регулювання», «час наростання».

Кнопка *Show parameters*викликає додаткове поле з панелями *Controller parameters*і *Performance and robustness*. Слід мати на увазі, що на правій панелі виводяться параметри завжди тільки для стандартних установок MATLAB: час наростання по діапазону 0.1–0.9, час регулювання по рівню 2%, тоді як на графіці ліворуч ті ж параметри відображуються при фактичних установках, заданих у вікні ControlSystemToolboxPreferences користувачем. Відповідно, значення *Risetime* і *Settlingtime* для однієї і тієї ж системи ліворуч і праворуч можуть бути різними.

Для незалежного налаштування смуги пропущення і запасу за фазою слід вибрати опцію **Extended** в меню **DesignMode**.

Для проектування регулювальника з командного рядка використовується функція **pidtune** в наступному виді:

- `rgp=pidtune(plant,'type')` – для SISO об’єкта *plant* проектується регулювальник паралельної форми вказаного типу *type* (p, i, pi, pd, pid, pidf), параметри якого вибрані збалансовано між вимогами найменшого часу регулювання і задовільної робасності (запасів стійкості);

- `rgc=pidtune(plant,cbase)` – проектування ведеться порівняно із зразковим регулювальником *cbase* і результується у вигляді регулятораа того ж типу і форми;

- `rgp=pidtune(plant,'type',options),rgc=pidtune(plant,cbase, options)` – визначає додаткові параметри налаштування;

- `[rgp,info] = pidtune(plant,...)` – повертає додаткові відомості по проектованому регулювальникові.

5.3 Приклади і задачі для самостійного розв'язання

Задача 5.1. Об'єкт управління має передавальну функцію:

$$W_{oy}(s)=\frac{K_{об}e^{-\tau s}}{(T_1 s+1)(T_2 s+1)},$$

де $K_{об}$=1,2, T_1=10, T_2=3, τ=3. Визначити параметри налаштування ПІД-регулятора і показники роботи системи регулювання.

Розв'язування

Представимо ПФ об'єкта у вигляді послідовного з'єднання двох ПФ:

$$W_1(s)=\frac{K_{об}}{T_1 s+1} \text{ та } W_2(s)=\frac{e^{-\tau s}}{T_2 s+1}.$$

У командному вікні MATLAB уводимо наступні команди.

Створення ПФ $W_1(s)$:

```
>>W1=tf([1.2], [10 1])
Transfer function:
  1.2
--------
10 s + 1
```

Створення ПФ $W_2(s)$:

```
>>W2=tf([1], [3 1],'ioDelay',3)
Transfer function:
                 1
exp(-3*s) * -------
             3 s + 1
```

Послідовне з'єднання $W_1(s)$ та $W_2(s)$:

```
>>W=series(W1, W2)

Transfer function:
                      1.2
exp(-3*s) * -----------------
             30 s^2 + 13 s + 1
```

Виклик програми PID Tuner:

>>pidtool(W,'pid')

У діалоговому вікні PID Tuner переміщенням повзунка *Responsetime* домагаємося бажаної форми перехідного процесу. Натиснувши кнопку

Showparameters, отримуємо додаткове поле з панелями *Controllerparameters* і *Performanceandrobustness*, з яких читаємо параметри налаштування регулятора і показники роботи системи регулювання.

Задача 5.2. Для об'єкта управління з ПФ, заданої в задачі 5.1, побудувати засобами ControlSystemToolboxLTI-модель. Параметри моделі взяти з табл. 5.1.

Таблиця 5.1 – Варіанти завдань

№	*K*	*T1*	*T2*	tau	№	*K*	*T1*	*T2*	tau
1	1,5	10	5	2	14	3,5	14	4	3
2	2,0	15	4	3	15	3,2	11	5	2
3	3,0	17	3	4	16	0,5	16	3	3
4	3,5	18	4	2	17	1,2	15	4	2
5	3,2	12	5	3	18	2,5	14	5	4
6	0,5	13	7	4	19	1,7	16	4	5
7	1,2	14	6	3	20	2,1	17	3	3
8	2,5	11	3	2	21	2,0	14	7	4
9	1,7	18	3	3	22	1,8	10	4	3
10	2,1	12	4	4	23	2,4	11	3	3
11	1,5	13	2	3	24	1,7	12	6	4
12	2,0	12	3	4	25	2,5	17	7	3
13	3,0	17	4	5					

З використанням PID Tuner визначити і занести у табл. 5.2параметри налаштування П-, ПІ-, ПДФ-, ПІД- і ПІДФ-регуляторівK_p, K_i, K_d, T_fта показники роботи системи регулювання:

а) час регулювання t_p (*Settingtime*);

б) перерегулюванняσ (*Overshoot*);

в) пікове значення $y_{\max}$ (*Peak*);

г) запас стійкості по модулю ΔA(*Gainmargin*);

д) запас стійкості по фазі $\Delta\varphi$(*Phasemargin*).

Таблиця 5.2

Параметри	K_p	K_i	K_d	T_f	t_p, с	σ, %	$y_{\max}$	ΔA, дБ	$\Delta\varphi$, град.	Характер перехідного процесу
П										
ПІ										
ПДФ										
ПІД										
ПІДФ										

Контрольні питання

1. Що таке LTI-модель?

2. Як створити LTI-модель у вигляді передавальної функції?

3. Назвіть призначення програми PID Tuner.

4. Наведіть формулу ПФ ПІД-регулятора у паралельній формі.

4. Наведіть формулу ПФ ПІД-регулятора у стандартній формі.

6. Як визначити параметри налаштування ПІД-регулятора за допомогою PID Tuner?

7. Які показники роботи системи регулювання дозволяє визначити PID Tuner?

6 ПІД-РЕГУЛЯТОР З ДОДАТКОВОЮ КЕРУЮЧОЮ ДІЄЮ ДИФЕРЕНЦІАТОРА

6.1 Побудова регуляторів з додатковою керуючою дією

Стандартні регулювальники не завжди забезпечують необхідну якість регулювання об'єктів із запізнюванням. Якість управління за допомогою ПІД-регулятора можна покращити, вводячи в структуру регулятора додаткові елементи.

Структурна схемаПІД-ПД-регулятора представлена на рис. 6.1.

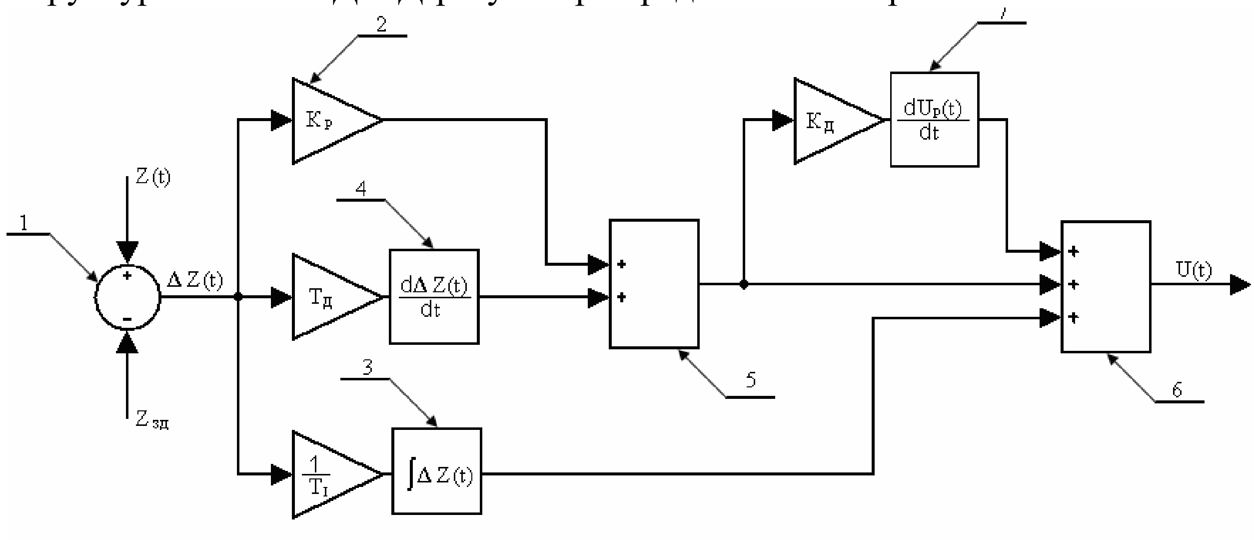


Рис. 6.1

На вхід додаткового диференціатора 7 подається вихідний сигнал суматора 5, який підсумовує вихідні сигнали блоку пропорційного перетворювання 2 та блоку диференціювання 4 стандартного ПІД-регулятора. Додаткова керуюча дія диференціатора 7 додається доосновного керуючого сигналу ПІД-регулятора всуматорі 6, вихідний сигнал якого є загальнимрегулюючим сигналом цього регулятора. Передатна функція такого регулятора

$$W(s)=K_p+\frac{1}{T_s\cdot s}+(T_Д+K_p\cdot K_д)\cdot s+T_д\cdot K_д\cdot s^2. \quad (6.1)$$

Схема ПІД-ІД регулятора показана на рис 6.2.

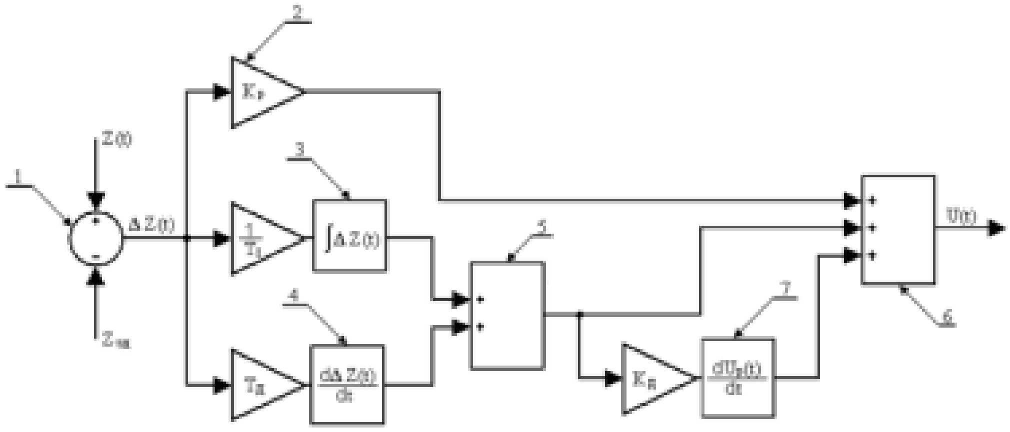


Рис. 6.2

Цей ПІД-регулятор відрізняється тим, що вихід блоку пропорційного перетворення 2 сполучений з входом другого суматора 6, а виходи блоку інтеграції 3 і першого блоку диференціювання 4 відповідно сполучені з входами першого суматора 5, вихід другого сполучений з входом другого блоку диференціювання 7. Вихід другого суматора 6 є керуючою дією цього регулятора.

Відповідно до представленої структурної схеми передавальна функція такого регулятора має вигляд:

$$W_{(s)} = K_P + \frac{K_Д}{T_i} + \frac{1}{T_i \cdot s} + T_Д \cdot s + T_Д \cdot K_Д \cdot s^2 . \tag{6.2}$$

ПІД-ЗД-регулятор виконує додаткове диференціювання всіх трьох вихідних складових стандартного ПІД-регулятора (рис. 6.3)

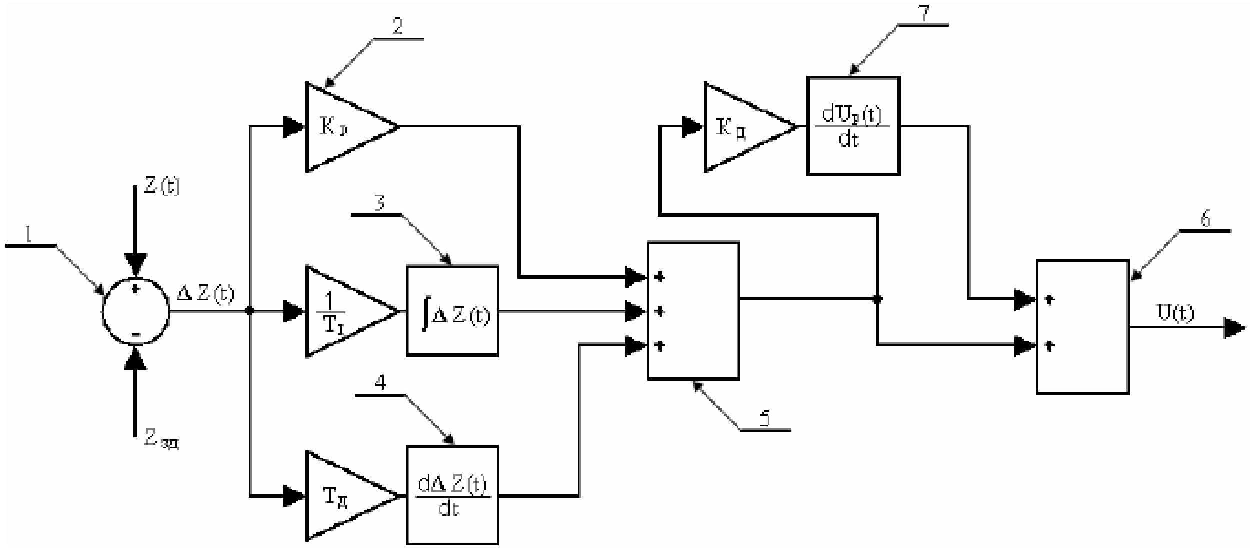


Рис. 6.3

Для цього регулятора маємо передатну функцію

$$W(s) = K_р + \frac{1}{T_i \cdot s} + T_Д \cdot s + (K_р + \frac{1}{T_i \cdot s} + T_Д \cdot s) \cdot K_Д \cdot s . \tag{6.3}$$

Як бачимо, для регуляторів з додатковою дією необхідне визначення чотирьох параметрів налаштування: $K_р$, T_i , $T_Д$та$K_Д$.

Формули розрахунку параметрів налаштування ПІД-ІД регулятора шукались методом статистичного комп'ютерного експерименту для двоємкісних об'єктів із запізненням, які мають передатну функцію

$$W_o(s) = \frac{K_o e^{-\tau s}}{(T_1 s + 1)(T_2 s + 1)}, \qquad (6.4)$$

де коефіцієнт передачі $K_о$ варіюється в межах від 0,4 до 4 %/%, сталі часу T1 і T2 від 1 до 35 одиниць часу. Що стосується чистого запізнення τ, то у ході статистичного експерименту використовувалось відносне запізнення $\tau_в=\tau/T_o$, де T_o – загальна стала часу об'єкта. Відносне запізнення виріювалося в межах від 0,05 до 1,0.

При опрацюванні отриманих статистичних даних з метою досягнення більш точних розрахунків досліджувані об'єкти ділили на дві групи за співвідношеннями τ/T_o<=0,3 та τ/T_o>0,3 і для кожної групи одержували свої рівняння регресії.

Для об'єктів зі співвідношенням τ/T_o<=0,3 отримані такі рівняння регресії:

- коефіцієнт підсилення регулятора:

$$K_{p1} = (0.298\text{-}72.0967\tau_в)/(1.0\text{-}18.0885\tau_в);$$

- стала часу інтегрування:

$$T_{i1}=(0.598+1.477\tau_в +12.857\ \tau_в^2)\cdot(T_o/18.5);$$

- стала часу диференціювання:

$$T_{d1} = (26.2\text{-}72.485\tau_в +105.714\ \tau_в^2)\cdot(T_o/18.5);$$

- коефіцієнт додаткового диференціювання:

$$K_{d1} = ((1.6443\text{-}16.636\tau_в)/(1\text{-}12.6592\tau_в))\cdot(T_o/15).$$

Для співвідношення τ/T_o>0,3:

$$K_{p1} = (9.9878\text{-}21.2\tau_в+12.982\tau_в^2);$$
$$T_{i1} = (0.583\text{-}1.12\tau_в+20.5714\tau_в^2)\cdot(T_o/18.5);$$
$$T_{d1} = (15.267\text{-}1.5929\tau_в\text{-}8.2143\tau_в^2)\cdot(T_o/18.5);$$
$$K_{d1} = (3.4798\text{-}6.2025\tau_в+2.9296\tau_в^2)\cdot(T_o/18.5).$$

Ці налаштування далі уточнюються наступними формулами:

- для співвідношенняT_1/T_2<=0,4:

$$K_p= K_{p1}/K_о;$$
$$T_i=T_{i1}\,(0.764+1.6298(T_1/T_2)–2.5973(T_1/T_2)^2K_о;$$
$$T_d= T_{d1}\,(0.7962\text{-}0.1333(T_1/T_2)+0.9416(T_1/T_2)^2/K_о;$$
$$K_d=K_{d1}\,(0.8304+0.4262(T_1/T_2));$$

- для співвідношенняT_1/T_2>0,4:

$$K_p= K_{p1}/K_о;$$
$$T_i=T_{i1}\,(1.4235–1.7065(T_1/T_2)+1.615(T_1/T_2)^2K_о;$$
$$T_d= T_{d1}\,(1.2068–0.8698(T_1/T_2)+0.877(T_1/T_2)^2/K_о;$$
$$K_d=K_{d1}\,(0.8333+0.4167(T_1/T_2)).$$

6.2 Класичний метод синтезу систем регулювання

Синтез системи автоматичного регулювання технологічного параметра звичайно починається з експериментального дослідження об'єкта управління і побудови кривої розгону, яку іноді називають розгінною характеристикою. Для її отримання треба на вхід об'єкта подати ступінчастий вплив. Визначають чинник, який впливає на регульовану величину, наприклад, температуру можна регулювати зміною подачі граючої пари. Стрибком змінюють вхідну дію на декілька відсотків ходу виконавчого механізму і спостерігають за зміною регульованої величини. Графік зміни регульованої величини обробляють, як показано на рис. 6.4. Тим самим реальний об'єкт (суцільна лінія) апроксимується об'єктом першого порядку з чистим запізненням L(пунктирна лінія).

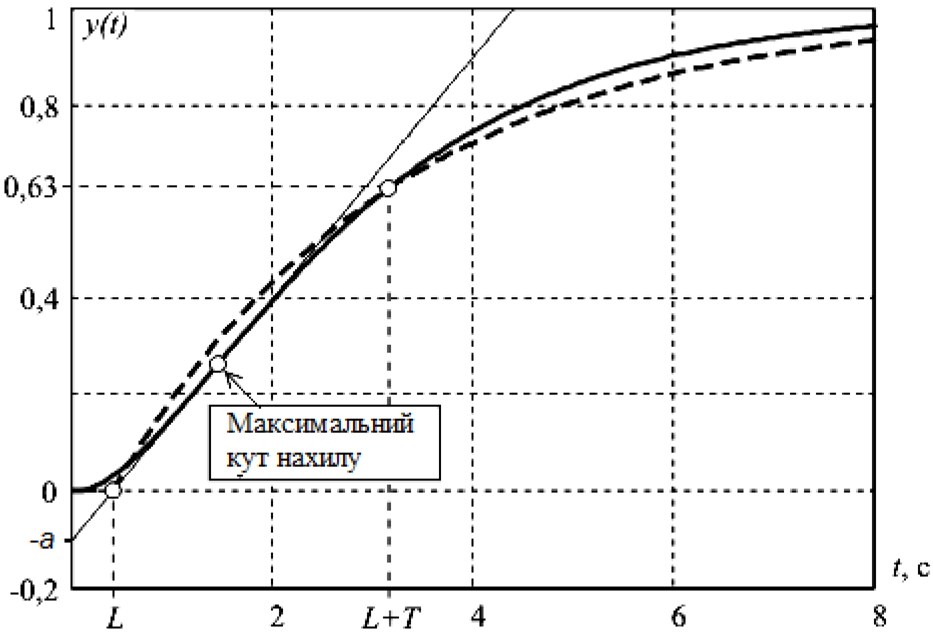


Рис. 6.4

За допомогою графіка визначають параметри a та L, відношення яких є рівним тангенсу кута нахилу дотичної прямої. Знаходять також коефіцієнт передачі як відношення приросту сталого значення вихідної змінної до амплітуди тестового стрибка.

Далі за допомогою ряду формул розраховують параметри налаштування регулятора. Найбільш вживаними безпошуковими методами розрахунку параметрів налаштувань регуляторів є метод Циглера-Нікольса (Ziegler-Nichols), метод Коена-Куна (Cohen-Coon), метод Чина-Хронса-Ресвіка (Chien-Hrones-Reswick – CHR).

Розрахувавши параметри, комп'ютерним моделюванням або на реальному об'єкті отримують перехідний процес системи регулятор-об'єкт.

6.3 Синтез з використанням графічного інтерфейсу PIDD

Графічний інтерфейс PIDD дозволяє в інтерактивному режимі швидко виконати проектування контуру регулювання з додатковою керуючою дією, починаючи з ідентифікації об'єкта управління і закінчуючи побудовою і аналізом перехідного процесу спроектованої системи.

Для використання графічного інтерфейсу треба зробити PIDD поточною директорією (Current folder) і ввести у командному вікні текст «PIDD». Відкривається вікно, показане на рис. 6.5.

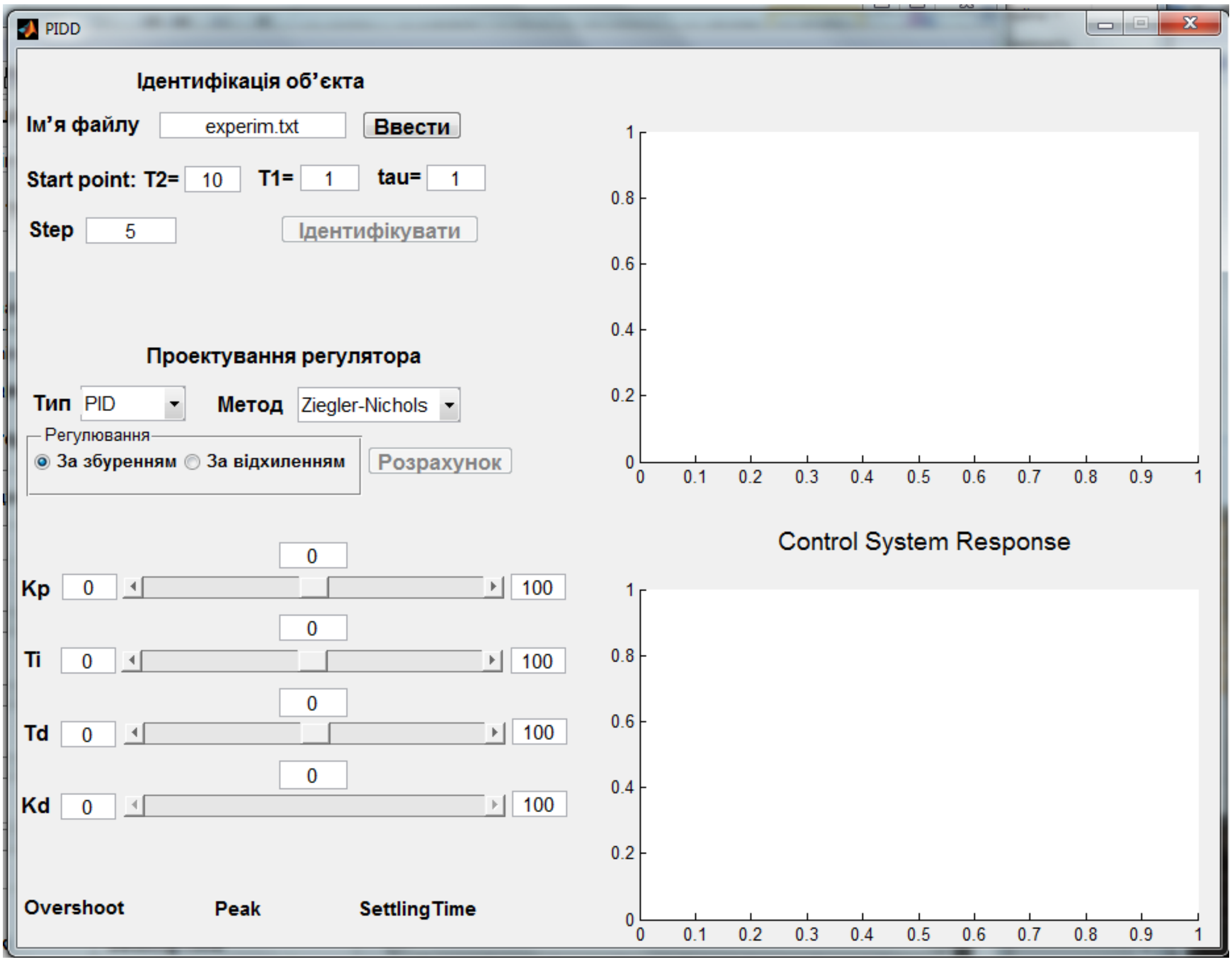


Рис. 6.5

Після натискання кнопки **Input** в програму завантажуються данні кривої розгону з текстового файлу, ім'я якого вписано в полі **Filename**, та у вікні axes1 будується графік кривої розгону. Текстовий файл має два рядки. У першому через кому записуються моменти часу, в які знімались відліки вихідної величини об'єкта, у другому – відповідні значення вихідної величини.

Апроксимація об'єкта моделлю у вигляді передатної функції (6.4) здійснюється шляхом мінімізації квадрату різниці значень вихідних величин моделі і об'єкта. Для цього використовується функція MATLAB:

```
[x,fval]=fmincon('TFobj',[T2,T1,tau],[],[],[],[],lb,ub,[],options);
```

Опції мінімізації задаються командою:

```
options=optimset('Algorithm','active-set','TolFun',1e-4,'TolX',1e-4);
```

Апроксимація запускається натисканням кнопки «Identification». В результаті будується графік вихідної величини моделі і виводиться вигляд передатної функції.

Далі можна вибрати тип регулятора: PID – стандартний ПІД-регулятор або PID-ID– регулятор з передатною функцією (6.2). Для стандартного ПІД-регулятора вибирається один з методів розрахунку Ziegler-Nichols, Cohen-Coon або CHR.

Комплект радіокнопок під заголовком «Регулювання» дозволяє вибрати тип перехідного процесу: за збуренням чи за відхиленням.

Натискання кнопки «Design» призводить до розрахунку параметрів налаштування вибраного типу регулятора і побудови графіка перехідного процесу САР. Параметри налаштування можна змінювати, пересуваючи движок відповідного слайдера в нижній частині вікна.

6.4 Приклади і задачі для самостійного розв'язання

Задача 6.1. Спроектувати ПІД-ІД регулятор для динамічного об’єкта, крива розгону якого представлена даними табл. 6.1.

Таблиця 6.1 – Вихідні дані до задачі 6.1

T	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
h	0	0	0	0.07	0.23	0.43	0.64	0.84	1.03	1.19	1.33	1.45	1.55	1.64	1.71	1.77

Розв'язування

1. Копіюємо у свою робочу теку директорію D:\Problem\PIDD.
2. У вікні Current folder робимо скопійовану директорію поточною.
3. Створюємо текстовий файл, що включає два рядки. В перший рядок заносимо моменти часу T, у другий – значення відгуку h з таблиці 6.1. Усі числа в обох рядках розділяємо комами.
4. У командному вікні MATLABвводимо PIDD.
5. У поле «Ім’я файлу» вводимо ім’я файлу, створеного за п. 3, і натискуємо кнопку «Ввести». Будується графік кривої розгону.
6. Підбираючи початкові значення для T2, T1 і tau та натискуючи кнопку «Ідентифікація», отримуємо графік виходу моделі, яка апроксимує експериментальну криву розгону (рис. 6.6).

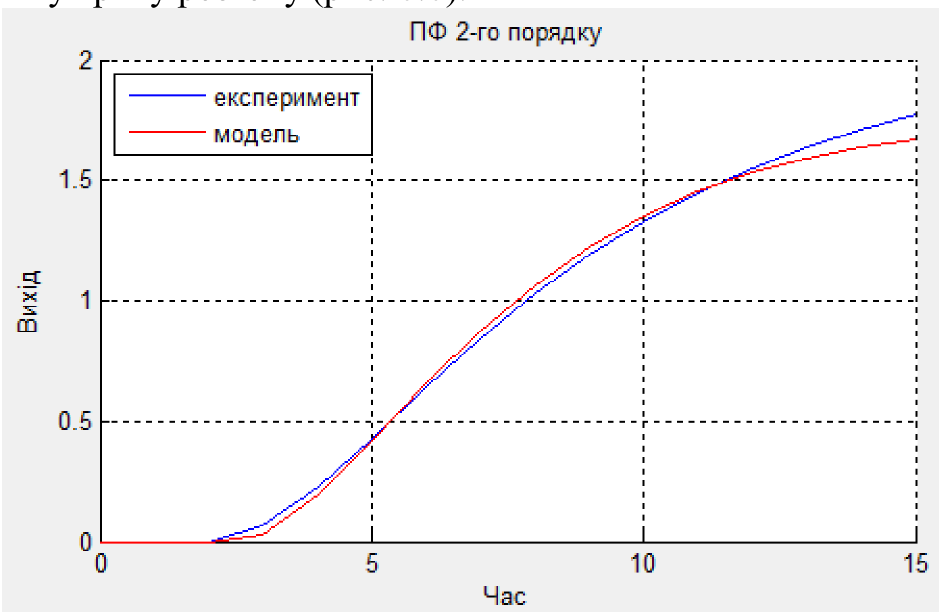


Рис. 6.6

7. Переписуємо отриману передатну функцію:

```
        0.354*exp(-2.4864*s)
Wo = -----------------------------
     (2.7178*s + 1)(2.7138*s + 1)
```

8 Підбираючи метод розрахунку Ziegler-Nichols, Cohen-Coon чи CHR, і натискуючи кнопку «Розрахунок», вибираємо бажаний вид перехідного процесу (рис. 6.7).

9. Заносимо у таблицю 6.2 параметри налаштування і характеристики перехідного процесу.

Таблиця 6.2

Параметри	K_p	T_i	$T_Д$	$K_Д$	σ	y_{max}	t_p
ПІД	4.09	3.84	0.79	–	53.9	1.54	48.4
ПІД-ІД	4.81	0.83	7.68	0.18	36.8	1.37	21.0

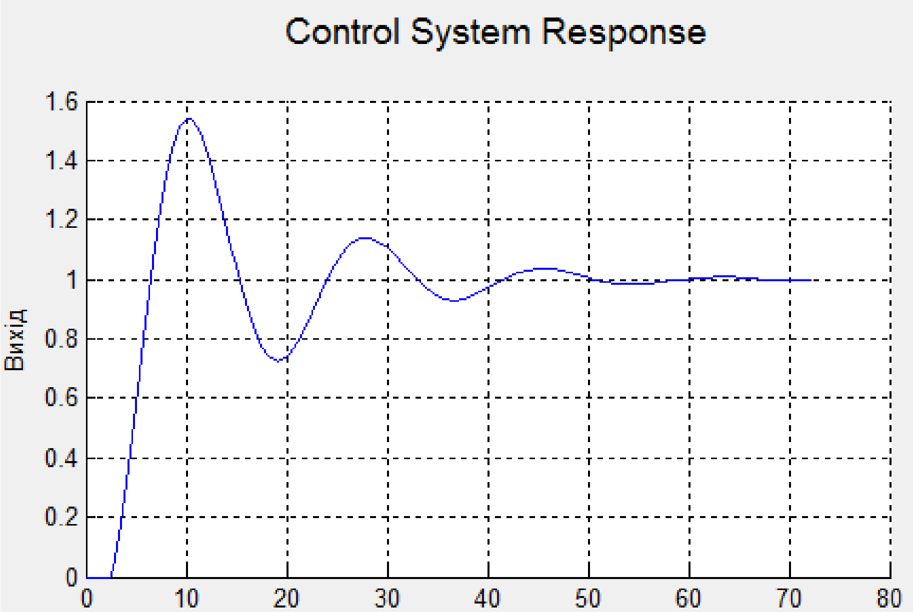


Рис. 6.7

Параметри налаштування K_p, T_i , $T_Д$ зчитуємо у віконцях над відповідними слайдерами Kp, Ti, Td. Характеристики перехідного процесу виводяться під слайдерами:

– Overshoot – перерегулюванняσ;
– Peak – пікове значення y_{max};
– Setting time – час регулювання t_p .

10. Встановлюємо тип регулятора PID-ID і натискуємо кнопку «Розрахунок». Заносимо у табл. 6.2 параметри налаштування K_p, T_i , $T_Д$ та $K_Д$ і характеристики перехідного процесу. Зберігаємо графік перехідного процесу (рис. 6.8).

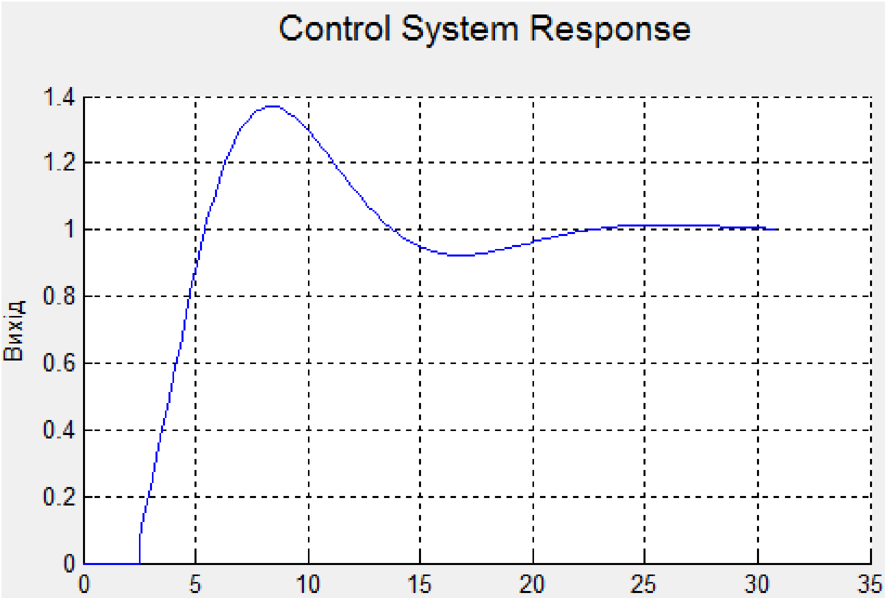


Рис. 6.8

11. Порівнюючи отримані результати, робимо висновок, що регулятор PID-ID забезпечує кращу якість регулювання, ніж PID.

Задача 6.2. Спроектувати ПІД-ІД регулятор для динамічного об’єкта, крива розгону якого представлена даними табл. 3.3.

7 ВИКОРИСТАННЯ ПАКЕТУ SISO DESIGN TOOL ДЛЯ ПРОЕКТУВАННЯ І АНАЛІЗУ СИСТЕМ УПРАВЛІННЯ

7.1 Основи роботи з SISO Design Tool

7.1.1 **Запуск SISO Design Tool**

Пакет SISO Design Tool призначений для синтезу одноконтурних лінійних систем автоматичного регулювання (SISO розшифровується як Single Input Single Output, тобто є інструментом для синтезу систем з одним входом і одним виходом).

Щоб запустити пакет, треба набрати у вікніMATLAB:

```
>>controlSystemDesigner
```

З'явиться графічний інтерфейс користувача Control and Estimation Tools Manager та вікно графіків, як показано на рис.7.1.

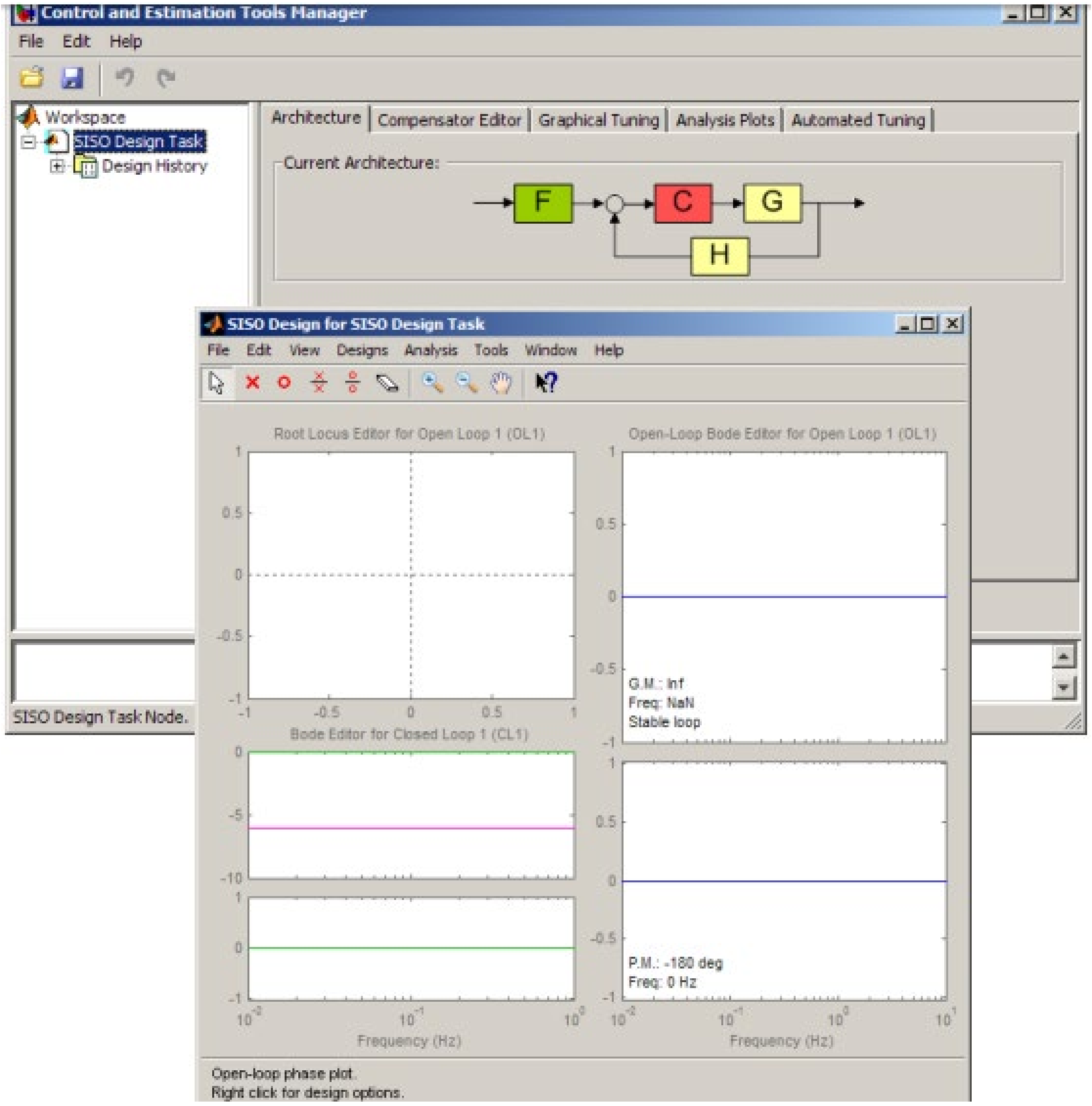


Рис. 7.1 – Графічний інтерфейс користувача

Основне робоче вікномістить п'ять вкладок:

а) Architecture – визначення архітектури системи регулювання, що синтезується;

б) Compensator Editor – редактор ланки-регулятора;

в) Graphical Tuning – налаштування графіків, що виводяться в основному вікні програми;

г) Analysis Plots – налаштування графіків, використовуваних для аналізу отримуваних якісних і кількісних характеристик системи регулювання, що синтезується, в цілому;

д) Automated Tuning – майстер автоматичного синтезу.

Відразу після старту у вікні менеджера є активною перша вкладка, яка містить чотири кнопки :

а) Control Architecture – налаштування архітектури одновимірного замкнутого контура регулювання (вибір його типу) і визначення вхідного сигналу та контрольних точок;

б) Loop Configuration – конфігурація додаткових відкритих контурів для здійснення синтезу багатовходової (декілька входів, один вихід) системи регулювання;

в) System Data – майстер імпортування частин системи регулювання;

г) Sample Time Conversion - конвертор стану системи з неперервного в дискретний або навпаки.

По-перше, необхідно визначити структуру системи регулювання, що синтезується. Це робиться натисненням кнопки Control Architecture першої вкладки менеджера і вибору потрібної архітектури. У пакеті передбачені чотири типи архітектури. Складові архітектури позначають символами:

а) символ F (Prefilter – попередній фільтр) – сукупність ланок, розташованих між сигналом завдання і суматором. Застосовується для корекції чи масштабування вхідного сигналу;

б) символи C, C1 і C2 (Compensator - компенсатор) – ланки регулятора, характеристики, що безпосередньо коригують і структури якого можутьмінятися для здійснення синтезу системи регулювання. У разі, якщо ланок, що коригують, дві і вони розташовані в різних гілках, застосовуються символи C1 і C2;

в) символ G (Plant – об’єкт управління) – сукупність ланок, що лежать в прямій гілці замкнутого контура і не включають регулятор та зворотний зв'язок;

г) символ H (Sensor – датчик) – сукупність ланок, розташованих в зворотному зв'язку системи регулювання і що не включають ланку, що коригує

За замовчанням передатні функції для усіх моделей є рівними 1.

Щоб задати ПФ тієї чи іншої ланки треба імпортувати її LTI-модель, створену у командному вікні MATLAB. Ви можете це зробити натисненням кнопки **System Data** (рис. 7.2).

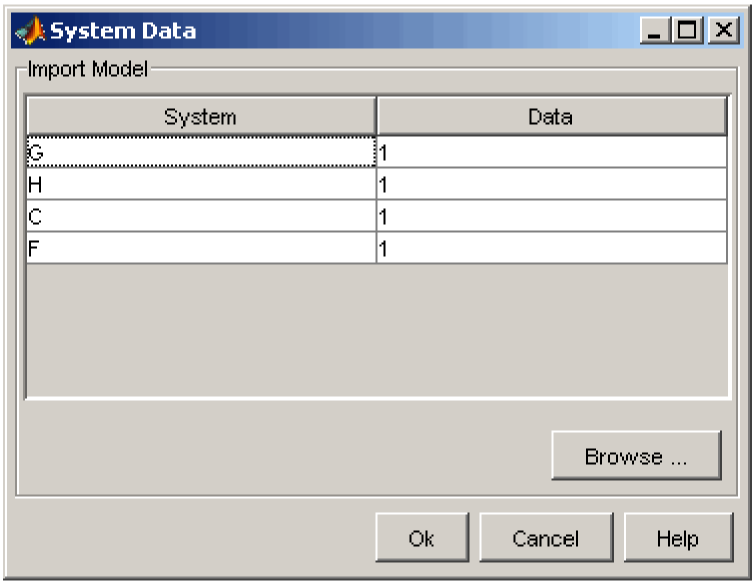


Рис. 7.2 – Вікно імпорту ПФ

Треба виділити необхідну складову і натиснути кнопку Browse. З'явиться вікно, яке містить моделі, наявні у робочому просторі MATLAB (рис. 7.3).

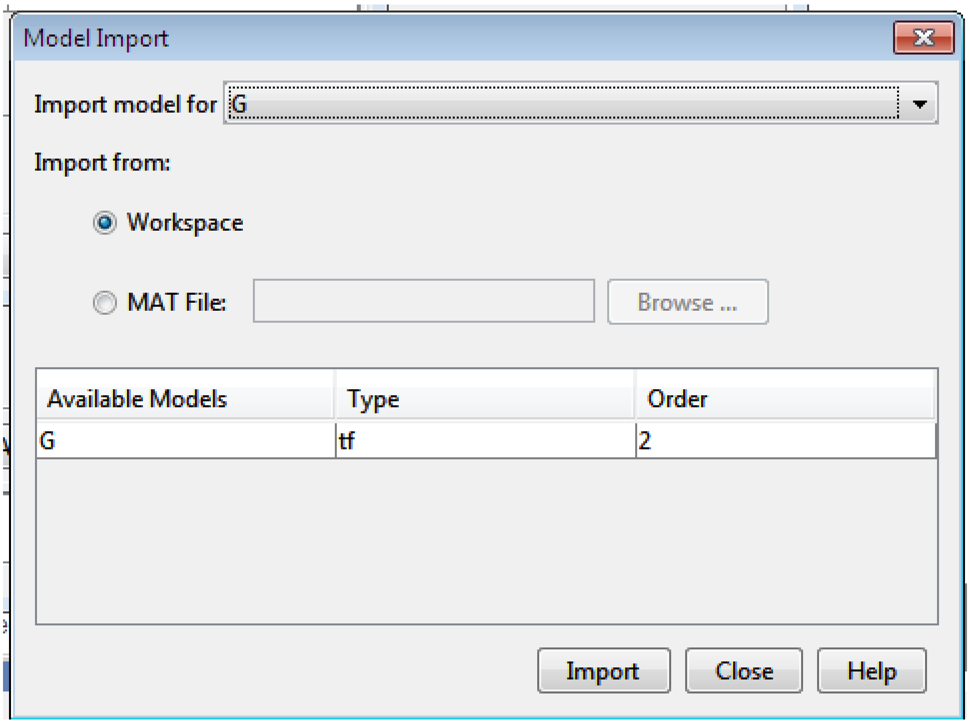


Рис. 7.3 – Імпорт моделей

Виділяємо необхідну LTI-модель і тиснемо кнопку **Import**.

7.1.2 Розрахунок ПІД-регулятора

Для синтезу ПІД-регулятора треба відкрити вкладку Automated Tuning (рис. 7.4) і вибрати один з методів синтезу, наприклад PID Tuning. Далі треба встановить специфікації синтезу: вибирається метод (Tuning method), тип регулятора (Controller type), метод розрахунку налаштувань (Formula).

Синтез регулятора виконується натисненням кнопки **Update Compensator.**

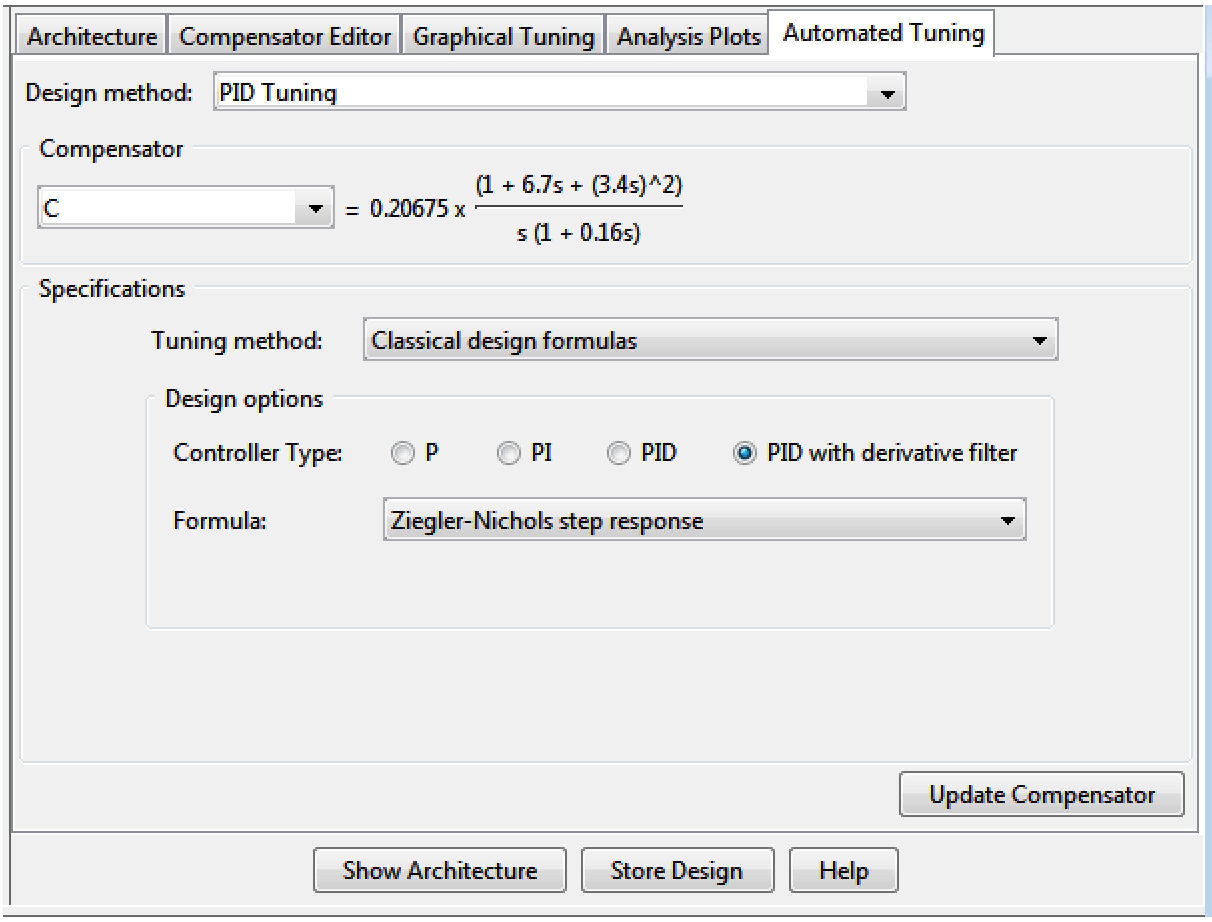


Рис. 7.4

7.1.3 Аналіз результатів

Відкриваємо вкладку Analysis Plots (рис. 7.5).

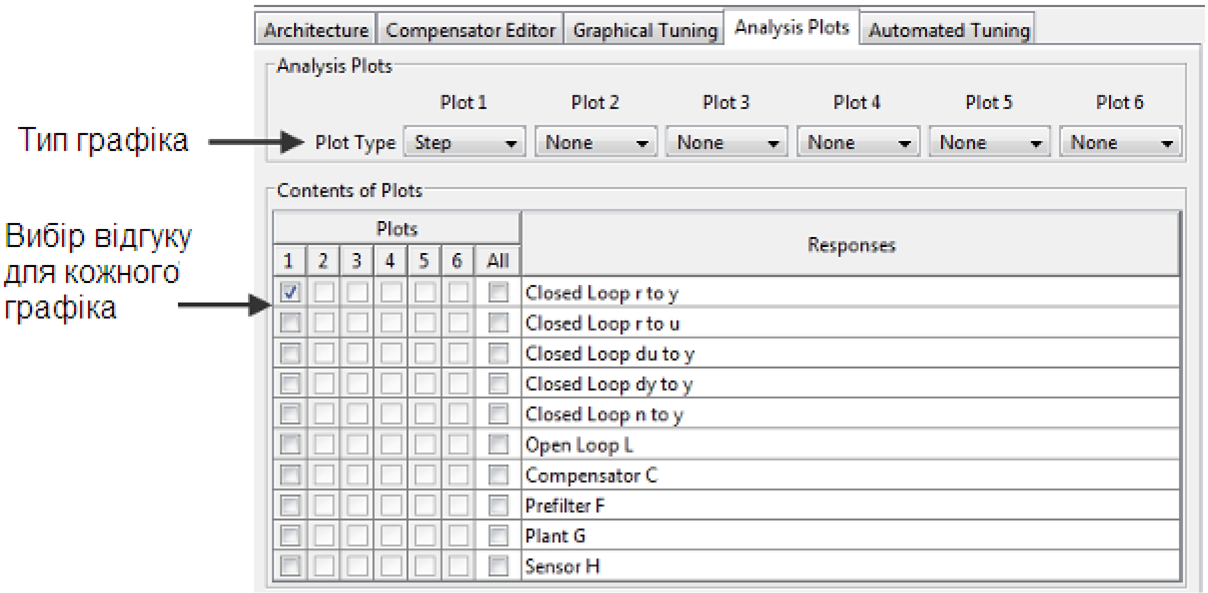


Рис. 7.5

Надається можливість побудови до шести графіків наступних типів:

а) Step – перехідний процес;

б) Impulse – вагова функція;

в) Bode – діаграми Боде;

г) Nyquist– годограф Найквиста;

д) Nichols – диаграмма Николса;

е) Pole/Zero – розміщення полюсів і нулів.

Для визначення числових характеристик системи треба клацнути правою кнопкою миші, щоб вибрати потрібні характеристики (рис. 7.6).

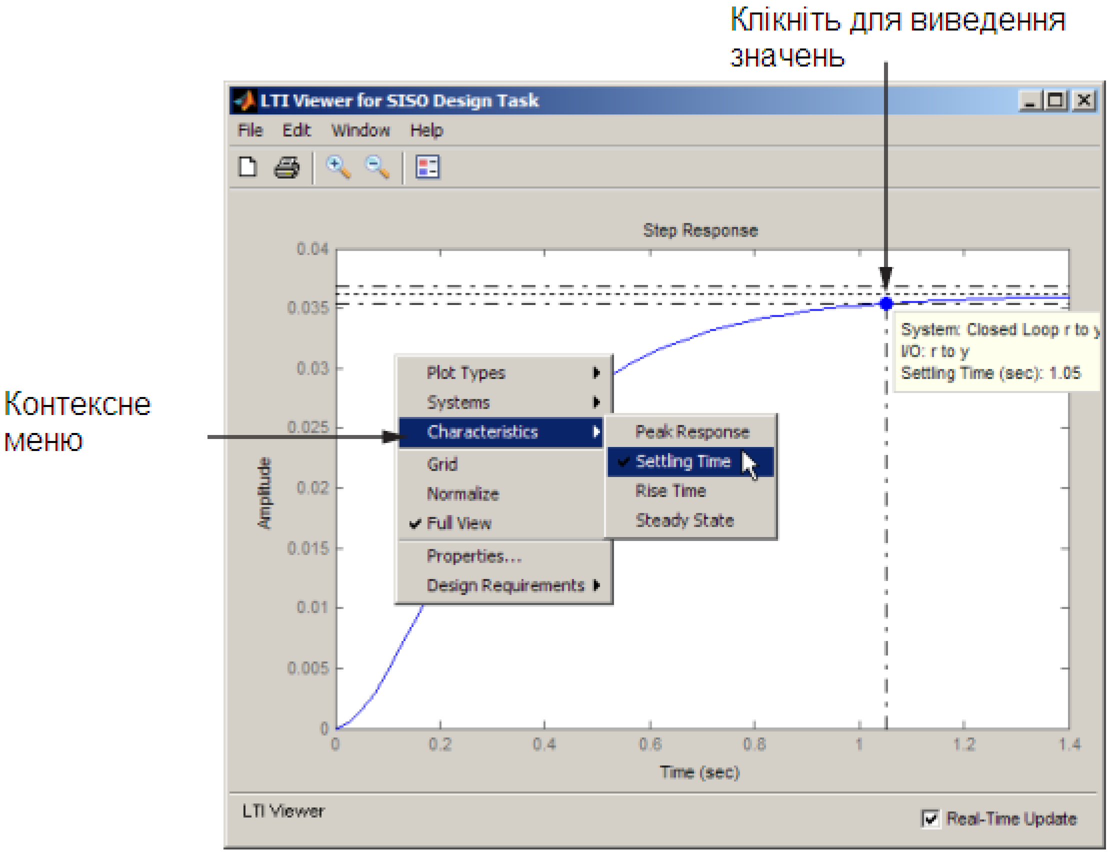


Рис. 7.6

Клацніть по точці, яка представляє характерне значення.

Існує можливість установити параметри (обмеження) перехідного процесу, для цього необхідно вибрати у контекстному меню "DesignRequirements–New" і у діалоговому вікні, що відкрилося, натиснути кнопку "OK".

В результаті будуть встановлені обмеження за умовчанням. Для зміни обмежень необхідно виконати у вікні графіку контекстне меню "Design Requirements – Edit". Відкриється діалогове вікно, в якому необхідно заповнити описані нижче поля і натиснути кнопку "Close":

- Overshoot – перерегулювання в %.
- SettingTime – загальний час.
- RiseTime – час першого блоку (не може бути рівним загальному часу).
- Undershoot – мінімальне значення (як правило, 0).
- Settings – ширина третього блоку, відносно вихідного значення.
- Rise – висота першого блоку в % (обов'язково менше 100).
- FinalValue – вихідне значення (як правило, 1).

В результаті у вікні графіку будуть встановлені обмеження, як показано на рис. 7.7.

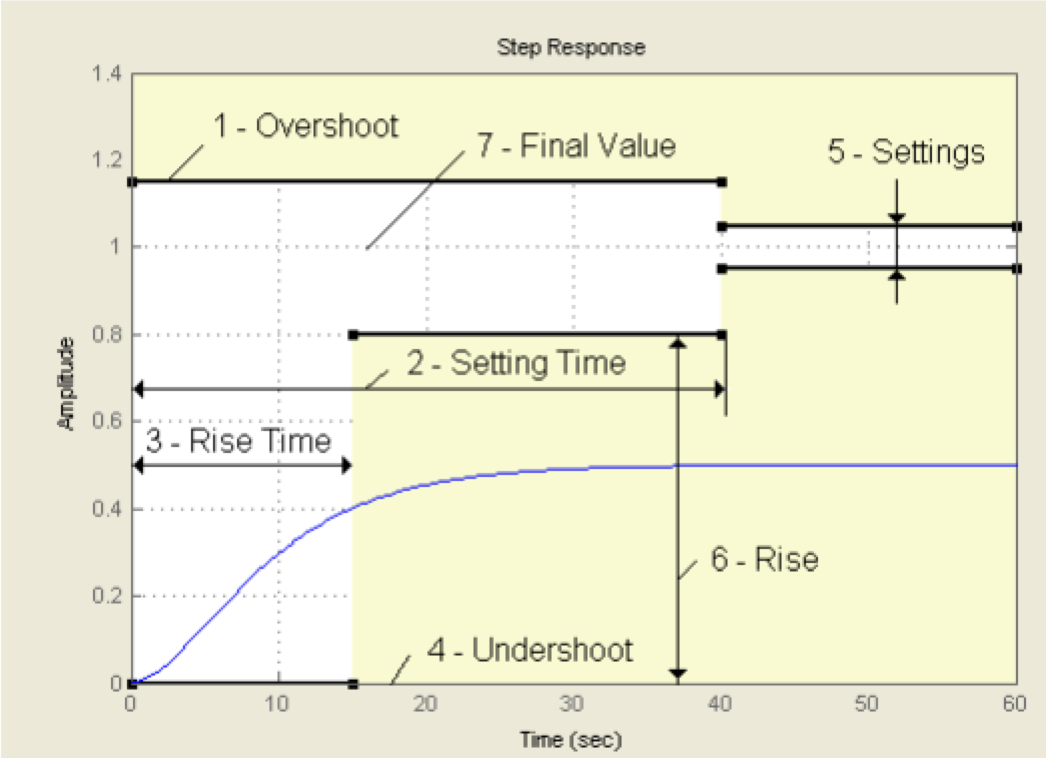


Рис. 7.7

7.2 Приклади і задачі для самостійного розв'язання

Задача 7.1. Об’єкт управління має передатну функцію:

$$G(s) = \frac{0{,}023}{(s+0{,}1)(s+10{,}2)},$$

Спроектувати ПІ-регулятор для таких вимог до якості:

- похибка усталеного стану дорівнює нулю;
- час регулювання <4 с;
- максимальне перерегулювання <20%.

Для цього запускаємо наступний скрипт:

```
>> s=tf('s');
>> G=0.023/(s+0.1)/(s+10.2);
>> sisotool(G)
```

У вікні «SISO Design for SISO Design Task» бачимо графіки, показані на рис. 7.8.

Клацаємо правою кнопкою миші на графіку «Root Locus Editor for Open Loop 1{OL1)» вибираємо Design requirement–> New. У вікні, що відкривається, вибираємо Settling time та вводимо значення 4 в поле Settling time. Потім повторюємо операцію та додаємо нові Design requirement. У вікні, що відкрилося, вибираємо Percent overshoot та вводимо значення 20.

У вікні «Control and Estimation Tools Manager» відкриваємо вкладку Compensator Editor. У полі під заголовком «Dynamics» клацаємо правою кнопкою миші і вибираємо Add Pole/Zero–> Real Zero. Аналогічно виконуємо Add Pole/Zero–>Integrator.

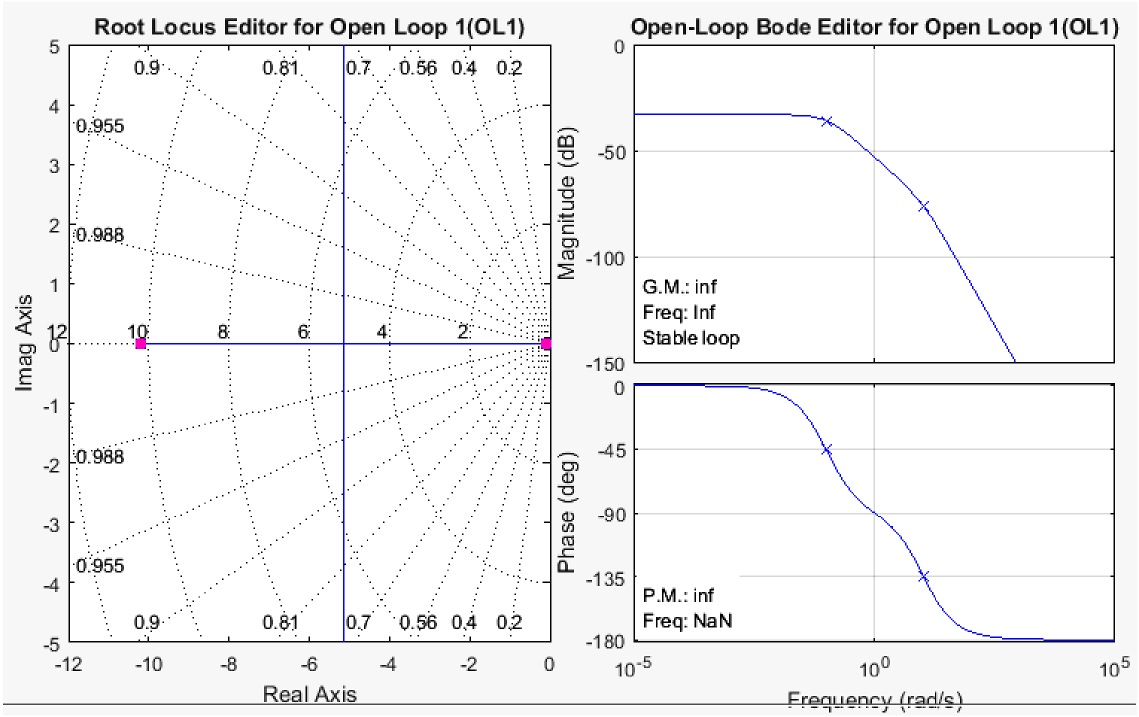


Рис. 7.8

При проектуванні регулятора можна контролювати поточну реакцію системи на стрибок вхідного сигналу і провести порівняння з прийнятими вимогами до якості. Для цього у вікні «Control and Estimation Tools Manager» відкриваємо вкладку Analysis Plots. У рядку Plot Type вибираємо для Plot 1 варіант Step. У таблиці Contents of Plots у рядку «Closed Loop *r* to *y*» відзначаємо «пташкою», що треба будувати графік Plot l. Тиснемо кнопку Show Analysis Plot.

Далі слід використовувати графік «Root Locus Editor for Open Loop 1{OL1)», доступний у вікні «SISO Design for SISO Designing Task». За допомогою миші зміщуємо полюси системи, щоб відповідали вимогам якості. Крім того, діаграма показує зони, за межами яких повинні розташовуватися полюси системи, щоб відповідати вимогам якості. Ці зони визначаються автоматично на основі визначених вимог до якості.

Таким чином отримуємо бажаний вигляд перехідного процесу (рис. 7.9)

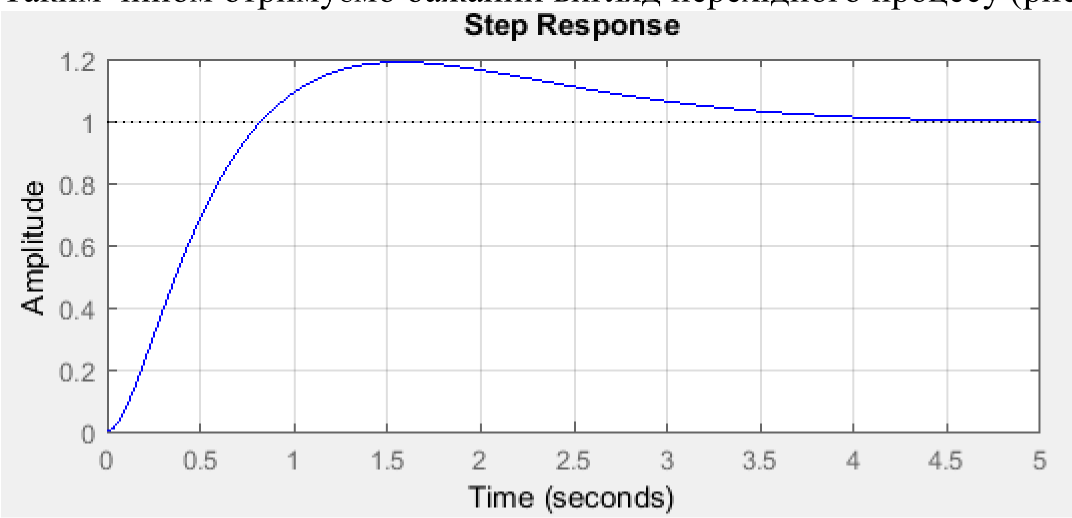


Рис. 7.9

Передатну функцію синтезованого регулятора читаємо на вклвдці Compensator Editor:

$$C = 697.39\frac{1+1.3s}{s}.$$

Задача 7.2. Об’єкт управління має передатну функцію:

$$G(s) = \frac{K_{об}}{T_{2об}s^2 + T_{1об}s + 1}e^{-\tau s},$$

де $K_{об}$=2,5, $T_{2об}$=8,4, $T_{2об}$=2,6, τ=0,5.

Спроектувати ПІД-регулятор і дослідити систему з цим регулятором, а саме:

а) побудувати перехідні процеси;

б) побудувати вагові характеристики.

Розв'язування

1. Створюємо LTI-модель об’єкта управління:

```
>> Plant=tf(2.5,[8.4 2.6 1] ,'ioDelay', 0.5)
Plant =
                       2.5
  exp(-0.5*s) * -------------------
                8.4 s^2 + 2.6 s + 1
Continuous-time transfer function.
```

2. Запускаємо SISO Design Tool

```
>>controlSystemDesigner
```

У вікні «Control and Estimation Tools Manager» тиснемо кнопку **System Data.** У вікні «System Data» тиснемо кнопку **Browse**. У вікні «Model import» виділяємо рядок, що містить модель Plant і тиснемо кнопки **Import** і **Close.** У вікні «System Data» у рядку G з’являється запис <Plant>. Тиснемо кнопку **ОК**.

3. У вікні «Control and Estimation Tools Manager» відкриваємо вкладку Automated Tuning. У віконечку Design method вибираємо PID Tuning. У списку Tuning method вибираємо Robust response time. Як Controller Type вибираємо PID. Відзначаємо «пташкою» Design with first order derivative filter. У списку Design mode вибираємо Automatic (balanced performance and robustness). Тиснемо кнопку **Update Compensator** і спостерігаємо, як під заголовком Compensator з’являється передатна функція регулятора.

Поточний проект було зберігаємо в папці Design History як вузол Design натисненням кнопки Store Design.

4. Відкриваємо вкладку Analysis Plots. У рядку Plot Type вибираємо для Plot 1 варіант Step, а для Plot 2 – Impulse. У таблиці Contents of Plots у рядку «Closed Loop *r* to *y*» відзначаємо «пташками», що треба будувати графіки Plot l і Plot 2. Тиснемо кнопку Show Analysis Plot і отримуємо:

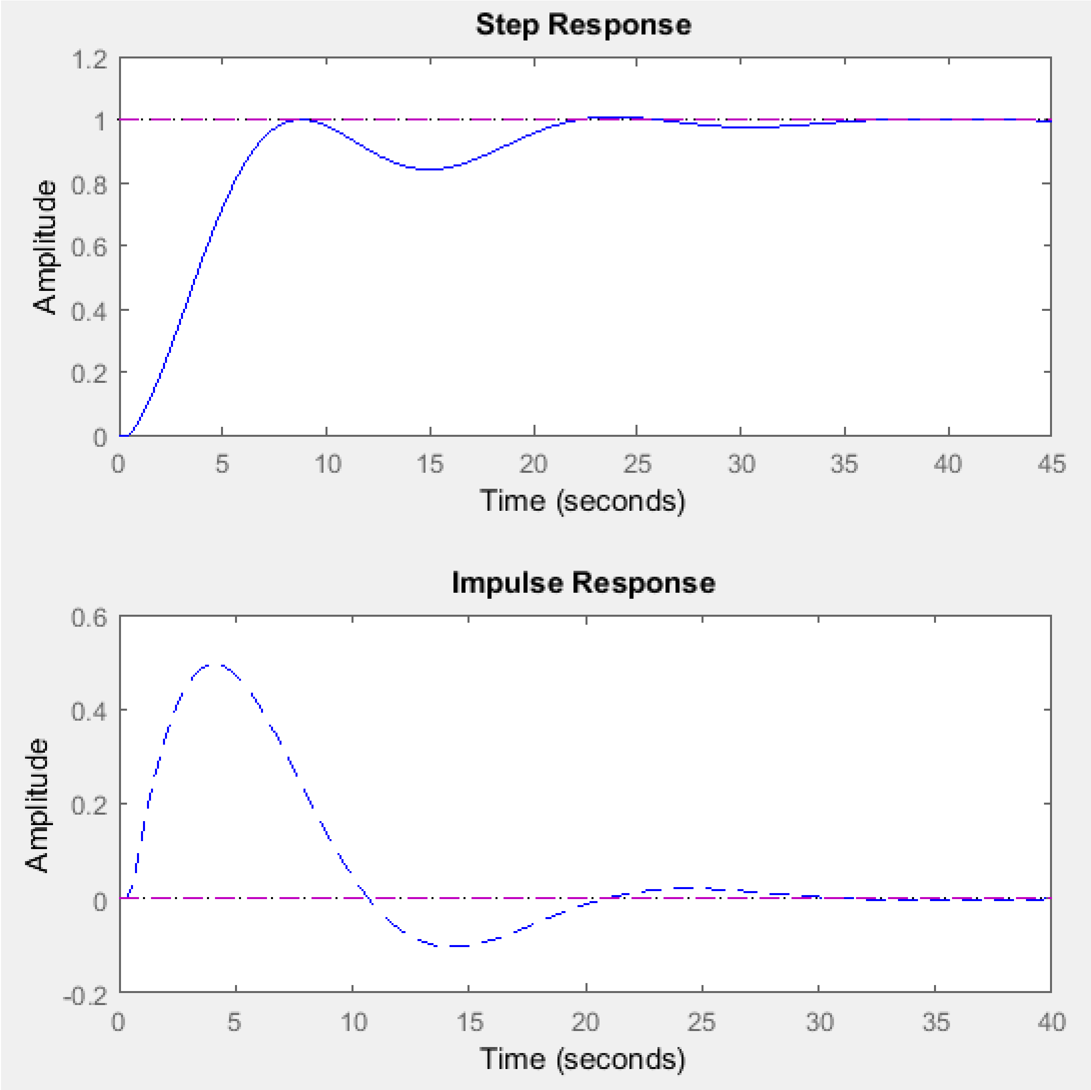


Рис. 7.10

Задача 7.3. Для умов задачі 7.2 побудувати діаграми Боде і визначити запаси по амплітуді та фазі.

Розв'язування

1. Виконуємо п.п. 1–3 розв'язання задачі 7.3.

2. Відкриваємо вкладку Analysis Plots. У рядку Plot Type вибираємо для Plot 1 варіант Bode. У таблиці Contents of Plots у рядку «Closed Loop *r* to *y*» відзначаємо «пташкою», що треба будувати графік Plot l. Тиснемо кнопку Show Analysis Plot

3. Клацаемо по графіку правою клавішею миші, у контексному меню вибираємо *Characteristics* та *Minimum Stability Margins*. Клацаємо мишею по кружечку на графіку амплітуд, читаємо значення запасу стійкості по амплітуді Gain Margin (dB): 27.7. Аналогічно на графіку фаз читаємо запас по фазі Phase Margin (deg): 80.7.

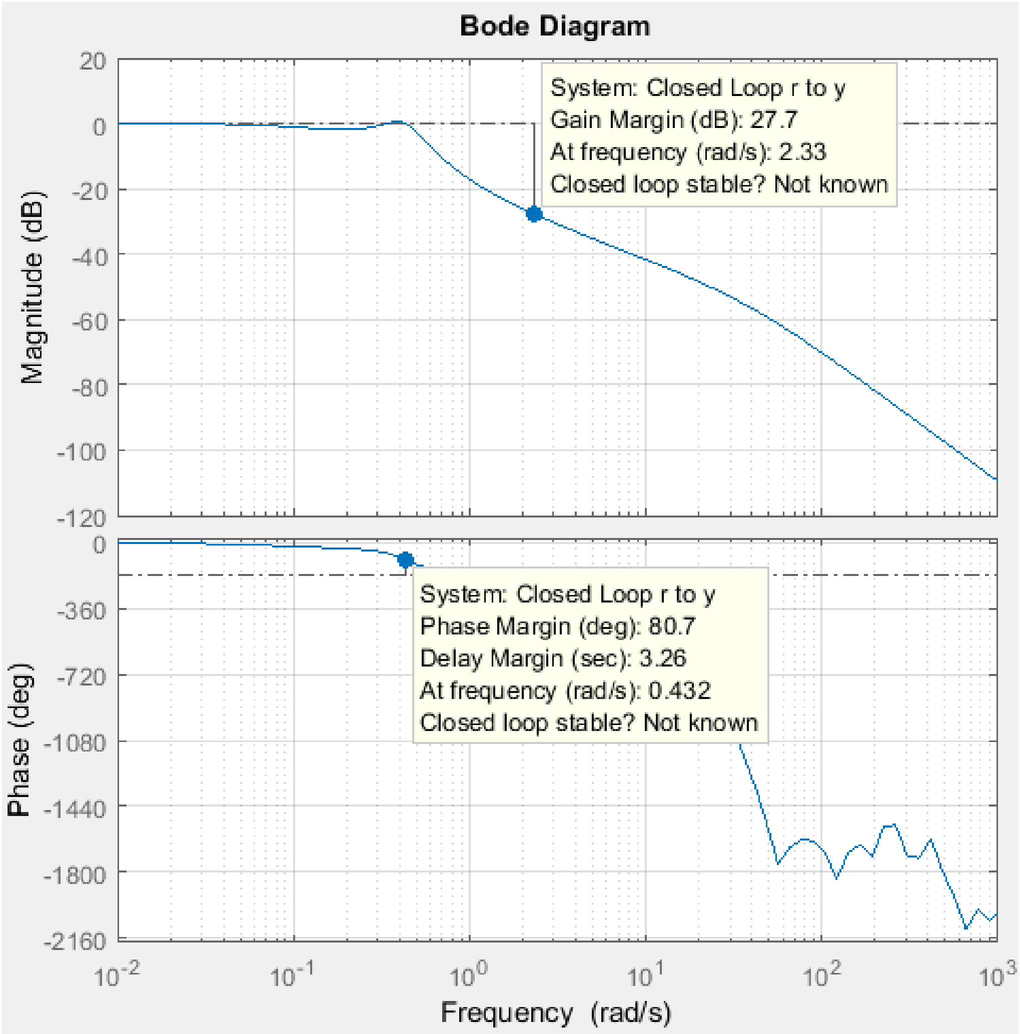


Рис. 7.11

Задача 7.4. Виконайте операції задач 7.2 та 7.3 для варіанту з табл. 7.1.

Таблиця 7.1 – Вихідні дані для задачі 7.4

№ варіанту	**Параметри об’єкта управління**			
	$K_{об}$	**$T_{1об}$, хв**	**$T_{2об}$, хв**	**$\tau_{об}$, хв**
1	1,1	23	7	2
2	52	55	17	8
3	35	21	7	3
4	6	62	18	9
5	2,7	23	10	5
6	0,75	14	5	1,2
7	0,8	3,5	1,1	0,1
8	2,4	17	6,3	0,5
9	0,12	19,5	4,7	2,2
10	0,9	2,4	0,8	0,1
11	1,4	27	7	2
12	48	52	17	4
13	30	24	7	4
14	7,8	58	18	5
15	3,6	25	10	7

16	0,6	12	5	1,5
17	0,9	3,8	1,1	0,2
18	2,7	21	6,3	0,4
19	0,21	16	4,7	2,8
20	0,72	2,1	0,8	0,3
21	1,5	24	7	3
22	50	53	17	4
23	36	24	7	5
24	5,8	55	18	5
25	2,5	22	10	2

8 МОДАЛЬНЕ УПРАВЛІННЯ

8.1 Короткі теоретичні положення

8.1.1 Принципи модального управління

Розглянемо одновимірну динамічну систему (SISO – Single Input Single Output), модель якої у просторі станів має вигляд:

$$\begin{cases} \frac{dX(t)}{dt} = AX(t) + Bu(t), \\ y(t) = CX(t) + Du(t), \end{cases} \quad (8.1)$$

де $\mathbf{X}(t)$ – вектор-стовпчик стану розмірності $[n\times1]$; $\mathbf{A}$ – матриця коефіцієнтів об’єкта $[n\times n]$; $\mathbf{B}$ – матриця входів $[n\times1]$; $u(t)$ – сигнал управління; $\mathbf{Y}$ – вектор виходів $[k\times1]$; $\mathbf{C}$ – матриця виходів $[1\times n]$; $\mathbf{D}$ – матриця обходу, тобто впливу входу безпосередньо на вихід системи $[n\times1]$. Звичайно $\mathbf{D} = 0$.

Система, описувана матрицями **A** і **B**, є керованою,якщо вона може бути переведена з довільного початкового стану x_0, у момент часу t_0, у довільний кінцевий стан x_1, у момент часу t_1, за обмежений проміжок часу $[t_0,\ t_1]$, за допомогою обмеженого управління $u(t)$, визначеного на інтервалі $t_0 < t < t_1$.

Для SISO-системи з одним входом і одним виходом вводиться поняття матриці керованості (розміром n×n):

$$\mathbf{Q} = [\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A}^2\mathbf{B} \quad \dots \quad \mathbf{A}^{n-1}\mathbf{B}]. \quad (8.2)$$

Система (8.1) повністю керована тоді і тільки тоді, коли ранг матриці керованості (8.2) дорівнює порядку системи: $\mathrm{rank}(\mathbf{Q}) = n$.

Нагадаємо, що під рангом матриці мається на увазі найвищий з порядків відмінного від нуля мінору цієї матриці. Ранг матриці дорівнює найбільшому числу лінійно незалежних рядків.

Модальне управління припускає формування таких зворотних зв’язків, при яких забезпечується задане розташування полюсів замкнутої системи. Модою називається та складова рішення диференціального рівняння, що є відповідною одному конкретному полюсу.

Розташування полюсів в основному визначає характер перехідного процесу в системі. Звичайно розглядаються такі кореневі оцінки якості перехідного процесу, як час перехідного процесу, степінь стійкості, коливальність і перерегулювання. Для оцінки швидкодії системи використовується поняття ступеня стійкості η, під яким розуміється абсолютне значення дійсної частини найближчого до уявної осі кореня (тому що корені, що мають якнайменшу по модулю дійсну частину, дають в перехідному процесі поволі самозатухаючу складову). Час перехідного процесу $t_п$ можна приблизно оцінити по формулі

$$t_п \approx \frac{3}{\eta}. \tag{8.3}$$

Запас стійкості системи оцінюється коливальністю. Система має схильність до коливань, якщо характеристичне рівняння містить комплексні корені $\eta_{1,2} = -\alpha \pm j\beta$. Коливальність оцінюється формулою:

$$\mu = \frac{\beta}{\alpha}. \tag{8.4}$$

По значенню коливальності можна оцінити перерегулювання:

$$\delta \le e^{-\frac{\pi}{\mu}} \cdot 100\%. \tag{8.5}$$

Охопимо систему (8.1) негативним зворотним зв’язком (рис. 8.1). У цій системі $g(t)$ – задаючий вплив, управління по стану описується виразом

$$u(t) = -\mathbf{KX}(t), \tag{8.6}$$

де **K** – вектор коефіцієнтів зворотного зв'язку.

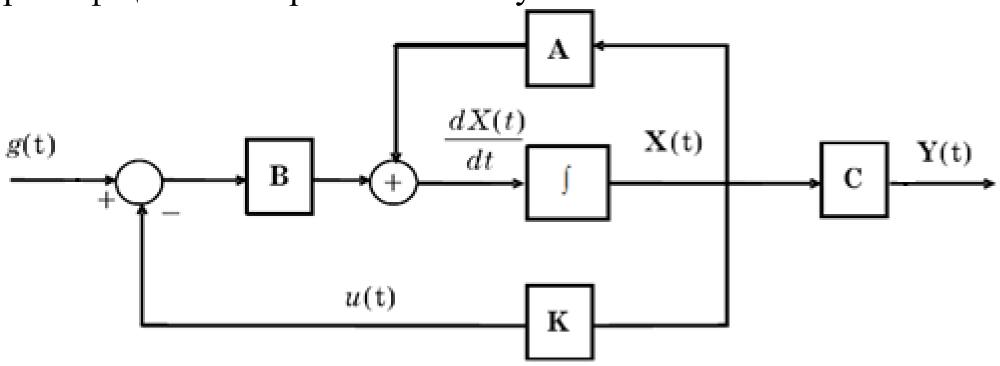


Рис. 8.1 – Структура системи зі зворотним зв’язком

Таким чином, система, замкнута регулятором, приводиться до наступного вигляду:

$$\frac{dX(t)}{dt} = (A - BK)X(t). \tag{6.7}$$

Основна теорема модального управління свідчить, що якщо лінійна динамічна система (8.1) є керованою, то лінійний зворотний зв'язок може бути вибраний таким чином, що матриця (**A–BK**) забезпечить бажане розташування коренів.

Аккерманом була запропонована формула, що дозволяє з допомогою перетворення подібності перевести модель довільної структури в канонічну форму керованості, визначити шукані коефіцієнти K, а потім перерахувати отримане рішення стосовно початкової структури. Формула Аккермана має вигляд [6]

$$K=[0 \;\; 0 \;\; \ldots \;\; 0 \;\; 1]\cdot\left[B \;\; AB \;\; A^2B \;\; \ldots \;\; A^{n-1}B\right]^{-1}\times$$
$$\times\left[A^n \;\; \beta_{n-1}A^{n-1} \;\; \ldots \;\; \beta_1 A \;\; \beta_0 I\right], \qquad (8.8)$$

де β_i – коефіцієнти характеристичного полінома матриці (**A– BK**). Таким чином, задача модального синтезу зводиться до вибору бажаних коренів характеристичного полінома замкнутої системи, при яких забезпечуються задані параметри перехідного процесу, після чого відповідно до стандартного алгоритму розраховуються коефіцієнти зворотних зв'язків по стану.

Матриця керованості може бути побудована за допомогою функції **ctrb**, яка викликається однією з команд:

```
>> Q = ctrb(A, B)
>> Q = ctrb(sys)
>> Q = ctrb(sys.A, sys.B)
```

В пакеті MATLAB є функція **acker**, за допомогою якої можна забезпечити бажане розташування полюсів одновимірної лінійної системи (відповідно до формули Аккермана):

```
>> k = acker(A,B,P),
```

де **A** и **B** – матриці системи; **P** – вектор, що задає бажане розміщення полюсів системи

Для багатовимірних систем в пакеті MATLAB є функція place (її можна використовувати також і для одновимірних систем). Функція

```
>>K=place(A,B,P)
```

розраховує матрицю коефіцієнтів зворотних зв'язків **K**, яка забезпечує бажане розташування полюсів системи. Довжина вектора **P** має дорівнювати числу рядків матриці **A**.

Слід зауважити, що метод модального управління не гарантує рівність сталої помилки нулю. Для забезпечення рівності задаючої дії і вихідного сигналу системи в сталому режимі вводиться коефіцієнт масштабуванняk_0. На цей коефіцієнт повинна множитися вхідна дія.

8.1.2 Побудова спостерігачів

Метод модального управління припускає, що усі компоненти вектора стану X можуть бути виміряні. Проте на практиці деякі компоненти можуть бути невідомі по одній з двох причин :

- вимірювальних приладів може бути недостатньо;
- деякі компоненти вектора X можуть не мати фізичного сенсу.

Проте якщо система є спостережуваною, то усі компоненти вектора X можуть бути відновлені за спостереженнями вектора Y.

Система, що показана на рис. 6.1, є спостережуваною тоді і тільки тоді, коли існує кінцевий час T такий, що початковий стан $x(0)$ може бути визначений в результаті спостереження вихідний змінної $y(t)$ при заданому управлінні $u(t)$. Спостережуваність системи описується умовою $\operatorname{rank}(\mathbf{N}) = n$, де **N** – матриця спостережуванностірозміром $n \times n$:

$$N = [C^T \;\; A^T C^T \;\; (A^T)^2 C^T \;\; \ldots \;\; (A^T)^{n-1} C^T]. \qquad (8.9)$$

Для системи з одним входом і одним виходом система спостережувана, якщо детермінант цієї матриці відмінний від нуля.

Для того, щоб визначити усі компоненти вектора стану об’єкта, можна використовувати його модель

$$\frac{d\hat{\mathbf{X}}(t)}{dt} = \mathbf{A}\hat{\mathbf{X}}(t) + \mathbf{BU}(t), \tag{8.10}$$

де $\widehat{\mathrm{X}}(t)$ − оцінка стану об’єкта.

Якщо початковий стан об’єкта і моделі співпадають і модель адекватна об’єкта, то можна вважати у будь-який момент часу, що $\widehat{\mathrm{X}}(t) = \mathrm{X}(t)$.

Проте практично добитися повної адекватності об’єкта і моделі неможливо, неможливою може бути і повна рівність початкових умов. Тому на практиці можна розраховувати лише на виконання умови

$$\lim_{t\to\infty} \hat{\mathbf{X}}(t) = \mathbf{X}(t) \tag{8.11}$$

Подібну властивість мають так звані асимптотичні спостерігаючі пристрої.Асимптотичний спостерігаючий пристрій використовує зворотний зв'язок за помилкою для відновлення вектора стану. Загальний вигляд системи управління із спостерігачем показаний на рис. 8.2.

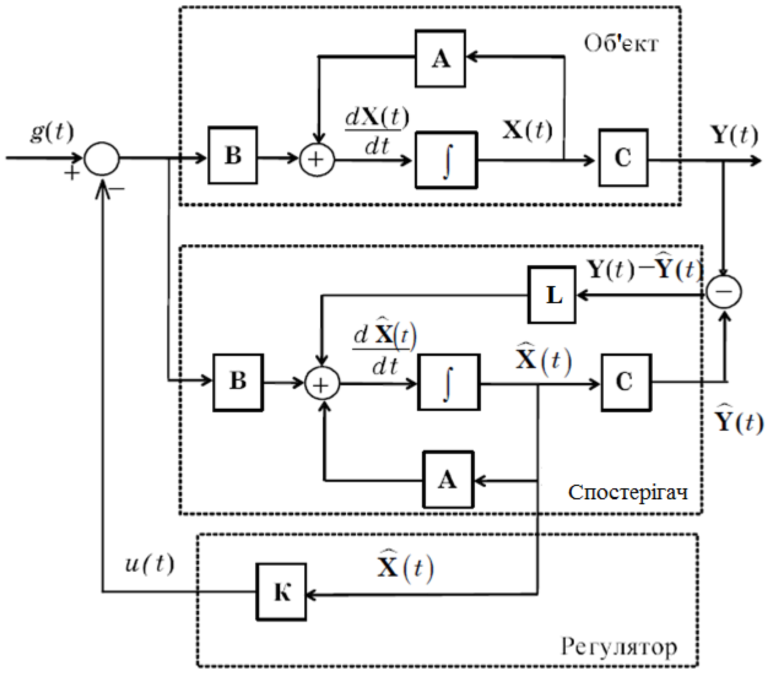


Рис. 8.2 – Структура системи управління із спостерігачем

Робота спостерігаючого пристрою описується рівнянням

$$\frac{d\hat{\mathbf{X}}(t)}{dt} = \mathbf{A}\hat{\mathbf{X}}(t) + \mathbf{BU}(t) + \mathbf{L}(\mathbf{Y} - \mathbf{C}\hat{\mathbf{X}}(t)), \tag{8.12}$$

де **L**– матриця параметрів спостерігаючого пристрою.

Параметри спостерігача і параметри регулювальника можуть розраховуватися незалежно.

Зрозуміло, що процеси в спостерігачі повинні протікати швидше, ніж перехідний процес в системі. Емпірично встановлено [7], що спостерігач повинен мати швидкодію, що в 2–4 рази перевищує швидкодію системи.

Матриця спостережуванності може бути побудована за допомогою функції **obsv**, яка також може викликатися в одному з варіантів :

```
>>N = obsv(A, C)
>> N = obsv(sys)
>> N = obsv(sys.A,sys.C)
```

Описана вище функція **acker** може бути застосована і для розрахунку коефіцієнтів зворотних зв'язків спостерігача одновимірної системи. Для цього потрібно транспонувати матрицю **A** і замінити **B** на $\mathbf{C}^T$:

```
>>L=acker(A^T,C^T,P),
```

де **P** – вектор бажаних полюсів спостерігача.

Для багатовимірних (і одновимірних) систем це ж завдання можна вирішити за допомогою функції **place:**

```
>>L=place(A^T,C^T,P)
```

В MATLAB існують спеціальні функції для формування спостерігача. Функція **estim**формує спостерігаючий пристрій у вигляді ss-об'єкта для оцінювання вектора змінних стану моделі об'єкта управління **sys** і для заданої матриці коефіцієнтів зворотних зв'язків спостерігача **L**:

```
>>est=estim(sys,L)
```

Функція **reg** формує регулятор для заданої в просторі станів моделі об’єкта управління **sys**, матриці коефіцієнтів зворотних зв'язків по змінних стану **K** і матриці коефіцієнтів зворотних зв'язків спостерігача **L**:

```
>>rsys=reg(sys,K,L)
```

8.2 Приклади і задачі для самостійного розв'язання

Задача 8.1. Система описується матрицями

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -8 & -9 \end{bmatrix}; \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}; \mathbf{C} = \begin{bmatrix} 8 & 0 & 0 \end{bmatrix}; \mathbf{D} = 0.$$

Обґрунтувати можливість модального управління за допомогою критерію керованості.

Розв'язування

Маємо систему третього порядку: n=3.Визначаємо матриці системи у робочому просторі MATLAB за допомогою команд

```
>>A=[0 1 0; 0 0 1; 0 -8 -9]; B=[0; 0; 1];
```

Будуємо матрицю керованості:

```
>> Q = ctrb(A, B);
```

Визначаємо ранг матриці Q:

```
>>rank(Q)
ans =
3
```

Оскільки rank(Q) =n, система є керованою і модальне управління можливе.

Задача 8.2. Розрахувати значення коефіцієнтів зворотних зв'язків для об’єкта управління з задачі 8.1. Бажані полюси задати вектором **`P=[-1,-2,-3]`**.

Розв'язування

Розрахувати значення коефіцієнтів зворотних зв'язків можна за допомогою команд

```
>>A=[0 1 0; 0 0 1; 0 -8 -9]; B=[0; 0; 1];
>>P=[-1,-2,-3];
>>K=acker(A,B,P)
K =
     6     3    -3
```

Таким чином, рівняння управління має бути сформоване у вигляді:

$$u(t) = -[K][X(t)] = [6 \quad 3 \quad -3]\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} = 6x_1(t) + 3x_2(t) - 3x_3(t).$$

Задача 8.3. Для об’єкта управління з задачі 8.1 з урахуванням результатів задачі 8.2 визначити передатну функцію системи модального управління і коефіцієнт масштабування та побудувати графік перехідного процесу.

Розв'язування

Після розв'язання задачі 8.2 вводимо матрицю С:

```
>>C=[8 0 0]; D=0;
```

Матриця А коефіцієнтів замкненої системи:

```
>> As=A-B*K;
```

Модель замкненої системи:

```
>>sys=ss(As,B,C,D);
```

ПФ замкненої системи:

```
>>sys=tf(sys)
Transfer function:
          8
----------------------
s^3 + 6 s^2 + 11 s + 6
```

Розрахунок перехідного процесу

```
>> [out,t]=step(sys);
```

Для обчислення коефіцієнта масштабування k_0

```
>> k0=1/out(length(out))
k0 =
```

```
    0.7530
```

Будуємо графік

```
>>plot(t, k0*out)
```

Задача 8.4. Для об’єкта управління, що описується матрицями **A**, **B**, **C** (див. Додаток А) визначити можливість модального управління і, у випадку позитивного висновку, синтезувати систему модального управління:

а) розрахувати значення коефіцієнтів зворотних зв'язків системи;

б)визначити передатну функцію системи модального управління;

в) розрахувати значення коефіцієнта масштабування, на який повинна множитися вхідна дія для забезпечення рівності задаючої дії і вихідного сигналу системи в сталому режимі;

г) побудувати графік перехідного процесу.

Задача 8.5. Визначити спостережуванність об’єкта управління, що описується матрицями задачі 8.1

Розв'язування

Виконуємо наступні команди.

```
>>A=[0 1 0; 0 0 1; 0 -8 -9]; B=[0; 0; 1];
>>C=[8 0 0]; D=0;
>> N = obsv(A, C);
>>det(N)
   512
```

Оскільки детермінант матриці спостережуванності відмінний від нуля, то система спостережувана.

Задача 8.6. Сформувати модель спостерігача для об’єкта управління задачі 8.1.

Розв'язування

Виконуємо наступні команди:

```
>>A=[0 1 0; 0 0 1; 0 -8 -9]; B=[0; 0; 1];
>>C=[8 0 0]; D=0;
Бажане розміщення полюсів спостерігача:
>>P=[-1,-2,-3];
```

Будуємо матрицю коефіцієнтів зворотних зв'язків спостерігача **L**:

```
>>Lt=acker(A', C', Pe); L=Lt'
L =
   -0.3750
    3.7500
  -30.0000
```

Матриця А коефіцієнтів замкненої системи:

```
>> Pp = [-10 -10 -20];
>> K = acker(A, B, Pp);
>> As=A-B*K;
```
Модель замкненої системи:
```
>>sys=ss(As,B,C,D);
```
Формуємо спостерігаючий пристрій у вигляді ss-об'єкта:
```
>>rsys=reg(sys,K,L)
a =
           x1_e   x2_e   x3_e
   x1_e     -51      1      0
   x2_e    -733      0      1
   x3_e   -4995   -900    -60

b =
             y1
   x1_e   6.375
   x2_e   91.63
   x3_e   124.4

c =
         x1_e   x2_e   x3_e
   u1   -4000   -892    -51

d =
        y1
   u1    0
```
Передатна функція спостерігача:
```
>> We=tf(rsys)
Transfer function from input "y1" to output "u1":
-1.136e005 s^2 - 1.405e006 s - 4e006
------------------------------------
 s^3 + 162 s^2 + 8186 s + 1.817e005
```
Будуємо перехідний процес спостерігача:
```
>> step(We)
```

Задача 8.7. Для об’єкта управління, що описується матрицями **A**, **B**, **C** (див. Додаток А) визначити спостережуванність об’єкта і, у випадку позитивного висновку, розрахувати значення коефіцієнтів зворотних зв'язків спостерігача, сформувати модель спостерігача и побудувати графік перехідного процесу.

ЛІТЕРАТУРА

1. Елизаров, И.А. Моделирование систем : учебное пособие / И. А. Елизаров, Ю.Ф. Мартемьянов, А.Г. Схиртладзе, А.А. Третьяков. – Тамбов : Изд-во ФГБОУ ВПО «ТГТУ», 2011. – 96 с.

2. Поляков, К.Ю. Лабораторные работы «Исследование систем автоматического управления в MATLAB» [Електронний ресурс] / К.Ю. Поляков. – Режим доступу : http://kpolyakov.spb.ru/uni/labs.htm. – Заголовок з екрану.

3. Ануфриев, И.Е. MATLAB 7 / И.Е. Ануфриев, А.Б. Смирнов, Е.Н. Смирнова. – СПб. : БХВ-Петербург, 2005. – 1104 с.

4. Медведев, В.С. Control System Toolbox. MATLAB 5 для студентов / В. С. Медведев, В. Г. Потемкин. – М: ДИАЛОГ-МИФИ, 1999. – 287с.

5. Теория автоматического управления. Методические указания к выполнению лабораторных работ № 1–9 / М.В. Бураков, Т.Г. Полякова, А.В. Подзорова. – СПб. : ГОУ ВПО СПбГУАП, 2006. – 62 с.

6. Дорф, Р.Современные системы управления / Р.Дорф, Р. Бишоп. М : Лаборатория базовых знаний, 2002. – 832 с.

7. Гудвин, Г.К. Проектирование систем управления / Г.К. Гудвин, С.Ф. Гребе, М.Э.Сальгадо. – М.: Бином, 2004. – 911 с.

8. Лабораторная работа № 1. Модальное управление и наблюдающие устройства. Автоматизация синтеза регуляторов и наблюдателей состояния в среде пакета MATLAB [Електронний ресурс]. – Режим доступу : http://gigabaza.ru/doc/191541.html. – Заголовок з екрану.

9. Методические указания к выполнению курсовой работе по дисциплине «Проектирование автоматизированных систем» для студентов, обучающихся по направлению 220301 «Автоматизация технологических процессов и производств» / Сост.: С.А. Милюшенко, И.В. Лазута. – Омск : Изд-во СибАДИ, 2012. – 54 с.

10. Buciakowski, M. Wstep do projektowania regulatorów w SISOTOOL® [Електронний ресурс] / Mariusz Buciakowski // Uniwersytet Zielonogórski. – Режим доступу : http://staff.uz.zgora.pl/mbuciakowski/mb/Materialy/sisotool_(MB).pdf. – Заголовок з екрану.

ДОДАТОК А

Таблиця А.1 – Вихідні дані для задач

№ варіанта	A	B	C
1	$\begin{bmatrix} -39 & -71 & -23 \\ 23 & 40 & 13 \\ -6,6 & -10,7 & -3,3 \end{bmatrix}$	$\begin{bmatrix} 0.228 \\ -0.127 \\ 0.038 \end{bmatrix}$	$[2800 \quad 84000 \quad 112000]$
2	$\begin{bmatrix} -44 & -78 & -25 \\ 26 & 43,7 & 14 \\ -7,5 & -11,8 & -3,7 \end{bmatrix}$	$\begin{bmatrix} 0.228 \\ -0.127 \\ 0.038 \end{bmatrix}$	$[2800 \quad 84000 \quad 112000]$
3	$\begin{bmatrix} -54 & -98 & -32 \\ 31 & 55 & 17,8 \\ -9 & -15 & -4,8 \end{bmatrix}$	$\begin{bmatrix} 0.228 \\ -0.127 \\ 0.038 \end{bmatrix}$	$[35000 \quad 105000 \quad 140000]$
4	$\begin{bmatrix} -60,8 & -107 & -34,7 \\ 35 & 60 & 19 \\ -10 & -16,7 & -5,3 \end{bmatrix}$	$\begin{bmatrix} 0.228 \\ -0.127 \\ 0.038 \end{bmatrix}$	$[35000 \quad 105000 \quad 140000]$
5	$\begin{bmatrix} -72,4 & -130,6 & -42,5 \\ 41,5 & 73 & 23,6 \\ -12 & -20,6 & -6,6 \end{bmatrix}$	$\begin{bmatrix} 0.228 \\ -0.127 \\ 0.038 \end{bmatrix}$	$[42000 \quad 126000 \quad 168000]$
6	$\begin{bmatrix} -80,4 & -142,5 & -46 \\ 45,9 & 79,6 & 25,6 \\ -13,6 & -22,6 & -7,2 \end{bmatrix}$	$\begin{bmatrix} 0.228 \\ -0.127 \\ 0.038 \end{bmatrix}$	$[42000 \quad 126000 \quad 168000]$
7	$\begin{bmatrix} -100,5 & -174,4 & -56 \\ 57 & 97 & 31 \\ -17 & -28 & -8,9 \end{bmatrix}$	$\begin{bmatrix} 0.228 \\ -0.127 \\ 0.038 \end{bmatrix}$	$[42000 \quad 126000 \quad 168000]$
8	$\begin{bmatrix} -13,2 & -16,7 & -4,7 \\ 8,6 & 9,5 & 2,6 \\ -2,4 & -1,6 & -0,3 \end{bmatrix}$	$\begin{bmatrix} 0.228 \\ -0.127 \\ 0.038 \end{bmatrix}$	$[3500 \quad 10500 \quad 14000]$
9	$\begin{bmatrix} -11,4 & -16,7 & -5,13 \\ 7,5 & 9,7 & 2,9 \\ -2 & -1,6 & -0,36 \end{bmatrix}$	$\begin{bmatrix} 0.228 \\ -0.127 \\ 0.038 \end{bmatrix}$	$[7000 \quad 21000 \quad 2800]$
10	$\begin{bmatrix} -16,2 & -26,3 & -8,3 \\ 10,2 & 15 & 4,6 \\ -2,9 & -3,2 & -0,9 \end{bmatrix}$	$\begin{bmatrix} 0.228 \\ -0.127 \\ 0.038 \end{bmatrix}$	$[14000 \quad 42000 \quad 56000]$
11	$\begin{bmatrix} -22,3 & -38,6 & -12,4 \\ 13,6 & 21,9 & 6,9 \\ -3,9 & -5,3 & -1,6 \end{bmatrix}$	$\begin{bmatrix} 0.228 \\ -0.127 \\ 0.038 \end{bmatrix}$	$[21000 \quad 63000 \quad 84000]$
12	$\begin{bmatrix} -34 & -54 & -17 \\ 20 & 30,5 & 9,4 \\ -5,9 & -7,9 & -2,3 \end{bmatrix}$	$\begin{bmatrix} 0.228 \\ -0.127 \\ 0.038 \end{bmatrix}$	$[21000 \quad 63000 \quad 84000]$

Продовження таблиці А.1

№ варіанта	A	B	C
13	$\begin{bmatrix} -34 & -54 & -17 \\ 20 & 30{,}5 & 9{,}4 \\ -5{,}9 & -7{,}9 & -2{,}3 \end{bmatrix}$	$\begin{bmatrix} 0{,}228 \\ -0{,}127 \\ 0{,}038 \end{bmatrix}$	$[2800 \quad 84000 \quad 112000]$
14	$\begin{bmatrix} -39 & -71 & -23 \\ 23 & 40 & 13 \\ -6{,}6 & -10{,}7 & -3{,}3 \end{bmatrix}$	$\begin{bmatrix} 0{,}228 \\ -0{,}127 \\ 0{,}038 \end{bmatrix}$	$[2800 \quad 84000 \quad 112000]$
15	$\begin{bmatrix} -44 & -78 & -25 \\ 26 & 43{,}7 & 14 \\ -7{,}5 & -11{,}8 & -3{,}7 \end{bmatrix}$	$\begin{bmatrix} 0{,}228 \\ -0{,}127 \\ 0{,}038 \end{bmatrix}$	$[35000 \quad 105000 \quad 140000]$
16	$\begin{bmatrix} -54 & -98 & -32 \\ 31 & 55 & 17{,}8 \\ -9 & -15 & -4{,}8 \end{bmatrix}$	$\begin{bmatrix} 0{,}228 \\ -0{,}127 \\ 0{,}038 \end{bmatrix}$	$[35000 \quad 105000 \quad 140000]$
17	$\begin{bmatrix} -60{,}8 & -107 & -34{,}7 \\ 35 & 60 & 19 \\ -10 & -16{,}7 & -5{,}3 \end{bmatrix}$	$\begin{bmatrix} 0{,}228 \\ -0{,}127 \\ 0{,}038 \end{bmatrix}$	$[42000 \quad 126000 \quad 168000]$
18	$\begin{bmatrix} -72{,}4 & -130{,}6 & -42{,}5 \\ 41{,}5 & 73 & 23{,}6 \\ -12 & -20{,}6 & -6{,}6 \end{bmatrix}$	$\begin{bmatrix} 0{,}228 \\ -0{,}127 \\ 0{,}038 \end{bmatrix}$	$[42000 \quad 126000 \quad 168000]$
19	$\begin{bmatrix} -80{,}4 & -142{,}5 & -46 \\ 45{,}9 & 79{,}6 & 25{,}6 \\ -13{,}6 & -22{,}6 & -7{,}2 \end{bmatrix}$	$\begin{bmatrix} 0{,}228 \\ -0{,}127 \\ 0{,}038 \end{bmatrix}$	$[42000 \quad 126000 \quad 168000]$
20	$\begin{bmatrix} -100{,}5 & -174{,}4 & -56 \\ 57 & 97 & 31 \\ -17 & -28 & -8{,}9 \end{bmatrix}$	$\begin{bmatrix} 0{,}228 \\ -0{,}127 \\ 0{,}038 \end{bmatrix}$	$[3500 \quad 10500 \quad 14000]$
21	$\begin{bmatrix} -13{,}2 & -16{,}7 & -4{,}7 \\ 8{,}6 & 9{,}5 & 2{,}6 \\ -2{,}4 & -1{,}6 & -0{,}3 \end{bmatrix}$	$\begin{bmatrix} 0{,}228 \\ -0{,}127 \\ 0{,}038 \end{bmatrix}$	$[7000 \quad 21000 \quad 2800]$
22	$\begin{bmatrix} -11{,}4 & -16{,}7 & -5{,}13 \\ 7{,}5 & 9{,}7 & 2{,}9 \\ -2 & -1{,}6 & -0{,}36 \end{bmatrix}$	$\begin{bmatrix} 0{,}228 \\ -0{,}127 \\ 0{,}038 \end{bmatrix}$	$[14000 \quad 42000 \quad 56000]$
23	$\begin{bmatrix} -16{,}2 & -26{,}3 & -8{,}3 \\ 10{,}2 & 15 & 4{,}6 \\ -2{,}9 & -3{,}2 & -0{,}9 \end{bmatrix}$	$\begin{bmatrix} 0{,}228 \\ -0{,}127 \\ 0{,}038 \end{bmatrix}$	$[21000 \quad 63000 \quad 84000]$
24	$\begin{bmatrix} -22{,}3 & -38{,}6 & -12{,}4 \\ 13{,}6 & 21{,}9 & 6{,}9 \\ -3{,}9 & -5{,}3 & -1{,}6 \end{bmatrix}$	$\begin{bmatrix} 0{,}228 \\ -0{,}127 \\ 0{,}038 \end{bmatrix}$	$[21000 \quad 63000 \quad 84000]$