

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

ВИКОРИСТАННЯ ПРОГРАМНИХ ЗАСОБІВ МАТЛАВ ДЛЯ РОЗВ’ЯЗАННЯ ТИПОВИХ ЗАДАЧ АНАЛОГОВОЇ АВТОМАТИЗАЦІЇ

Навчальний посібник
для студентів спеціальностей 141 – «Електроенергетика, електротехніка
та електромеханіка», 173 – «Авіоніка», 151 – «Автоматизація та
комп’ютерно-інтегровані технології (промислова автоматика)» всіх
форм навчання

Запоріжжя • 2019

УДК 004.4 Matlab:621.31(075.8)

Д53

*Рекомендовано до друку вченою радою
Національного університету «Запорізька політехніка»
(протокол № 10 від 03 червня 2019 року).*

Рецензенти:

Тиховод С.М., д.т.н., завідувач кафедри теоретичних основ електро-техніки Національного університету «Запорізька політехніка»;

Орловський І.А., д.т.н., професор Технологічно-природничого університету у м. Бидгощ, Польща;

Поднебенна С.К., к.т.н., доцент кафедри систем автоматизації та електроприводу ДВНЗ «Приазовський державний технічний університет», м. Маріуполь.

Арсеньєва С. І.

Д53

Використання програмних засобів MATLAB для розв'язання типових задач аналогової автоматизації: навчальний посібник для студентів вищих навчальних закладів / С. І. Арсеньєва – Запоріжжя: НУ «Запорізька політехніка», 2019. – 118 с.

ISBN 978-617-529-244-0

Навчальний посібник присвячено отриманню практичних навичок з командами пакету MATLAB, зокрема його розділу Control System Toolbox. На прикладах розв'язання конкретних задач для двомасових систем, літаків, ПІД – регуляторів розглянуті засоби оцінки динамічних характеристик, які дозволяють наочно відстежити вплив зміни окремих параметрів систем. Особлива увага надається векторно – матричному підходу як найбільш відповідному методам, закладеним у основу пакету. Окрім командного підходу, детально розглянуті популярні браузері (ltiviewer, sisotool та інші). Усі програми, наведені у посібнику, можуть бути легко використані шляхом копіювання з його електронної версії у М-файл, бажаного редагування та запуску у вікні Command Window.

Призначено для студентів бакалаврату та магістратури за напрямками навчання «Електроенергетика, електротехніка та електромеханіка», «Авіоніка», «Автоматизація та комп'ютерно-інтегровані технології (промислова автоматика)», для яких теорія автоматичного керування є профільною дисципліною за обсягом навчального навантаження, але в той же час закладає основи професійного підходу до майбутнього фаху.

УДК 004.4 Matlab:621.31(075.8)

ISBN 978-617-529-244-0

© Арсеньєва С.І.,

© Національний університет

«Запорізька політехніка»,

НУ «Запорізька політехніка»,

2019

ЗМІСТ

Вступ	7
1. Основні поняття та принципи роботи пакету MATLAB, необхідні для роботи з підкаталогом CONTROL TOOLBOX	8
1.1 Короткі відомості про пакет MATLAB	8
1.1.1 Інтерфейс середовища MATLAB	8
1.1.2 Введення даних та операції з ними	10
1.1.3 Основні базові команди та системні змінні (clc, help elfun, type, pi, ans)	12
1.1.4 Оформлення результатів обчислень (plot, subplot, figure, grid, axis, title, text)	13
1.1.5 Збереження даних для відновлення роботи	17
2 Деякі загальні положення та моменти при моделюванні аналогових систем керування у MATLAB	18
2.1 Форми опису моделей, перетворення між ними (tf, zpk, ss)	18
2.1.1 Основні форми подання моделей неперервних систем	18
2.1.2 Змінні стану динамічної системи	19
2.1.3 Диференційні рівняння стану	25
2.1.4 Стійкість у просторі станів (det, roots, poly)	28
2.1.5 Завдання передатних функцій та поліномів	29
2.1.6 Введення різних форм моделей	34
2.1.7 Команди для перетворення між неперервними та неперервними та дискретними системами (tf2ss, ss2tf, ss2zp, zp2ss, tf2zp, zp2tf, c2d, d2c, d2d)	36
2.2 Типові ланки та їхні динамічні властивості	38
2.3 Типові вхідні сигнали та отримання реакції системи на їх вплив у MATLAB (step, grid, lsim)	39
2.4 Використання частотних характеристик	43
2.4.1 Команди побудови та аналізу частотних характеристик (bode, margin, nyquist, nicols)	44
2.5 Практичне дослідження динамічних характеристик за допомогою команд пакету	45
2.6 Використання браузерa LTI Viewer для аналізу динамічних характеристик	48
2.7 Завдання з'єднань блоків у MATLAB (series,	53

parallel, feedback, append, connect)	
2.8 Використання команд пакету Control System Toolbox для створення моделі системи	56
3 Приклади розв'язання деяких задач з аналоговою формою сигналу	60
3.1 Дослідження впливу гнучкого зворотного зв'язку на якісні показники САК	60
3.2 Дослідження впливу чистого запізнювання (pade)	62
3.3 Дослідження САК з ПД – регулятором	66
3.3.1 Розрахунок ПД-регулятора методом настроювання на модульний оптимум	68
3.3.2 Програма для аналізу впливу зміни коефіцієнтів КІ, КД та КП	72
3.3.3 Розрахунок ПД регулятора за допомогою браузера PIDTool	74
3.3.4 Розрахунок коефіцієнтів ПД регулятора за допомогою MATLAB/Simulink (версія 7.04)	75
3.4 Розрахунок системи з двома масами (векторно – матрична модель)	82
3.5 Аналіз систем керування креном літака за допомогою MATLAB/Simulink	87
3.5.1 Аналіз системи керування креном гвинтового літака	87
3.5.2 Аналіз системи керування креном реактивного винищувача	90
3.5.3 Визначення області стійкості у випадку нестійкого об'єкту	93
4 Використання кореневого годографа (КГ) та браузера SISO Design Tool задля аналізу та синтезу систем керування	95
4.1 Поняття кореневого годографа та його основні властивості (rlocus, rlocfind, residue)	95
4.2 SISO Design Tool	102
4.2.1 Структура зі зворотним зв'язком	103
4.2.2 Альтернативні структури зворотного зв'язку	104
4.2.3 Налаштування компенсаторів	105
4.2.4 Розташування коренів	105

4.2.5 Використання меню проектування компенсаторів	108
4.2.6 Додавання нулів та полюсів до компенсатора	108
4.2.7 Додавання передфільтра	109
4.3 Дослідження СК збудження за допомогою SISO Design Tool	110
Додаток А. Довідкова таблиця функцій MATLAB	112
Додаток Б. Параметри для зображення кривих у функції plot	116
Перелік посилань	117

ПЕРЕЛІК СКОРОЧЕНЬ

САК - система автоматизованого керування;
ТАК - теорія автоматичного керування;
ПФ - передатна функція;
ЧХ - частотна характеристика;
АЧХ, ФЧХ - амплітудна та фазочастотна характеристики;
ЛЧХ - логарифмічна частотна характеристика;
ЛАЧХ, ЛФЧХ - логарифмічні амплітудна та фазова частотні характеристики;
ЛКМ, ПКМ - ліва або права кнопки миші;
ПД - пропорційно-інтегруючий-диференціюючий (регулятор);
КГ - кореневий годограф;
SISO - single input, single output – одновимірні об'єкти з одним входом та одним виходом;
LTI - об'єкти - лінійні системи з постійними параметрами.

ВСТУП

Метою створення даного посібника є отримання навичок для швидкого перегляду якісних характеристик систем керування, наочного аналізу впливу параметрів системи на ці характеристики з подальшим підбиранням корегуючих елементів, якщо не існує можливості поліпшити якість системи зміною її власних параметрів.

При тому наведений нижче матеріал не є повторенням основних понять теорії автоматичного регулювання [1-2, 5, 6 та інші]. У даному випадку мова йде про використання апарату програмування мовою MATLAB для більш детального, але в той самий час і нескладного та наочного з точки зору зручних у роботі, редагуванні та роздрукуванні графіків залежностей параметрів якості САК від досліджуваних параметрів. У даному підході не розглядається детально моделювання за допомогою Simulink, оскільки таких робіт також досить багато. Як відомо, пакет містить спеціалізований набір програм, розміщених у підкаталозі Control System Toolbox, що використовує матричні структури MATLAB і формує на їх основі спеціальні функції, призначені для задоволення потреб спеціаліста в області автоматичного керування. Згодом, Control System Toolbox - це набір алгоритмів, розміщених в М-файлах, які широко використовують для проектування, аналізу і моделювання системи керування, деякі з яких і будуть розглянуті нижче на прикладах.

1 ОСНОВНІ ПОНЯТТЯ ТА ПРИНЦИПИ РОБОТИ ПАКЕТУ MATLAB, ПОТРІБНІ ДЛЯ РОБОТИ З ПІДКАТАЛОГОМ CONTROL TOOLBOX

1.1 Короткі відомості про пакет MATLAB

Система MATLAB є універсальним засобом для виконання математичних обчислень. Її робота заснована на виконанні операцій з матрицями, що одержало віддзеркалення в назві: **MATLAB** – це скорочення від **Matrix Laboratory** (Матрична Лабораторія).

У галузі автоматичного керування пакет MATLAB використовують для моделювання та оптимізації параметрів динамічних систем та побудови частотних характеристик.

1.1.1 Інтерфейс середовища MATLAB

На рисунку 1.1 зображено вікно робочого середовища MATLAB студентської версії, яка встановлена у навчальних лабораторіях кафедри. Воно включає наступні елементи:

- рядок меню, в якому знаходяться меню **File** (Файл), **Edit** (Правка), **Debug** (Налагодження), **Desktop** (Робочий стіл), **Window** (Вікно), **Help** (Довідка);
- панель інструментів з кнопками, що дозволяють виконати деякі найбільш поширені операції;
- панель **Shortcuts** (Ярлики), де можна створити кнопки для швидкого запуску найбільш часто використовуваних команд або наборів команд;
- командне вікно (**Command Window**), де відображаються команди, що вводяться користувачем, і результати їх виконання, тексти виконуваних програм, а також повідомлення про помилки;
- вікно з вкладками **Workspace** (дозволяє дістати доступ до вмісту робочого простору) і **Current Directory** (служить для установки поточного каталогу);

- вікно **Command History** (призначено для перегляду і ви-
клику раніше введених команд);
- рядок стану, де відображаються повідомлення системи.

Командне вікно системи MATLAB (Command Window) при-
значене для введення чисел, змінних, виразів і команд, а також
здля перегляду результатів обчислень, відображення текстів
виконуваних програм та виведення повідомлень про помилки.

Про те, що система готова до введення чергової команди,
свідчить знак запрошення (>>), що знаходиться в текстовому полі
командного вікна.

Для того, щоб програма MATLAB виконала введену команду
або обчислила заданий вираз, після введення команди або виразу
слід натиснути клавішу <Enter>.

Клавіші <↓> і <↑>, дозволяють відобразити в рядку введення
команди, раніше введені з клавіатури, і вирази з метою їх повто-
рного виконання або редагування. При цьому клавіша <↑> відо-
бражає попередню команду, а клавіша <↓> виконує прокрутку в
інший бік.

Крім того, тієї ж мети можна досягнути звичайним копію-
ванням потрібних даних та впровадженням їх у потрібному міс-
ці.

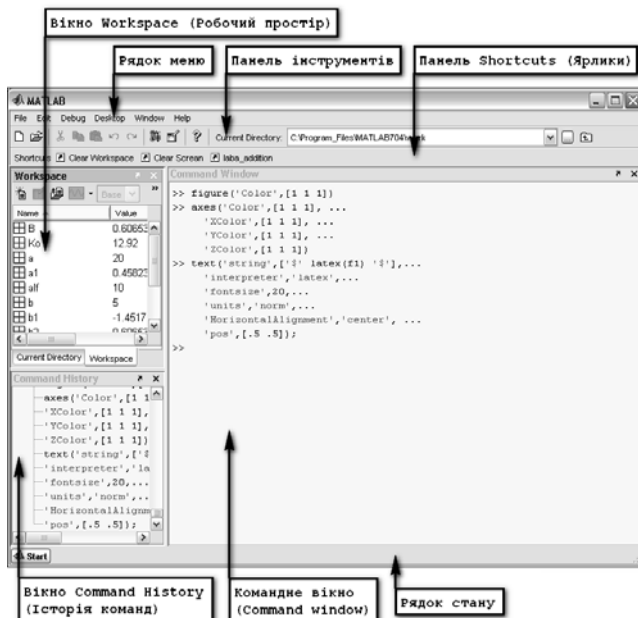


Рисунок 1.1 – Вікно робочого середовища MATLAB

1.1.2 Введення даних та операції з ними

Введення даних виконується у вікні Command Window. Результат обчислення виразу привласнюється змінній за допомогою символу «=». Ім'я змінної може складатися з символів латинського алфавіту, цифр та символу «_». Завершується введення натисненням клавіші «Enter», при цьому у вікні Command Window з'явиться результат.

Пакет MATLAB має деякі особливості:

- вбудовані функції (наприклад $\sin x$) записуються стрічними буквами, їх аргументи вказуються у круглих дужках, подивитись, які саме елементарні тригонометричні, експоненціальні та інші функції можна використати за допомогою команди `help elfun` ;
- значення змінних, назви файлів, записуються стрічковими і заголовними буквами **тільки латинськими літерами**, причому в залежності від регістру (тобто A и a) вони будуть різними;

- для блокування виводу результату обчислень деякого вираження після нього треба встановити знак «;» (крапка з комою);
- для введення коментарів потрібно поставити спочатку позначку «%».

Приклад:

$a = \sin(45)$ % перевірити, це у градусах чи у радіанах

$a = 0.8509$

Коефіцієнти рівнянь (поліномів) записуються у квадратних дужках та розділяються між собою пробілами або комами. Наприклад, щоб записати рівняння $p = 3s^3 + 0.5s^2 + 2s + 1$ потрібно ввести коефіцієнти у порядку від старшого до молодшого (якщо елемент відсутній-ввести нуль)

$p = [3 \ 0.5 \ 2 \ 1];$

Щоб знайти корені цього рівняння (зазвичай у ТАК знаходять корені характеристичного рівняння) потрібно обрати функцію $root(p)$, яка видасть і дійсні, і комплексні корені.

Елементи векторів і матриць записуються аналогічно. Елементи рядків розділяються проміж собою пробілами або комами, а самі рядки розділяються символом «;» (крапка з комою). Елементи, які відрізняються один від одного на постійну величину, задаються через їх початкове значення, шаг та кінцеве значення, що розділяються символом «:» (двокрапка). Якщо шаг дорівнює 1, то його можна опустити. Це схоже на цикли у програмуванні.

Приклад:

$A = [1 \ 2 \ 3 \ 4];$

$A = [1, 2, 3, 4];$

$A = 1:1:4;$

$A = 1:4$

$A =$

1 2 3 4

% Всі чотири записи еквівалентні

$V = [1:6; 1 \ 2.5 \ 5:5:20;]$

$V =$

1.0000 2.0000 3.0000 4.0000 5.0000 6.0000

1.0000 2.5000 5.0000 10.0000 15.0000 20.0000

Пакет MATLAB є насамперед програмою для роботи з матрицями, тому звичайні математичні операції $*$, $/$, $+$, $-$, $^$ виконуються в матричному виді. Для поелементного множення, ділення, додавання, віднімання та возведення до ступеню використовується спеціальний оператор «.» (крапка).

Приклад:

```
% Обчислення суми квадратів чисел від одного до десяти
```

```
sum([1:10].^2)
```

```
ans =
```

```
385
```

Зверніть увагу на дві обставини:

- попереду команди `sum` не було привласнення і MATLAB видав відповідь у «свою» змінну `ans`;

- попереду зведення у степінь 2 стоїть крапка, яка повідомляє про те, що зліва стоїть скалярний, а не матричний вираз і обчислювати слід по правилам скалярних величин.

1.1.3 Основні базові команди та системні змінні (`clc`, `help`, `elfun`, `type`, `pi`, `ans`)

До найбільш вживаних далі базових команд та системних змінних середовища MATLAB можна віднести наступні (таблиці 1.1 та 1.2). Всі команди (функції) надані у Додатку А.

Таблиця 1.1 – Базові команди середовища MATLAB

clc	очищує екран і розміщує курсор у лівому верхньому куті порожнього екрана;	
help	вивід довідкової інформації у вікні Command Window;	
	help help	% Вивід інформації по використанню команди help
	help sin	% Вивід інформації по функції sin (або іншій потрібній)
	help elfun	% Показує перелік елементарних функцій
	help specfun	% Показує перелік спеціальних функцій
	help type (<i>ім'я файлу або команди</i>)	% Вивід всіх розділів довідкової системи % показує текст відповідної програми, наприклад type step

Таблиця 1.2 – Основні **системні змінні**, застосовувані в системі MATLAB:

i або j	мніма одиниця (квадратний корінь з -1)
pi	число $\pi = 3.14159265358979$
inf	значення машинної нескінченності, наприклад: <code>>>k=2/cosd(90) %тут функція cosd() має аргументом кут у градусах, дивись <i>help elfun</i></code> Warning: Divide by zero <code>k = -Inf</code>
ans	змінна, що зберігає результат останньої операції без знака привласнення; <code>sin(pi/4) %функція sin() з аргументом у радіанах</code> <code>ans = 0.7071</code>
NaN	указівка на нечисловий характер даних (Not-a-Number), а також невизначеність 0/0; <code>x=-pi:pi/3:pi;</code> <code>z=sin(x)./x</code> Warning: Divide by zero. <code>z = 0.0000 0.4135 0.8270 NaN 0.8270 0.4135</code> <code>0.0000</code>

1.1.4 Оформлення результатів обчислень (plot, subplot, figure, grid, axis, title, text)

Значною перевагою пакета MATLAB є можливість легкого виводу результатів розрахунків у вигляді графіків на екран.

Найпростішою формою подання інформації в графічному вигляді є створення двовимірного графіка. Для побудови таких графіків у середовищі пакета передбачено декілька функцій[1-3], розглянемо найбільш поширену **plot** – побудову графіка в декартовій системі координат.

Приклад:

% Розрахунок даних для побудови графіків

`x = 0:0.05*pi:2*pi;` % введення змінної x , що змінюється від 0 до 2π з кроком 0.05π

`y1 = sin(x);`

`y2 = cos(x);`

% Побудова графіків $y1 = \sin(x)$ та $y2 = \cos(x)$ на спільній координатній площині

```
plot(x,y1,x,y2)
```

Функція `plot` автоматично відкриває нове вікно виводу графіків, якщо до цього воно ще не було відкрите. Якщо вікно виводу графіків вже було відкрите, то функція `plot` використовує його як задане і попередній графік зникне. Для відкриття нового вікна для виводу необхідно спочатку виконати команду **figure**.

% Побудова графіків в окремих вікнах

```
figure
```

```
plot(x,y1)
```

```
figure
```

```
plot(x,y2)
```

Для оформлення графіків призначені наступні функції:

subplot- розташування у одному графічному вікні декількох графіків;

axis - зміна масштабу побудови на вісі x та y ;

text - розташування підпису у заданому місці графічного вікна;

title - показ назви графіку;

xlabel - визначення назви горизонтальної вісі;

ylabel - визначення назви вертикальної вісі;

grid - нанесення (відміна нанесення) вимірювальної сітки.

Всі дії, що виконують функції **axis**, **text**, **title**, **xlabel**, **ylabel**, **grid** можна виконати *безпосередньо на графіку, використовуючи меню вікна **Графік*** (але тоді при повторенні цієї роботи зі змінами ви повинні бути всі ці редагування знов проробити).

Спробуйте додати до попередніх графіків ці функції і проаналізуйте результат.

Ще одним засобом подання графічної інформації є функція **subplot**, яка ділить вікно виводу графіків на задане число частин задля більш компактного наочного представлення інформації.

% Поділ вікна на дві частини по вертикалі

```
subplot(211), plot(x,y1)
```

```
subplot(212), plot(x,y2)
```

У дужках після назви цієї функції перші дві цифри (у нас це 21) – на скільки частин ділиться вікно по вертикалі(отримаємо у нашому прикладі дві стрічки) та горизонталі (у нас буде один

стовпчик) , відповідно, а третя цифра (спочатку 1, у наступній стрічці 2) – номери активної частини.

Розглянемо це на простому прикладі:

```
x=0:pi/10:2*pi;  
subplot(221)  
plot(x,sin(x)),grid  
subplot(222)  
plot(x,sin(x+pi/2)),grid  
subplot(223)  
plot(x,sin(x),x,cos(x)),grid  
subplot(224)  
plot(x,tan(x)),grid
```

Приклад оформлення графіків згідно наведеній нижче програмі [4] наданий на рисунку 1.2 (при копіюванні тексту у М-файл зверніть увагу, що всі коментарі повинні починатися з позначки «%»), навіть коли це перенесення на наступний рядок):

```
% Результати експерименту (слідкуйте, щоб кількість точок  
% по осях абсцис та ординат була однаковою!)  
x=[1.5, 2.13, 2.6, 3.09, 3.6, 4.15, 4.52, 4.78, 5.2, 5.59, 6.77, 7.74,  
8.55, 9.13, 10.58];  
y=[80:20:180, 190, 200, 212, 220, 240, 252, 260, 264, 276];  
% Апроксимаційний вираз  
y1=239.984*atan (0.222*x) ;  
% Вивід графіків та їх оформлення  
plot(x,y,'ko',x,y1,'k:') % перелік даних, що будуть виведені по  
% осях для першої та другої залежності: x, y та x, y1, а  
% також необов'язкова позначка для визначення кольору,  
% маркеру та типу лінії у вигляді 'color_style_marker', більш  
% детальніше можна подивитись у help або у примітках  
% даного посібника  
grid % додавання сітки  
axis ([0, 11, 0, 286]) % масштаб осей x від 0 до 11 та y від 0  
% до 286  
ylabel('U,[B]') % назва вісі y  
xlabel('Im, [A] ') % назва вісі x  
text (3,60,'o - експериментальні дані','FontName','Arial Unicode  
MS')
```

text (3,40,'--- - $U=239.984\arctg(0.2221I_m)$ ','FontName','Arial Unicode MS')

title ('Апроксимація характеристики неробочого ходу генератора', ...

'FontName','Arial Unicode MS') % підпис графіку

При виводі заголовків, підписів до осей або інших *кириличних написів* може виникнути проблема – програма MATLAB неправильно відображає кириличні символи у вікні графіку. Вирішенням цієї проблеми є наочна вказівка шрифту у функціях, що виводять текст (так само, як це зроблено у наведеному вище прикладі). Для цього в функцію послідовно передається два додаткових параметра:

- назва властивості, що встановлюється наступним параметром (в даному випадку 'FontName');
- нове значення цього параметру ('Arial Unicode MS' або інший Unicode шрифт, що має потрібні символи).

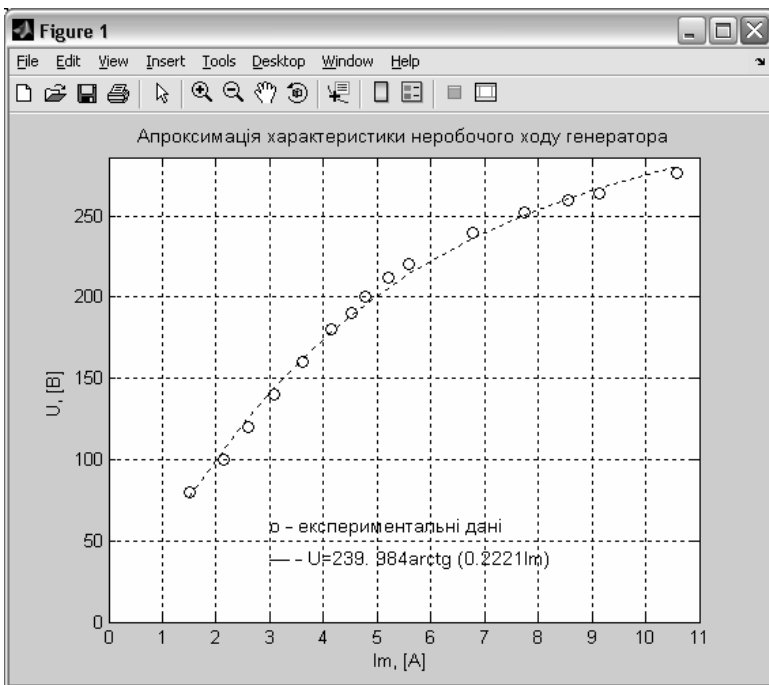


Рисунок 1.2 – Вигляд графічного вікна після виконання прикладу

1.1.5 Збереження даних для відновлення роботи

До сих пір мова йшла про роботу з MATLAB у *інтерактивному режимі*, на зразок «питання-відповідь». Всі ці результати зникають при вимиканні MATLAB і не з'являються при повторному включенні, але на етапі розробки це зручно. Щоб зберегти дані, їх краще оформити у вигляді так званого М-файлу. Його різновиди: *script* (повністю закінчений фрагмент, який не потребує введення додаткових даних) та *файл-функція* (потребує вхідних даних і повертає, але не обов'язково, вихідні дані).

Для створення такого файлу і вміщення до нього власних результатів у меню File оберіть пункт «Створити»/«M-file» (або script) та скопіюйте у *вікно Редактора*, що з'явиться на екрані, свою інформацію, яку отримали у сеансі роботи у інтерактивному режимі. Відредагуйте її згідно з правилами MATLAB: заберіть позначки на початку кожної стрічки «>>», а також результати розрахунків, додайте наприкінці кожного рядка позначку «;». Зберігати файл краще попередньо задавши ім'я **тільки латинськими літерами**. Розширення «.m» комп'ютер додає самостійно. Для запуску скопіюйте всю відредаговану інформацію з файлу і розмістіть її знову у командне вікно MATLAB після позначки «>>». Натисніть клавішу Enter і ваша програма виконається. Якщо знайдуться помилки, внесіть *виправлення у вікні Редактора файлу*, знов скопіюйте і запустіть до виконання або збережіть у каталог *work MATLAB* або на свій носій.

Контрольні питання

1. Яка різниця між інтерактивним та програмним режимами роботи MATLAB?
2. Які основні команди для побудови двовимірних графіків? Які додаткові можливості редагування існують у графічному вікні.
3. Як зберегти програми та дані розрахунків? Чим відрізняються mat-файли від m-файлів?

2 ДЕЯКІ ЗАГАЛЬНІ ПОЛОЖЕННЯ ТА МОМЕНТИ ПРИ МОДЕЛЮВАННІ АНАЛОГОВИХ СИСТЕМ КЕРУВАННЯ

2.1 Форми опису моделей, перетворення між ними (tf, zpk, ss)

2.1.1 Основні форми подання моделей неперервних систем

З курсу ТАК [1-4] відомо, що систему керування можна подати як у вигляді передатної функції, так і у векторно-матричній формі чи у формі подання за допомогою нулів і полюсів, що дозволяє використовувати як класичні, так і сучасні методи синтезу САК. Під час дослідження можуть розглядатися як неперервні, так і дискретні системи, для яких доступні перетворення між різноманітними формами подання систем.

Найбільш поширений метод, коли досліджувану систему можна описати в такому вигляді:

• *передатною функцією (tf)*, наприклад,

$$W(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} \quad (2.1)$$

Практично теж саме з точки зору математики – надати багаточлени чисельника та знаменника через їхні корені:

• *у термінах нулів, полюсів системи та коефіцієнта підсилення (zpk)*, наприклад,

$$W(s) = K \frac{(s - z_1) \cdot (s - z_2) \dots (s - z_{m-1}) \cdot (s - z_m)}{(s - p_1) \cdot (s - p_2) \dots (s - p_{n-1}) \cdot (s - p_n)} \quad (2.2)$$

де z – нулі полінома;

p – полюси полінома;

k – коефіцієнт підсилення.

Найбільш зручний для матричної лабораторії MATLAB спосіб це

• **моделі простору станів (ss)**, наприклад,

$$\begin{aligned} \dot{x} &= A \cdot x + B \cdot u && \text{- рівняння стану} \\ y &= C \cdot x + D \cdot u && \text{- рівняння виходу} \end{aligned} \quad (2.3)$$

де x - вектор стану системи,

u - вектор керуючих і збуджуючих впливів,

y - вектор вихідних сигналів системи.

A - матриця, що пов'язує внутрішні змінні системи;

B - матриця, що пов'язує вектор керування з внутрішніми змінними системи;

C - матриця, що пов'язує вектор вихідних сигналів з внутрішніми змінними системи;

D - матриця, що пов'язує вихідні сигнали з сигналами керування.

Всі ці моделі у пакеті Control System Toolbox, який використовується MATLAB для опису моделей систем керування об'єднані у новий клас об'єктів - LTI об'єкти[2] - лінійні системи з постійними параметрами. В залежності від вибору моделі лінійна система може бути завданою або парою многочленів чисельник/знаменник передатної функції, або трійкою (нулі, полюси, узагальнений коефіцієнт передачі), або четвіркою (A, B, C, D) для моделей у просторі стану.

2.1.2 Змінні стану динамічної системи

У зв'язку з тим, що такий підхід є практично основним у MATLAB (особливо при розгляданні задач з спостережувачем та інших [3]), і що цілком відповідає його назві (матрична лабораторія), приділимо трохи більше уваги цьому опису. Більш докладно його можна знайти у [1-4].

Аналіз та синтез систем керування у часовій області базується на понятті стану системи.

Стан системи – це сукупність таких змінних, знання яких разом зі вхідними функціями та рівняннями, що описують динаміку системи, дозволяє визначити її майбутній стан та вихідну змінну.

Для динамічної системи її стан описується набором *змінних стану* [$x_1(t), x_2(t), \dots, x_n(t)$]. Це такі змінні, що визначають майбут-

ню поведінку системи, якщо відомий її поточний стан та усі зовнішні впливи. Розглянемо систему, що зображена на рисунку 2.1,

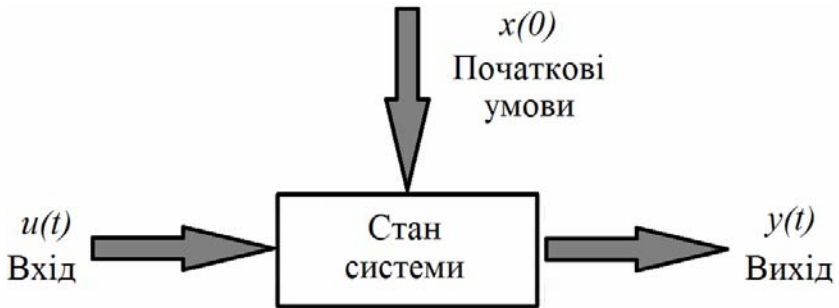


Рисунок 2.1 а) – Динамічна система

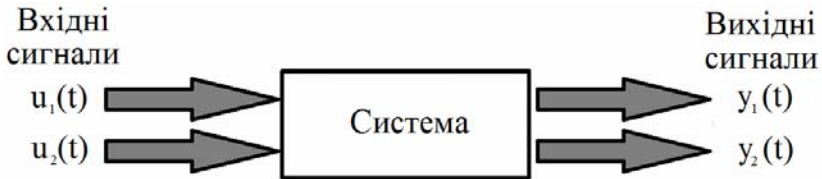


Рисунок 2.1 б) – Структурна схема динамічної системи керування

де $y(t)$ та $u(t)$ є вихідні та вхідні змінні, відповідно.

Змінні стану описують поведінку системи у майбутньому, якщо відомі поточний стан, зовнішні впливи та рівняння динаміки системи.

Розглянемо цей опис спочатку на прикладі механічної системи «маса – пружина» з затуханням [7] (рисунок 2.2).

Тертя об
стілки

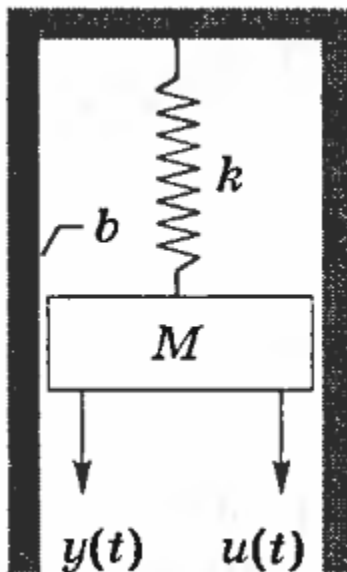


Рисунок 2.2 – Механічна система «маса – пружина» з затуханням

Кількість змінних стану, що обирають для опису системи, повинна бути по можливості мінімальною. Для даної системи цілком достатньо мати **дві** змінних стану – *положення у та швидкість руху маси* dy/dt (це дорівнює порядку диференційного рівняння, що описує обрану систему). Таким чином, маємо сукупність (x_1, x_2) , де

$$y(t) = x_1(t) \text{ та } x_2(t) = dy/dt.$$

Зверніть увагу – у наших випадках так буде найчастіше, коли одна зі змінних стану дорівнює *вихідний змінній*, а наступна – *похідній від неї*. Тобто будемо найчастіше розглядати канонічні системи, де змінні стану є переміщення системи, швидкість та прискорення (для рівнянь третього порядку) або кут, кутові швидкість та прискорення для систем з обертанням.

Для нашого прикладу диференціальне рівняння, що описує поведінку системи, зазвичай описується у вигляді

$$M \, d^2y/dt^2 + b \, dy/dt + k \, y = u(t), \quad (2.5)$$

де M – маса вантажу, b – коефіцієнт тертя, k – жорсткість пружини, $u(t)$ – вхідна величина.

З врахуванням записаних вище змінних стану це рівняння приймає вигляд:

$$M \, dx_2/dt + b \, x_2 + k \, x_1 = u(t) \quad (2.6)$$

Таким чином, вихідне рівняння другого порядку (2.5) ми можемо записати у вигляді еквівалентної системи двох диференціальних рівнянь першого порядку:

$$dx_1/dt = x_2 \quad (2.7)$$

$$dx_2/dt = -k/M \, x_1 - b/M \, x_2 + 1/M \, u \quad (2.8)$$

Ці рівняння по суті описують поведінку системи у термінах швидкості зміни кожної змінної стану. Зверніть увагу, що такий запис - це просто по-іншому записані рівняння (2.4) та (2.5).

Таким же чином *перетворюють будь-які диференціальні рівняння n -го порядку до системи n рівнянь першого порядку*. Саме за рахунок закладених багатьох методів комп'ютерного розв'язання таких систем рівнянь і є цінним пакет MATLAB.

Отримаємо тепер з рівнянь (2.7) та (2.8) матриці A та B , що надходять до *рівняння стану* (зрозуміло, що x з крапкою зверху дорівнює dx/dt):

$$\dot{x} = A \cdot x + B \cdot u \quad (2.9)$$

Із порівняння цього запису і (2.7), (2.8) видно, що матриця A – це коефіцієнти праворуч від позначки « $\Rightarrow$ » поперед x_1 та x_2 , що надходять, а матриця B – поперед u , тобто:

$$A = [0 \quad 1; -k/M \quad -b/M]; \quad B = [0; 1/M]. \quad (2.10)$$

Як уже відмічалось вище, матриця A квадратна, має розмірність $n \times n$, а матриця B – розмірність $n \times m$ (де m – кількість вхідних сигналів, тобто для *одного* сигналу вона має вигляд матриці-стовпця). У загальному випадку вихідні сигнали лінійної системи

пов'язані зі змінними стану та вхідними сигналами *рівнянням виходу*:

$$y = Cx + Du, \quad (2.11)$$

де y – сукупність вихідних сигналів у вигляді вектора-стовпця. Тоді матриця $C = [1 \ 0]$, якщо ми вимірюємо першу змінну стану, яка у нашому випадку співпадає з y (записували $x_1(t) = y_1(t)$). Якби вимірювали ще і швидкість x_2 , було б $C = [1 \ 1]$. Матриця $D=0$, якщо не має прямого зв'язку входу з виходом [1].

Ще приклад – для рівняння руху $F = m \cdot a$:

$$F = m \frac{d^2 z}{dt^2} \Rightarrow \begin{cases} \frac{dz}{dt} = v \\ \frac{dv}{dt} = \frac{F}{m} \end{cases}$$

Розглянемо також відомий в електротехніці RLC-ланцюжок (рисунок 2.3) [7]:

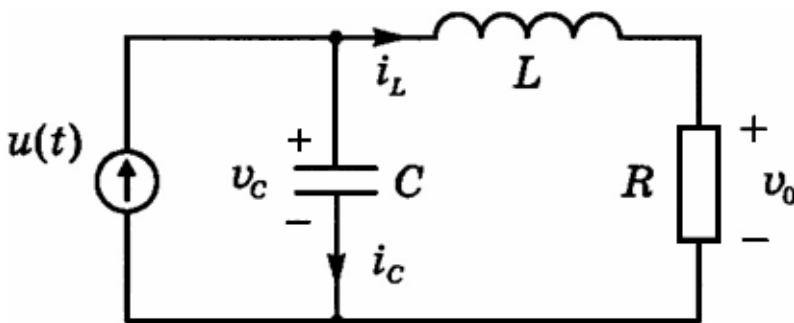


Рисунок 2.3 – RLC – ланцюг ($u(t)$ – джерело струму)

Стан системи характеризується двома змінними $x_1 = V_C(t); x_2 = i_L(t)$, причому такий вибір інтуїтивно робиться із міркувань, що у них запасається спільна енергія кола. Таким чином, x_1 та x_2 несуть інформацію про повну початкову енергію системи.

Потрібно для опису кількості необхідних змінних стану до-
рівнює кількості незалежних елементів, де накопичується енергія.
Згідно закону Кірхгофа для струму запишемо диференціальні рів-
няння швидкості змінення напруги на конденсаторі (нагадаємо,
що $U(t)$ це джерело струму) :

$$i_c = c \frac{dv_c}{dt} = U(t) - i_L \quad (2.12)$$

Закон Кірхгофа для напруги для правого контуру визначить:

$$L \frac{di_L}{dt} = Ri_L + Vc \quad (2.13)$$

Вихід системи, тобто напругу , що знімається з R , визна-
чимо лінійним алгебраїчним рівнянням:

$$V_0 = Ri_L(t)$$

Перепишемо рівняння (2.6) та (2.7) через змінні стану :

$$\left\{ \begin{array}{l} \frac{dx_1}{dt} = -\frac{1}{c}x_2 + \frac{1}{c}U(t) \end{array} \right. \quad (2.14)$$

$$\left\{ \begin{array}{l} \frac{dx_2}{dt} = -\frac{1}{L}x_1 + \frac{R}{L}x_2 \end{array} \right. \quad (2.15)$$

Тоді вихідний сигнал буде:

$$y_1(t) = V_0(t) = Rx_2 \quad (2.16)$$

Векторно-матрична модель: (2.8), (2.9), (2.10)

$$\left\{ \begin{array}{l} \dot{x} = Ax + BU - \text{рівняння стану} \\ y = Cx - \text{рівняння виходу} \end{array} \right.$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad A = \begin{bmatrix} 0 & -\frac{1}{c} \\ -\frac{1}{L} & +\frac{R}{L} \end{bmatrix} \quad B = \begin{bmatrix} \frac{1}{c} \\ 0 \end{bmatrix} \quad C = \begin{bmatrix} 0 & R \end{bmatrix} \quad (2.17)$$

Змінні стану можна було вибрати і по-іншому. На практиці у якості внутрішньої змінної вибирають те, що легко вимірюється.

2.1.3 Диференційні рівняння стану

Як вже відмічалось, стан системи описується диференційним рівнянням n -го порядку, а використання змінних стану дозволяє поміняти таке рівняння на систему n диференційних рівнянь 1-го порядку. Саме використання способів чисельного розв'язання таких систем рівнянь і надають цінності пакету MATLAB.

В загальному вигляді:

$$\begin{aligned} \dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + \dots a_{1n}x_n + b_{11}u_1 + \dots + b_{1m}u_m \\ \dot{x}_2 &= a_{21}x_1 + a_{22}x_2 + \dots \\ \dot{x}_n &= a_{n1}x_1 + a_{n2}x_2 + \dots a_{nn}x_n + b_{n1}u_1 + \dots + b_{nm}u_m \\ \frac{d}{dt} \begin{bmatrix} x_1 \\ x_n \end{bmatrix} &= \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_n \end{bmatrix} + \begin{bmatrix} b_{11} & \dots & b_{1m} \\ b_{n1} & \dots & b_{nm} \end{bmatrix} \begin{bmatrix} u_1 \\ u_m \end{bmatrix} \\ \dot{x} &= Ax + Bu \\ y &= Cx + Du \\ \begin{bmatrix} y_1 \\ y_n \end{bmatrix} &= [c_{11} \dots c_{1m}] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \quad (2.18)$$

На наступному рисунку 2.4 наведене графічне представлення векторно – матричної моделі.

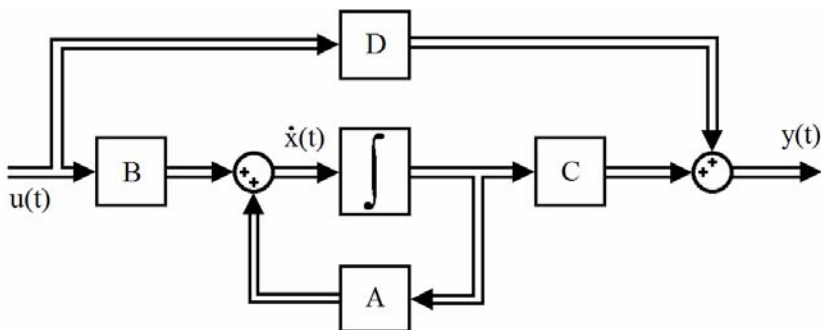


Рисунок 2.4 – Графічне представлення векторно-матричної моделі

Тут позначка $\Rightarrow$ відріжає матрично-векторний характер. У пакеті MATLAB, як вже відмічалось раніше, передатна функція системи, що описується через змінні стану (space state) записується у вигляді $\text{sys} = \text{ss}(A, B, C, D)$, де sys – ім'я, що завдає сам користувач.

Розглянемо приклад з електроприводу [6], а саме запишемо рівняння стану, якщо відомий оператор (коефіцієнт передачі) розімкненої системи.

$$K_p(p) = \frac{K_y}{p(T_g T_m p^2 + T_m p + 1)} = \frac{\varphi(t)}{U_o(t)}.$$

Тут K_y – коефіцієнт підсилення розімкненої системи;

$\Phi(t)$ – кут повороту валу двигуна.

$$T_g T_m \frac{d^3 \varphi}{dt^3} + T_m \frac{d^2 \varphi}{dt^2} + \frac{d\varphi}{dt} = K_y U_o(t)$$

$$\frac{d^3 \varphi}{dt^3} + \frac{1}{T_g} \frac{d^2 \varphi}{dt^2} + \frac{1}{T_g T_m} \frac{d\varphi}{dt} = \frac{K_y}{T_g T_m} U_o(t)$$

У якості змінних беремо $\varphi(t)$, $\dot{\varphi}(t)$, $\ddot{\varphi}(t)$ та позначимо їх змінною z :

$$z_1 = \varphi$$

$$z_2 = \dot{\varphi} = \dot{z}_1$$

$$z_3 = \ddot{\varphi} = \dot{z}_2$$

$$\dot{z}_3 = \ddot{\varphi} = \frac{1}{T_{\varphi}} \ddot{\varphi} - \frac{1}{T_{\varphi} T_M} \dot{\varphi} + \frac{K_y}{T_{\varphi} T_M} U_{\delta}$$

$$\dot{z}_1 = 0 \cdot z_1 + 1 \cdot z_2 + 0 \cdot z_3 + 0 \cdot U_{\delta}$$

$$\dot{z}_2 = 0 \cdot z_1 + 0 \cdot z_2 + 1 \cdot z_3 + 0 \cdot U_{\delta}$$

$$\dot{z}_3 = 0 \cdot z_1 - \frac{1}{T_{\varphi} T_M} z_2 - \frac{1}{T_{\varphi}} z_3 + \frac{K_y}{T_{\varphi} T_M} U_{\delta}$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\frac{1}{T_{\varphi} T_M} & -\frac{1}{T_{\varphi}} \end{bmatrix}; B = \begin{bmatrix} 0 \\ 0 \\ \frac{K_y}{T_{\varphi} T_M} \end{bmatrix}; C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; D = 0.$$

Більш докладніше все це можна подивитись у літературі [3, 4]. Крім того, нижче наведені команди, що дозволяють за допомогою команд MATLAB отримати одну форму запису з іншої.

Наприкінці зауважимо, що у якості змінних стану можна обирати різні лінійно незалежні комбінації (наприклад, для RLC - ланок дві напруги на елементах, а не напруга на конденсаторі та струм скрізь індуктивність, як частіше), але інженери у першу чергу обирають такі змінні, які визначають енергію, що накопичується у системі. Але оскільки поняття змінних стану застосовується до аналізу не тільки фізичних, але і біологічних, соціальних та економічних систем, у цих випадках до вибору змінних стану підходять ширше, трактуючи їх відповідним чином.

2.1.4 Стійкість у просторі станів (det, roots, poly)

Якщо немає вхідного сигналу, тобто $u = 0$, рівняння стану (2.3) розглядається у такому вигляді

$$\dot{x} = A \cdot x$$

де x – вектор стану. Враховуючи векторний характер рівняння та не наводячи подробиць [7] відмітимо, що цей запис фактично являє собою відоме характеристичне рівняння

$\det(sI - A) = 0$, де I – одинична матриця, s_i – корені цього рівняння.

Якщо реальна частина для всіх коренів $\operatorname{Re}(s_i) < 0$ – система стійка, якщо > 0 – то нестійка.

Таким чином, коли система представлена у змінних стану, перш за все треба знайти її характеристичний поліном. Це можна зробити декількома способами. Перший спосіб – розкрити визначник $\det(sI - A) = 0$ вручну. Після цього можливо обчислити корені характеристичного рівняння за допомогою функції MATLAB *roots(s)*, але при великій розмірності матриці A можуть виникнути труднощі.

Другий спосіб – використати функцію *poly* в MATLAB, яка дозволяє створити поліном n -го порядку по вектору його коренів та може бути використана для розрахунку характеристичного рівняння матриці A . Тобто потрібно шукати детермінант матриці A .

Наприклад, нехай матриця A задана у вигляді:

$$A = \begin{bmatrix} -8 & -16 & -6 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Після використання $p = \operatorname{poly}(A)$ отримаємо коефіцієнти характеристичного поліному

$$p = [1 \ 8 \ 16 \ 6], \text{ отже поліном має вигляд } s^3 + 8s^2 + 16s + 6.$$

Після використання функції $roots(p)$ отримаємо n коренів характеристичного поліному:

-5.0861; -2.4280; -0.4859 - числа від'ємні, система стійка.

Якщо характеристичне рівняння являє собою функцію якогось параметра, то за критерієм Рауса-Гурвіца [5-7] можливо визначити діапазон значень цього параметру, при котрих система буде стійка. Взагалі ж критерій Рауса-Гурвіца говорить тільки про те, скільки коренів знаходиться у правій частині графіку «уявна частина $\text{Im}(p)$ – дійсна частина $\text{Re}(p)$ », тобто додатних значень дійсної частини кореня. При цьому він не вказує конкретного положення цих коренів. Це ми можемо легко зробити за допомогою MATLAB, використавши формули переходу від однієї форми запису до іншої (дивись далі).

2.1.5 Завдання передатних функцій та поліномів у MATLAB

Диференційні рівняння, що описують динаміку фізичної системи, отримуються на підставі фундаментальних фізичних законів. Цей метод у рівній мірі застосовується до механічних, електричних, гідравлічних та інших систем. У загальному випадку, при моделюванні замкнених систем керування, що зазнають вплив різноманітних зовнішніх дій, а також при різних початкових умовах, аналітичне рішення бува отримати дуже складно. Тут можна вдатися до допомоги MATLAB, який чисельно розв'яже задачу та надасть результат у графічній формі.

У інженерній практиці в основному розглядають лінійні системи (або лінеаризовані), що дозволяє використовувати апарат перетворення Лапласа. Метод перетворення Лапласа дозволяє замінити досить складне розв'язання диференційних рівнянь відносно простим розв'язанням алгебраїчних рівнянь.

MATLAB дозволяє досліджувати системи, що описуються передатними функціями (ПФ) [1-7], тобто через відношення вихідного сигналу до вхідного. Згідно визначенню ПФ лінійної сис-

теми - це відношення перетворення Лапласа вихідної змінної до перетворення Лапласа вхідної змінної за умови, що всі початкові умови дорівнюють нулю. ПФ системи однозначно описує динамічний зв'язок між цими змінними. ПФ існує тільки для лінійних стаціонарних (с постійними параметрами) систем. У нестаціонарних системах один чи декілька параметрів залежать від часу, тому перетворенням Лапласа скористатися неможливо. ПФ описує поведінку системи у термінах вхід-вихід і не несе ніякої інформації про внутрішні змінні та характер їхнього змінення.

Таким чином, таке представлення систем більш компактне та має меншу кількість змінних, чим в просторі стану. Оскільки до моделі входять тільки вхідні та вихідні змінні, її називають зовнішнім описом.

Зовнішньо ПФ має вигляд відношення двох поліномів, тому розглянемо спочатку, як MATLAB оперує з алгебраїчними поліномами. Поліноми у MATLAB надаються у вигляді «вектор-стрічка», що складаються із коефіцієнтів за убувальним порядком степенів. Наприклад, поліном $p(s) = s^3 + 3s^2 + 4$ задається у вигляді $p=[1 \ 3 \ 0 \ 4]$. Зверніть увагу, що якщо навіть коефіцієнт біля якої небудь степені дорівнює нулю, він все рівно включається у представлення полінома $p(s)$. Функція $\text{roots}(p)$ визначає вектор-стовпчик, що вміщує корені цього поліному. І навпаки, якщо r - вектор – стовпчик, що вміщує корені поліному, то функція $\text{poly}(r)$ дає вектор-стрічку із коефіцієнтів поліному у убувальному порядку степенів.

$p=[1 \ 3 \ 0 \ 4]$; $r=\text{roots}(p)$ % завдання поліному обчислення коренів

$r = -3.3553$

$0.1777+1.0773i$

$0.1777-1.0773i$

$p=\text{poly}(r)$ % відновлення поліному згідно заданих коренів

$1.0000 \ 3.0000 \ 0.0000 \ 4.0000$

Додаток поліномів отримується за допомогою функції conv .

$p=[3 \ 2 \ 1]$; $q=[1 \ 4]$; % завдання поліномів $p=3s^2+2s+1$ та

$q=s+4$

$n=\text{conv}(p,q)$ % перемноження поліномів p та q

$n = 3 \ 14 \ 9 \ 4$ % $n(s)=3s^3+14s^2+9s+4$

$\text{value}=\text{polyval}(n,-5)$ % обчислення $n(s)$ при $s=-5$

value= - 66

Розглянемо на прикладі, як подивитись на графіку вплив зміни одного з коефіцієнтів поліному на дійсну та уявну частини його коренів (нагадаємо, що у даному випадку це поліном 3-го порядку, тому кількість коефіцієнтів дорівнює 4, причому коефіцієнт біля невідомої x^0 змінюється у діапазоні від значення нуля до 20 з кроком 0.5):

k=[0:0.5:20];

for i=1:length(k) % і змінюється від 1 до довжини вектору k

q=[1 2 4 k(i)];

p(:,i)=roots(q)

and

plot(real(p), image(p));

Додамо декілька слів про передатні функції.

Як відмічалось, опис системи може бути зроблений через диференційне рівняння порядку n, для якого з використанням перетворення Лапласу можна отримати алгебраїчне рівняння та запис через передатну функцію [1-3, 7]:

$$\frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_n y = b_0 \frac{d^m U}{dt^m} + b_1 \frac{d^{m-1} U}{dt^{m-1}} + \dots + b_m U \quad (2.19)$$

$$\begin{aligned} (s^n + a_1 s^{n-1} + \dots + a_n) Y(s) &= (b_0 s^m + b_1 s^{m-1} + \dots + b_m) U(s) \\ W(s) = \frac{Y(s)}{U(s)} &= \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_m}{s^n + a_1 s^{n-1} + \dots + a_n} \end{aligned}$$

причому $m \leq n$ - для реальної системи

Без доказів [7] передатну функцію можна розрахувати безпосередньо через змінні стану:

$$W(s) = \frac{Y(s)}{U(s)} = C(sI - A)^{-1}B + D \quad (2.20)$$

$$\text{де } I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Тоді, наприклад, для схеми на рис. 2.3 та відповідних матриць A,B,C,D отримаємо таку передатну функцію:

$$W(s) = (R/LC)/(s^2 + (R/L)s + 1/(LC)),$$

а для механічної системи (дивись приклад з рівнянням руху $F = m \cdot a$):

$$W(s) = \frac{z(s)}{F(s)} = \frac{1}{ms^2}$$

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}; B = \begin{bmatrix} 0 \\ m^{-1} \end{bmatrix}$$

$$W(s) = C(sI - A)^{-1}B = (1 \ 0) \begin{pmatrix} s & -1 \\ 0 & s \end{pmatrix} \begin{bmatrix} 0 \\ m^{-1} \end{bmatrix} = \frac{1}{ms^2}$$

Для подібних розрахунків у символьному вигляді краще скористатися символьною математикою MATLAB [3, 5].

Оскільки опис через вхід/вихід вміщує менше коефіцієнтів, чим у просторі стану, його завжди можна отримати з останнього, але зворотне перетворення нерівнозначне. Це пов'язано з тим, що вектор стану x отримується за допомогою підставлення нових

змінних, які обираються довільно, а вхід (U) та вихід (Y) залежать від фізичної природи та визначені рівнозначно.

Відомо, що знаменник передатної функції називають характеристичним рівнянням, фундаментальні значення у ТАК мають значення його коренів (полюсів), які ідентичні власним значенням матриці A. Корені чисельника називаються нулями.

Таким чином, передатна функція може бути також записана у такому вигляді, який відповідає третій прийнятій у пакеті MATLAB формі подання, а саме у термінах нулів, полюсів системи та коефіцієнта підсилення (zpk):

$$W(s) = \frac{K(s - z_1)(s - z_2) \dots (s - z_m)}{(s + p_1) \dots (s - p_n)} = \frac{a_1}{s - p_1} + \dots + \frac{a_n}{s - p_n} \quad (2.21)$$

Для реальної системи, як вже відмічалось, $n \geq m$, z_i – нулі (m штук), p_i – полюси (n штук). Останній запис у цій формулі відповідає розкладу на прості дробки (команда `residue` [1-3]), які у часовому просторі дають можливість представити вихідний сигнал $Y(t)$ як суму показових функцій – так званих мод, складових руху, тобто такому виразу:

$$W(s) = \frac{a_1}{s - p} + \dots + \frac{a_n}{s - p_n}$$

що відповідає у часовому просторі

$$y(t) = C_1 e^{-p_1 t} + C_2 e^{-p_2 t} + \dots + C_n e^{-p_n t} + [n(t)], \quad (2.22)$$

де $n(t)$ залежить від зовнішньої сили.

Нагадуємо, що дійсні полюси p_i відповідають складовій (2.22) з дійсним показником степеня, а два комплексно – спряжених корені завжди можна представити у вигляді:

$$p_{k,k1} = -\sigma \pm j\omega$$

Цій парі у (2.22) відповідає складова у (t) $C_k e^{\sigma t} \cdot \sin(\omega t)$.

Полюси (p_i) або власні значення матриці А лінійної системи повністю визначають її стійкість. Якщо дійсні частини полюсів (σ) від'ємні, реакція на обмежений вхідний сигнал теж завжди обмежена, тобто система стійка.

Нулі визначають значення коефіцієнтів (C_k) біля експоненціальних функцій, але при цьому не впливають на стійкість. Якщо полюс розміщений близько до нуля, відповідна мода мала. Якщо полюс та нуль співпадають, відповідна мода зникає.

2.1.6 Введення різних форм моделей

Відповідні команди пакета мають вигляд [1, 2-7] (дивись також Додаток А):

`sys = tf (num, den)` - передатна функція (transfer function), де num, den – поліноми чисельника та знаменника, згідно (2.1);

`sys = zpk (z, p, k)` - опис у термінах нулів та полюсів системи та коефіцієнта підсилення (тут у дужках визначені перші літери від zero/pole/gain), згідно (2.2);

`sys = ss (a, b, c, d)` - модель простору станів (space state), згідно (2.3).

Усі ці функції пов'язані між собою, можна переходити від однієї форми до іншої. Через яку функцію зробити опис лінійної системи чи окремої ланки залежить від того, які дані є в наявності: якщо чисельник та знаменник, тоді з функції *tf*, якщо задані нулі та полюси, тоді з *zpk*, якщо матриці А,В,С,Д – тоді з *ss*. Далі перейти від однієї до іншої форми можна за наведеними нижче прикладами. Нагадуємо, при копіюванні цих прикладів у М-файл, коментарі потрібно видалити, або додати перед ними позначку «%».

`z=0; p=[1 2]; k=3;` - задаємо відомі значення нулів, полюсів та коефіцієнта підсилення k.

`W1=zpk(z,p,k)` - викликаємо опис у відповідному вигляді, не ставимо позначку «;» наприкінці виразу, щоб переглянути у перший раз наочно передатну функцію W1

3 s

(s-1) (s-2)

W2=tf(W1) переходимо до опису у вигляді *tf* і теж перегляда-

ємо

Transfer function:

3 s

s^2 - 3 s + 2

W3=ss(W1) переходимо до опису у просторі стану ss

a =

b =

c =

d =

x1 x2

u1

x1 x2

u1

x1 1 1

x1 0

y1 1.5 1.5

y1 0

x2 0 2

x2 2

Аналогічно здійснюємо введення та перетворення у інших випадках. Більш детальні знання щодо пріоритету описів та перетворень, наприклад, типу, *tf2ss*, *ss2tf*, *ss2zp*, *zp2ss*, *tf2zp*, *zp2tf* можна отримати, наприклад, з [2].

Деякі приклади пояснюють їхнє використання:

num = [0.25 0.50 0.75];

den = [1.00 1.25 1.50];

[a b c d]=tf2ss(num,den) отримуємо одразу a,b,c,d через задані num, den

a = [-1.25 -1.50 1.00 0];

b = [1.00 ; 0];

c = [0.19 0.38];

d = 0.25;

[num,den]=ss2tf(a,b,c,d) отримуємо num, den через задані a,b,c,d

num = 0.25 0.50 0.75

den = 1.00 1.25 1.50

$[z,p,k]=ss2zp(a,b,c,d)$ отримуємо нулі, полюси та коефіцієнт передачі через задані a, b, c, d

$z = -1.00$

-1.00

$p = -0.63$

-0.63

$k = 0.25$

$[num,den]=zp2tf(z,p,k)$ отримуємо num,den через задані нулі, полюси та коефіцієнт передачі

$num = 0.25 \quad 0.50 \quad 0.75$

$den = 1.00 \quad 1.25 \quad 1.50$

2.1.7 Команди для перетворення між неперервними та дискретними системами (tf2ss, ss2tf, ss2zp, zp2ss, tf2zp, zp2tf, c2d, d2c, d2d)

Команди $c2d, d2c$ і $d2d$ виконують, відповідно, перетворення неперервної (continuous) до дискретної (discrete), дискретної до неперервної і дискретної до дискретної системи.

$sysd = c2d(sysc, Ts[,method])$ перехід від заданої раніше ПФ неперервної системи $sysc$ до дискретної, Ts - період дискретизації в секундах, $method$ - не обов'язковий параметр;

$sysc = d2c(sysd)$ зворотний перехід до еквівалентної неперервної моделі;

$sysd1 = d2c(sysd, Ts)$ - встановлення іншого періоду дискретизації Ts ;

У пакеті можливі різноманітні *методи дискретизації / інтерполяції*, включаючи кусково-постійне 'zoh' (Zero-order hold on the inputs – *приймається за замовчуванням*) та кусково-лінійне перетворення 'foh', підстановку Тастина 'tustin' та 'impulse' (Im-

pulse-invariant discretization). Так, команда `sysd = c2d (sysc, Ts, 'foh')` передбачає використання кусково - лінійного перетворення. Розглянемо приклади.

`s1=c2d(W,0.001)` дискретна модель з періодом дискретизації 0.001

отримуємо z - перетворення

Transfer function:

$$0.25 z^2 - 0.4995 z + 0.2495$$

$$z^2 - 1.999 z + 0.9988$$

`S2=d2c(S1)` перехід від дискретної моделі до неперервної

Transfer function:

$$0.25 s^2 + 0.5 s + 0.75$$

$$s^2 + 1.25 s + 1.5$$

`S3=d2d(S1,0.1)` змінюємо період дискретизації з попереднього значення на 0.1с

Transfer function:

$$0.25 z^2 - 0.4477 z + 0.2048$$

$$z^2 - 1.868 z + 0.8825$$

2.2 Типові ланки та їхні динамічні властивості

– *Підсилююча ланка* відповідає рівнянню $y = kx$

$$\text{Передатна функція } W(s) = \frac{y}{x} = k. \quad (2.23)$$

– *Інтегруюча ланка.*

Інтегруючий елемент описується рівнянням

$$Ty' = x, \text{ або } y = \frac{1}{T} \int_0^t x dt + y_0,$$

де x, y – вхідний та вихідний сигнали,

T – стала часу, y_0 – початкова умова.

Передатна функція ідеальної інтегруючої ланки має вигляд:

$$W(s) = \frac{K}{s}, \quad (2.24a)$$

де s – оператор Лапласа. При цьому $K = \frac{1}{T}$.

Передатна функція реальної інтегруючої ланки має вигляд

$$W(s) = \frac{K}{s(Ts + 1)} \quad (2.24b)$$

– *Інерційна ланка 1-го порядку.*

Рівняння, яким описується інерційна ланка 1-го порядку, має вигляд:

$$Ty' + y = Kx,$$

де x, y – вхідний та вихідний сигнали,

T – стала часу, K – коефіцієнт передачі.

Попередньому рівнянню відповідає передатна функція

$$W(s) = \frac{K}{s(Ts + 1)} \quad (2.25)$$

де $X(s), Y(s)$ – перетворені за Лапласом x та y .

– *Коливальна ланка* може бути описана рівнянням

$$T^2 y'' + 2dT y' + y = Kx,$$

де x та y – вхідний та вихідний сигнали, T – стала часу, d – параметр згасання, K – коефіцієнт передачі.

Якщо $d=0$ – ланка має назву консервативна, якщо $0<d<1$ – це коливальна ланка, якщо $d \geq 1$ – це інерційна (аперіодична) ланка другого порядку.

Попередньому рівнянню відповідає передатна функція

$$W(s) = \frac{K}{T^2 s^2 + 2dT s + 1}, \quad (2.26)$$

– ланка, що диференціює

$$W(s) = \frac{K_\partial s}{(T_\partial s + 1)} \quad (2.27)$$

Якщо знаменник відсутній – маємо *ідеальну* ланку, що диференціює, якщо T_d не дорівнює нулю – *реальну* ланку.

Існує ще декілька ланок – «цеглин», з яких по суті будується структурна схема системи керування, вони будуть розглянуті на момент використання.

2.3 Типові вхідні сигнали та отримання реакції системи на їх вплив у MATLAB (step, grid, lsim)

Під час аналізу якості систем керування найбільшу цікавість представляє поведінка у часі, оскільки ці системи об'єктивно динамічні. Перш за усе необхідно з'ясувати, чи є система керування *стійкою* (методи аналізу стійкості розглянемо детальніше трохи пізніше, оцінка стійкості згідно коренів характеристичного рівняння вже розглядалася).

Якщо система *стійка*, тоді показники якості можна оцінити по реакції системи на певний вхідний сигнал. Але оскільки заздалегідь невідомо, яким у реальних умовах буде цей сигнал, під час аналізу якості обирається деякий *тестовий вхідний сигнал*. Такий підхід оправданий, тому що є кореляція між реакцією системи на типовий вхідний сигнал та її поведінкою у реальних робочих умовах. Крім того, використання типового вхідного сигналу дозволяє проектувальнику порівнювати декілька варіантів створюваної системи. До того ж більшість систем керування у процесі експлуатації зазнає зовнішніх впливів, які за виглядом дуже близькі до тестових сигналів (таблиця 2.1).

Таблиця 2.1 – Тестові входні сигнали

Тестовий сигнал	$r(t)$	$R(s)$
Ступінчастий	$r(t) = A, t > 0$ $= 0, t < 0$	$R(s) = A/s$
Лінійний	$r(t) = At, t > 0$ $= 0, t < 0$	$R(s) = A/s^2$
Параболічний	$r(t) = At^2, t > 0$ $= 0, t < 0$	$R(s) = 2A/s^3$

У якості тестового сигналу може також служити одинична імпульсна функція, що отримується з прямокутного імпульсу

$$f(t) = \begin{cases} \frac{1}{\varepsilon}, & -\varepsilon/2 < t < \varepsilon/2 \\ 0, & \text{при інших } t, \end{cases}$$

де $\varepsilon > 0$. При $\varepsilon \rightarrow 0$ імпульс $f(t)$ прагне до одиничної імпульсної функції $\delta(t)$ з відомими властивостями.

Типові тестові сигнали мають загальний вигляд $r(t) = t^n$, відповідне перетворення Лапласа $R(s) = n!/s^{n+1}$. Звідси виходить, що реакцію на один тестовий сигнал завжди можна виразити через реакцію на інший тестовий сигнал. Від того, що ступінчастий входний сигнал є найбільш простим, його звичай і обирають задля оцінки якості системи.

Приклад.

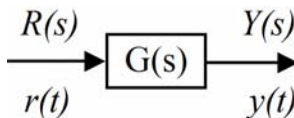


Рисунок 2.5 – Структурна схема розімкненої системи

Знайдемо реакцію розімкненої системи на ступінчастий сигнал, якщо її передатна функція

$$G(s) = 9/(s+10).$$

Тоді перетворення по Лапласу для вихідного сигналу

$$Y(s) = G(s) \cdot R(s) = G(s) \cdot 1/s = 9/s(s+10).$$

Перехідний процес описується виразом (дивись таблицю зворотних перетворень Лапласа [2])

$$y(t) = 0.9(1 - e^{-10t}),$$

а його встановлене значення $y(\infty)=0.9$. Для встановленої похибки

$$E(s) = [R(s) - Y(s)] \quad \text{маємо} \quad e_{ss} = \lim_{s \rightarrow 0} E(s) = 0.1.$$

Всю цю процедуру, ще і з побудовою графіку перехідного процесу легко отримати за допомогою команд MATLAB `tf` та `step` (рисунок 2.6):

```
G=tf(9,[1 10]);    %дивись (2.25)
step(G),grid
```

Графік перехідного процесу викликається командою `step` (ім'я передатної функції_1 [, ім'я передатної функції_2, ім'я передатної функції_3, ...]). Тут у квадратних дужках – необов'язкові параметри, тобто якщо будується один графік-вони не потрібні, а якщо ви хочете порівняти декілька графіків на одних висях, вкажіть їхні імена через кому. Всі ці функції повинні бути *попередньо задані* (як у нашому прикладі спочатку ввели `G`).

Якщо хочете більш наглядно, з можливою оцінкою числових даних графіка, подивитись перехідні процеси, додайте сітку, для чого після виразу `step(..)` поставте команду `grid`, тобто, наприклад, так:

```
step(W1, W2), grid
```

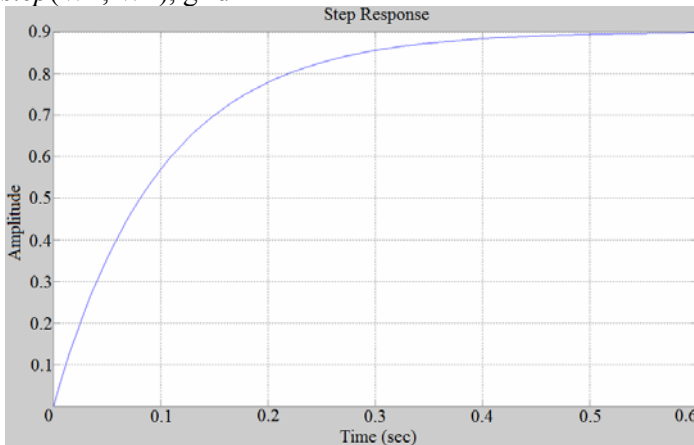


Рисунок 2.6 – Графік перехідного процесу для прикладу з ПФ $G(s) = 9/(s+10)$

Якщо заданий відрізок часу (або ви самі хочете детальніше або, навпаки, більш загально подивитись процес), потрібно спочатку задати кінцеве значення (щоб його легко було змінювати при потребі), а потім вектор часу з якимось кроком, наприклад, 0.1, тобто

```
tk=5; t=[0:0.1:tk];
```

та ввести це значення у команду *step(W1, W2,t)*, *grid*

Реакція на імпульсний вплив отримується командою *impulse* цілком аналогічно попередньому.

Більш цікава команда *lsim* - реакція системи на довільно заданий вхідний сигнал;

Приклад:

```
t=0:0.01:10; %задаємо зміну часу від 0 до 10 секунд
```

```
u1=sin(t); %вхідний сигнал у вигляді sin(t)
```

```
u2=4*t+5; %лінійний вхідний сигнал
```

```
w=tf(1,[1 3]); %задаємо передаточну функцію (2.23 – 2.27),
```

```
lsim(w,u1, t); %викликаємо команду для вхідного сигналу u1
```

```
figure %відкриваємо нове вікно задля сигналу u2
```

```
lsim((1,[1 3]),u2,t),grid %ПФ можна задати безпосередньо у lsim
```

На рис. 2.7 приведений результат роботи програми для лінійного вхідного сигналу. Деякі поширені види сигналів можуть бути встановлені за допомогою команди *gensig*, подробиці роботи з якою легко опрацювати за допомогою команди *help gensig* пакету або у [1, 3, 4].

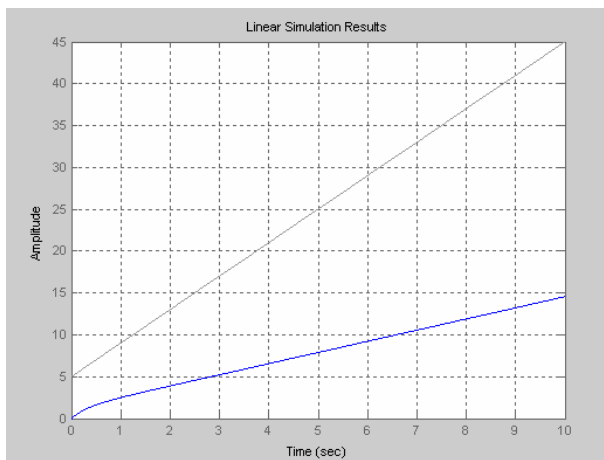


Рисунок 2.7 – Лінійний вхідний сигнал та реакція на нього

2.4 Використання частотних характеристик

Як відомо [1-3, 5, 7], це альтернативний метод аналізу та синтезу систем керування. Частотна характеристика визначається як реакція системи у встановленому режимі на синусоїдальний вхідний сигнал при зміні його частоти у всьому можливому діапазоні. При цьому в лінійній системі як вхідний сигнал, так і сигнал у будь якій іншій точці в усталеному режимі являються синусоїдальними, вони відрізняються від вхідного тільки по амплітуді та фазі.

Основні переваги використання цього методу [7].

1. Згідно експериментально отриманим частотним характеристикам системи можна визначити передатну функцію системи, крім того, при синтезі у частотних областях інженер отримує цінну інформацію про смугу пропускання системи та може оцінити її реакцію на небажані шуми та збурення.

2. Поведінку системи в усталеному режимі при синусоїдальному вхідному сигналі можна описати шляхом заміни $s=j\omega$ в передатній функції системи, в результаті чого отримуємо комплексну функцію, модуль та аргумент якої (частотні характеристики), якщо їх представити у графічній формі, дають корисну інформацію, яка необхідна задля аналізу та синтезу систем керування.

До недоліків методу частотних характеристик можна віднести той факт, що відсутній прямий зв'язок між властивостями системи у часовій та частотній областях. Такий зв'язок відстежується лише частково і на практиці вигляд частотних характеристик звичай підбирається таким чином, щоб вони у якійсь мірі задовольняли поведінці системи у часовій області.

Детальнішу інформацію про частотні характеристики можна отримати майже у всіх підручника з ТАК, тому не будемо на цьому зупинятися. Розглянемо особливості використання їх за допомогою MATLAB задля розв'язання типових задач.

2.4.1 Команди побудови та аналізу частотних характеристик (*bode*, *margin*, *nyquist*, *nicols*)

Основні частотні характеристики, що розглядаються у сучасній ТАК це амплітудно-фазові - АФЧХ, але є ряд труднощів у використанні їх задля аналізу складних систем, особливо пов'язаних зі зміною складу системи керування (додавання або віднімання складових системи). Більш зручні у використанні у пакеті логарифмічні частотні характеристики.

Задля побудови логарифмічних частотних характеристик (ЛФЧХ) використовується команда *bode()*, *grid*, де у дужках наданий список (або одна!) передатних функцій, попередньо заданих. Запас стійкості обчислюється після виконання команди *bode(..)* за допомогою команди *margin()*, де у дужках ім'я тієї самої ПФ, що і у команді *bode()*. Наприклад

`K=2; T=0.1; %введення параметрів`

`W=tf(K,[T 1]); %введення ПФ інерційної ланки першого порядку`

`bode(W),grid; margin(W);`

Тут здійснюється побудова ЛЧХ, причому запаси стійкості вказані на самому графіку вгорі у такому порядку: запас стійкості за модулем G_m у дБ, далі у дужках частота f , на якій він розрахований, потім запас за фазою G_f та у дужках відповідна частота (частота зрізу).

Задля подальшого використання у тексті програми цих запасів та відповідних їм частот можна вказати їхні імена у квадратних дужках попереду команди *margin(W)*:

`[ZapasPoMod,fZm,ZapasPoFaze,fZf]=margin(W) % вивід запасів стійкості`

Ці команди виведуть вам одразу запас по модулю (ми назвали його латинськими літерами *ZapasPoMod*) у дБ, частоту, на якій він обчислюється (*fZm*), запас по фазі (*ZapasPoFaze*) у градусах та частоту, на якій він обчислюється (*fZf*).

Інші частотні характеристики (критерії Найквіста, Нікольса) задаються аналогічно відповідними командами *nyquist*, *nikols* і будуть розглянуті, коли будуть потрібні.

2.5 Практичне дослідження динамічних характеристик за допомогою команд пакету

Для моделювання типових динамічних ланок (2.23 – 2.27) частіше за інші використовується запис передатних функцій у вигляді [1-3, 7]:

$$W_{(s)} = \frac{25s^2 + 2s + 1}{20s^2 + 1}$$

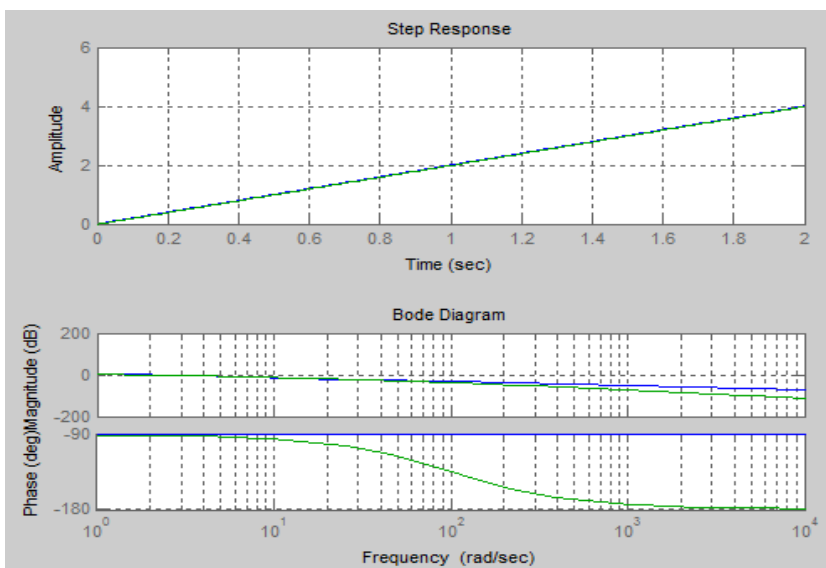
Задля введення такого виразу у середовище MATLAB відповідні коефіцієнти при операторі Лапласа чисельника та знаменника записуються так, як вище було написано задля коефіцієнтів рівнянь – у квадратних дужках, зліва направо, від старшого до молодшого, через пробіли або коми. Загальна кількість їх дорівнює $n+1$, де n - ступінь поліному чисельника або знаменника. При цьому, як уже відмічалось вище, якщо якийсь ступінь відсутній у виразі, це означає що, коефіцієнти при ньому буде дорівнювати нулю.

Так, для наведеного вище прикладу передатної функції чисельник потрібно записати у вигляді: [25 2 1], а знаменник у вигляді [20 0 1]. Сама передатна функція задається командою MATLAB **tf(чисельник,знаменник)**. Тут літери «tf», як відмічалось вище, є першими літерами від слів *transfer function* (передатна функція).

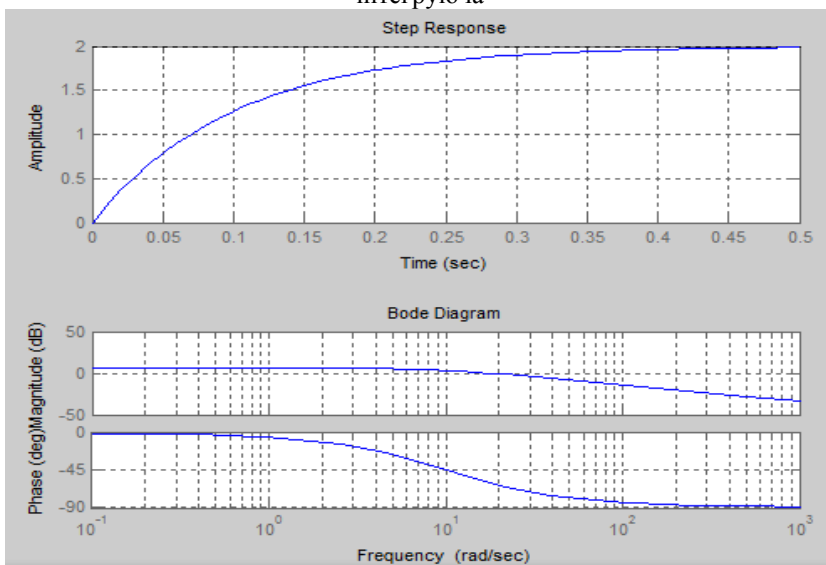
W1=tf ([25 2 1] ,[20 0 1]); W2=tf (1 ,[2 0]);

Як уже відмічалось, якщо ставимо після виразу позначку «;», то MATLAB не буде повторювати введений вираз, а якщо ні - повторить і ви зможете перевірити себе (не забудьте тоді під час копіювання у М-файл цей вираз видалити, а позначку «;» поставити!).

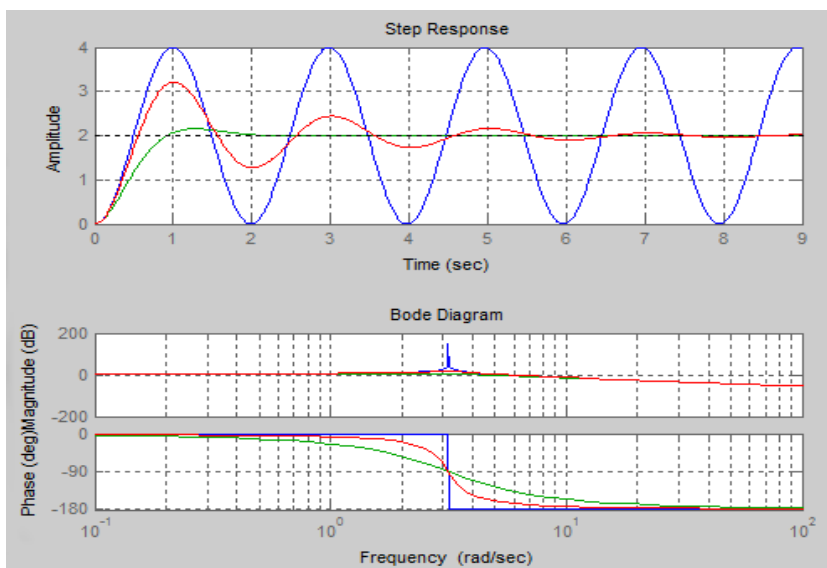
Для закріплення отриманих знань самостійно введіть передатні функції типових ланок і отримайте їхні динамічні характеристики. Параметри К та Т задайте з діапазонів 1-5 та 0.1-0.5, відповідно. Спробуйте навести всі на одному аркуші з використанням subplot (повинні отримати подібне нижче наведеному рисунку 2.8).



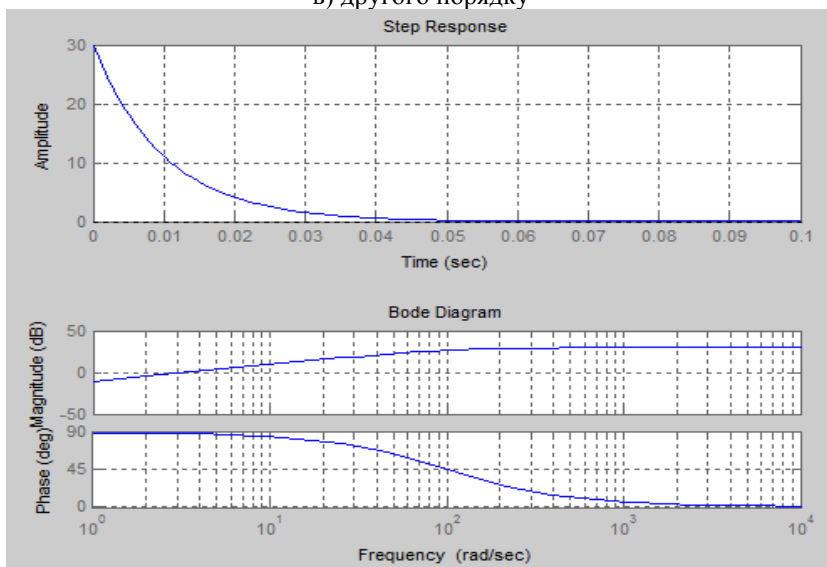
а) ідеальна та реальна
інтегруюча



б) інерційна першого порядку



в) другого порядку



г) реальна, що диференціює

Рисунок 2.8 – Динамічні характеристики типових ланок

Наберіть або **просто скопіюйте** написану нижче програму, спочатку у М-файл, після виправлення можливих помилок знов скопіюйте та помістіть у Command Window, після позначки >> та натисніть клавішу Enter. Самостійно додайте коментарі щодо типів ланок.

```
w1=tf(2,[1 0]);w11=tf(2,[0.01 1 0]);  
t1=[0:0.01:2];t2=[0:0.01:0.5];t3=[0:0.01:9];t4=[0:0.001:0.1];  
subplot(211)  
step(w1,w11,t1),grid  
subplot(212)  
bode(w1,w11),grid  
figure  
w2=tf(2,[0.1 1]);  
subplot(211)  
step(w2,t2),grid  
subplot(212)  
bode(w2),grid  
figure  
w31=tf(2,[0.1 0 1]);  
w32=tf(2,[0.1 0.4 1]);  
w33=tf(2,[0.1 0.1 1]);  
subplot(211)  
step(w31,w32,w33,t3),grid  
subplot(212)  
bode(w31,w32,w33),grid  
figure  
w4=tf([0.3 0],[0.01 1]);  
subplot(211)  
step(w4,t4),grid  
subplot(212)  
bode(w4),grid
```

2.6 Використання браузера LTI Viewer для аналізу динамічних характеристик

На відміну від розглянутого **командного способу** є другий спосіб аналізу динамічних властивостей – використання **інтерактивного інтерфейсу користувача LTI Viewer**.

У цьому разі ви можете переглянути всі замовлені характеристики, попередньо обов'язково задавши передаточні функції досліджуваних систем, але не зможете «витягти» і допрацювати окремий графік, на відміну від розглянутого «ручного» командного режиму. LTI Viewer забезпечує

- різні засоби аналізу поведінки систем;
- вивід на один графік характеристик різних моделей;
- масштабування в області графіка;
- обчислення параметрів якості регулювання, наприклад, часу регулювання;
- вивід залежностей для різних каналів вхід - вихід;
- обробку поданих графіків.

Для ініціалізації LTI Viewer необхідно набрати в командному рядку пакета MATLAB команду `ltiview`. Цю команду можна також викликати з додатковими аргументами, що дозволяють визначити тип характеристик системи, що будуть показані при відкритті вікна LTI Viewer. У загальному випадку структура команди матиме вигляд

`ltiview(plottype, sys, extras),`

де `sys` – назва моделі у робочій області пакета MATLAB; `plottype` – одна з команд для дослідження часових та частотних характеристик системи: `step`, `impz`, `initial`, `lsim`, `bode`, `nyquist`, `nichols`, `sigma`; `extras` – додаткові вхідні аргументи (наприклад, початкові умови, час звернення розрахунку).

Набір команди `ltiview` у командному рядку пакета відкриває вікно LTI Viewer зображене на рисунку 2.9.

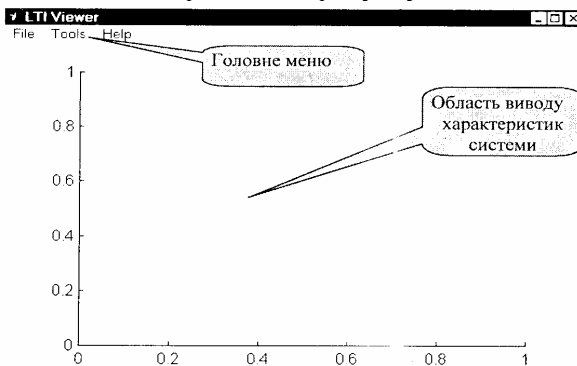


Рисунок 2.9 – Вигляд вікна LTI Viewer

LTI Viewer оперує з *власною робочою областю* вікна, що безпосередньо пов'язана, але не є цілком залежною від робочої області пакета MATLAB. LTI Viewer може маніпулювати тільки з системами у власній робочій області. Отже, для здійснення досліджень вибрану систему необхідно розмістити в робочій області LTI Viewer. Для цього використовується команда **Import** меню **File**.

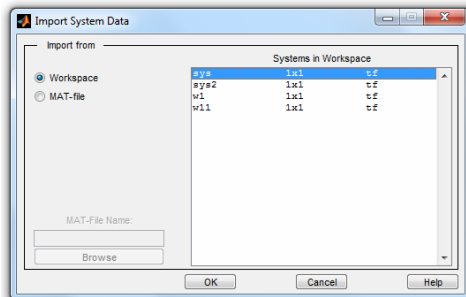


Рисунок 2.10 – Вікно перегляду LTI Browser

Внаслідок виконання цієї команди на екрані з'явиться вікно перегляду LTI Browser (рисунок 2.10), яке містить всі моделі, розміщені в робочій області MATLAB у момент відкривання LTI Viewer. Зі списку доступних моделей потрібну вибирають так:

- у вікні перегляду двічі натискають на клавішу «мишки» на назві моделі у випадку дослідження однієї моделі;
- вибирають необхідні моделі і натискають клавішу OK. Для вибору необхідних моделей необхідно, утримуючи натиснутою клавішу Ctrl, натиснути ліву кнопку «мишки» на назві моделі або, якщо назви моделей розміщені одна за одною, переміщати курсор «мишки» по назвах моделей, утримуючи натиснутою ліву кнопку «мишки».

Після вибору моделей у вікні виводу графіка з'являється графік характеристики, встановленої в області Plot Type (за замовчуванням це step). Для зміни вигляду вікна виводу графіків та проведення інших досліджень системи застосовується команда **View configuration** меню **Tools**. Внаслідок виконання цієї команди на екрані з'явиться вікно (рисунок 2.11).

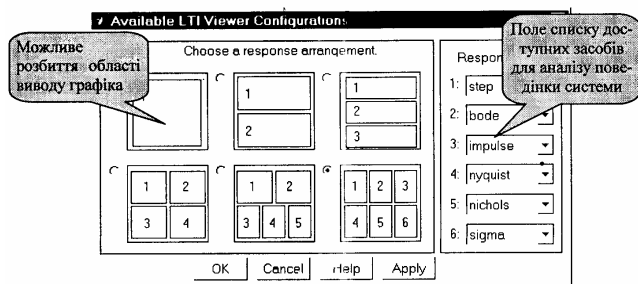


Рисунок 2.11 – Вигляд вікна вибору виводу графіків та засобів аналізу

Іншим способом визначення вибраного засобу аналізу та виводу залежностей на графік є вибір відповідних команд у меню, що з'являється після натискання правої клавіші «мишки» на області графіка.

Після натискання лівої кнопки «мишки» на виведеній залежності можна отримати координати цієї точки, а натискання правої клавіші «мишки» на виведеній залежності виводить інформацію про назву системи, а також вказує вхід та вихід системи, для яких отримана ця залежність.

Для встановлення параметрів графіків застосовується команда **Response Preferences** меню **Tools**. Внаслідок виконання цієї команди на екрані з'являється вікно Response Preferences (рисунок 2.12).

У цьому вікні можна визначити часовий або частотний діапазони для виводу характеристик системи; задати одиниці для виводу частотних характеристик; визначити межі осі Y для часових характеристик; задати значення для визначення параметрів регулювання.

Можливим є використання додаткових засобів для оформлення стилю ліній залежностей, що виводяться на графік, доступних у вікні Linestyle Preferences (рис. 2.13), що відкривається за допомогою команди Linestyle Preferences меню Tools.

Response Preferences

Time Domain

Time vector (sec.)

☒ Generate automatically
 ☐ Define: [T1,dT,Tf]

Y-axis Range

☒ Generate automatically
 ☐ Define: [Ymin,Ymax]

Plot Option

Settling time within: %
 Rise time from: to %

Frequency Domain

Frequency vector (rad/sec)

☒ Generate automatically
 ☒ Recalculate for each response type
 ☐ Define: logspace(d1,d2,N)

Magnitude in:

☒ Decibels
 ☐ Absolute
 ☐ Logarithmic

Phase in:

☒ Degrees
 ☐ Radians

Frequency in:

☒ Radians/second
 ☐ Hertz

OK

Cancel

Help

Рисунок 2.12 – Вигляд вікна Response Preferences

LineStyle Preferences

Distinguish by:

	Color	Marker	Line style	No Distinction
Systems	☒	☐	☐	☐
Inputs	☐	☐	☒	☐
Outputs	☐	☐	☐	☒
Channels	☐	☒	☐	☐

Color Order

green

red

cyan

magenta

yellow

blue

black

Marker Order

none

x

o

+

*

s

d

p

h

LineStyle Order

solid

dashed

dash-dot

dotted

none

OK

Cancel

Help

Apply

Рисунок 2.13 – Вигляд вікна LineStyle Preferences

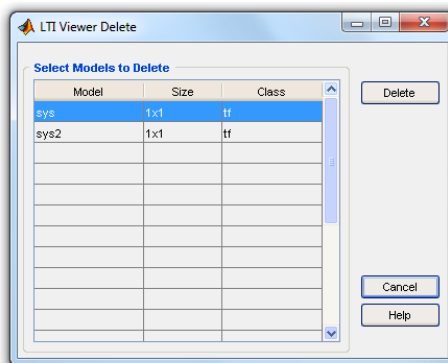


Рисунок 2.14 – Вигляд вікна LTI Browser

Якщо в робочій області LTI Viewer з’явилися моделі систем, які не будуть аналізуватися в подальшій роботі, їх можна стерти з робочої області. Для цього необхідно виконати команду Delete System меню File. Внаслідок виконання команди відкривається вікно (рисунок 2.14) в якому необхідно вибрати за допомогою одного з описаних раніше методів назви систем, які потрібно стерти і натиснути клавішу ОК.

Для поновлення параметрів аналізованих систем, служить команда Refresh Systems меню File.

Для виходу з середовища графічного інтерфейсу користувачу необхідно виконати команду Close Viewer меню File.

2.7 Завдання з’єднань блоків у MATLAB (series, parallel, feedback, append, connect)

За допомогою команд пакета Control System Toolbox можна створювати моделі системи, в якій є різноманітні з’єднання між її елементами (дивись Додаток А). До таких команд належать:

append (sys1, sys2,...) - об’єднує вектори входів і виходів систем для створення однієї системи:

$$sysA = \begin{bmatrix} sys1A & 0 \\ 0 & sys2A \end{bmatrix}$$

parallel (sys1, sys2) – реалізує паралельне з'єднання ланок;

series (sys1, sys2) – реалізує послідовне з'єднання ланок;

feedback (sys1, sys2) – реалізує зворотній зв'язок у системі;

connect (sys,q) – формує модель системи в просторі станів на основі матриці з'єднань q. Матриця q утворюється з рядків для кожного з виходів. Першим елементом кожного рядка є номер входу, а інші елементи зі знаком «-» або без нього вказують як формується вхід.

Більш традиційним є спосіб попереднього спрощення багатоконтурної структурної схеми з використанням команд задля реалізації послідовного, паралельного та зворотного зв'язків та відомих правил переносів вузлів та суматорів [2,7] до розімкненої системи.

Послідовне і паралельне з'єднання елементів системи можна виконувати за допомогою відповідних команд або операцій, відповідно, множення і додавання передаточних функцій $W1, W2$:

- series($W1, W2$), або просто $W1 * W2$ - для послідовного зв'язку;

- parallel($W1, W2$), або просто $W1 + W2$ - для паралельного зв'язку;

Аналогічно можна записати відношення задля зустрічно-паралельного з'єднання (зворотного зв'язку) блоків:

- feedback($W1, W2, [-1]$), або просто $W1 / (1 + W1 * W2)$ для від'ємного зворотного зв'язку (для додатного зв'язку потрібно змінити знак у дужках). $W1$ -передатна функція блоку, що розміщений у прямому ланцюгу, а $W2$ – у зворотному.

У випадку систем з багатьма входами і багатьма виходами додатково формуються вектори входів і виходів системи.

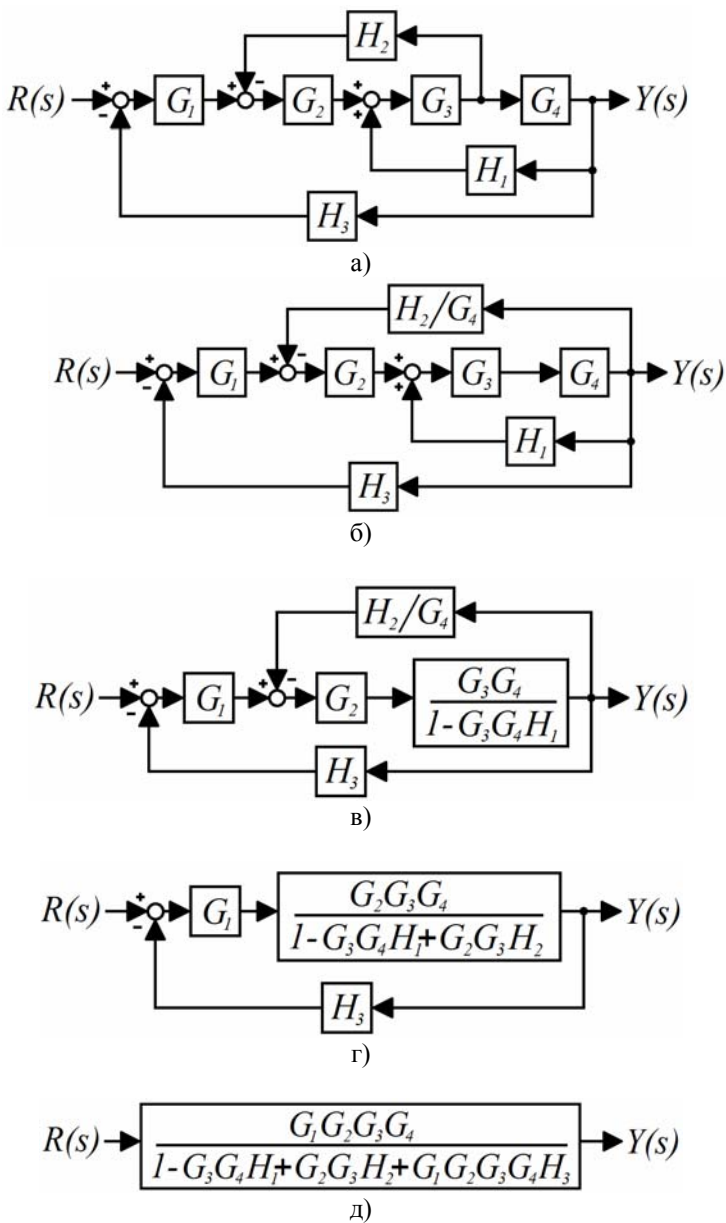


Рисунок 2.15 – Спрощення структурної схеми системи

2.8 Використання команд пакета Control System Toolbox для створення моделі системи

На прикладі наведеної схеми (рисунок 2.16) створимо модель системи, в якій є різноманітні з'єднання між її елементами, двома способами. Далі для наочності порівняння кінцевих результатів вони зведені у одну таблицю.

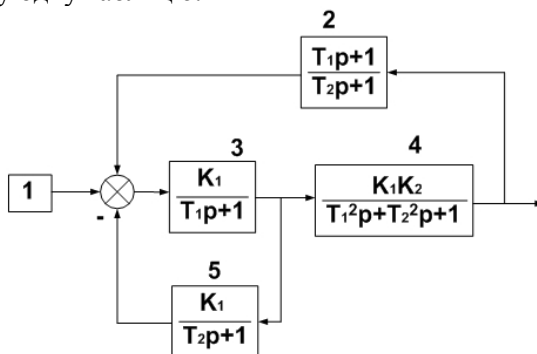


Рисунок 2.16 – Задана структурна схема

На вхід системи подається ступінчастий сигнал. Для використання першого засобу перетворимо спочатку систему – перенесемо вузол після блоку 3 згідно існуючих правил [1, 3, 7]. Після перетворення схема буде мати вигляд:

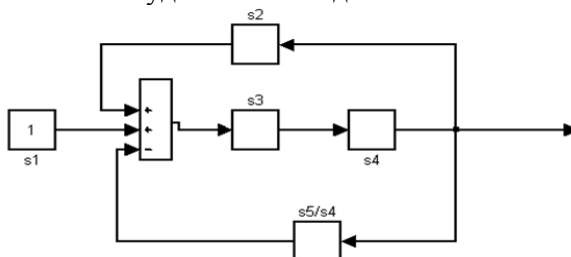


Рисунок 2.17 – Структурна схема в узагальненому вигляді

Далі послідовно знаходимо передаточну функцію системи за допомогою ТАК та реалізуємо це у MATLAB (спосіб 1).

Другим способом формуємо *модель системи в просторі станів* на основі матриці з'єднань q , що добре відстежується на прикладі у таблиці. Зверніть увагу, що у цьому випадку не потрібні перенесення вузлів у схемі, що спрощує розв'язання задачі.

Порядок роботи у цьому методі такий.

Спочатку записуємо передатні функції всіх блоків через команду $s1=tf(); \dots sn= tf()$. Потім записується функція $sln = append(s1, s2, \dots, sn)$, щоб описати кожен блок без врахування перехресних зв'язків. Далі записується матриця q , яка призначена для опису зв'язків блоків на структурній схемі, причому кожна стрічка цієї матриці відповідає одному входу системи sys ; перший елемент стрічки – це номер входу, наступні елементи вказують номери виходів, які алгебраїчно підсумовуються по цьому вході; від'ємні елементи позначають підсумування з позначкою мінус. Наприклад, якщо на вхід 5 поступають сигнали з виходів 4, 7, 8, причому сигнал з виходу 8 від'ємний, відповідна стрічка матриці зв'язків q має вигляд $[5 \ 4 \ 7 \ -8]$. Вектори *input* та *output* визначають, які входи та виходи є зовнішніми, і підсумкова модель $sysn$ має саме ці входи та виходи.

Додамо, що іноді буває зручно створити фіктивний блок (підсилювач з коефіцієнтом підсилення одиниця).

Наприкінці використовується функція $sysn = connect(sln, q, input, output)$

Як видно з Таблиці 2.2, відповіді, отримані обома способами, збігаються.

Використана у прикладі функція *minreal* спрощує, якщо це можливо, вираз (скорочує однакові нулі та полюси). Наведені у лівій частині («Перший спосіб») записи вигляду %або... відповідають спрощеному варіанту, без використання команд *series* та *feedback*.

Як ці програми, так і М-файли можна не набирати самостійно, а просто копіювати та вставляти у робочий простір (Workspace) після позначки >> та подальшого натискання кнопки Enter для виконання.

Зверніть увагу, що при роботі з передаточними функціями у формулах позначка «.» не використовується при виконанні операцій $*$, $/$ та $^$ на відміну від роботи зі скалярними змінними.

Таблиця 2.2 – Порівняння способів створення моделі

Перший спосіб	Другий спосіб
$k1=5;$ $k2=10;$ $t1=0.5;$ $t2=0.15;$ $s1=tf(1,1);$ $s2=tf([t1 \ 1],[t2 \ 1]);$ $s3=tf([k1],[t1 \ 1]);$ $s4=tf([k1*k2],[t1^2 \ t2^2 \ 1]);$ $s5=tf([k1],[t2 \ 1]);$ $ss1=s5/s4;$ $ss2=series(s3,s4);$ $\% \text{або } ss2=s3*s4;$ $ss3=feedback(ss2, s2, +1);$ $\% \text{або } ss3=ss2/(1+ss2*ss1);$ $ss4=feedback(ss3, ss1, -1);$ $\% \text{або } ss4=ss3/(1-ss3*s2);$ $ss5=series(s1,ss4);$ $\% \text{або } ss5=s1*ss4;$ $ss6=minreal(ss5)$	$k1=5;$ $k2=10;$ $t1=0.5;$ $t2=0.15;$ $s1=tf(1,1);$ $s2=tf([t1 \ 1],[t2 \ 1]);$ $s3=tf([k1],[t1 \ 1]);$ $s4=tf([k1*k2],[t1^2 \ t2^2 \ 1]);$ $s5=tf([k1],[t2 \ 1]);$ $sln=append(s1,s2,s3,s4,s5);$ $q=[2 \ 4 \ 0 \ 0; 3 \ 1 \ 2 \ -5; 4 \ 3 \ 0 \ 0; 5 \ 3 \ 0 \ 0];$ $input=1;$ $output=4;$ $sysn=connect(sln,q,input,output);$ $sss1=tf(sysn);$ $sss2=minreal(sysn)$
<i>Transfer function:</i> $\frac{2000s + 1.333e004}{s^4 + 8.76s^3 + 351.4s^2 - 6601s - 1.19e004}$	

Для кращого розуміння теми скопіюйте з наведеного вище тексту програми, виконані першим (тут краще обрати спрощений варіант після `%або...`) і другим способами спочатку у М-файл (як його створити, написано вище), а потім у Робочий простір MATLAB. Запустіть по черзі обидва варіанти і порівняйте результати. Побудуйте перехідні процеси за допомогою команди `step`, змініть декілька разів параметри `k` та `t` ланок, що у складі структурної схеми, самі рівняння ланок (візьміть інші ланки). Оформіть результати з використанням різних, наведених вище, засобів (команд `plot`, `subplot`, `ltiview` тощо).

Увага! При копіюванні у М-файл та подальшому запуску програми можливі помилки у передачі деяких символів, що пов'язано з особливостями Word (у вікні Command Window

з'явиться повідомлення). Поверніться у редактор М-файлу та перенаберіть символи рожевого кольору. Вони повинні стати бузковими (найчастіше це символи «'» або деякі літери). Знов скопіюйте текст програми і розмістіть його у Command Window.

Контрольні питання

1. Форми опису моделей систем, зв'язок між ними, ієрархія [2].
2. Засоби перегляду динамічних характеристик систем - командний та з використанням браузерів.
3. Назвіть основну різницю між двома засобами розрахунку багатоконтурних систем керування.

3 ПРИКЛАДИ РОЗВ'ЯЗАННЯ ДЕЯКИХ ЗАДАЧ З АНАЛОГОВОЮ ФОРМОЮ СИГНАЛУ

3.1 Дослідження впливу гнучкого зворотного зв'язку на якісні показники САК

На рисунку 3.1 наведена структурна схема системи, що досліджується, де x - вхідний сигнал, $x_l = x - y - y_l$, ГЗЗ - гнучкий зворотний зв'язок [4].

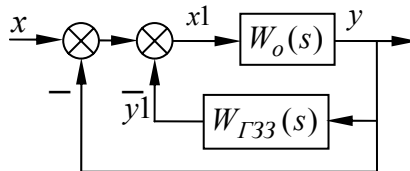


Рисунок 3.1 – Структурна схема системи

В цій системі

$$W_o(s) = \frac{K_{op}}{(T_1s + 1) \cdot (T_2s + 1) \cdot (T_3s + 1)} \quad (3.1)$$

$$W_{ГЗЗ}(s) = \frac{K_{ГЗЗ}s}{(T_{ГЗЗ}s + 1)} \quad (3.2)$$

Гнучкий зворотній зв'язок – це зв'язок, який діє лише під час перехідного процесу і не впливає на усталене значення сигналу на виході системи. Ясно, що для цього у чисельнику ПФ такої ланки повинен бути вираз, що вміщує s , оскільки цей вираз відповідає похідній, і як тільки вона дорівнює нулю-процес встановився. В нашому випадку у ланцюгу зворотного зв'язку стоїть

ланка $\frac{Ks}{(T_{ГЗЗ}s + 1)}$. яка є реальною ланкою, що диференціює. Зна-

менник у цьому виразі відповідає інерційній ланці, що має добрі фільтруючі властивості. Ця ланка введена для підвищення завадо

захищеності, яка погіршується за рахунок ланки $K_{ГЗЗ}S$ (див. (3.2)).

Користуючись наведеними у попередніх підрозділах правилами, складемо приблизний варіант програми для дослідження впливу зворотнього зв'язку на обраний об'єкт керування. Як і раніше, відпрацюємо спочатку програму у інтерактивному режимі, а після цього збережемо її у М-файлі.

```
Kop= 4;T1= 0.1 ;T2=0.2 ;T3=0.2 ; st=3;t=[0:0.1:st];
%Ввести свої дані T, Kop, st
Kgzz1= 0; Kgzz2=0.05 ; Kgzz3= 0.1 ;Tgzz= 0.05;
%Три значення коеф.передачі та стала часу ГЗЗ
Wo1=tf(1,[T1 1]; Wo2=tf(1,[T2 1]; Wo3=tf(Kop,[T3 1];
%Вв. Заменника ПФ об'єкту керування
Wo=Wo1*Wo2*Wo3; % ПФ об'єкту керування
Wgzz1=tf([Kgzz1 0],[ Tgzz 1]);
Wgzz2=tf([Kgzz2 0],[ Tgzz 1]);
Wgzz3=tf([Kgzz3 0],[ Tgzz 1]);
% ПФ дифференціуючої ланки ГЗЗ для Kgzz1,Kgzz2,Kgzz3
Wz1=Wo/(1+Wo* Wgzz1);
%Урахування локального зворотного зв'язку
Wz2=Wo/(1+Wo* Wgzz2);
Wz3=Wo/(1+Wo* Wgzz3);
Wwse1=Wz1/(1+Wz1); %Урахування головного зворотного
зв'язку
Wwse2=Wz2/(1+Wz2);
Wwse3=Wz3/(1+Wz3);
step(Wwse1,Wwse2,Wwse3,t),grid %Побудова перехідного
процесу.
```

Якщо ви правильно все зробили та побачили відповідний графік згідно останній команді *step*, ви легко зможете зробити висновки щодо впливу гнучкого зворотного зв'язку на якісні характеристики у часовій області. Змініть диференційну ланку, наприклад, на інерційну. Тобто зробіть зв'язок жорстким та порівняйте результати. Для цього у чисельнику передатних функцій $Wgzz1$, $Wgzz2$, $Wgzz3$ заберіть 0 (тобто отримайте $Wgzz1 = tf([Kgzz1],[Tgzz 1])$) та перезапустіть програму.

3.2 Дослідження впливу чистого запізнювання на якісні показники САК (pade)

Системою з запізнюванням називають систему, в якій реакція на виході відстає від вихідного сигналу на деякий час τ .

Запізнювання суттєво впливає на стійкість. По суті це проміжок часу між моментом, коли у якомусь місті системи відбудеться деяка подія та моментом, коли ця подія виявить себе десь у іншому місті (рисунок 3.2).

Ідеальне запізнювання характеризується передатною функцією вигляду $Ga(s) = e^{-st}$ (T – час запізнювання). $T = \frac{d}{V}$ (d – відстань, V – швидкість)

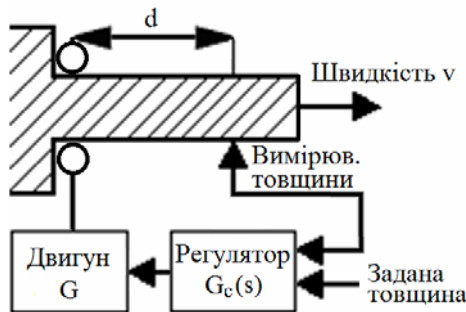


Рисунок 3.2 – Керування прокатним станом

Критерій Найквіста залишається діючим і задля таких систем, оскільки e^{-st} не призводить до появи додаткових полюсів або нулів розімкненої системи та тільки вносить додатковий від'ємний зсув без впливу на амплітудні характеристики.

Відстань d намагаються зменшити, але вона завжди буде.

Передатна функція розімкненої системи у нашому випадку має вигляд [7]:

$$G(s) \cdot G_c(s) \cdot e^{-sT}$$
$$G \cdot G_c(j\omega) \cdot e^{-j\omega T}$$

З'являється додатковий фазовий зсув - ωT рад.
Розглянемо як це впливу на частотні характеристики

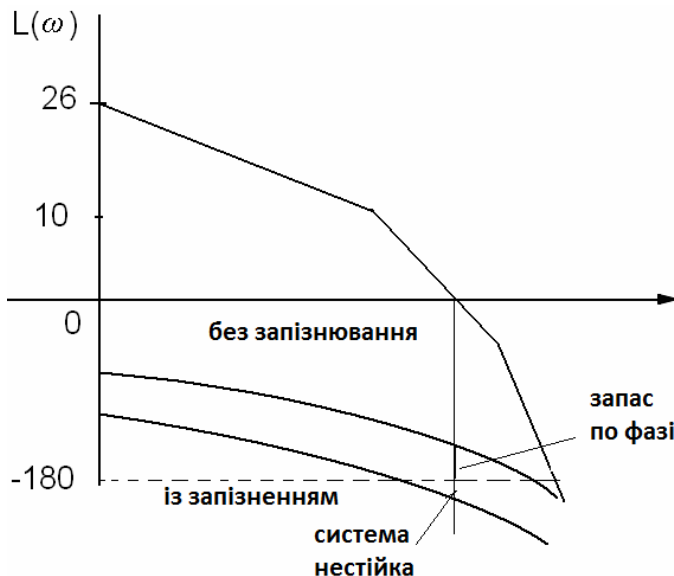


Рисунок 3.3 – Вплив запізнювання на частотні характеристики

При запізненні фазовий зсув переходить за 180^0 – система не стійка, тому, щоб мати достатній запас по фазі, потрібно зменшити коефіцієнт підсилення (тобто криву $L(\omega)$ опустити).

Таким чином, ланка запізнювання завжди вносить від'ємний фазовий зсув та погіршує стійкість замкненої системи, оскільки запізнювання об'єктивно майже завжди суттєво задля забезпечення стійкості. При цьому доводиться зменшувати коефіцієнт підсилення. Розплатою за це є неминуче збільшення встановленої похибки.

Розглянемо на конкретному прикладі.

Структурна схема САК з наявністю чистого запізнювання в об'єкті приведена на рисунку 3.4 (K_p – коефіцієнт підсилення регулятора) [4].

Передатна функція об'єкту має вигляд:

$$W(s) = \frac{K}{(T_1s+1) \cdot (T_2s+1)} \cdot e^{-s\tau} \quad (3.3)$$

де τ - значення запізнювання, с.

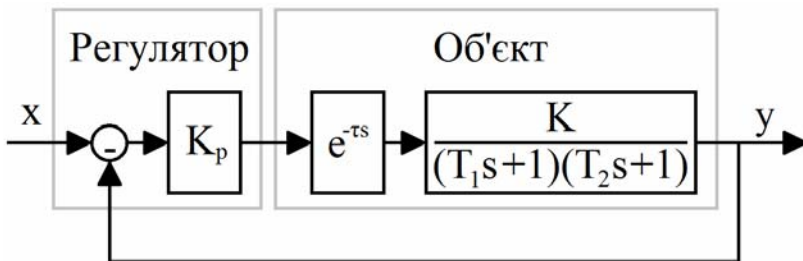


Рисунок 3.4 – Структурна схема САК при наявності чистого запізнювання в об'єкті

Об'єкт може бути описаний диференціальним рівнянням з аргументом, що відповідає за запізнювання:

$$T_1 * T_2 * y'' + (T_1 + T_2) * y' + y = K * x(t - \tau).$$

Наявність запізнювання в розімкненій системі приводить до зростання часу перехідного процесу на значення τ .

Наявність запізнювання в об'єкті для замкненої системи приводить до зростання коливальності та тривалості перехідного процесу.

Як вже відмічалось, наявність запізнювання в об'єкті не змінює вигляду АЧХ, але змінює ФЧХ - додає додатковий фазовий зсув $\varphi(\omega) = -\omega\tau$, що, зокрема, знижує запас сталості по фазі γ , який визначається за формулою:

$$\gamma = 180^\circ - |\varphi(\omega_c)|,$$

де $\varphi(\omega_c)$ - фаза на частоті зрізу ω_c , тобто на частоті, де коефіцієнт передачі системи дорівнює одиниці (або $20 \lg(A(\omega_c)) = 0$ дБ).

Для того, щоб врахувати запізнення у програмі MATLAB треба привести передатну функцію об'єкту

$$W(s) = \frac{K}{(T_1s+1) \cdot (T_2s+1)} \cdot e^{-s\tau} \text{ до такого вигляду, щоб чисельник і}$$

знаменник мали вигляд алгебраїчних поліномів . Для цього використаємо апроксимацію e^{-st} за допомогою функції *pade*. Вона має такий вигляд: $[num,den] = pade(T,n)$. Тут T – запізнювання у часі, n – порядок апроксимації, а num,den – поліноми чисельника та знаменника, відповідно. Наприклад, $T=1\text{с}$ і ми бажаємо отримати апроксимацію другого порядку ($n=2$), тоді використання функції *pade* (1,2) дасть результат:

$e^{-s}=(s^2-6s+12)/(s^2+6s+12)$, який ми підставимо замість e^{-st} у передатну функцію W .

Примірник програми досліджень впливу запізнення для трьох значень часу запізнення τ , з побудовою графіку залежності запасів стійкості від цих значень:

```
Kp=11; Kop=0.1; K=Kp*Kop;T1=0.2; T2=0.2;
tau1=0; tau2=0.5*(T1+T2); tau3=(T1+T2);st=4; t=[0:0.1:st ];
%vvod danix
W1=tf(1, [T1 1]); W2=tf(K, [T2 1]);W=W1*W2;
%vvod peredatochnoy funkzii objekta
[ num1,den1]=pade(tau1,2);
%perevod zadergki v formu vida polnomov dlja zadergki tau1
Wz1=tf(num1,den1);%peredatochnaja funkzija zadergki
Wposl1=W* Wz1;
%peredatochnaja funkzija razomknutoj zepi dlja zadergki tau1
Wzamz1= Wposl1/(1+ Wposl1);
%peredatochnaja funkzija zamknutoj zepi dlja zadergki tau1
[ num2,den2]=pade(tau2,2);%
Wz2=tf(num2,den2);%
Wposl2=W* Wz2;%
Wzamz2= Wposl2/(1+ Wposl2);
[ num3,den3]=pade(tau3,2);%
Wz3=tf(num3,den3);%
Wposl3=W* Wz3;%
Wzamz3= Wposl3/(1+ Wposl3);%
step(Wzamz1, Wzamz2, Wzamz3,t),grid
% stroim perehodnij procesdlja wseh
figure
bode(Wposl1, Wposl2, Wposl3), grid
%LogCastotHarakteristiki dlja wseh tau
figure
```

```

[zap_po_mod1,f_zm1,zap_po_faz1,wsr1]=margin(Wposl1);
% вывод запасов длja tau1
[zap_po_mod2,f_zm2,zap_po_faz2,wsr2]=margin(Wposl2);
% вывод запасов длja tau2
[zap_po_mod3,f_zm3,zap_po_faz3,wsr3]=margin(Wposl3);
%вывод запасов длja tau3
Zapas_po_Faze=[ zap_po_faz1, zap_po_faz2, zap_po_faz3];
%massiv danih po zapasam
TAU=[tau1, tau2, tau3];%massiv danih po zapiznennju
plot(TAU, Zapas_po_Faze),grid%вывод графика запас по фазе от
zapiznennja

```

Примітка. Для підвищення точності визначення частоти зрізу та запасу з фази треба збільшити масштаб зображення логарифмічних характеристик наступним чином:

- натиснути ПКМ, обрати *Zoom*, а в цьому *In-X* (збільшення масштабу по осі абсцис)
- ЛКМ виділяємо область збільшення (у якій ЛАЧХ перетинає вісь абсцис).

Скопіюйте програму у М – файл, відредагуйте, якщо потрібно (змінить числові параметри, *tau* або порядок апроксимації) та запустіть через Command Window.

3.3 Дослідження САК з ПІД – регулятором

Структурна схема САК з ПІД - регулятором показана на рисунку 3.5[4], де g - вхідний сигнал, $x = g - u$ - помилка системи, u - керуючий сигнал, $W_{\text{під}}(s)$ - передатна функція ПІД –регулятора, $W_o(s)$ - передатна функція посилювача потужності, виконавчого органу та об'єкта керування.

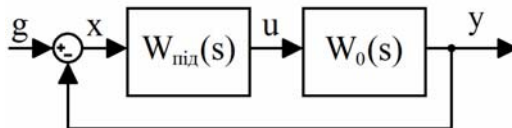


Рисунок 3.5 – Структурна схема САК з ПІД - регулятором

Передатна функція ідеального ПД - регулятора має вигляд:

$$Wp(s) = K_{II} + \frac{K_I}{s} + K_D s \quad (3.4)$$

При цьому перше складове являє собою пропорційну ланку(П), друге складове - інтегруючу ланку (І), третє складове - ланка(Д), що диференціює. На підставі цього регулятор має назву ПД - регулятор. Існують також П-, ПІ-, та ПД- регулятори, структура яких складається відповідно до відомостей , наведених вище.

Структурна схема ідеального ПД - регулятора має вигляд (рисунок 3.6 а)):

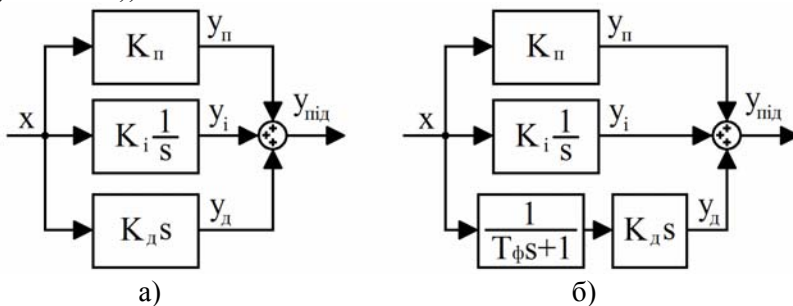


Рисунок 3.6 – Структурні схеми а) ідеального та б) реального ПД - регулятора

Для покращення заводо захищеності на практиці третє складове у змінюють на ланку, що диференціює з уповільненням . Тоді реальний ПД - регулятор має вигляд як на рисунку 3.6 б) і формула (3.4) приймає вигляд:

$$Wp(s) = K_{II} + \frac{K_I}{s} + \frac{K_D s}{T_\phi s + 1} \quad (3.4a)$$

3.3.1 Розрахунок ПІД-регулятора методом настроювання на модульний оптимум [6].

Розрахунок коефіцієнтів ПІД-регулятора у багатьох випадках проводять з огляду на те, щоб передатна функція замкненої системи та перехідний процес мали *наперед заданий вигляд*.

Наприклад, розглянемо формування перехідного процесу в контурі регулювання швидкості двигуна постійного струму з ПІД-регулятором (рисунок 3.7) за так званим модульним оптимумом [4]. При цьому *бажана* передатна функція розімкненої системи має вигляд:

$$W_{rho}(s) = \frac{1}{2T_{\mu}s(T_{\mu}s + 1)} \quad (3.5)$$

а замкненої:

$$W_{zmo}(s) = \frac{1}{2T_{\mu}^2s^2 + 2T_{\mu}s + 1} \quad (3.6)$$

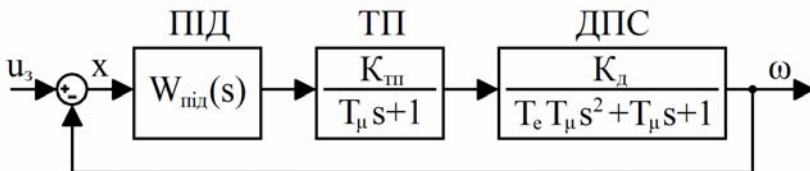


Рисунок 3.7 – Контур регулювання швидкості двигуна постійного струму з ПІД-регулятором

Перехідний процес в системі при цьому має вигляд (рисунк 3.8):

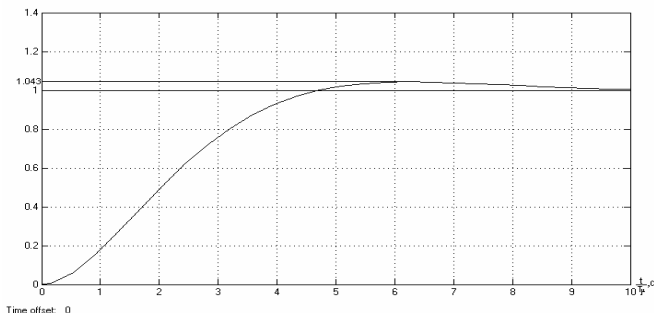


Рисунок 3.8 – Перехідний процес в системі

В такому випадку компенсуються постійні часу T_e та T_M , а T_μ є такою, що не компенсується. Для забезпечення цього передатна функція ПІД-регулятора згідно рисунка 3.8 повинна мати вигляд:

$$W_{pid} = \frac{W_{pmo}}{W_0} \quad (3.7)$$

Тут $W_0 = W_{tp} * W_{dps}$ згідно структурної схеми на рис. 3.8 і після введення даних з рисунку

$$W_{pid}(s) = \frac{T_M T_e s^2 + T_M s + 1}{2T_\mu K_{TP} K_\partial s} \quad (3.8)$$

Тобто після порівняння з формулою 3.4 і рисунками 3.6 та 3.7 отримаємо:

$$K_D = \frac{T_e T_M}{2T_\mu K_{TP} K_\partial}; K_I = \frac{1}{2T_\mu K_{TP} K_\partial}; K_P = \frac{T_M}{2T_\mu K_{TP} K_\partial}, \quad (3.9)$$

при $K_d = \frac{1}{c\Phi}$;

Проведемо розрахунки згідно заданим у [4] технічним даним.

Таблиця 3.1 – Таблиця варіантів технічних даних

Дані/Варіанти	1	2	3	4	5	6	7
Тип двигуна	П102	П91	П111	П112	П111	П52	П72
Тип тиристорного перетворювача	КТЭУ	КТЭУ	КТЭУ	КТЭУ	КТЭУ	КТЭУ	КТЭУ
Стала часу ТП T_m ,с	0.003	0.003	0.0035	0.003	0.004	0.0035	0.003
Коефіцієнт підсилення, K_{TP}	27.27	27.675	51.3	51.3	51.3	27.68	27.68
Електромагнітна стала часу, T_e	0.023	0.037	0.032	0.0116	0.079	0.014	0.014
Електромеханічна стала часу, T_M	0.115	0.183	0.067	0.04	0.11	0.142	0.112
Коефіцієнт двигуна, $C\Phi$	1.953	1.91	5.23	2.688	2.68	1.27	1.84

Ввести дані з таблиці 3.1 згідно варіанту можна було так, як робили раніше, через оператор присвоювання (тобто $T_m=0.003$ і таке далі), а можна з використанням оператору введення *input*, дія якого зрозуміла з наступного(такий метод універсальний і відповідає будь-якому варіанту, потрібно тільки ввести *свої дані*):

```
Tu=input('Введіть сталу часу ТП Tu ')
Ktp=input('Коефіцієнт підсилення, Ktp ')
Te=input('Електромагнітна стала часу, Te ')
Tm=input('Електромеханічна стала часу, Tm ')
CF=input('Коефіцієнт двигуна, CF ')
% Знаходження коефіцієнтів ПІД-регулятора
Kd=1/CF;
K1=Tm/(2*Tu*Ktp*Kd);%K пропорційної складової з ф-ли
(3.9)
K2=1/(2*Tu*Ktp*Kd);%K інтегруючої складової з ф-ли (3.9)
K3=Te*Tm/(2*Tu*Ktp*Kd);%K диференціюючої складової з
ф-ли(3.9)
%Передатна функція ПІД-регулятора
```

```
Wpid=tf([K3 K1 K2], [1 0])
% згідно ф-лі (3.4) після приведення до загального знамен-
ника
```

```
%Передатні функції ТП и ДПС
```

```
Wtp=tf([Ktp], [Tu 1]);
```

```
Wdps=tf([Kd], [Te*Tm Tm 1]);
```

```
%-----
```

```
Wo=Wtp*Wdps;% послідовне з'єднання ТП та ДПС
```

```
Wpaz=Wpid*Wo;% ПФ розімкненої системи
```

```
Wzam=feedback(Wpaz,1,-1);% ПФ замкненої системи
```

```
step(Wzam);grid % графік перехідного процесу з ПІД-
регулятором
```

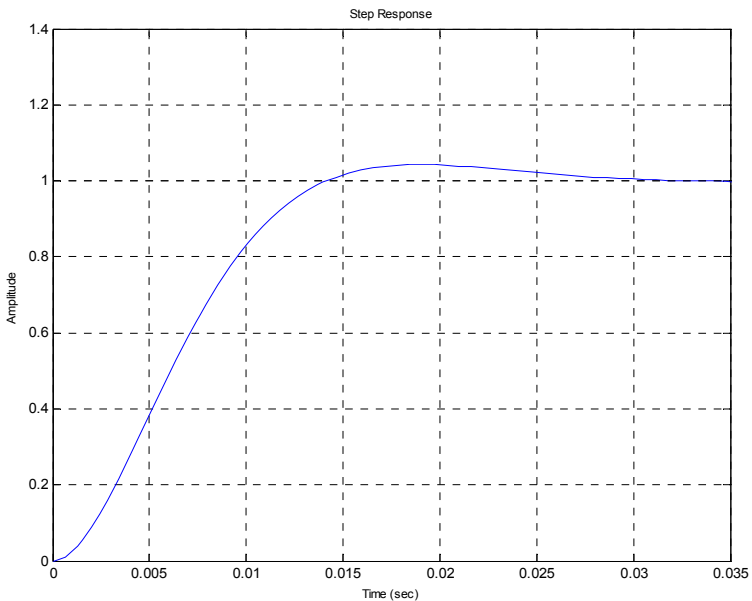


Рисунок 3.9 – Перехідний процес у контурі швидкості з ПІД-регулятором

Порада: порівняйте задля наочності, як буде виглядати перехідний процес без ПІД-регулятора, виведіть на одному графіку з перехідним процесом з регулятором. Для цього поверніться до

структурної схеми на рисунку 3.7. Щоб не враховувати ПІД-регулятор, просто задайте його передаточну функцію рівною 1, тобто $W_{pid}=tf(1,1)$, внесіть ці зміни у скопійований нижче фрагмент своєї програми зі зміною назви W_{paz} на W_{paz1} та W_{zam} на W_{zam1} і виведіть `step(Wzam, Wzam1),grid`

Самостійно порівняйте у програмі частотні характеристики для обох випадків.

Більш детальніше вплив коефіцієнтів K_i , K_d , K_p можна, якщо використати програму, наведену далі.

Надана далі програма необов'язкова при першому читанні посібника, вона призначена для аналізу окремого впливу інтегральної, диференційної та пропорційної частин ПІД – регулятора.

3.3.2 Програма для аналізу впливу зміни коефіцієнтів K_i , K_d та K_p

%Як змінюється перехідний процес при зміні K_3

```
clear Wpid Wtp Wdps Wprivoda W1(i) T K3 i
s=zpk('s')
Tu=0.003;
Ktp=27.27;
Te=0.023;
Tm=0.115;
CF=1.953;
Kd=1/CF;
K1=Tm/(2*Tu*Ktp*Kd);
K2=1/(2*Tu*Ktp*Kd);
K3=0:0.011:0.033;
figure
for i=1:1:3
    Wpid(i)=K2*((K3(i)/K2)*s^2+(K1/K2)*s+1)/s;
    Wtp=Ktp/(Tu*s+1);
    Wdps=Kd/(Te*Tm*s^2+Tm*s+1);
    Wprivoda=Wtp*Wdps;
    W1(i)=Wpid(i)*Wprivoda;
    W2(i)=feedback(W1(i),1,-1)
    hold on
    step(W2(i));grid
```

```

end
%-----
% Як змінюється перехідний процес при зміні K2
%-----
clear Wpid Wtp Wdps Wprivoda W1(i) K3 K2 i
K3=Te*Tm/(2*Tu*Ktp*Kd);
K2=4:4:24;
figure
for i=1:1:4
    Wpid(i)=K2(i)*((K3/K2(i))*s^2+(K1/K2(i))*s+1)/s;
    Wtp=Ktp/(Tu*s+1);
    Wdps=Kd/(Te*Tm*s^2+Tm*s+1);
    Wprivoda=Wtp*Wdps;
    W1(i)=Wpid(i)*Wprivoda;
    W2(i)=feedback(W1(i),1,-1)
    hold on
    step(W2(i));grid
end
%-----
% Як змінюється перехідний процес при зміні K1
clear Wpid Wtp Wdps Wprivoda W1(i) K1 i
K3=Te*Tm/(2*Tu*Ktp*Kd);
K2=1/(2*Tu*Ktp*Kd);
K1=0:0.5:1.5;
figure
for i=1:1:3
    Wpid(i)=K2*((K3/K2)*s^2+(K1(i)/K2)*s+1)/s;
    Wtp=Ktp/(Tu*s+1);
    Wdps=Kd/(Te*Tm*s^2+Tm*s+1);
    Wprivoda=Wtp*Wdps;
    W1(i)=Wpid(i)*Wprivoda;
    W2(i)=feedback(W1(i),1,-1)
    hold on
    step(W2(i));grid
end

```

Зробіть висновки про вплив складових частин в ПІД-регуляторі на вигляд перехідного процесу в системі.

3.3.3 Розрахунок ПІД регулятора за допомогою браузера PIDTool (версія MATLAB R2012a)

Альтернативним методом розрахунку ПІД-регулятора є використання браузера PIDTool, що з'явився у останніх версіях MATLAB (рисунок 3.10).

Попередньо у М-файлі задамо, наприклад, параметри ДПТ та ТП у вигляді передатної функції і викличемо **PIDTool**, задавши назву передатної функції та вигляд необхідного регулятора (*pi*, *pid* і т.д.). Відповідна програма наведена нижче:

```
Tu=4; % введення початкових даних
```

```
Tm=0.0254;Te=0.0153;Ktp=22;
```

```
Kd=1/1.23;
```

```
a=tf(Ktp,[Tu 1]); % отримання передатних функцій
```

```
b=tf(Kd,[Te*Tm Tm 1]);
```

```
c=feedback(series(a,b),1,-1) % отримання ПФ замкненої системи
```

```
pidtool(c,'pid') % виклик браузера
```

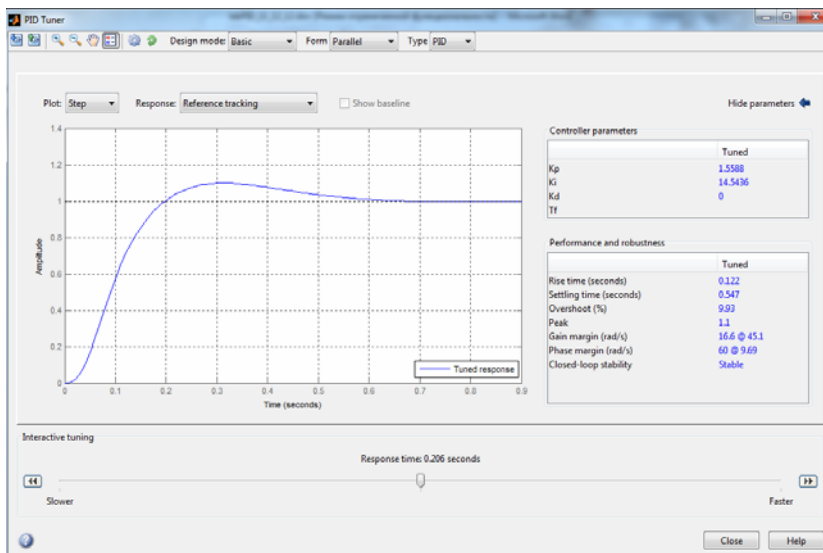


Рисунок 3.10 – Загальний вигляд PIDTool

Змінюючи параметр *Response time*, спостерігаємо зміну коефіцієнтів ПД-регулятора та параметрів перехідного процесу. Також можна позначити на перехідному процесі час наростання, пік перехідного процесу, коридор входження у зону $\pm \delta$ відносно сталого значення. Для цього необхідно натиснути ПКМ на графіку та у характеристиках відмітити необхідні.

У **PIDTool** передбачено 2 вхідних сигнали у пункті *Plot* (рисунк 3.10). У *Response*, що знаходиться правіше *Plot*, можна обрати різні види заданої системи.

Також можна дослідити 2 типи зміни параметрів в *Design mode* у рядку меню (рисунк 3.11).

Задамо в *Design mode* пункт *Extended* (розширений, на відміну від *Basic* (основний)) та задамо за допомогою бігунків нові параметри передатної функції. Результат роботи показаний на рисунку 3.12.

3.3.4 Моделювання ПД регулятора за допомогою MATLAB Simulink (версія 7.04)

У складі MATLAB Simulink є пакет **Simulink Response Optimization**, де обираємо блок **Signal Constraint**, за допомогою якого можна виконати оптимізацію параметрів ПД-регулятора, якщо мається модель об'єкта керування [2] (рисунк 3.13).

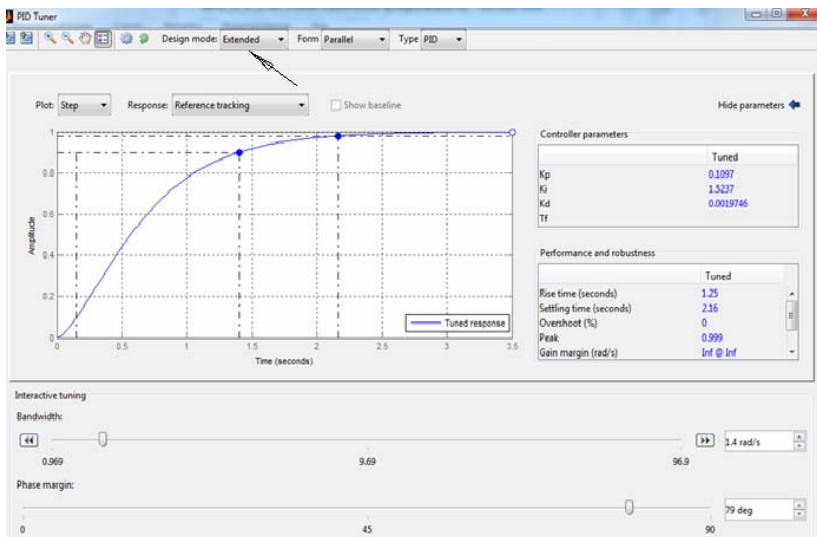


Рисунок 3.11– Змінена передатна функція

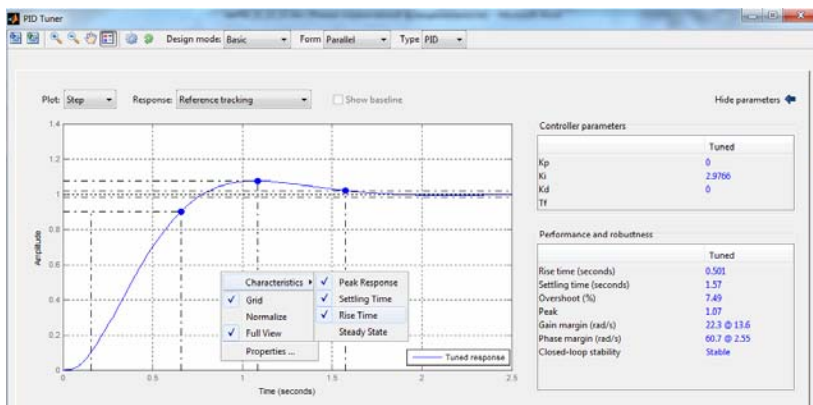


Рисунок 3.12 – Перехідний процес для передатної функції, зміненої за допомогою параметрів Extended

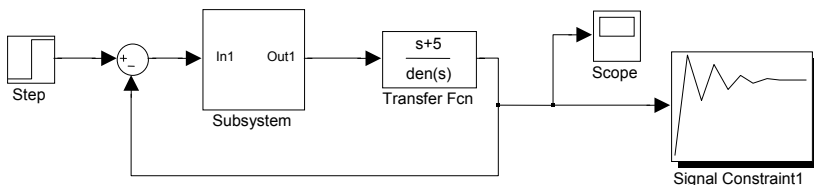


Рисунок 3.13 – Модель для оптимізації

Передатна функція може, наприклад, мати такий вигляд:

$$\frac{s+5}{2s^3+5s^2+3s+1}.$$

Підсистема (Subsystem на рисунку 3.13) має вигляд:

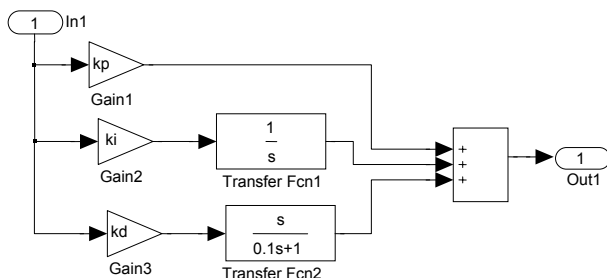


Рисунок 3.14 – ПІД-регулятор

Зверніть увагу, що у підсилювачах Gain1-Gain3 вводимо числові значення, а позначення коефіцієнтів регулятора kp , ki , kd .

Порядок роботи такий:

- в **Command Window** вводимо початкові значення $kp=1$, $ki=1$, $kd=1$ і запускаємо модель Simulink;
- відкриваємо блок **Signal Constraint** та натискаємо **Plot current response** (позначено стрілкою на рисунку 3.15), щоб з'явився перехідний процес заданої передатної функції (рисунок 3.16);
- у рядку меню вікна рисунку 3.16 натискаємо **Optimization** і

обираємо пункт **Tuned Parameters**. У наступному вікні натискаємо *Add* та виділяємо k_p , k_i , k_d (рисунок 3.17), додаємо їх натиснувши *Ok*. Підтверджуємо зміни у **Tuned Parameters** натисканням *Ok* (рисунок 3.18)

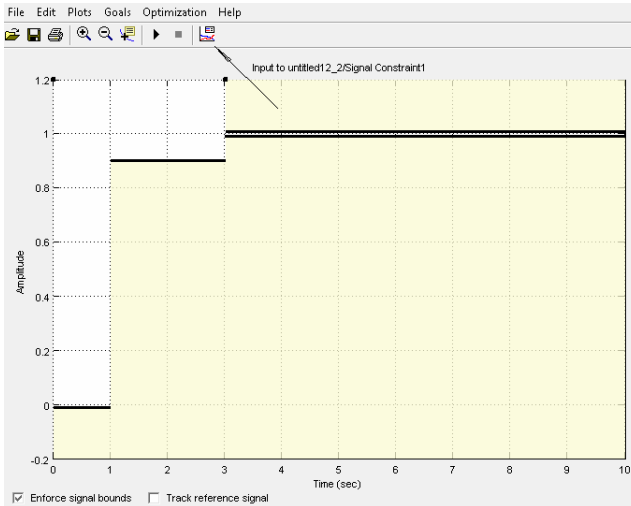


Рисунок 3.15 – Загальний вигляд параметрів Signal Constraint

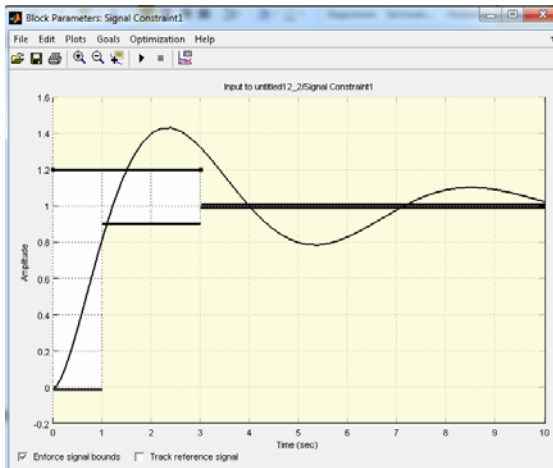


Рисунок 3.16 – Вигляд перехідного процесу у блоці Signal Constraint після використання Plot current response

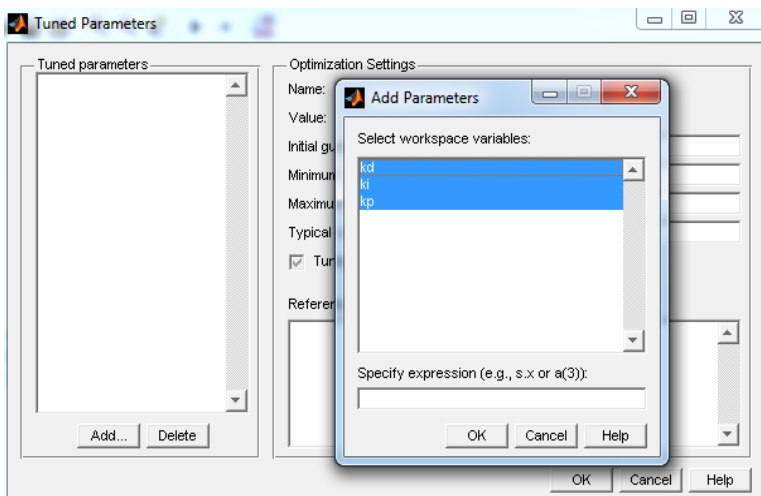


Рисунок 3.17 – Додавання коефіцієнтів ПД-регулятора

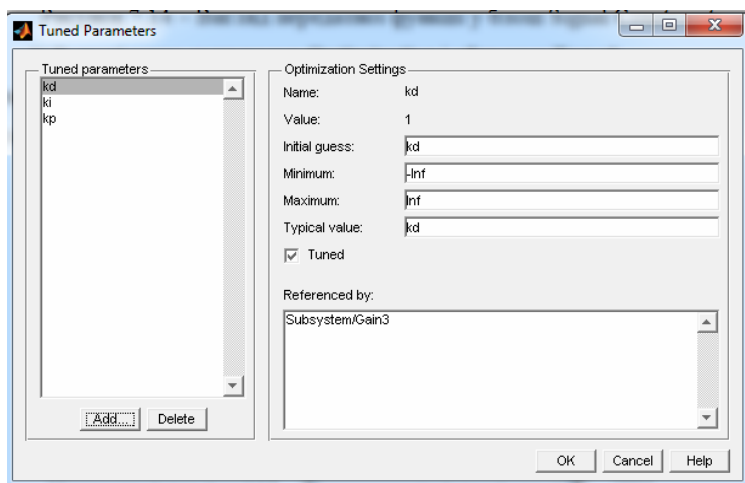


Рисунок 3.18 – Підтвердження змін у Tuned Parameters

– запускаємо блок **Signal Constraint** натисканням кнопки , спостерігаємо підбір оптимальних параметрів перехідних процесів, а також зміни передатної функції (рисунки 3.19, 3.20).

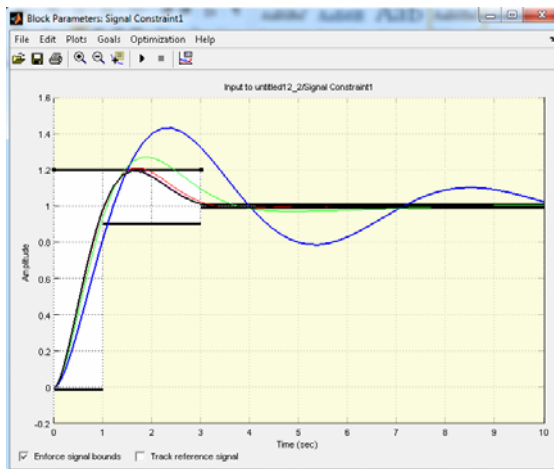


Рисунок 3.19 – Зміни перехідних процесів

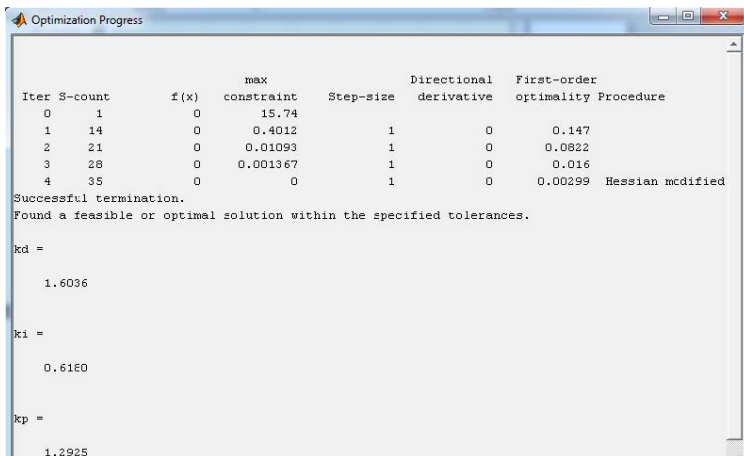


Рисунок 3.20 – Параметри оптимізованого перехідного процесу та коефіцієнти ПД-регулятора

– у блоці **Signal Constraint** можна змінити параметри перерегулювання та коридор входження у зону $\pm \delta$ відносно сталого значення і отримати нові параметри оптимізації, перетягнувши відповідні границі (рисунки 3.21, 3.22).

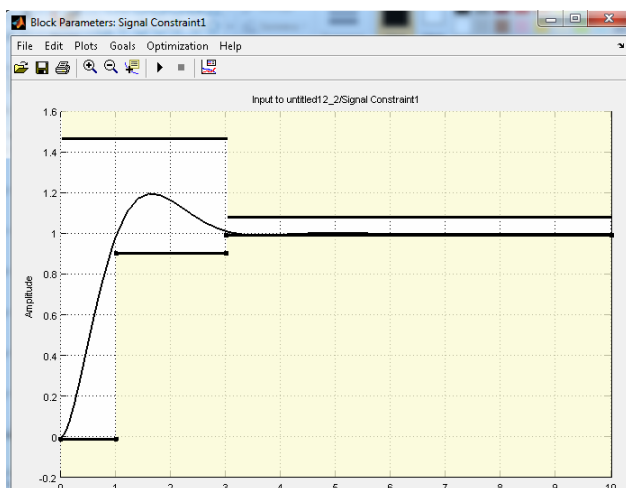


Рисунок 3.21 – Подальші зміни перехідного процесу

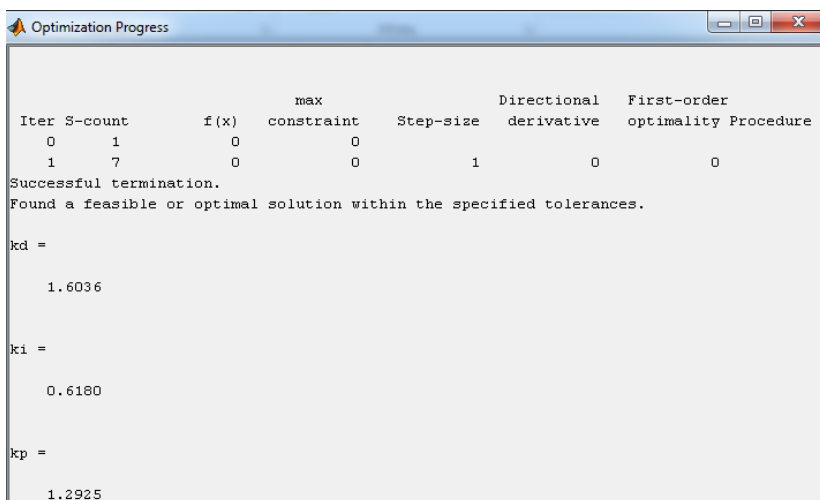


Рисунок 3.22 – Нові параметри оптимізованого перехідного процесу та коефіцієнти ПД-регулятора

3.4 Розрахунок системи з двома масами (векторно-матрична модель)

У цьому розділі на прикладі розробки моделі керування дисководом комп'ютера (позиціонування головки, що зчитує, точно на задану доріжку) в змінних стану[7], врахуємо ефект вигину пластини, який буде зведений до системи з двома масами.

Розглянемо конструкцію пристрою, що зчитує, зображену нижче на рисунку 3.23.

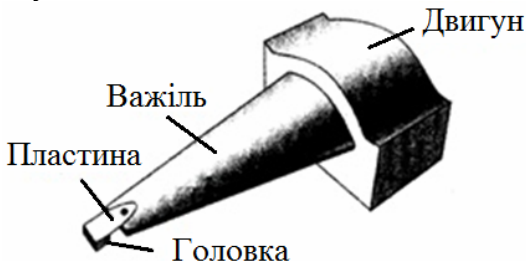


Рисунок 3.23 – Головка дисковода, що зчитує.

Оскільки для швидкого переміщення головки необхідно мати малу масу важеля, то нам доведеться врахувати ефект вигину пластини, виготовленої з дуже тонкої пружної сталевих стрічки. Ще раз відзначимо, що нам необхідно з високою точністю управляти положенням головки $y(t)$, як це показано на рисунку 3.24 та врахувати, що сигнал похибки формується на підставі завданого номеру доріжки.

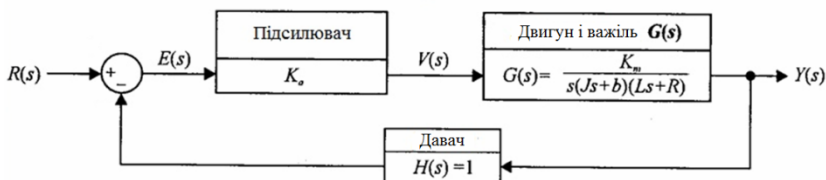


Рисунок 3.24 а) – Структурна схема системи дисковода, що зчитує дані



Рисунок 3.24 б) – Функціональна схема системи дисководу

Розробимо модель системи, що відповідає цим вимогам (рис. 3.25):

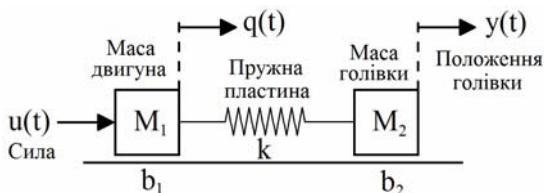


Рисунок 3.25 а) – Модель системи з двома масами і пружною пластиною;

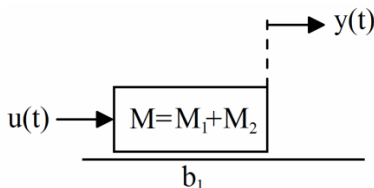


Рисунок 3.25 б) – Спрощена модель з жорсткою пластиною

Позначимо масу двигуна через M_1 , а масу голівки через M_2 . Вигин пластини будемо характеризувати за коефіцієнтом пружності k . Сила $u(t)$, що приводить до руху масу M_1 , створюється двигуном постійного струму. Якщо пластина є абсолютно жорсткою (не схильною до вигину), то ми отримаємо спрощену модель, зображену на рис. 3.25 б). Типові параметри цієї системи з двома масами наведені у таблиці 3.2 [7].

Таблиця 3.2 – Типові параметри системи з двома масами

Параметр	Позначення	Величина
Маса двигуна	M	20 г = 0.02 кг
Коефіцієнт пружності пластини	k	$10 \leq k \leq \infty$
Маса головки	M_2	0.5 г = 0.0005 кг
Положення головки	$y(t)$	переміщене, мм
Коефіцієнт тертя маси 1	b	$410 \cdot 10^{-3}$ кГ/м/с
Опір обмотки збудження	R	1 Ом
Індуктивність обмотки збудження	L	1 мГн
Постійна електродвигуна	K_m	125 Н•м/А
Коефіцієнт тертя маси 2	b	$4.1 \cdot 10^{-3}$ КГ/м/с

Спочатку ми отримаємо передатну функцію спрощеної моделі на рис. 3.25 6). Врахуємо, що загальна маса $M=M_1+M_2=20.5$ г = 0.0205 кг. Тоді ми маємо:

Отже,

$$M \frac{d^2 y}{dt^2} + b_1 \frac{dy}{dt} = u(t) \quad (3.10)$$

передатна функція моделі:

$$\frac{Y(s)}{U(s)} = \frac{1}{s(Ms + b_1)} \quad (3.11)$$

З урахуванням параметрів в таблиці 3.2 отримаємо:

$$\frac{Y(s)}{U(s)} = \frac{1}{s(0.0205s + 0.410)} = \frac{40}{s(s + 20)}$$

Структурна схема пристрою, що зчитує з урахуванням обмотки електродвигуна приведена на рис.3.26. При $R=1$ Ом, $L=1$ мГн и $K_m=125$ ми маємо:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{5000}{s(s + 20)(s + 1000)} \quad (3.12)$$

Векторно-матрична модель для цього випадку, побудована для диференційного рівняння другого порядку (3.10) буде мати вигляд:

```

k=10;
M1=0.02; M2=0.0005;
M=M1+M2;
b1=410e-03;b2=4.1e-03;
t=[0:0.001:1.5];
%Модель в змінних стану
A=[0 1 ; 0 -b1/M] ;
B=[0;1/M]; C=[ 0 1]; D=[0]; sys=ss(A,B,C,D);
%Обчислення перехідної характеристики
step(sys,t), grid
xlabel('Час(с)')
ylabel('Швидкість(м/с)')

```

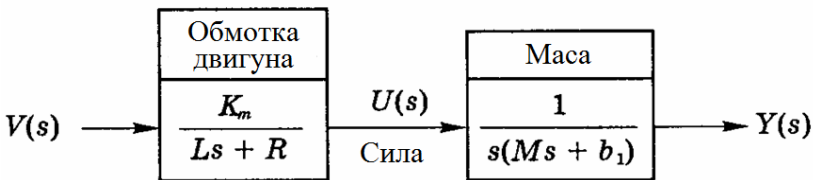


Рисунок 3.26 – Структурна схема пристрою, що зчитує, з жорсткою пластиною

Тепер отримаємо модель в змінних стану для системи з двома масами, зображеної на рисунку 3.25а). Диференційне рівняння має такий вигляд[7]:

$$\text{Для маси } M1: M_1 \frac{d^2 q}{dt^2} + b_1 \frac{dq}{dt} + k(q - y) = u(t) \quad (3.13)$$

$$\text{Для маси } M2: M_2 \frac{d^2 y}{dt^2} + b_2 \frac{dy}{dt} + k(y - q) = 0$$

Вибираємо в якості змінних стану

$$x_1 = q \quad \text{та} \quad x_2 = y.$$

$$\text{Тоді } x_3 = dq/dt; \quad x_4 = dy/dt$$

Рівняння стану у векторно-матричній формі [7]:

$$dx_i/dt = Ax + Bu, \quad (3.14)$$

де $x = [q \ y \ dq/dt \ dy/dt]^T$, $B = [0 \ 0 \ 1/M_1 \ 0]$,

$$A = \begin{bmatrix} 0 & 0 & 1 & 0; \\ 0 & 0 & 0 & 1; \\ -k/M_1 & k/M_1 & -b_1/M_1 & 0; \\ k/M_2 & -k/M_2 & 0 & -b_2/M_2 \end{bmatrix}.$$

Зауважимо, що виходом системи є $dy(t)/dt = x_4$. Крім того, нехтуючи індуктивністю обмотки двигуна ($L = 0$), маємо $u(t) = K_m v(t)$. Вибравши значення $k = 10$ і використовуючи інші параметри з табл. 3.2, отримаємо:

$$B = [0 \ 0 \ 50 \ 0]$$

$$A = [0 \ 0 \ 1 \ 0; 0 \ 0 \ 0 \ 1; -500 \ 500 \ -20.5 \ 0; 20000 \ -20000 \ 0 \ -8.2].$$

Реакція змінної dy/dt при $v(t) = 1B$, $t > 0$ зображена на рис. 3.27. Характер процесу є суттєво коливальним, тому ясно, що необхідно мати пластину з великою жорсткістю, тобто вибрати $k > 100$.

```
%Параметри моделі
k=10;
M1=0.02; M2=0.0005;
b1=410e-03; b2=4.1e-03;
t=[0:0.001:1.5];
%Модель в змінних стану
A=[0 0 1 0; 0 0 0 1; -k/M1 k/M1 -b1/M1 0; k/M2 -k/M2 0 -
b2/M2]
B=[0;0;1/M1;0]; C=[0 0 0 1]; D=[0]; sys=ss(A,B,C,D);
%Обчислення перехідної характеристики
step(sys,t), grid
xlabel('Час(c)')
ylabel('Швидкість(м/с)')
```

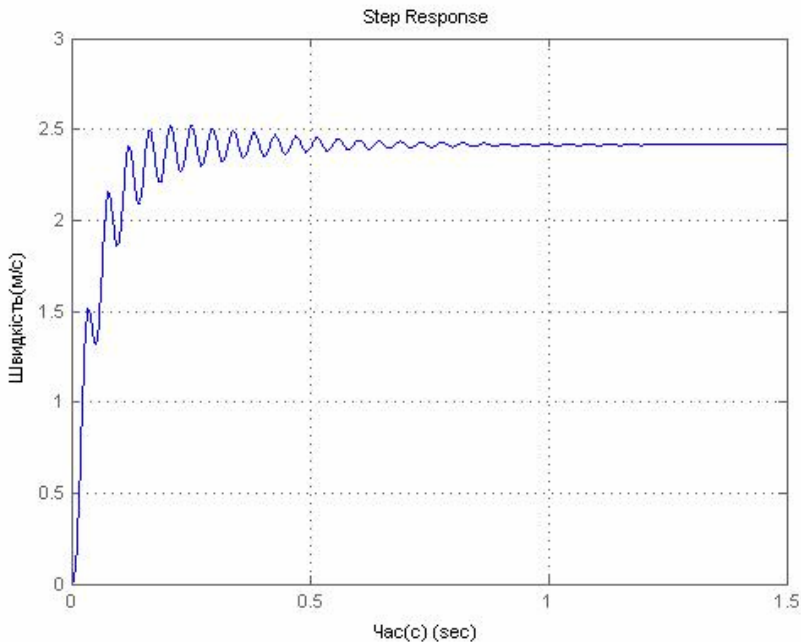


Рисунок 3.27 – Перехідний процес у моделі з двома масами при $k = 10$

Побудуйте для порівняння характеристику з $k=100$ та з однією сумарною масою на цьому ж графіку та зробіть висновки.

3.5 Аналіз систем керування креном літака за допомогою MATLAB/ Simulink

3.5.1 Аналіз системи керування креном гвинтового літака

Кожного разу, коли ви летите на літаку, ви на своєму досвіді відчуваєте усі переваги систем автоматичного керування, які допомагають пілотам вести машину. Точна математична модель, яка описує поведінку літака в повітрі, являє собою систему нелінійних диференціальних рівнянь зі змінними параметрами. Розглянемо спрощену модель динаміки літака у вигляді передатної фу-

нкції, яка пов'язує відхилення елеронів та кута крену літака, як це показано на рисунку 3.28 [7, 9].

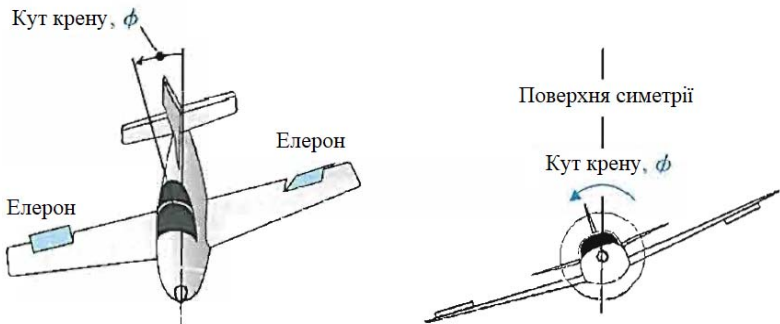


Рисунок 3.28 – Керування кутом крену літака за допомогою елеронів

На рис. 3.28 зображена замкнута система керування положенням літака в повітрі, ціль якої підтримувати кут крену, близького до нуля градусів, тобто $\phi_d=0$, в умовах непередбачених зовнішніх збуджень.



Рисунок 3.29 – Замкнена система керування положенням літака

Основними елементами системи керування є регулятор,

$$G_c(s) = K, \quad (3.15)$$

привод елеронів,

$$G_u(s) = \frac{10}{s+10}, \quad (3.16)$$

та літак,

$$G(s) = \frac{11.4}{s^2 + 1.4s}. \quad (3.17)$$

Для наших цілей використовуємо спрощену динамічну модель для автопілоту. Допустимо, є жорсткий літак з площиною симетрії. Будемо ігнорувати будь-які роторні гіроскопічні ефекти, обумовлені маніпуляціями на літальних апаратах (гвинти або турбіни). Ці припущення дозволяють відокремити поздовжньо-обертальний рух від бічного поворотного руху. Для цього припускаємо, що літак летить на низькій швидкості у стійкому положенні рівня крила. Наприклад, спроектуємо систему керування автопілотом, в якому перегулювання не буде перевищувати 20% від вхідного сигналу.

Проектуємо математичну модель, схема якої наведена на рисунках 3.29 та 3.30 а) [7, 9] та аналітично визначимо значення параметру підсилювача $K[1-2]$.

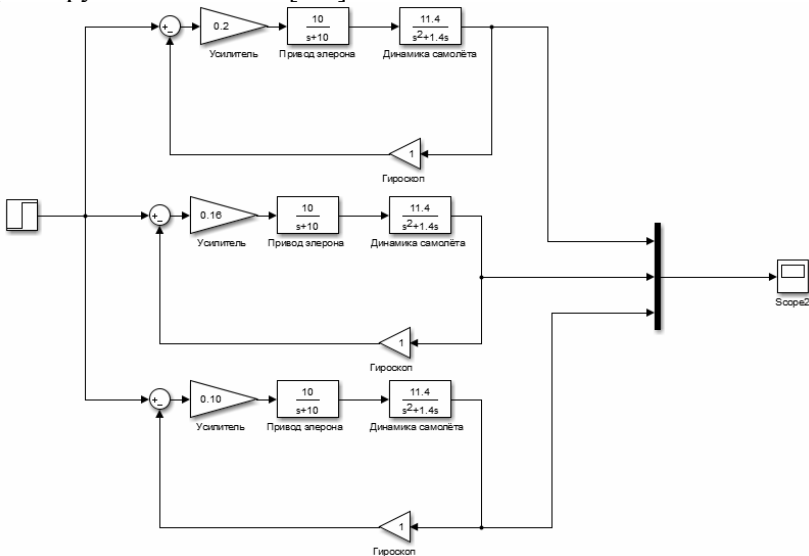


Рисунок 3.30 а) – Математична модель автопілоту

В результаті моделювання отримуємо перехідні процеси (рис. 3.30) з різними значеннями коефіцієнтів підсилювача K .

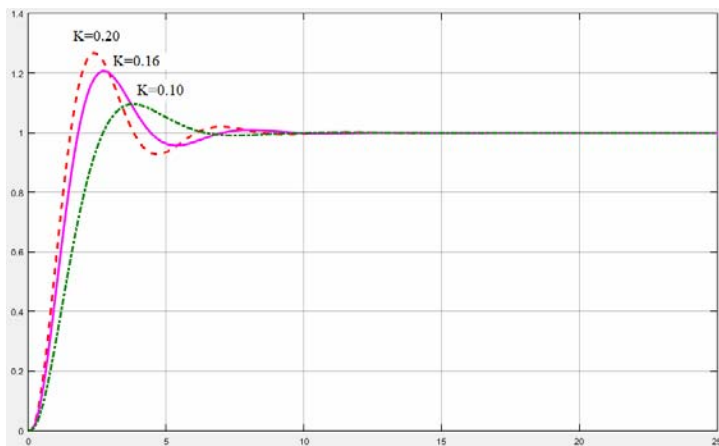


Рисунок 3.30 б) – Графіки перехідних процесів

З графіку (рис. 3.30 б)) бачимо, що необхідний для нашої умови коефіцієнт $K=0.16$, при якому перерегулювання не перевищує 20%.

3.5.2 Аналіз системи керування креном реактивного винищувача

Розберемо систему керування креном реактивного винищувача, схема якого приведена на рис. 3.31.

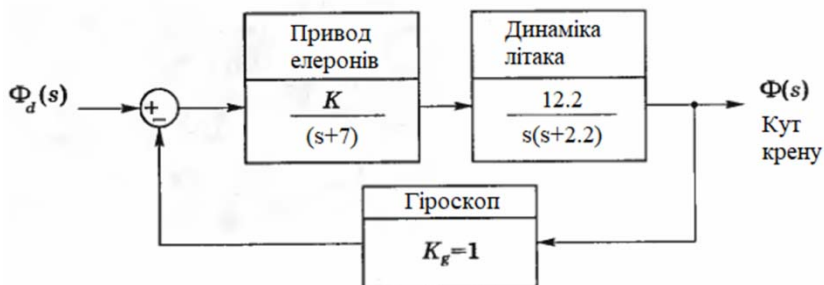


Рисунок 3.31 – Структурна схема керування креном реактивного винищувача

Для цього визначимо наступні задачі:

1. Виберемо коефіцієнт K з перерегулюванням не більше 20%;
2. Визначимо передатну функцію замкненої системи
3. Знайдемо корні характеристичного рівняння при $K=0.7$; 3; 6;
4. Передбачимо очікуване перерегулювання та час максимуму перехідної характеристики в апроксимованій системі другого порядку.
5. Побудуємо графік реакцій вихідної системи та порівняємо показники якості з отриманими.
6. Виберемо значення K з відносним перерегулюванням 16%.
Визначаємо передатну функцію замкненої системи [7]

Знаходимо корні характеристичного рівняння при $K=0.7$:
 $s^3 + 9.2s^2 + 15.4s + 8.54 = 0$.

Корні рівняння $s_1 = -7.23$, $s_{2,3} = -0.98 \pm 0.46j$.

Знаходимо корні характеристичного рівняння при $K=3$:
 $s^3 + 9.2s^2 + 15.4s + 36.6 = 0$.

Тут корні $s_1 = -7.83$, $s_{2,3} = -0.68 \pm 2.05j$.

Знаходимо корні характеристичного рівняння при $K=6$:
 $s^3 + 9.2s^2 + 15.4s + 73.2 = 0$.

Тепер корні $s_1 = -8.4$, $s_{2,3} = -0.4 \pm 2.9j$.

Допустимо, що складна сполучена пара є сполученими коренями, очікуємо наступне:

для $K=0.7$: відносне перерегулювання (ВП) = 0.13% з часом максимуму (T_m) = 6.8 сек.;

для $K=3$: ВП = 35%, T_m = 1.5 сек.;

для $K=6$: ВП = 65.2%, T_m = 1.1 сек.

Обираємо $K=1.71$, щоб отримати ВП=16% та $T_m=2.18$ сек.

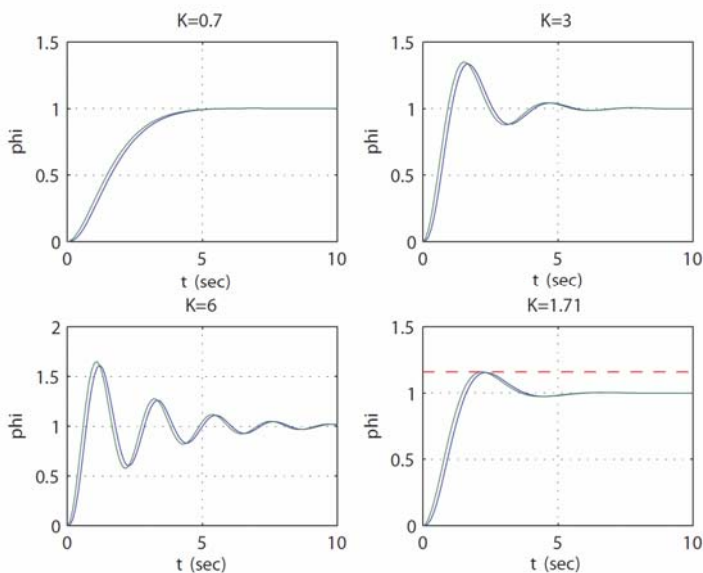


Рисунок 3.32 – Результати моделювання вихідної системи

Всі варіанти ($K=0.7; 3; 6; 1.71$) показані на рисунку 3.32. В кожному випадку наближена передатна функція, яка отримана за умови нехтування недомінантним реальним полюсом і регулює підсилення нульової стаціонарної похибки.

Для $K = 0.7$:

$$T(s) = \frac{1.18}{s^2 + 1.965s + 1.18} = \frac{0.7908}{(s + 0.98 + 0.46j)(s + 0.98 - 0.46j)}$$

Для $K = 3$:

$$T(s) = \frac{4.67}{s^2 + 1.37s + 4.67} = \frac{3.299}{(s + 0.68 + 2.05j)(s + 0.68 - 2.05j)}$$

Для $K = 6$:

$$T(s) = \frac{8.71}{s^2 + 0.796s + 8.71} = \frac{6.399}{(s + 0.4 + 2.9j)(s + 0.4 - 2.9j)}$$

Для $K = 1.71$:

$$T(s) = \frac{2.77}{s^2 + 1.679s + 2.77} = \frac{1.458}{(s + 0.83 + 1.43j)(s + 0.83 - 1.43j)}$$

3.5.3 Визначення області стійкості у випадку нестійкого об'єкту

Розглянемо систему керування реактивним літаком (рис.3.33). Допустимо, що $z > 0$ та $p > 0$. У розімкненому стані система нестійка, тому що її характеристичне рівняння має вигляд:

$$s(s-1)(s+p) = s[s^2 + s(p-1) - p] = 0$$

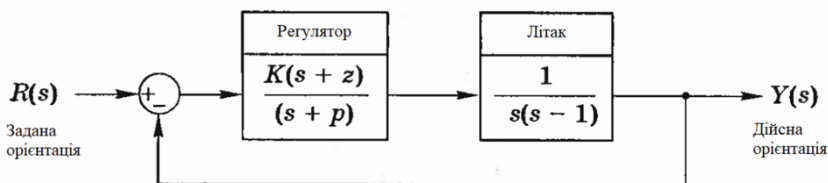


Рисунок 3.33 – Система керування реактивним літаком

Оскільки одна складова частина у квадратних дужках є від'ємною [5], характеристичне рівняння має хоча б один корінь у правій півплощині. Характеристичне рівняння замкненої системи має вигляд:

$$s^3 + (p-1)s^2 + (K-p)s + Kz = 0$$

Визначимо область стійкості у просторі параметрів K , p , z . Складемо таблицю Рауса:

$$\begin{array}{l|l} s^3 & 1 \quad (K-p) \\ s^2 & (p-1) \quad (Kz) \\ s^1 & b_2 \\ s^0 & Kz \end{array}$$

$$\text{де } b_2 = \frac{(p-1)(K-p) - Kz}{p-1}$$

Згідно критерію Рауса-Гурвіца, потрібно виконати наступну вимогу: $Kz > 0$, $p > 1$. Допустимо $b_2 = 0$, отримаємо:

$$(p-1)(K-p) - Kz = K[(p-1)-z] - p(p-1) = 0.$$

Тоді повинна виконуватись наступна умова:

$$K = \frac{p(p-1)}{(p-1)-z}.$$

Розглянемо два випадки:

1. $z \geq p-1$: не існує таких K , при яких система була б стійка;
2. $z < p-1$: при заданих p та z існують K , які задовольняють

вимогам стійкості.

Лістинг програми в MATLAB:

```
[p,z]=meshgrid(1.2:0.2:10,0.1:0.2:10);
```

```
k=p.*(p-1)./(p-1-z);
```

```
mesh(p,z,k).
```

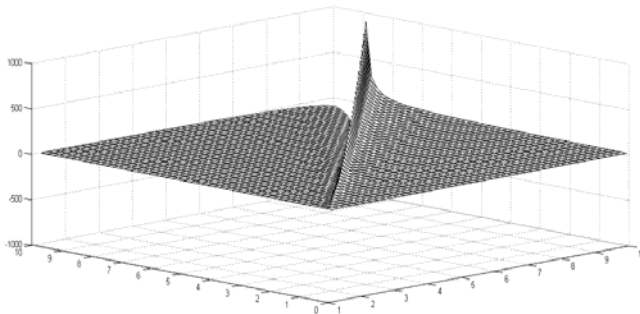


Рисунок 3.34 – Межі стійкості в тривимірному просторі

На рис. 3.34 приведена тривимірна діаграма для визначення області стійкості в координатах K , p , z .

Таким чином можна зробити такі висновки .

Літальні апарати – це нестійкі системи, які компенсуються зворотним зв'язком у вигляді пілота. Використання даних методів розрахунку систем керування дозволяє швидко та зручно виконувати прості розрахунки з авіоніки за допомогою середовища MATLAB та проаналізувати вплив параметрів системи керування на динамічні характеристики ЛА.

4 ВИКОРИСТАННЯ КОРЕНЕВОГО ГОДОГРАФА (КГ) ТА БРАУЗЕРА SISO DESIGN TOOL ЗАДЛЯ АНАЛІЗУ ТА СИНТЕЗУ СИСТЕМ КЕРУВАННЯ

4.1 Поняття кореневого годографа та його основні властивості [7] (rlocus, rlocfind, residue)

При замиканні системи з передаточною функцією $W_p(s)$ одиничним зворотним зв'язком, передатна функція замкненої системи $W_z(s)$ приймає вигляд

$$W_z(s) = \frac{W_p(s)}{1 \pm W_p(s)}, \quad (4.1)$$

де знак «+» відповідає від'ємному зворотному зв'язку; а знак «-» відповідає додатному зворотному зв'язку.

Структурна схема зі зворотним зв'язком приведена на рисунку 4.1.

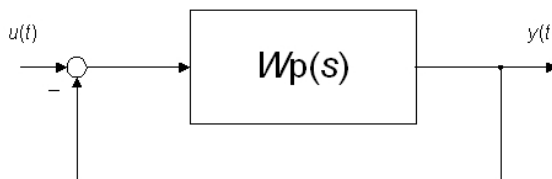


Рисунок 4.1 – Структурна схема САК

З формули (4.1) видно, що нулі передатної функції замкненої системи дорівнюють нулям передатної функції розімкненої системи.

Для визначення полюсів замкненої системи необхідно розв'язати рівняння:

$$W_p(s) = -1. \quad (4.2)$$

Оскільки $W_p(s)$ являється функцією комплексної змінної s , то рівняння (4.2) розпадається на два рівняння - рівняння модулів

$$|W_p(s)| = 1; \quad (4.3)$$

та рівняння аргументів для від'ємного зворотного зв'язку

$$\arg W_p(s) = \pm(2\nu+1)\pi, \quad \nu=0, 1, 2, \dots \quad (4.4)$$

і для додатного зворотного зв'язку

$$\arg W_p(s) = \pm 2\nu\pi, \quad \nu=0, 1, 2, \dots \quad (4.5)$$

У більшості випадків попереду блоку $W_p(s)$ ставлять підсилювач K і підбирають його значення як буде показано далі.

КГ – це траєкторія коренів характеристичних рівнянь системи на s - площині при зміні якого-небудь параметру системи [1, 3], наприклад K . Таким чином, метод КГ дозволяє інженеру оцінити чутливість полюсів системи до змінення якого-небудь параметра. Велику користь має у поєднанні з критерієм Рауса-Гурвіца.

Не наводячи подробиць приведемо основні властивості кореневих годографів:

- гілки кореневого годографа неперервні та розташовані на комплексній площині симетрично відносно дійсної вісі;

- кількість гілок КГ дорівнює порядку системи n . Гілки починаються в n полюсах розімкненої системи при $K=0$. При збільшенні K від 0 до нескінченності полюси замкненої системи рухаються по гілках КГ;

- відрізки дійсної вісі, по яких рухаються дійсні полюси замкненої системи, являються дійсними гілками кореневого годографа. Ці гілки знаходяться в тих частинах дійсної вісі, праворуч від яких розташовано непарне загальне число дійсних полюсів та нулів розімкненої системи;

- m гілок КГ при зростанні K від 0 до нескінченності закінчуються в m нулях $W_p(s)$, а $(n-m)$ гілок при K , що прямують до нескінченності, віддаляються від полюсів вздовж асимптот;

- при розташуванні гілок кореневого годографа у лівій півплощині s САК стійка. При перетині гілок КГ уявної вісі зліва направо САК стає нестійкою. Нехай при $K=K^{kp}$ перетин КГ з уявною віссю відбудеться в деякій точці $i\omega^{kp}$. Назвемо її значення коефіцієнта підсилення критичним K^{kp} , а величину ω^{kp} критичною кутовою частотою, на якій система стає нестійкою.

Метод КГ дає можливість вибрати коефіцієнти підсилення САК, підібрати розташування полюсів та нулів передатної функції корегуючих ланок, визначити параметри домінуючих полюсів

САК (найближчих до початку координат площини s). Розглянемо на прикладі спочатку командний підхід.

Нехай є замкнена система керування такого вигляду.

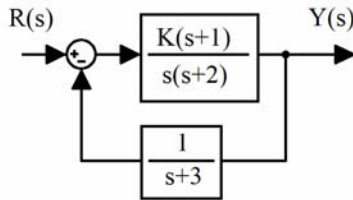


Рисунок 4.2 – Замкнена система керування

Передатна функція системи

$$T(s) = \frac{Y}{R} = \frac{K(s+1)(s+3)}{s(s+2)(s+3) + K(s+1)},$$

де R – вхідний сигнал.

Характеристичне рівняння можна представити у вигляді:

$$s(s+2)(s+3) + K(s+1) = 0$$

$$1 + K \frac{s+1}{s(s+2)(s+3)} = 0$$

Саме у такому вигляді повинно бути записане це рівняння, щоб можна було скористатися функцією *rlocus* (тобто ця функція застосовується до $1+KG$):

$$1 + K \frac{p(s)}{g(s)},$$

де $0 < K < \infty$

Слід мати на увазі, що можуть бути два варіанти використання цієї функції:

- $rlocus(sys)$ – дасть графік годографа;
- $[r,k]=rlocus(sys)$ – дасть параметри r та k , де r – розміщення положення комплексних коренів, k – вектор коефіцієнтів.

Етапи побудови годографу в  MATLAB:

- записати характеристичне рівняння у вигляді $1 + KG = 0$;
- використати $rlocus$ задля побудови кореневого годографа

$$1 + K \frac{s + 1}{s(s + 2)(s + 3)} = 0, \text{ тобто } 1 + K \frac{p(s)}{g(s)}, \text{ де } p(s), g(s) - \text{по-}$$

ліноми чисельника та знаменника, відповідно.

$$p = [1 \quad 1];$$

$$g = [1 \quad 5 \quad 6 \quad 0]; \quad \% \text{ після розкриття } s(s+2)(s+3), \text{ зліва напра-}$$

во

$$sys = tf(p, g); \quad \% \text{ зробити з поліномів передатну функцію } sys$$

$$rlocus(sys)$$

Замість перемноження можна скористатися функцією $conv$ – множення поліномів (множити потрібно тільки по двоє):

$$g1 = [1 \quad 0]; g2 = [1 \quad 2]; g3 = [1 \quad 3];$$

$$g11 = conv(g1, g2);$$

$$g = conv(g11, g3)$$

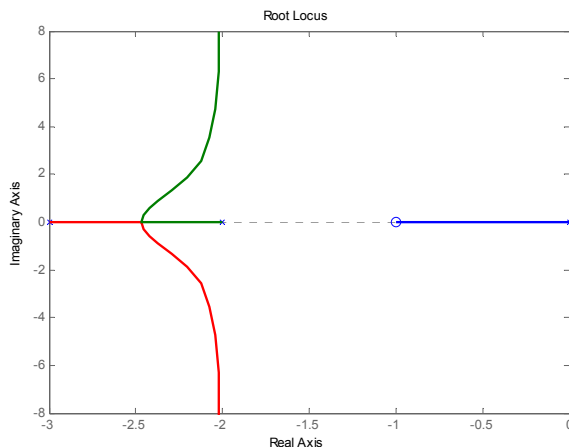


Рисунок 4.3 – Кореневий годограф функції *sys*

На рисунку 4.3 позначками «0» задані нулі, а позначками «x» – полюси.

Другий спосіб використання *rlocus*:

$[r, k] = rlocus(sys)$

plot(r)

Після цього використовуємо функцію *rlocfind*, що призведе до появи на графіку кореневого годографа маркеру у вигляді червоної смуги, яка перетинає траєкторію вертикально. Підводимо її до тих коренів, що нас цікавлять та натискаємо клавішу ENTER. На дисплеї виведеться відповідне значення K та координати (корені – нулі та полюси) цієї точки. Далі їх потрібно підставити в передатну функцію $T(s)$ та побудувати перехідний процес за допомогою функції *step*.

Так, у нашому прикладі при $K=20.5775$ передатна функція замкненої системи має 3 полюси та 2 нулі.

$$\text{Полюси } s = \begin{vmatrix} -2.05 & +j4.32 \\ -2.05 & -j4.32 \\ -0.8989 \end{vmatrix}$$

$$\text{Нулі } s = \begin{vmatrix} -1 \\ -3 \end{vmatrix}$$

Виходячи з того, щоб чим ближче полюс до уявної вісі, тим він важливіший полюс -0.8989 повинен грати домінуючу роль. Для перевірки має сенс дослідити реакцію системи на ступінчастий сигнал $R(s) = 1/s$.

У цьому випадку після підставлення у $T(s)$ отриманого значення K

$$T(s) = \frac{20.5775(s+1)(s+3)}{s(s+2)(s+3) + 20.5775(s+1)} \cdot \frac{1}{s}$$

До того як перейти у часову область, тобто отримати функцію $Y(t)$,  використовується функція *residue*, яка задає розклад $Y(s)$ на прості дробі вигляду:

$$Y(s) = \frac{r(1)}{s-p(1)} + \frac{r(2)}{s-p(2)} + \frac{r(3)}{s-p(3)} = \frac{r(s)}{s-p(4)} + k(s)$$

Тут r – коефіцієнти чисельника – лишки (російською вычеты), p – полюси, k – постійний член.

Функція *residue* має такий загальний вигляд $[r,p,k]=\text{residue}$

Використаємо її для нашого випадку:

$$k = 20.5775; \text{num} = k \cdot [1 \ 4 \ 3]; \text{den} = [1 \ 5 \ 6 + k \ k]$$

$$[r, p, k] = \text{residue}(\text{num}, \text{den})$$

$$r = \begin{pmatrix} -1.38 + j1.7 \\ -1.38 - j1.7 \\ -0.2429 \\ 3.00 \end{pmatrix} \quad p = \begin{pmatrix} -2.5 + j4.34 \\ -2.5 - j4.34 \\ -0.89 \\ 0 \end{pmatrix} \quad k = 0$$

$$Y(s) = \frac{-1.37 + j1.7}{s + 2.5 + j4.34} + \frac{-1.37 - j1.7}{s + 2.5 - j4.34} + \frac{-0.2429}{s + 0.89} + \frac{3.00}{s}$$

Порівнюючи значення лишків бачимо, що коефіцієнт у чисельнику для полюса $s = -0.89$ значно менше всіх інших, тому реакція цієї складової вихідного сигналу не може бути домінуючою і час встановлення перехідного процесу буде визначатися іншими полюсами.

На рисунку 4.4 побудований графік перехідного процесу за допомогою функції *step*, перерегулювання отримали майже 60%.

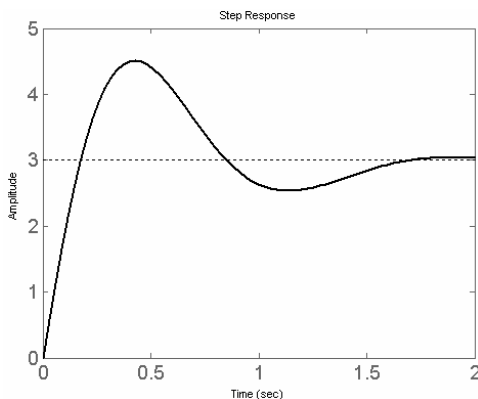


Рисунок 4.4 – Перехідний процес згідно розрахунку

Якщо потрібно, шукаємо нове значення K за допомогою `find` і повторюємо розрахунки та перетворення.

Крім командного підходу для роботи з КГ можна використати браузер *sisotool*.

4.2 SISO Design Tool

SISO Design Tool являє собою графічний інтерфейс користувача, який дозволяє аналізувати і налаштовувати SISO системи керування зі зворотним зв'язком [2, 7]. Завдяки SISO Design Tool можна графічно налаштовувати коефіцієнти підсилення і динамічні властивості (dynamics) компенсатора (C) й передфільтра (F), використовуючи комбінацію методу розташування коренів і методу формування контуру.

Наприклад, можна використовувати графік розташування коренів, щоб стабілізувати контур зворотного зв'язку і посилити мінімальне згасання, а діаграми Бode використовувати для корекції смуги пропускання, визначення запасів стійкості за амплітудою і фазою або додавання режекторного фільтра для придушення перешкод. Також можна викликати графіки Нікольса для розімкненого контуру або діаграми Бode передфільтра (F), обравши відповідні пункти меню View. Всі графіки динамічно пов'язані між собою. Наприклад, якщо змінити коефіцієнт підсилення на графіку розташування коренів, то діаграма Бode також одразу зміниться.

SISO Design Tool, розроблений для роботи з LTI Viewer, дозволяє вносити зміни в схему і тут же відслідковувати їх результат в LTI Viewer. Наприклад, якщо змінити параметри компенсатора, то LTI Viewer, пов'язаний з SISO Design Tool, автоматично оновить обрані вами графіки. За замовчуванням SISO Design Tool виводить на екран графік розташування коренів і діаграму Бode розімкненого контуру системи.

Імпортовані системи можуть включати будь-які елементи схеми структури зі зворотним зв'язком, розташованої в правому куті панелі поточного компенсатора (рисунк 4.5). Не можна змінити імпортовані моделі об'єкта керування (G) та датчика (H), але можна використовувати SISO Design Tool для розробки нового (або зміни існуючого) передфільтра (F) або компенсатора (C) для

імпортованих об'єкта керування та датчика.

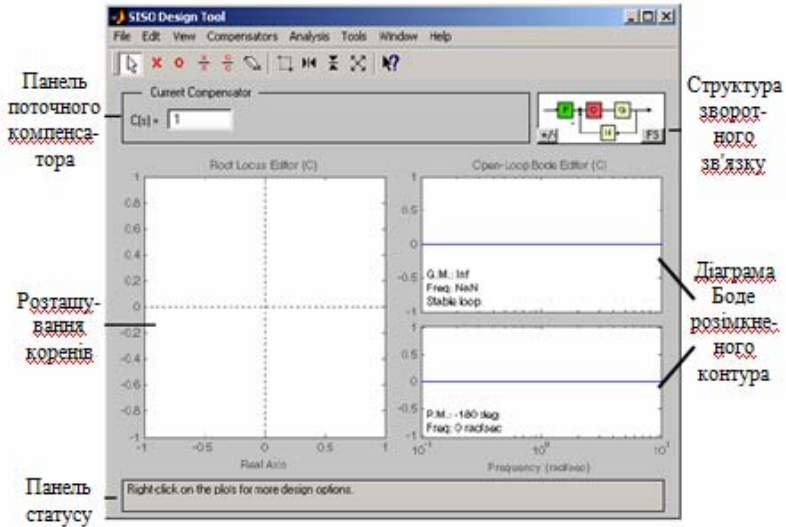


Рисунок 4.5 – Основне вікно SISO Tool

4.2.1 Структура зі зворотним зв'язком

В SISO Design Tool за замовчуванням компенсатор знаходиться в послідовному зв'язку, тобто структура зі зворотним зв'язком виглядає наступним чином:

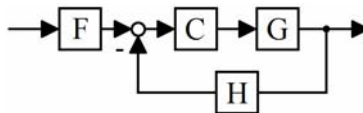


Рисунок 4.6 – Структура зворотного зв'язку за замовчуванням

На цьому рисунку прийняті наступні позначення:

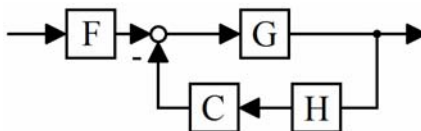
- G – об'єкт керування;
- H – датчик;
- F – передфільтр;
- C – компенсатор.

За замовчуванням значення для блоків F, H та C дорівнюють

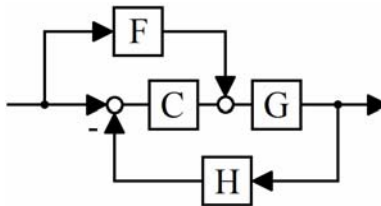
1. В якості об'єкта керування (G) задана модель двигуна постійного струму sys_dc.

4.2.2 Альтернативні структури зворотного зв'язку

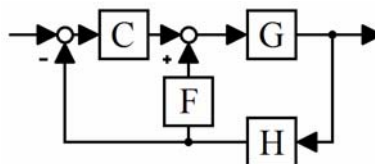
На подальших рисунках 4.6 наведені основні структури зворотного зв'язку.



а) компенсатор в колі зворотного зв'язку;



б) контролер з попередженням;



в) каскадна система з фільтром (F) у внутрішньому контурі.

Рисунок 4.6 – Основні структури зворотного зв'язку

Структуру зворотного зв'язку можна змінити, натиснувши кнопку FS, яка циклічно викликає різні конфігурації системи.

4.2.3 Налаштування компенсаторів

SISO Design Tool спрощує процедуру проектування компенсаторів. В залежності від того, які графіки відкриті – розташування коренів, діаграма Боде, графіки Нікольса – існує три засоби зміни структури компенсатора:

- Інтерактивні графіки дозволяють налаштовувати коефіцієнт підсилення компенсатора і регулювати динамічні характеристики (нулі та полюси).

- Меню правої кнопки миші дозволяє додавати/видаляти нулі і полюси і налаштовувати вигляд графіка (наприклад, додавати сітку, змінювати масштаб, додавати обмеження або налаштовувати властивості графіка).

- Вікно редактора компенсатора (Edit Compensator) являє GUI з областями для введення з клавіатури коефіцієнта підсилення і розташування нулів / полюсів.

В SISO Design Tool можна використовувати будь-який з цих засобів на графіку розташування коренів, діаграмах Боде розімкнутого контуру і передфільтра. Після додавання нулів і полюсів в компенсаторі, можна змінювати їх місце розташування, перетягуючи їх курсором миші. SISO Design Tool розроблений так, що зміна будь-якого з графіків автоматично призводить до зміни інших. Зокрема, панель Current Compensator завжди відображає поточну схему компенсатора.

4.2.4 Розташування коренів

Можна налаштувати компенсатор, використовуючи метод розташування коренів. На рисунку 4.7 нижче показані імпортовані моделі компенсатора і об'єкт керування; використовуємо меню правої кнопки і властивості інтерактивної графіки, щоб додати, змінити і видалити нулі і полюси компенсатора.

Найпростішим методом налаштування є зміна коефіцієнта підсилення компенсатора, який за замовчуванням дорівнює 1. Щоб зробити це, просто виділіть мишкою на графіку червоні квадрати і перемістіть їх уздовж кривої.

Щоб було зручніше відстежувати вплив зроблених змін на характеристики системи, одночасно з графіком розташування ко-

ренів побудуйте перехідну характеристику замкненого контуру. Для цього оберіть в меню Analysis/Other Loop responses і у вікні натисніть ОК.

Однак, можна помітити, що збільшення коефіцієнту підсилення (КП) призводить до підвищення коливальності системи. Подальше збільшення КП призводить до нестійкого стану системи, отже досягти всіх вимог тільки за допомогою зміни КП неможливо. Крім КП компенсатор характеризується також нулями і полюсами. Тому після налаштування КП можна додати нулі або полюси в компенсатор.

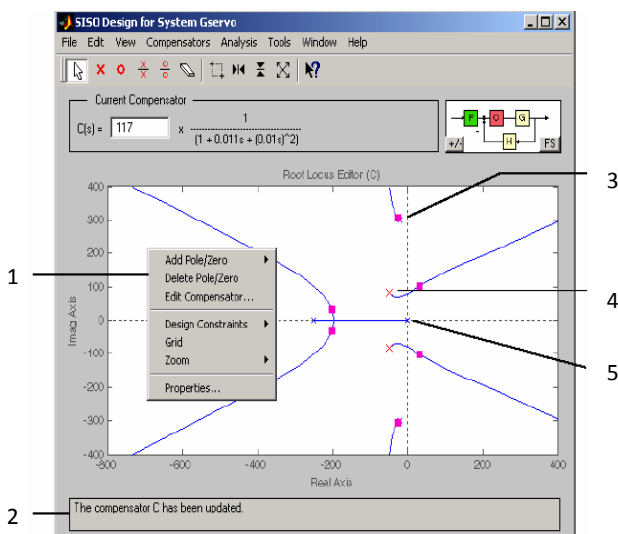


Рисунок 4.7 – Графік розташування коренів, де:

1 - використовуйте меню правої кнопки щоб додати нулі і полюси в схему компенсатора; 2 - панель статусу показує поради по використанню SISO Design Tool; 3 - пересуньте будь-який з червоних квадратів щоб змінити коефіцієнт підсилення компенсатора; 4 - можна змінити положення нулів і полюсів компенсатора (червоні 0 і x відповідно) за допомогою миші; 5 - сині x і 0 позначають полюси і нулі об'єкта керування. їх не можна пересувати.

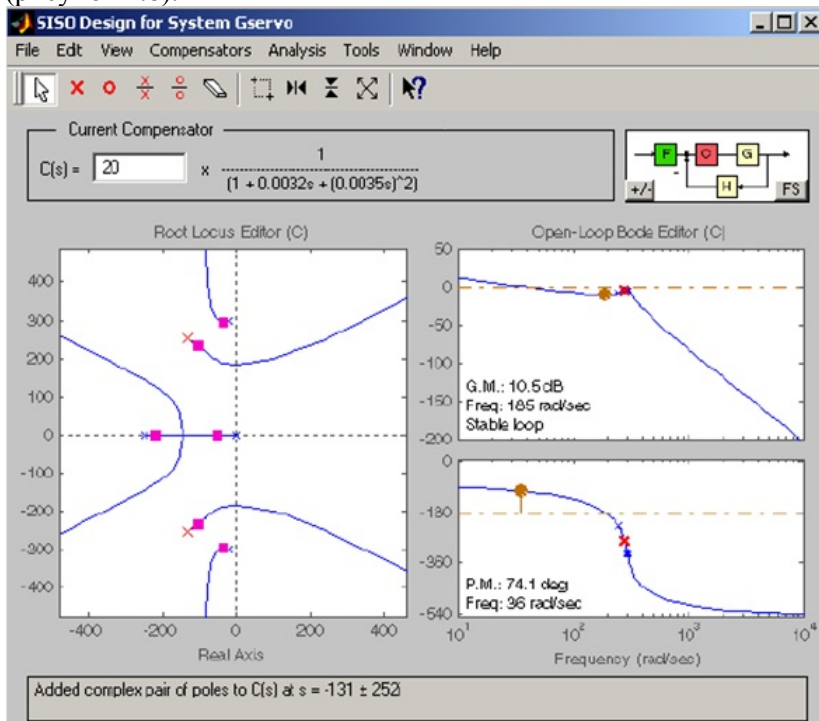
Спробуйте додати комплексно спряжену пару полюсів. Для цього в меню правої кнопки оберіть Add Pole/Zero>Complex Pole (рисунок 4.7). Намагайтесь розташувати полюс лівіше пари по-

люсів, розташованих близько до уявної вісі. При цьому графіки Боде, розташування коренів і перехідної характеристики зміняться з урахуванням доданого полюса. На графіку розташування коренів нові полюси будуть позначені червоними 'x'.

Додавання нулів в компенсатор здійснюється аналогічним чином. Намагайтеся додати пару комплексно спряжених нулів трохи лівіше пари тільки що доданих полюсів.

Процес проектування компенсатора може включати деякі помилки та неточності. Тому переміщуйте нулі і полюси компенсатора або полюси замкнутого контуру по графіку розташування коренів доки не досягнете необхідних характеристик.

Щоб поставити точні значення нулів і полюсів, використовуйте вікно Edit Compensator, за допомогою якого можна змінити значення КП і місце розташування нулів і полюсів компенсатора (рисуюнок 4.8).



Рисуюнок 4.8 – Результат додавання комплексно спряженої пари полюсів

Вікно Edit Compensator можна викликати трьома способами: Обрати пункт Edit Compensator меню Edit; обрати пункт Edit Compensator в меню правої кнопки; двічі клацнувши мишкою по панелі Current Compensator.

4.2.5 Використання меню проектування компенсаторів

Найбільш швидким засобом проектування компенсаторів є проектування за допомогою меню правої кнопки. SISO Design Tool включає меню правої кнопки для всіх доступних у додатках графіків, включаючи графік розташування коренів, діаграми Бode розімкненого контуру та діаграми Бode передфільтра. Пункти меню для всіх цих графіків ідентичні, але, обмеження об'єкта відрізняються в залежності від графіка, в якому викликане це меню.

Меню правої кнопки можна використовувати, щоб додати нулі, полюси, попередження, запізнювання, та режекторні фільтри.

4.2.6 Додавання нулів та полюсів до компенсатора

За допомогою пункту Add Pole/Zero меню правої кнопки можна додати до схем компенсатора дійсні та комплексні нулі чи полюси, контури попередження та запізнювання, режекторні фільтри (рисунок 4.9).

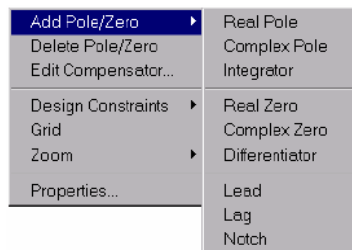


Рисунок 4.9 – Підменю Add Pole/Zero

Доступними являються наступні конфігурації нулів та полюсів:

- Real Pole (дійсний полюс).
- Complex Pole (комплексний полюс).
- Integrator.
- Real Zero (дійсний нуль).
- Complex Zero (комплексний нуль).
- Lead (попередження).
- Lag (запізнювання).
- Notch (режекція).

При виборі будь-якого пункту, окрім integrator та differentiator, курсор миші перетворюється на «х». Щоб додати вибраний пункт в схему компенсатора, наведіть «х» на необхідне місце на графіку та натисніть ліву кнопку миші. Графік автоматично зміниться.

4.2.7 Додавання передфільтра

В SISO Design Tool передфільтр зазвичай виконує наступні функції:

- відстеження прямого контуру для зниження навантаження контуру зворотного зв'язку (при низьких запасах стійкості);
- фільтрація високочастотної складової у керуючому впливі для обмеження пере регулювання та уникнення збуджуючого резонансного режиму об'єкта керування.

Звичайний передфільтр – це фільтр низької частоти, що знижує шум у вхідному сигналі.

У передфільтр можна додавати нулі та полюси, а також налаштовувати КП за допомогою методів, що використовуються для налаштування компенсатора.

Якщо передфільтр не був імпортований, то всі його параметри дорівнюють нулю. Щоб швидко створити фільтр низької частоти, що згладжує, можна просто додати декілька комплексно спряжених полюсів.

4.3 Дослідження СК збудження за допомогою SISO Design Tool

Для того щоб скористатися редактором, потрібно привести контур збудження до виду однієї передатної функції, як це зроблено, наприклад, на рисунку 2.17. Нагадуємо, що з шести описаних далі у програмі очевидних блоків перший, з передатною функцією $s1$, додано для зручності, він буде виконувати функцію вхідного ($input=1$), а вихідним є блок 5. Структурну схему легко побудувати згідно матриці q з програми (дивись розділ 2.8).

Лістинг програми:

```
k1=23.2;k2=0.264;k3=0.004;  
T1=1.1;T2=0.0007766;T3=0.017;  
T4=0.0085;T5=0.0455;T6=6.71;  
s1=tf(1,1);s2=tf([T1 1],[T2 T3 0]);s3=tf([k1],[T4 1]);  
s4=tf([T5 1],[T6 6.1]);s5=tf([k3],[T5 1]);s6=tf([0.264], [1]);  
sln=append(s1,s2,s3,s4,s5,s6);  
q=[2 1 -6 0 0;3 2 0 0 0;4 3 0 0 0;5 4 0 0 0;6 4 0 0 0];  
input=1;output=5;  
sysn=connect(sln,q,input,output);  
sss1=tf(sysn)  
W1=minreal(sss1)  
s345=s3*s4*s5;  
s65=s6*(1/s5)
```

Отримаємо передатну функцію контуру збудження G і відкриємо SISO Design Tool, рисунок 4.8. Далі необхідно вибрати необхідну структуру з фільтром та регулятором і внести дані до ланок та обрати відповідний вид регулятора. У вікні SISO Design task буде побудований кореневий годограф та частотні характеристики (рисунок 4.9). Плавню переміщуючи повзунок по гілці КГ можемо спостерігати в реальному часі зміни перехідного процесу.

За допомогою пункту Add Pole/Zero меню правої кнопки можна додати до схем компенсатора дійсні та комплексні нулі чи полюси, контури попередження та запізнювання, режекторні фільтри.

Вікно Edit Compensator можна викликати трьома способами: Обрати пункт Edit Compensator меню Edit; обрати пункт Edit Compensator в меню правої кнопки; двічі клацнувши мишкою по панелі Current Compensator.

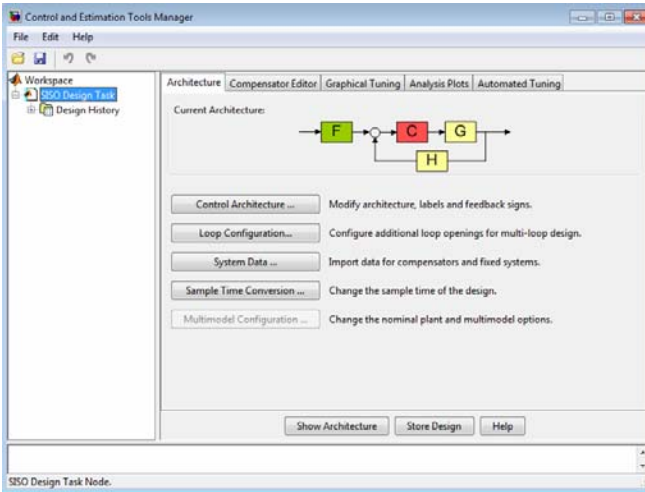


Рисунок 4.9 – Вікно SISO Design task

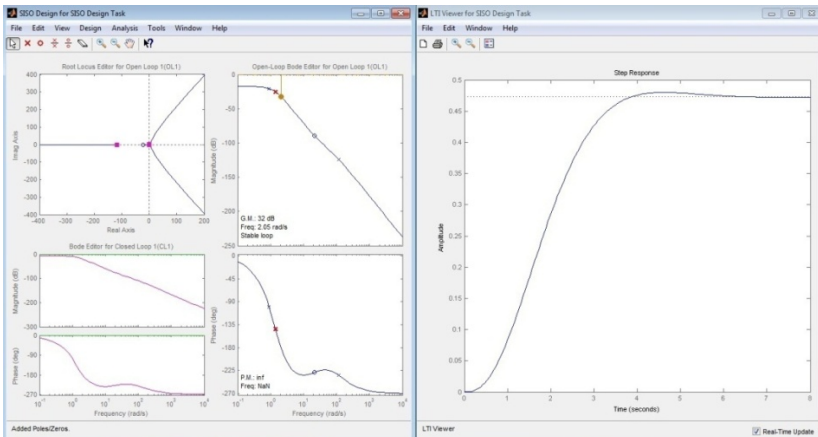


Рисунок 4.10 – Діалогове вікно SISO Design Tool

Додаток А. Довідкова таблиця функцій MATLAB.

Функція	Призначення
Формування LTI- об'єктів	
ss	Створення моделі простору стану
zpk	Створення моделі нулі-полюси-коефіцієнти передачі
tf	Створення моделі у вигляді передавальної функції
dss	Спеціальний опис моделі простору стану
filt	Створення моделі цифрового фільтра
set	Установка (модифікація) атрибутів LTI-моделі
ltiprops	Отримання детальної довідки про атрибути LTI-моделей
Витяг даних	
ssdata	Витяг матриць A, B, C, D простору стану
zpkdata	Витяг даних про нулі, полюси, коефіцієнти передачі
tfddata	Витяг чисельника і знаменника передавальної функції
dssdata	Отримання інформації про версію описувача dss
get	Отримання інформації про значення властивостей LTI-моделі
Інформація про окремі характеристики моделі	
class	Отримання даних про тип моделі ('ss', 'zpk', чи 'tf')
size	Отримання даних про розміри матриць входу і виходу
isempty	Перевірка, чи є LTI-модель порожньою
isct	Перевірка, чи є модель неперервною
isdt	Перевірка, чи є модель дискретною
isproper	Перевірка, чи є модель правильною
issiso	Перевірка, чи має модель один вхід і один вихід
isa	Перевірка, чи є LTI-об'єкт моделлю заданого типу
Перетворення одного виду моделі в інший	
ss	В модель простору стану
zpk	В модель нулі-полюси-коефіцієнт передачі
tf	В модель передавальної функції
c2d	Перехід від неперервної моделі до дискретної моделі
d2c	Перехід від дискретної моделі до неперервної моделі

d2d	Перевизначення дискретної системи або додавання затримок
Арифметичні операції над моделями	
+, -	Додавання і віднімання LTI-систем (паралельне з'єднання)
*	Множення LTI-систем (послідовне з'єднання)
inv	Звернення LTI-системи
\	Ліве ділення sys1/sys2, рівносильно inv(sys1)*sys2
/	Праве ділення sys1/sys2, рівносильно sys1*inv(sys2)
'	Перетранспонування
.'	Транспонування карти входу-виходу
[...,...,...]	Горизонтальне об'єднання LTI-систем
[...;...;...]	Вертикальне об'єднання LTI-систем
Характеристики динамічної системи	
pole, eig	Визначення полюсів системи
tzero	Визначення нулів системи
pzmap	Складання карти нулів-полюсів, виклик сітки
dcgain	Знаходження коефіцієнта передачі при нульовій частоті
norm	Визначення норми LTI-системи
covar	Обчислення коваріації відгуку на білий шум
damp	Визначення частоти власних коливань і демпфування
esort	Сортування полюсів неперервної системи за їх величиною $\text{Re}(p_i)$
dsort	Сортування полюсів дискретної системи за їх величиною $\text{Re}(p_i)$
pade	Апроксимація Паде затримок у часі
Моделі простору стану	
rss, drss	Генерування випадкових моделей простору стану
ss2ss	Перетворення змінних стану
canon	Перехід до канонічних форм простору стану
ctrb, obsv	Матриці керованості та спостережності системи
gram	Визначники Грама (керованості та спостережності)
ssbal	Діагональне балансування матриць простору стану
balreal	Балансування входу-виходу на основі визначника Грама

modred	Редукція стану моделі
minreal	Мінімальна реалізація та скорочення нулів і полюсів
augstate	Вивід на вихід системи всіх змінних стану
Відгук (реакція системи) у часі	
step	Визначення відгуку $h(t)$ на одиничний стрибок $1(t)$
impulse	Визначення відгуку $\omega(t)$ на одиничний імпульс $\delta(t)$
initial	Визначення відгуку на задані початкові умови стану
lsim	Визначення відгуку при довільних вхідних впливах
ltiview	Аналіз відгуків за допомогою графічного інтерфейсу
gensig	Генерування періодичних сигналів $u(t)$ для функції $lsim$
stepfun	Генерування одиничного стрибка $1(t)$
Частотні характеристики САК	
bode	Побудова діаграми Бode - частотної характеристики САК
sigma	Побудова частотного графіка сингулярних значень
nyquist	Побудова діаграми Найквіста-ЧХ в полярних координатах
nichols	Побудова діаграми Ніколса, виклик сітки L_3 , ϕ_3 -ngrid
ltiview	Аналіз характеристик САК за допомогою графічного інтерфейсу
evalfr	Розрахунок ЧХ САК на заданій частоті (точки ЧХ)
freqresp	Розрахунок ЧХ САК в заданому діапазоні частот
margin	Визначення запасів по фазі та амплітуді L_3 , ϕ_3
Об'єднання (комутація) систем	
append	Об'єднання ЛТІ-систем шляхом об'єднання входів і виходів
parallel	Узагальнене паралельне з'єднання (див. також +)
series	Узагальнене послідовне з'єднання (див. також *)
feedback	З'єднання двох систем у вигляді петлі зворотного зв'язку
star	З'єднання зіркою Редхейфера
connect	Отримання ss - моделі з опису блок-схеми
Синтез методами кореневого годографа та розміщенням полюсів	
rlocus	Побудова діаграми Еванса - кореневого годографа (КГ) САК

rlocfind	Вибір коефіцієнта передачі САК з її полюсів на КГ
rltool	Виклик графічного інтерфейсу який реалізує метод КГ
acker	Синтез регулятора К одновимірної системи (або спостерігача L)
place	Синтез регулятора К багатовимірної системи (або спостерігача L)
estim	Синтез спостерігача за заданими коефіцієнтами $L=[l_1;\dots;l_n]$
reg	Синтез регулятора за заданим матрицям зворотного зв'язку К та L
Синтез оптимальних лінійно-квадратичних регуляторів (LQG)	
lqr, dlqr	Синтез регуляторів (для неперервної і дискретної САК)
lqry	Синтез регулятора LQR зі зважуванням виходу у та входу u
lqrd	Синтез дискретного регулятора LQR для неперервної системи
kalman	Синтез неперервного фільтра Калмана
kalmd	Синтез дискретного фільтра Калмана для неперервної системи
lqgreg	Синтез LQG регулятора за коефіцієнтами К і фільтру Калмана
Матричні рівняння	
lyap	Рішення неперервних рівнянь Ляпунова
dlyap	Рішення дискретних рівнянь Ляпунова
care	Рішення неперервних алгебраїчних рівнянь Ріккати
dare	Рішення дискретних алгебраїчних рівнянь Ріккати
Демонстраційні приклади в Control System Toolbox	
ctrldemo	Введення в Control System Toolbox
jetdemo	Класичне проектування САК кутот рискання
diskdemo	Цифрове проектування контролера приводу жорсткого диска

Додаток Б. Параметри для зображення кривих у функції plot.

При виведенні двомірних графіків деколи використовується конструкція вигляду:

`'color_style_marker'`, де:

Одно -, дво -, або тризначними змінними символного типу ідентифікують колір, тип лінії та маркер:

- допустимим є використання кольорів 'c' – cyan (голубий), 'm' – magenta (бузковий), 'y' – yellow (жовтий), 'r' – red (червоний), 'g' – green (зелений), 'b' – blue (синій), 'w' – white (білий), 'k' – black (чорний).

- знаки стилів лінії є такими: '-' – неперервна, '--' – штрихова, ':' – пунктирна, '-.' – штрих пунктирна, 'none' – не проводити лінії.

- знаки найбільш вживаних маркерів '+', 'o', '*' та 'x'.

Наприклад, команда:

`plot(x,y, 'y:+')` виведе залежність $y(x)$ жовтою (y) пунктирною лінією (:), на якій кожна пара даних буде відмічена маркером (+). У випадку, якщо визначено тип маркера, але не визначено стиль лінії, на екран будуть виведені тільки точки залежності, задані у вихідних векторах.

ПЕРЕЛІК ПОСИЛАНЬ

1. Лозинський А., Мороз В., Паранчук Я. Розв'язування задач електромеханіки в середовищах пакетів MathCAD і MATLAB : Навчальний посібник. Львів: Видавництво Державного університету «Львівська політехніка», 2000. – 166 с.
2. Медведев В. С., Потемкин В. Г. Control System Toolbox.MATLAB 5 для студентів, Под общей ред. В. Г. Потемкина, М. : ДИАЛОГ-МИФИ, 1999. – 287 с. (Пакеты прикладных программ).
3. Попович М. Г., Ковальчук О. В. Теорія автоматичного керування. Київ, «Либідь», 1997. – 544 с.
4. Методичні вказівки до лабораторних робіт з дисципліни «Теорія автоматичного керування» для студентів спеціальності 8.092203 денної форми навчання /Укл. Є.М. Потапенко, С.Г. Десев, В.І. Левикіна. – Запоріжжя: ЗНТУ, 2007. – 48 с.
5. Олссон Г. Цифровые системы автоматизации и управления / Г. Олссон, Д. Пиани. Санкт-Петербург: Невский Диалект, 2001. – 557 с.
6. Потапенко Е. М., Казурова А. Е. Основы теории автоматического управления. Запорожье: ЗНТУ, 2007. – 162 с.
7. Р. Дорф. Современные системы управления. Р.Дорф, Р.Бишоп: Пер с англ.-Б.И.Копылова. М. : Лаборатория Базовых Знаний, 2004. – 832 с.

Навчальне видання

Арсеньєва Світлана Іванівна

**ВИКОРИСТАННЯ ПРОГРАМНИХ ЗАСОБІВ МАТЛАВ
ДЛЯ РОЗВ’ЯЗАННЯ ТИПОВИХ ЗАДАЧ АНАЛОГОВОЇ
АВТОМАТИЗАЦІЇ**

Навчальний посібник

Укладач: *Арсеньєва С.І.*

Відповідальний за випуск: *Пилипенко К.І.*

Верстання: *Дяченко О.О.*

Підписано до друку 09.12.2019. Формат 60×84/16. Ум. друк. арк. 6,86.
Тираж 300 прим. Зам. № 1418.

Національний університет «Запорізька політехніка»
Україна, 69063, м. Запоріжжя, вул. Жуковського, 64
Тел.: (061) 769–82–96, 220–12–14

Свідоцтво суб’єкта видавничої справи ДК № 6952 від 22.10.2019.