

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут»

Динамічна оптимізація з використанням MATLAB та SIMULINK

Київ
НТУУ «КПІ»
2010

Жученко А. І., Ладієва Л. Р., Дубік Р. М.
Динамічна оптимізація з використанням MATLAB та SIMULINK. –
К.: НТУУ “КПІ”, 2010. – 209 с.

У монографії розглядається застосування пакету MATLAB та його розширення Control System Toolbox і SIMULINK (моделювання нелінійних систем) для розв’язання задач оптимального керування. Розглядається оптимальне керування для систем, що описуються диференціальними рівняннями (лінійними і нелінійними та з аргументами, що запізнюються). Приділено увагу оптимальному оцінюванню стану та стохастичному керуванню. Викладення ілюстроване прикладами задач оптимального керування.

Книга призначена для студентів, аспірантів та спеціалістів, що спеціалізуються в області автоматизованого керування технологічними процесами та виробництвом, а також інших спеціальностей, які пов’язані з розробкою систем оптимального керування.

Відповідальний редактор

д. т. н., професор,
завідувач кафедрою АХВ,
НТУУ “КПІ” Жученко А. І.

Жученко А. І., Ладієва Л. Р., Дубік Р. М. 2010

Синтез систем автоматичного керування – одна з найважливіших технічних задач. Метою синтезу являється створення системи, що складається з реальних елементів, яка задовольняла б певним показникам якості. Останні можуть бути задані в часовій області, наприклад, у вигляді величини перерегулювання і величини зміни перехідної характеристики. Коли представляє інтерес поведінки системи в перехідному режимі, її якість оцінюється по динамічним характеристикам. Тому очевидно, що процедуру синтезу будемо виконувати в часовій області.

Пакет Control System має велику кількість алгоритмів MATLAB для моделювання, аналізу і проектування систем автоматичного керування. Функції пакету включають в себе традиційні методи передатних функцій та сучасні методи простору станів. За допомогою пакету Control System можливо моделювати та аналізувати як неперервні, так і дискретні системи. Частотний та 2- часовий відгук, діаграми розташування нулів / полюсів можуть бути швидко обчислені та відображені на екрані.

Основні властивості

Моделювання:

- Неперервні та дискретні системи;
- Формати моделей: передатна функція, простір станів, нулі / полюси;
- Формування лінійних моделей систем;
- Перетворення моделей в різні формати.

Аналіз:

- Повний набір засобів для аналізу single - input / single - output (SISO) – систем з одним входом / одним виходом або multiple - input / multiple - output (MIMO) – систем багатьма входами / багатьма виходами;
- Динамічні характеристики: передатна і перехідна функція, реакції на довільні дії;
- Частотні характеристики: діаграми Боде, Нікольса, Найквіста та ін.

Проектування:

- Розробка зворотних зв'язків;
- Проектування лінійно-квадратичних регуляторів та оцінювачів;
- Характеристики моделей: керованість, спостережуваність, зниження порядку моделі;
- Підтримка систем з запізненням.

Середовище моделювання. Пакет Control System дозволяє проектувати неперервні та дискретні системи. Додаткові функції побудови моделей дозволяють конструювати більш складні моделі.

Об'єкти. Користуючись об'єктно-орієнтованими можливостями MATLAB, пакет дозволяє обробляти SISO або MIMO лінійні системи одною командою, також проводити операції з матрицями.

Аналіз часового і частотного відгуків. Часовий відгук може бути розрахованим для імпульсного входу, одиничного скачка або довільного вхідного сигналу. Серед багатьох функцій – діаграми Боде, Найквіста і Ніколаса для частотного аналізу. Також є функції для аналізу сингулярних чисел.

Настроювання зворотних зв'язків. Пакет Control System містить засоби для обрання параметрів зворотного зв'язку. Серед традиційних методів: аналіз власних точок, визначення коефіцієнтів підсилення та затухання. Серед сучасних методів: лінійно-квадратичне регулювання та ін.

Можливість настроювання користувача. Пакет Control System включає в себе велику кількість алгоритмів для проектування та аналізу систем керування. Крім цього, він має середовище, що настроюється, яке дає можливість програмування власних M – файлів.

Базис для проектування систем керування. Пакет Control System часто використовується в комбінації з іншими пакетами для проектування систем керування. Це дозволяє вдосконалювати створювані системи.

SIMULINK – це інтерактивне середовище для моделювання та аналізу широкого класу динамічних систем. SIMULINK надає

графічний інтерфейс для конструювання моделей з стандартних блоків за допомогою технології “drag-and-drop” – перетягни і відпусти. Велика бібліотека блоків дозволяє моделювати системи дуже швидко.

Основні засоби для моделювання і аналізу

- Бібліотека блоків для створення лінійних і нелінійних, дискретних і неперервних, гібридних, SISO і MIMO моделей;
- Ієрархічна структура моделей з необмеженими можливостями;
- Скалярні та векторні зв'язки;
- Засіб для створення блоків користувача та бібліотек;
- Інтерактивне моделювання з відображенням на екрані;
- Методи інтегрування з фіксованим та змінним кроком;
- Лінеаризація;
- Моделювання Монте-Карло;
- Визначення точок рівноваги;
- Різноманітні способи виводу на екран та бібліотека вхідних сигналів.

В виданні надані теоретичні відомості та приклади програм, наступних задач динамічної оптимізації: проектування лінійно-квадратичних регуляторів, оцінювання стану за допомогою фільтра Калмана, проектування лінійно-квадратичного гауссіана за заданими коефіцієнтами фільтра Калмана, розміщення полюсів за допомогою зворотного зв'язку за станом, застосування формули Акермана, оптимізація нелінійних систем, розв'язання нелінійних рівнянь та систем рівнянь, розв'язання нелінійних диференціальних рівнянь та систем рівнянь. Приклади програм для синтезу регуляторів, оцінювання стану, розміщення полюсів за допомогою зворотного зв'язку за станом, що наведені у виданні можуть бути використані для оптимізації n – вимірних систем з багатьма входами та виходами. Використовуючи додаток можна проводити проектування та оптимізацію більш складних систем керування.

1. Оптимальні системи керування

Якість системи керування можна характеризувати інтегральними оцінками. Синтез системи має бути оснований на мінімізації оцінки якості. Системи, в яких забезпечується мінімум складової оцінки якості, називають **оптимальними системами керування**. Система описується змінними стану, метод розв'язку задачі оснований на вимірюванні змінних стану і формування з них керуючого сигналу $U(t)$, що оптимізує систему.

Відома постановка задачі оптимального керування динамічними системами, що полягає в знаходженні траєкторії $U(t)$ $t \in [t_0, t_f]$, яка приводить стан системи (приведеної на рис. 1.1.) $X(t)$ за допомогою мінімізації функціоналу якості:

$$J = G[X(t), t] \Big|_{t=t_0}^{t=t_f} + \int_{t_0}^{t_f} F[X(t), U(t), t] dt, \quad (1.1)$$

з заданими початковими умовами $X(t=t_0) = X_0$ в кінцеву умову $N[X(t_f), t_f] = 0$ оптимальним способом. З точки зору використання є два альтернативних варіанти отримання оптимальних результатів.

У випадку оптимального програмного керування одержуємо оптимальну траєкторію керування, як постійну функцію часу, та в часовому проміжку оптимізації для реального процесу. Очевидно, що ця оптимальна траєкторія залежить від заданого початкового стану X_0 . Із-за дії можливих перешкод або неточностей моделі, на які не звертають увагу при оптимізації, буде перебігати в стан $X(t)$, що

відрізняється від очікуваного, отриманого за моделлю $X(t)$. При цьому також можливе відхилення кінцевих умов.

У випадку оптимального керування зі зворотнім зв'язком з необхідних умов випливає рішення так званої синтез-проблеми оптимального закону керування:

$$U(t) = R[X(t), t].$$

Ця формула використовується для прямих розрахунків керуючого впливу $U(t)$ за вимірними величинами станів $X(t)$. Істотна властивість оптимального регулювання випливає з того, що закон керування є незалежним відносно можливих початкових станів X_0 . За відсутності несподіваних перешкод або неточностей моделі керування зі зворотнім зв'язком для заданих початкових станів X_0 дає однакові результати. Встановлено, що при заданій постановці проблеми використання оптимального закону керування з-за своєї універсальності та легкості обчислень з точки зору практичного використання є більш бажаним, ніж застосування оптимальної траєкторії керування.

Важливим спеціальним випадком оптимального керування динамічними системами є лінійно-квадратична (ЛК) оптимізація (ЛКО), яка пов'язана з лінійними рівняннями станів та квадратичними функціоналами якості.

1.1. Проектування оптимального лінійно-квадратичного регулятора

В загальному випадку якість системи (приведеної на рис. 1.1.) керування можна характеризувати інтегральною оцінкою (1.1).

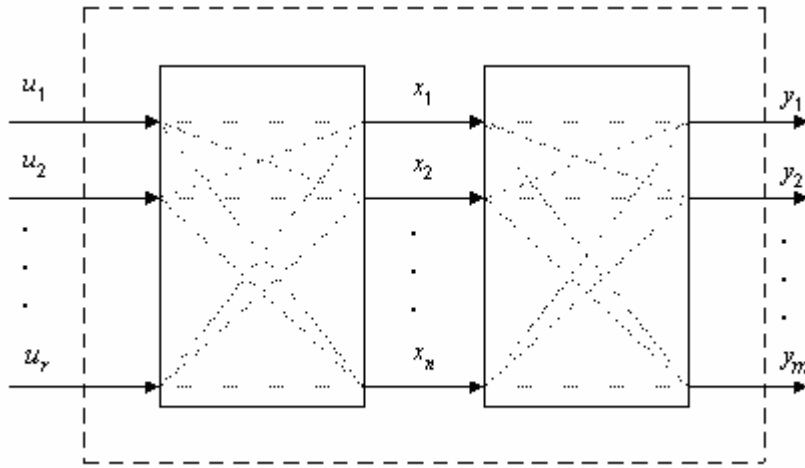


Рис. 1.1. Система керування

$$\frac{dX}{dt} = AX + BU \quad - \quad \text{рівняння стану}; \quad (1.2)$$

$$Y = CX + DU \quad - \quad \text{рівняння виходу}. \quad (1.3)$$

Тут $X = [x_1 \ x_2 \ \dots \ x_n]^T \in \mathfrak{R}^n$ – вектор змінних стану;

$U = [u_1 \ u_2 \ \dots \ u_r]^T \in \mathfrak{R}^r$ – вектор керування;

$Y = [y_1 \ y_2 \ \dots \ y_m]^T \in \mathfrak{R}^m$ – вектор параметрів, що змінюються;

A – характеристична матриця розміром $n \times n$; B – матриця розподілу керування розміром $n \times r$; C – матриця виходу розміром $m \times n$; D – матриця динаміки розміром $r \times r$. Вважається, що відомий початковий стан $X(t_0) = X_0$, де t_0 – початковий момент часу.

Потрібно забезпечити досягнення заданого цільового стану $X_d = 0$ з найменшими можливими витратами ресурсу керування (у випадку ненульового стану X_d приводиться до розгляду заміни вектора змінних стану на $X - X_d$). Якість керування визначається квадратичним критерієм якості:

$$J = \frac{1}{2} X(t_f)^T S_f X(t_f) + \frac{1}{2} \int_0^{t_f} (X^T Q X + U^T R U) dt, \quad (1.4)$$

де S_f, Q, R – матриці вагових коефіцієнтів.

Для розв'язку задачі запишемо гамільтоніан та рівняння для спряжених змінних:

$$H(X, U, \lambda, t) = \frac{1}{2} (X^T Q X + U^T R U) + \lambda^T (A X + B U), \quad (1.5)$$

$$\frac{d\lambda}{dt} = -\frac{\partial H}{\partial X} = -Q X - A^T \lambda, \quad \lambda(t_f) = S_f X(t_f). \quad (1.6)$$

Необхідна умова $\partial H / \partial U = 0$ дає рівняння $R U + B^T \lambda = 0$, або

$$U = -R^{-1} B^T \lambda. \quad (1.7)$$

Підставляючи (1.7) в (1.2), (1.6), отримаємо систему:

$$\begin{aligned} \frac{dX}{dt} &= AX - B R^{-1} B^T \lambda, \quad X(t_0) = X_0; \\ \frac{d\lambda}{dt} &= -Q X - A^T \lambda, \quad \lambda(t_f) = S_f X(t_f). \end{aligned} \quad (1.8)$$

Дана система визначає лінійну доточкову крайову задачу, для розв'язання якої застосуємо *перетворення Ріккати*:

$$\dot{\lambda} = P X, \quad (1.9)$$

де P – симетрична ($p_{ij} = p_{ji}$) додано визначена ($n \times n$) матриця.

Скористувавшись (1.9), з (1.8) отримаємо:

$$\frac{dX}{dt} = AX - B R^{-1} B^T P X, \quad (1.10)$$

$$\frac{dP}{dt} + \frac{dX}{dt} P = -Q X - A^T P X. \quad (1.11)$$

Тепер замінімо dX / dt в (1.11) його виразом (1.10):

$$\left(\frac{dP}{dt} + PA + A^T P - PBR^{-1}B^T P + Q \right) X = 0 \quad (1.12)$$

Для того щоб дане рівняння виконувалось при довільних X , необхідно, щоб обертався в нуль матричний множник при X . Таким чином, отримуємо *рівняння Ріккати*:

$$\frac{dP}{dt} = -PA - A^T P + PBR^{-1}B^T P - Q, \quad P(t_f) = S_f. \quad (1.13)$$

Умова трансверсальності отримується з порівняння (1.8) та (1.9). *Оптимальний закон керування із зворотнім зв'язком за станом*, або просто оптимальний лінійний регулятор, задається виразом:

$$U = -K(t)X, \quad (1.14)$$

де $K(t) = R^{-1}B^T P$.

Рівняння Ріккати (1.13) є матричним диференціальним нелінійним рівнянням. Аналітичний розв'язок рівняння такого виду ускладнений. Тому на практиці застосовують числові методи розв'язку (інтегрування) цих рівнянь: прямий метод інтегрування, метод Калмана-Енглара, метод діагоналізації, метод Ньютона-Рафсона і інші.

Виведені залежності, визначають лінійний оптимальний регулятор з матричним коефіцієнтом підсилення K . Цей регулятор мінімізує критерій (1.4) на траєкторіях системи (1.2), при цьому:

1. Матричний коефіцієнт підсилення K може бути визначеним поза контуром керування, так як він не залежить ні від X , ні від U ; для визначення K необхідно розв'язати (1.13) в зворотному часі.
2. При постійних матрицях A, B, R, Q і при $t \rightarrow \infty$, P прямує до усталеного значення, яке можна знайти розв'язавши алгебраїчне нелінійне матричне рівняння:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0. \quad (1.15)$$

Це в свою чергу забезпечує сталість матричного коефіцієнту підсилення регулятора K .

3. Пояснимо смисл критерію якості (1.4). Зрозуміло, що квадратичне зваження кінцевого стану дозволяє досягнути бажаної якості керування, але квадратичне зваження керування не так обґрунтовано, особливо якщо вартість ресурсів не велика. В деяких випадках квадратичне зваження заміняє собою явні обмеження на величину керуючих дій, та дозволяє отримати оптимальний закон зворотного зв'язку в аналітичному вигляді. Крім того, завдання занадто великих вагових матриць R викликає відхилення фактичного кінцевого стану від заданого, а занадто малих – призводить до дуже великих значень U . Основним обмеженням являються вимоги додатної визначеності R і неможливість завдання явних обмежень на X, U .
4. Вводячи більш загальний, ніж (1.4), критерій якості

$$J = \frac{1}{2} X(t_f)^T S_f X(t_f) + \frac{1}{2} \int_0^{t_f} (X^T Q X + U^T R U + 2X^T N U) dt, \quad (1.16)$$

в якому враховуються взаємозв'язки між керуваннями та станом, можна показати, що оптимальний регулятор визначається, як і раніше, виразом (1.4) з коефіцієнтом підсилення

$$K = R^{-1} (B^T P + N^T) \quad (1.17)$$

причому P визначається рівнянням Ріккати:

$$\frac{dP}{dt} = -A^T P - PA - (PB + N)R^{-1}(B^T P + N^T) - Q, \quad P(t_f) = S_f. \quad (1.18)$$

Це більш загальний вираз для оптимального закону зворотного зв'язку буде використано для синтезу нелінійних регуляторів.

Якщо $t \rightarrow \infty$ рівняння (1.18) набуває вигляду:

$$A^T P + PA - (PB + N)R^{-1}(B^T P + N^T) + Q = 0, \quad (1.19)$$

а критерій якості

$$J = \frac{1}{2} \int_0^{\infty} (X^T Q X + U^T R U + 2X^T N U) dt. \quad (1.20)$$

Вимоги, що до існування оптимального лінійно квадратичного регулятора:

- пара матриць $[A, B]$ має бути повністю керована;
- мають виконуватися нерівності $R > 0$, $Q - NR^{-1}N^T \geq 0$;
- пара матриць $[Q - NR^{-1}N^T, A - BR^{-1}N^T]$ не може мати спостережувані моди з власними значеннями на дійсній осі.

Слід відмітити, що при практичній реалізації алгоритмів оптимального керування часто не представляється можливим отримати оптимальні точнісні характеристики оптимізації. Це, найчастіше, обумовлене тим, що при побудові вихідної математичної моделі процесу керування звичайно використовують наближені чи апроксимуючі моделі, що враховують основні домінуючі зв'язки і в той же час відрізняються від реальних. В реальних умовах похибка алгоритмів оптимізації може зростати (особливо на великих інтервалах часу), що приведе до нестійкості алгоритмів оптимізації. Явище нестійкості алгоритмів оптимізації пов'язане з умовами реалізації рівняння Ріккати, розв'язок якого губить властивість симетричності.

Приклади

Розглянемо прилад для системи з одним входом і одним виходом третього порядку, що описується рівняннями у просторі стану:

$$\frac{dX}{dt} = A(X - Xd) + B(U - Ud), \quad X(0) = [-0,1 \quad 0,1 \quad 0,5]^T,$$

$$\text{де } A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -10 & -60 & -41 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix},$$

і квадратичний критерій якості

$$J = \int_0^{\infty} ((X - Xd)^T Q (X - Xd) + (U - Ud)^T R (U - Ud)) dt,$$

де $Xd = [0,3 \quad 0,2 \quad 0,5]^T$; $Ud = 0,3$; $Q = I$; $R = 1/2$.

Наведемо приклад програми для розв'язку цієї задачі з назвою "Lin_Reg".

```
function Lin_Reg
disp('Матриця A:')
A = [0 1 0; 0 0 1; -10 -60 -41] % Введення
матриці A
disp('Матриця B:')
B = [0; 0; 2] % Введення матриці B
% Параметри критерія якості:
disp('Матриця Q:')
Q = eye(3) % Формування одиничної матриці
disp('Матриця R:')
R = 1/2 % Введення матриці R
disp('Матриця N:')
N = [] % Введення матриці N
disp('Вектор початкових умов x0:')
x0 = [-0.1 0.1 0.3]' % Завдання вектора
початкових умов
disp('Вектор заданих значень відхилень xd:')
xd = [0.3 0.2 0.5]' % Завдання вектора заданих
значень відхилень
disp('Вектор заданих значень відхилень
керування Ud:')
ud = 0.3 % Завдання відхилень значень керування
disp('Час спостереження tf:')
```

```

tf = 30 % Завдання часу спостереження
disp('Довжина кроку dt:')
dt = 0.01 % Завдання довжини кроку
disp('Кількість кроків n:')
n = tf/dt
[K P E] = lqr(A,B,Q,R,N); % Синтез регулятора
disp('Матриця зворотнього зв'язку K:'); K
disp('Розв'язок рівняння Ріккати P:'); P
disp('Власні значення замкненої системи E:'); E
% Визначення розмірності задачі
SA = size(A); SA = SA(1); SB = size(B); SB =
SB(2);
% Формування векторів x u
x = zeros(SA,n); u = zeros(SB,n-1);
% Формування початкового стану
x(:,1) = x0;
% Знаходження оптимального керування u та змінних
стану x
for i=1:n-1,
    u(:,i) = ud-K*(x(:,i)-xd);
    x(:,i+1) = (A*(x(:,i)-xd)+B*(u(:,i)-
ud))*dt+x(:,i);
end
% Подова динаміки змінних стану
plot(0:dt:tf-dt,x), grid
title('динаміка руху змінних стану');
xlabel('Час, t');
ylabel('Змінні стану, x(t)');
legend('x_1','x_2','x_3')
% Побудова вектора керування
figure(2)
plot(dt:dt:tf-dt,u), grid
title('динаміка зміни керування'); xlabel('Час,
t'); ylabel('Керування, U(t)'); legend('u')

```

Результат виконання програми

Матриця A:

```

A =
     0     1     0
     0     0     1
    -10    -60   -41

```

Матриця B:

```

B =
     0
     0
     2

```

Матриця Q:

```

Q =
     1     0     0
     0     1     0
     0     0     1

```

Матриця R:

```

R =
    0.5000

```

Матриця N:

```

N =
    []

```

Вектор початкових умов x0:

```

x0 =

```


-0.1000
0.1000
0.3000

Вектор заданих значень відхилень x_d :

$x_d =$

0.3000
0.2000
0.5000

Вектор заданих значень відхилень керування u_d :

$u_d =$

0.3000

Час спостереження t_f :

$t_f =$

30

Довжина кроку dt :

$dt =$

0.0100

Кількість кроків n :

$n =$

3000

Матриця зворотного зв'язку:

$K =$

0.1962 0.1760 0.0530

Розв'язок рівняння Ріккати:

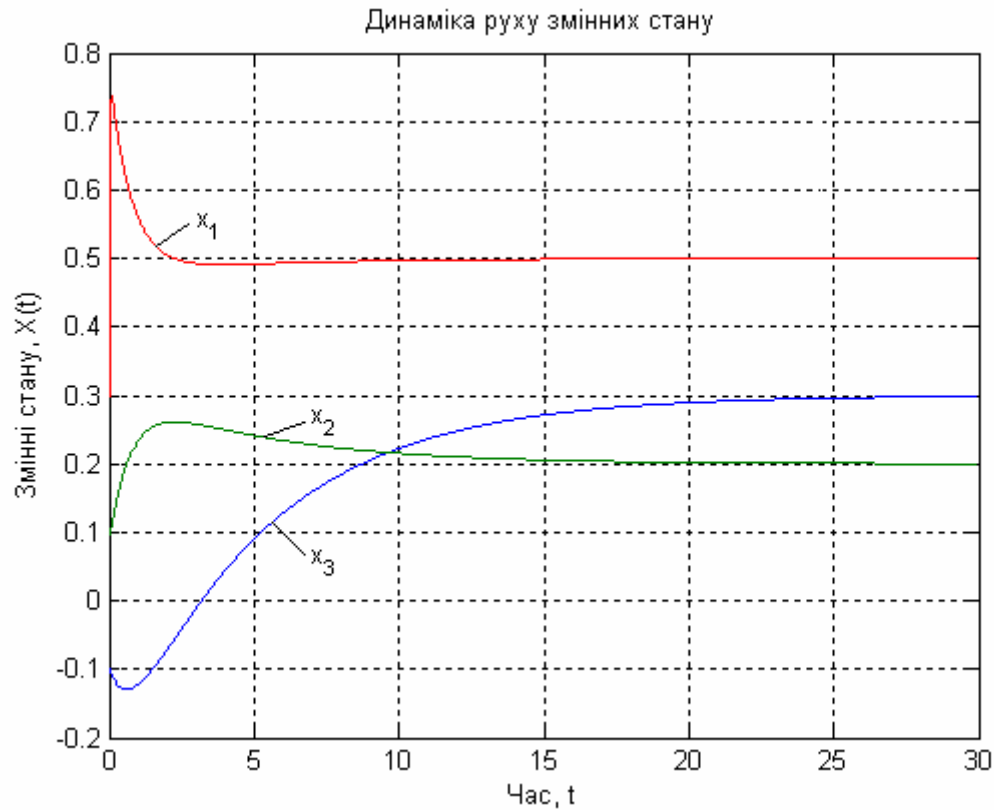
$P =$

3.3996 2.1483 0.0490
2.1483 2.5551 0.0440
0.0490 0.0440 0.0133

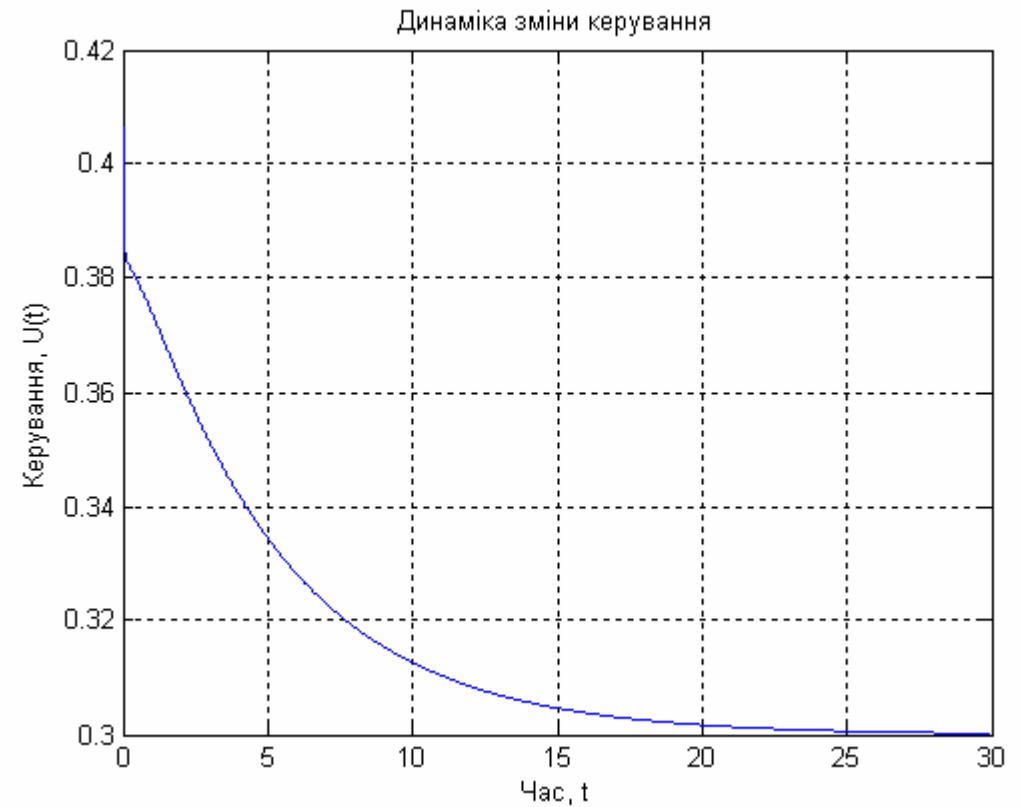
Власні значення замкненої системи:

$E =$

-39.5881
-1.3188
-0.1990



а)



б)

Рис. 1.2. Динаміка руху змінних стану а) і керування б)

Досліджували якість отриманого регулятора для наступних значеннях параметрів: $X(0) = [-0,1 \ 0,1 \ 0,5]^T$; $X_d = [0,3 \ 0,2 \ 0,5]^T$; $U_d = 0,3$; $Q = I$; $R = 1/2$; $N = 0$. Коефіцієнт підсилення K отриманий за допомогою розв'язку алгебраїчного рівняння Ріккати. Виходи оптимального регулятора, показані на рис. 1.2 а). Оптимальне керування, представлене на рис. 1.2 б), забезпечує мінімальне значення статичної похибки для X_d ,

Ud за умови, що $t_f \rightarrow \infty$, причому R вибрано так щоб для $u(t)$ виконувалася фізична умова невід'ємності. Таким чином, якість отриманого регулятора суттєво залежить від вагових коефіцієнтів Q, R, N .

Розглянемо приклад порівняння системи з одним входом і одним виходом, третього порядку для звичайних систем і для систем з врахуванням зв'язку між станом та керуванням. Система описується рівняннями у просторі стану:

$$\frac{dX}{dt} = A(X - Xd) + B(U - Ud), \quad X(0) = [-0,1 \quad 0,1 \quad 0,3]^T,$$

$$\text{де } A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -3 & -6 & -4 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix},$$

і квадратичний критерій якості

$$J = \int_0^{\infty} \left((X - Xd)^T Q (X - Xd) + (U - Ud)^T R (U - Ud) + 2(X - Xd)^T N (U - Ud) \right) dt$$

$$\text{де } Xd = [0,4 \quad 0,2 \quad 0,6]^T; \quad Ud = 0,3; \quad Q = I; \quad R = 2;$$

$$N = [1,1 \quad 0,5 \quad 0,4]^T.$$

Наведемо приклад програми для розв'язку цієї задачі з назвою "Lin_RegN".

```
function Lin_RegN
disp('Матриця A:')
A = [0 1 0; 0 0 1; -3 -6 -4] % Введення матриці A
disp('Матриця B:')
B = [0 0 2]' % Введення матриці B
% Параметри критерію якості:
disp('Матриця Q:')
Q = eye(3) % Формування одиничної матриці
disp('Матриця R:')
R = 2 % Введення матриці R
```

```
disp('Матриця N:')
N = [1.1 0.5 0.4]' % Введення матриці N
disp('Вектор початкових умов X0:')
x0 = [-0.1 0.1 0.3]' % Завдання вектора
початкових умов
disp('Вектор заданих значень відхилень Xd:')
xd = [0.4 0.2 0.6]' % Завдання вектора заданих
значень відхилень
disp('Вектор заданих значень відхилень керування
Ud:')
ud = 0.3 % Завдання відхилень значень керування
disp('Час спостереження tf:')
tf = 8 % Завдання часу спостереження
disp('Довжина кроку dt:')
dt = 0.001 % Завдання довжини кроку
disp('Кількість кроків n:')
n = tf/dt
disp('Синтез лінійного регулятора:')
[K P E] = lqr(A,B,Q,R); % Синтез регулятора
disp('Матриця зворотного зв'язку K:'); K
disp('Розв'язок рівняння Ріккати P:'); P
disp('Власні значення замкненої системи E:'); E
disp('Синтез нелінійного регулятора:')
[Kn Pn En] = lqr(A,B,Q,R,N); % Синтез регулятора
з нелінійною складовою
disp('Матриця зворотного зв'язку Kn:'); Kn
disp('Розв'язок рівняння Ріккати Pn:'); Pn
disp('Власні значення замкненої системи En:'); En
% Визначення розмірності задачі
SA = size(A); SA = SA(1); SB = size(B); SB =
SB(2);
% Формування векторів x u
x = zeros(SA,n); u = zeros(SB,n-1); xn =
zeros(SA,n); un = zeros(SB,n-1);
% Формування початкового стану
x(:,1) = x0; xn(:,1) = x0;
% Знаходження оптимального керування u та змінних
стану x
for i=1:n-1,
    u(:,i) = ud-K*(x(:,i)-xd);
```

```

x(:,i+1) = (A*(x(:,i)-xd)+B*(u(:,i)-
ud))*dt+x(:,i);
un(:,i) = ud-Kn*(xn(:,i)-xd);
xn(:,i+1) = (A*(xn(:,i)-xd)+B*(un(:,i)-
ud))*dt+xn(:,i);
end
% Побудова динаміки змінних стану
plot(0:dt:tf-dt,x), grid
hold on
plot(0:dt:tf-dt,xn,'--')
title('Динаміка руху змінних стану');
xlabel('Час, t');
ylabel('Змінні стану, X(t)');
legend('x_1', 'x_2', 'x_3', 'xn_1', 'xn_2', 'xn_3')
hold off
% Побудова вектора керування
figure(2)
plot(dt:dt:tf-dt,u), grid
hold on
plot(dt:dt:tf-dt,un,'--')
title('Динаміка зміни керування'); xlabel('Час,
t'); ylabel('Керування, U(t)'); legend('u', 'un')

```

Результат виконання програми

Матриця A:

```

A =
    0    1    0
    0    0    1
   -3   -6   -4

```

Матриця B:

```

B =
    0
    0

```

2

Матриця Q:

```

Q =
    1    0    0
    0    1    0
    0    0    1

```

Матриця R:

```

R =
    2

```

Матриця N:

```

N =
    1.1000
    0.5000
    0.4000

```

Вектор початкових умов X0:

```

x0 =
   -0.1000
    0.1000
    0.3000

```

Вектор заданих значень відхилень xd:

```

xd =
    0.4000
    0.2000
    0.6000

```

Вектор заданих значень відхилень керування Ud:

$ud =$

0.3000

Час спостереження tf :

$tf =$

8

Довжина кроку dt :

$dt =$

1.0000e-003

Кількість кроків n :

$n =$

8000

Синтез лінійного регулятора:
Матриця зворотного зв'язку K :

$K =$

0.1583 0.2788 0.1860

Розв'язок рівняння Ріккаті P :

$P =$

1.8744 1.2503 0.1583
1.2503 2.1767 0.2788
0.1583 0.2788 0.1860

Власні значення замкненої системи E :

$E =$

-1.5642 + 0.4691i
-1.5642 - 0.4691i
-1.2436

Синтез нелінійного регулятора:
Матриця зворотного зв'язку Kp :

$Kp =$

0.5976 0.4726 0.3500

Розв'язок рівняння Ріккаті Pp :

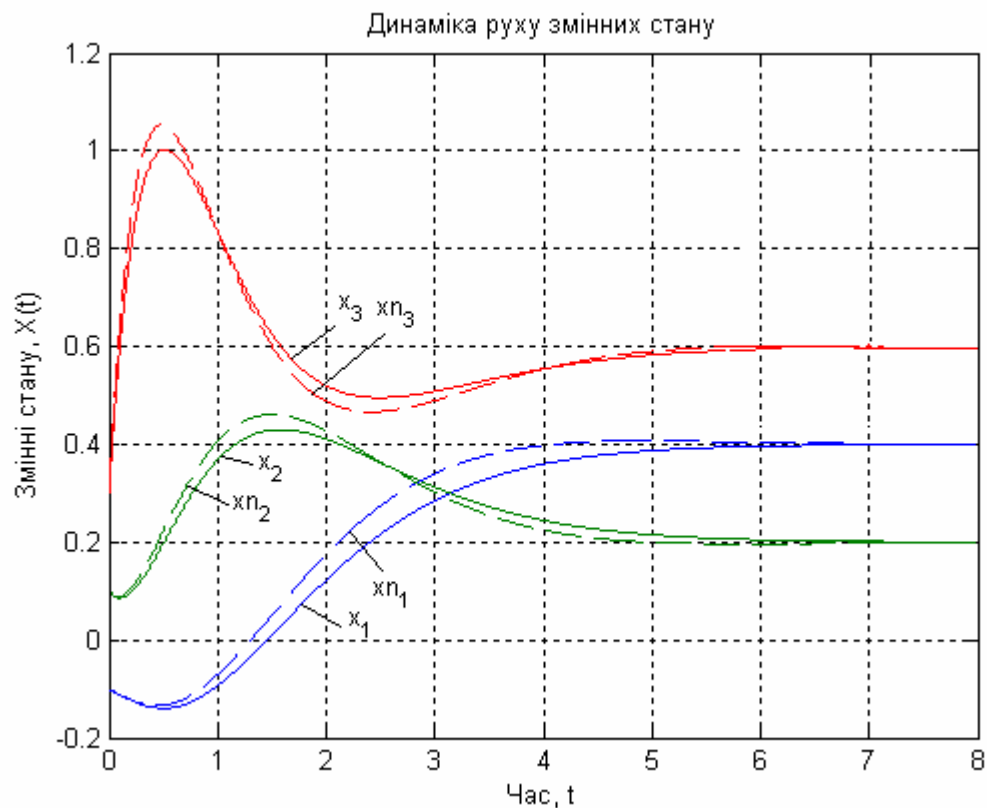
$Pp =$

1.5183 1.0589 0.0476
1.0589 2.0737 0.2226
0.0476 0.2226 0.1500

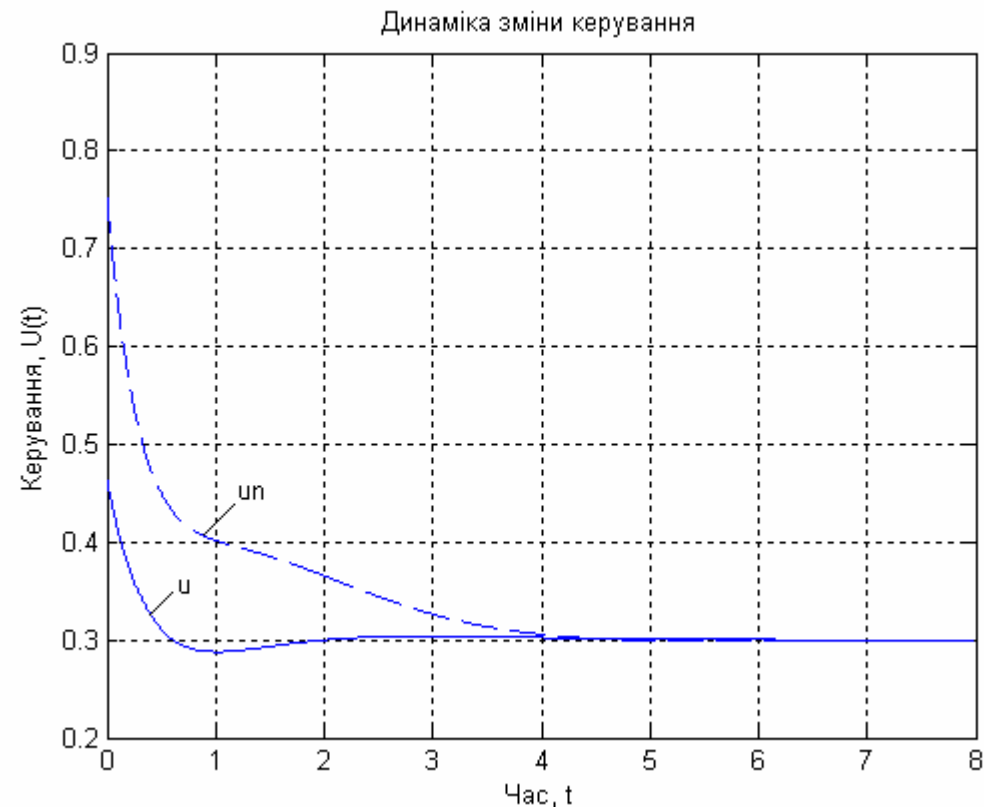
Власні значення замкненої системи E_p :

$E_p =$

-2.7068
-0.9966 + 0.7461i
-0.9966 - 0.7461i



а)



б)

Рис. 1.3. Динаміка руху змінних стану а) і керування б)

Дослідимо якість отриманих регуляторів для звичайної системи, та системи з урахуванням зв'язку між станом і керуванням (на графіках позначається штриховою кривою, а у програмі та формулах з індексом "n") наступних значеннях параметрів:

$$X(0) \equiv X_n(0) = [-0,1 \quad 0,1 \quad 0,3]^T;$$

$$Xd \equiv Xd_n = [0,4 \quad 0,2 \quad 0,6]^T; \quad Ud \equiv Ud_n = 0,3; \quad Q = I; \quad R = 2;$$

$N = [1, 1 \ 0, 5 \ 0, 4]^T$. Коефіцієнт підсилення K отриманий за допомогою розв'язку алгебраїчного рівняння Ріккати. Виходи оптимального регулятора, показані на рис. 1.3 а). Оптимальне керування, представлене на рис. 1.3 б), забезпечує мінімальне значення статичної похибки для $Xd \equiv Xd_n$, $Ud \equiv Ud_n$ за умови, що $t_f \rightarrow \infty$, причому R вибрано так щоб для $u(t)$ виконувалася фізична умова невід'ємності, а N вказує на наявність деякого взаємозв'язку між станом і керуванням. Як видно з графіків траєкторій змінних стану, якість регулятора з урахуванням взаємозв'язку між станом і керуванням по ряду параметрів краща, ніж без нього, та суттєво залежить від вагових коефіцієнтів Q, R, N .

1.2. Проектування оптимального лінійно-квадратичного регулятора за виходом системи

Для проектування лінійно-квадратичного регулятора по виходу системи, справедливі всі попередні до того твердження, окрім, що об'єкт керування описується рівняннями вигляду:

$$\frac{dX}{dt} = AX + BU; \quad (1.21)$$

$$Y = CX + DU;$$

та критерій якості

$$J = \frac{1}{2} Y(t_f)^T S_f Y(t_f) + \frac{1}{2} \int_0^{t_f} (Y^T Q Y + U^T R U + 2 Y^T N U) dt. \quad (1.22)$$

Додатково розв'язується наступне рівняння:

$$\begin{bmatrix} \bar{Q} & \bar{N} \\ \bar{N}^T & \bar{R} \end{bmatrix} = \begin{bmatrix} C^T & 0 \\ D^T & I \end{bmatrix} \begin{bmatrix} Q & N \\ N^T & R \end{bmatrix} \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}. \quad (1.23)$$

Вимоги, що до існування оптимального лінійно квадратичного регулятора:

- пара матриць $[A, B]$ має бути повністю керована;
- мають виконуватися нерівності $\bar{R} > 0$, $\bar{Q} - \bar{N}\bar{R}^{-1}\bar{N}^T \geq 0$;
- пара матриць $[\bar{Q} - \bar{N}\bar{R}^{-1}\bar{N}^T, A - B\bar{R}^{-1}\bar{N}^T]$ не може мати спостережувані моди з власними значеннями на дійсній осі.

Приклади

Розглянемо прилад для системи з одним входом і одним виходом третього порядку, що описується рівняннями у просторі стану:

$$\frac{dX}{dt} = A(X - Xd) + B(U - Ud), \quad X(0) = [-0,1 \ 0,1 \ 0,5]^T,$$

$$Y - Yd = C(X - Xd) + D(U - Ud)$$

$$\text{де } A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -10 & -60 & -41 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}; C = [0 \ 0 \ 1]; D = 0,$$

і квадратичний критерій якості

$$J = \int_0^{\infty} ((Y - Yd)^T Q (Y - Yd) + (U - Ud)^T R (U - Ud)) dt,$$

де $Xd = [0,1 \ 0,2 \ 0,3]^T$; $Yd = 0,5$; $Ud = 0,5$; $Q = I$; $R = 1/2$.

Наведемо приклад програми для розв'язку цієї задачі з назвою "Lin_Reg_Y".

```
function Lin_reg_Y;
A = [0 1 0; 0 0 1; -10 -60 -41] % Введення матриці A
B = [0; 0; 2] % Введення матриці B
C = [0 0 1] % Введення матриці C
D = 0 % Введення матриці D
% Параметри критерія якості:
```

```

disp('Матриця Q:')
Q = 1 % Введення матриці Q
disp('Матриця R:')
R = 1/2 % Введення матриці R
disp('Матриця N:')
N = [] % Введення матриці N
disp('Вектор початкових умов x0:')
x0 = [-0.1 0.1 0.5]' % Завдання вектора
початкових умов
disp('Вектор заданих значень відхилень Xd:')
xd = [0.1 0.2 0.3]' % Завдання вектора заданих
значень відхилень
disp('Вектор заданих значень відхилень
керування Ud:')
ud = 0.25 % Завдання відхилень значень керування
disp('Вектор заданих значень відхилень вихідної
величини Yd:')
yd = 0.5 % Завдання відхилень значень вихідної
величини
disp('Час спостереження tf:')
tf = 30 % Завдання часу спостереження
disp('Довжина кроку dt:')
dt = 0.01 % Завдання довжини кроку
disp('Кількість кроків n:')
n = tf/dt
disp('Модель системи у просторі станів:')
Sys = ss(A,B,C,D) %Створення моделі системи у
просторі станів
[K,P,E] = lqr(A,B,C,D,Q,R,N); % Синтез
регулятора
disp('Матриця зворотного зв'язку:'); K
disp('Розв'язок рівняння Ріккати:'); P
disp('Власні значення замкненої системи:'); E
% Визначення розмірності задачі
SA = size(A); SA = SA(1); SB = size(B); SB =
SB(2); SC = size(C); SC = SC(1);
% Формування векторів x u
x = zeros(SA,n); u = zeros(SB,n-1); y =
zeros(SC,n-1);
% Формування початкового стану
x(:,1) = x0;

```

```

% Знаходження оптимального керування u, змінних
стану x та виходу системи y
for i=1:n-1,
    u(:,i) = ud-K*(x(:,i)-xd);
    x(:,i+1) = (A*(x(:,i)-xd)+B*(u(:,i)-
ud))*dt+x(:,i);
    y(:,i) = yd+C*(x(:,i)-xd);
end
% Побудова динаміки змінних стану
figure(1)
plot(0:dt:tf-dt,x), grid
title('Динаміка руху змінних стану');
xlabel('Час, t');
ylabel('Змінні стану, X(t)');
legend('x_1','x_2','x_3')
% Побудова виходу системи
figure(2)
plot(dt:dt:tf-dt,y), grid
title('Динаміка виходу системи'); xlabel('Час,
t'); ylabel('Вихід, Y(t)'); legend('y')
% Побудова вектора керування
figure(3)
plot(dt:dt:tf-dt,u), grid
title('Динаміка зміни керування'); xlabel('Час,
t'); ylabel('Керування, U(t)'); legend('u')

```

Результат виконання програм

Матриця Q:

$Q = 1$

Матриця R:

$R = 0.5000$

Матриця N:

$N = []$

Вектор початкових умов x_0 :

$x_0 =$
-0.1000
0.1000
0.5000

Вектор заданих значень відхилень x_d :

$x_d =$
0.1000
0.2000
0.3000

Вектор заданих значень відхилень керування u_d :

$u_d =$
0.2500

Вектор заданих значень відхилень вихідної величини y_d :

$y_d =$
0.5000

Час спостереження t_f :

$t_f =$
30

Довжина кроку dt :

$dt =$
0.0100

Кількість кроків n :

$n =$
3000

Модель системи у просторі станів:

$$a = \begin{matrix} & x1 & x2 & x3 \\ x1 & 0 & 1 & 0 \\ x2 & 0 & 0 & 1 \\ x3 & -10 & -60 & -41 \end{matrix}$$

$$b = \begin{matrix} & u1 \\ x1 & 0 \\ x2 & 0 \\ x3 & 2 \end{matrix}$$

$$c = \begin{matrix} & x1 & x2 & x3 \\ y1 & 0 & 0 & 1 \end{matrix}$$

$$d = \begin{matrix} & u1 \\ y1 & 0 \end{matrix}$$

Continuous-time model.
Матриця зворотнього зв'язку:

$$K = \begin{matrix} & & & \\ & -0.0000 & 0.0082 & 0.0489 \end{matrix}$$

Розв'язок рівняння Ріккати:

$$P = \begin{matrix} & & & \\ & 0.0204 & 0.1223 & -0.0000 \\ & 0.1223 & 0.8176 & 0.0020 \\ & -0.0000 & 0.0020 & 0.0122 \end{matrix}$$

Власні значення замкненої системи:

$$E = \begin{matrix} & & & \\ & -39.5882 & & \\ & -1.3180 & & \\ & -0.1917 & & \end{matrix}$$

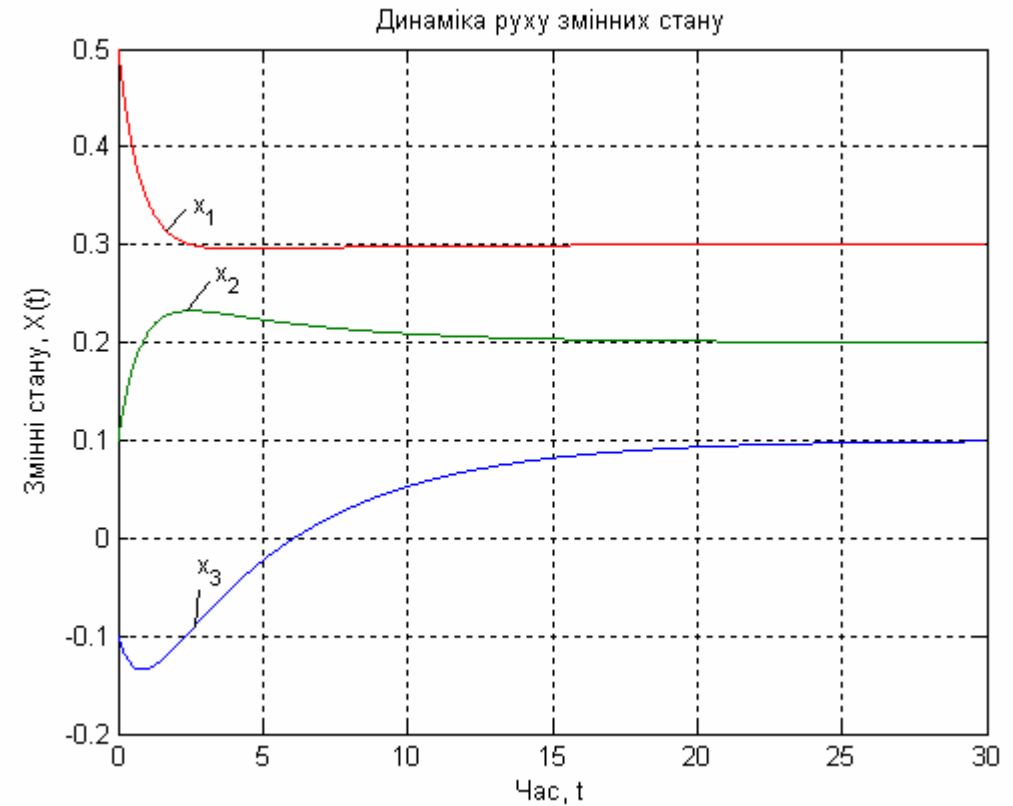
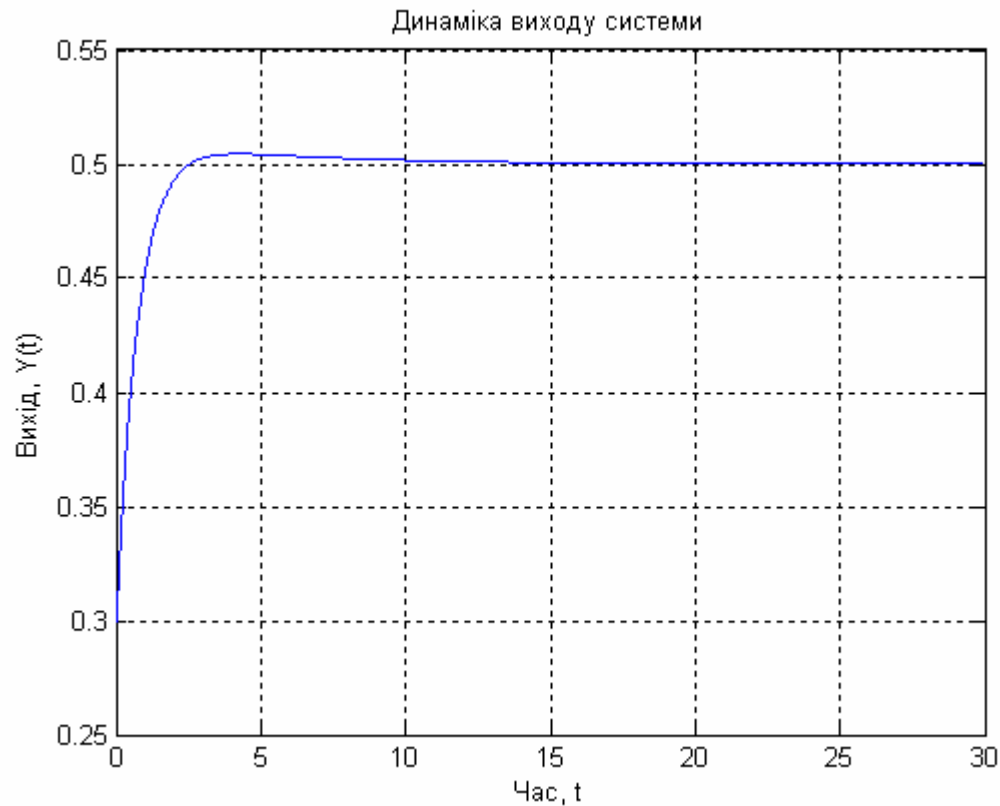
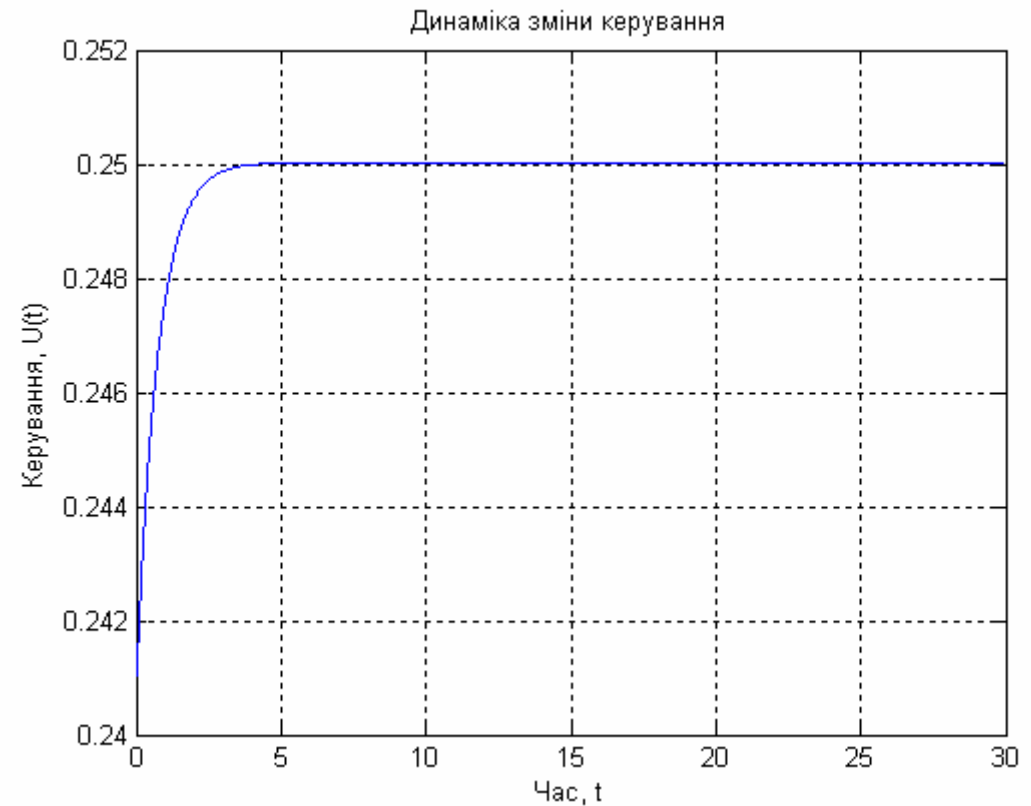


Рис. 1.4. Динаміка руху змінних стану



а)



б)

Рис. 1.5. Динаміка виходу системи а) і керування б)

Досліджували якість отриманого регулятора для наступних значеннях параметрів: $X(0) = [-0,1 \ 0,1 \ 0,5]^T$; $X_d = [0,1 \ 0,2 \ 0,3]^T$; $Y_d = 0,5$; $U_d = 0,5$; $Q = I$; $R = 1/2$. Коефіцієнт підсилення K отриманий за допомогою розв'язку алгебраїчного рівняння Ріккати. Вихід оптимального регулятора, показан на рис. 1.4, змінні стану показані на рис. 1.5 а). Оптимальне керування, представлене на рис. 1.5 б), забезпечує мінімальне

значення статичної похибки для X_d , Y_d , U_d за умови, що $t_f \rightarrow \infty$.

Розглянемо приклад порівняння системи з одним входом і одним виходом третього порядку для звичайних систем, і для систем з врахуванням зв'язку між станом та керуванням. Система описується рівняннями у просторі стану:

$$\frac{dX}{dt} = A(X - X_d) + B(U - U_d), \quad X(0) = [-0,1 \quad 0,1 \quad 0,3]^T$$

$$Y - Y_d = C(X - X_d) + D(U - U_d),$$

$$\text{де } A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -5 & -6 & -4 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}; C = [0 \quad 0 \quad 1]; D = 0,$$

і квадратичний критерій якості

$$J = \int_0^{\infty} \left((Y - Y_d)^T Q (Y - Y_d) + (U - U_d)^T R (U - U_d) + 2(Y - Y_d)^T N (U - U_d) \right) dt,$$

де $X_d = [0,4 \quad 0,2 \quad 0,6]^T$; $U_d = 0,3$; $Q = 1$; $R = 1/3$; $N = 0,2$.

Наведемо приклад програми для розв'язку цієї задачі з назвою "Lin_RegN_Y".

```
function Lin_RegN_Y
A = [0 1 0; 0 0 1; -5 -6 -4]; % Введення матриці A
B = [0 0 2]'; % Введення матриці B
C = [0 0 1]; % Введення матриці C
D = 0; % Введення матриці D
% Параметри критерія якості:
disp('Матриця Q:')
Q = 1 % Формування одиничної матриці
disp('Матриця R:')
R = 1/3 % Введення матриці R
disp('Матриця N:')
```

```
N = 0.2 % Введення матриці N
disp('Вектор початкових умов X0:')
x0 = [-0.1 0.1 0.3]' % Завдання вектора
початкових умов
disp('Вектор заданих значень відхилень Xd:')
xd = [0.4 0.2 0.6]' % Завдання вектора заданих
значень відхилень
disp('Вектор заданих значень відхилень
керування Ud:')
ud = 0.3 % Завдання відхилень значень керування
disp('Вектор заданих значень відхилень вихідної
величини Yd:')
yd = 0.5 % Завдання відхилень значень вихідної
величини
disp('Час спостереження tf:')
tf = 10 % Завдання часу спостереження
disp('Довжина кроку dt:')
dt = 0.001 % Завдання довжини кроку
disp('Кількість кроків n:')
n = tf/dt
disp('Модель системи у просторі станів:')
sys = ss(A,B,C,D) % Створення моделі системи у
просторі станів
disp('Синтез лінійного регулятора:')
[K P E] = lqru(A,B,C,D,Q,R); % Синтез регулятора
disp('Матриця зворотного зв'язку K:'); K
disp('Розв'язок рівняння Ріккати P:'); P
disp('Власні значення замкненої системи E:'); E
disp('Синтез нелінійного регулятора:')
[Kn Pn En] = lqru(A,B,C,D,Q,R,N); % Синтез
регулятора з нелінійною складовою
disp('Матриця зворотного зв'язку Kn:'); Kn
disp('Розв'язок рівняння Ріккати Pn:'); Pn
disp('Власні значення замкненої системи En:'); En
% Визначення розмірності задачі
SA = size(A); SA = SA(1); SB = size(B); SB =
SB(2); SC = size(C); SC = SC(1);
% Формування векторів x u y
x = zeros(SA,n); u = zeros(SB,n-1); y =
zeros(SC,n-1); xn = zeros(SA,n);
un = zeros(SB,n-1); yn = zeros(SC,n-1);
```

```

% Формування початкового стану
x(:,1) = x0; xn(:,1) = x0;
% Знаходження оптимального керування u та змінних
стану x
for i=1:n-1,
    u(:,i) = ud-k*(x(:,i)-xd);
    x(:,i+1) = (A*(x(:,i)-xd)+B*(u(:,i)-
ud))*dt+x(:,i);
    y(:,i) = yd+c*(x(:,i)-xd);
    un(:,i) = ud-Kn*(xn(:,i)-xd);
    xn(:,i+1) = (A*(xn(:,i)-xd)+B*(un(:,i)-
ud))*dt+xn(:,i);
    yn(:,i) = yd+c*(xn(:,i)-xd);
end
% Побудова динаміки змінних стану
plot(0:dt:tf-dt,x), grid
hold on
plot(0:dt:tf-dt,xn,'--')
xlabel('Time'); ylabel('x(t)');
legend('x1','x2','x3','x1n','x2n','x3n')
hold off
% Побудова виходу системи
figure(2)
plot(dt:dt:tf-dt,y), grid
hold on
plot(dt:dt:tf-dt,yn,'--')
xlabel('Time'); ylabel('Y(t)'); legend('y','yn')
hold off
% Побудова вектора керування
figure(3)
plot(dt:dt:tf-dt,u), grid
hold on
plot(dt:dt:tf-dt,un,'--')
xlabel('Time'); ylabel('U(t)'); legend('u','un')
hold off

```

Результат виконання програми

Матриця Q:

Q =

1

Матриця R:

R =

0.3333

Матриця N:

N =

0.2000

Вектор початкових умов x0:

x0 =

-0.1000

0.1000

0.3000

Вектор заданих значень відхилень xd:

xd =

0.4000

0.2000

0.6000

Вектор заданих значень відхилень керування Ud:

ud =

0.3000

Вектор заданих значень відхилень вихідної величини Yd:

yd =

0.5000

Час спостереження t_f :

$t_f =$

10

Довжина кроку dt :

$dt =$

1.0000e-003

Кількість кроків n :

$n =$

10000

Модель системи у просторі станів:

$a =$

	x_1	x_2	x_3
x_1	0	1	0
x_2	0	0	1
x_3	-5	-6	-4

$b =$

	u_1
x_1	0
x_2	0
x_3	2

$c =$

x_1	x_2	x_3
-------	-------	-------

$y_1 \quad 0 \quad 0 \quad 1$

$d =$

	u_1
y_1	0

Continuous-time model.

Синтез лінійного регулятора:

Матриця зворотного зв'язку K :

$K =$

-0.0000	0.5719	0.7517
---------	--------	--------

Розв'язок рівняння Ріккати P :

$P =$

0.4766	0.6264	-0.0000
0.6264	1.2763	0.0953
-0.0000	0.0953	0.1253

Власні значення замкненої системи E :

$E =$

-4.0421
-0.7307 + 0.8385i
-0.7307 - 0.8385i

Синтез нелінійного регулятора:

Матриця зворотного зв'язку K_n :

$K_n =$

0.0000	0.4154	1.1329
--------	--------	--------

Розв'язок рівняння Ріккати P_n :

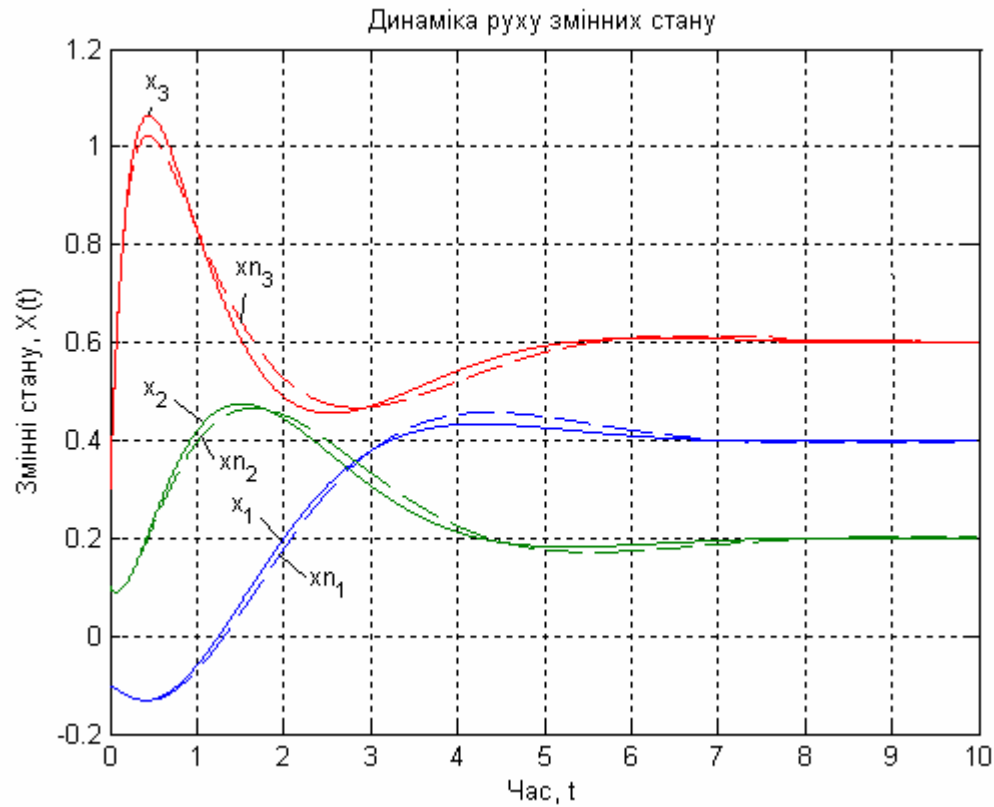
$P_n =$

0.3461	0.4441	0.0000
0.4441	0.9667	0.0692
0.0000	0.0692	0.0888

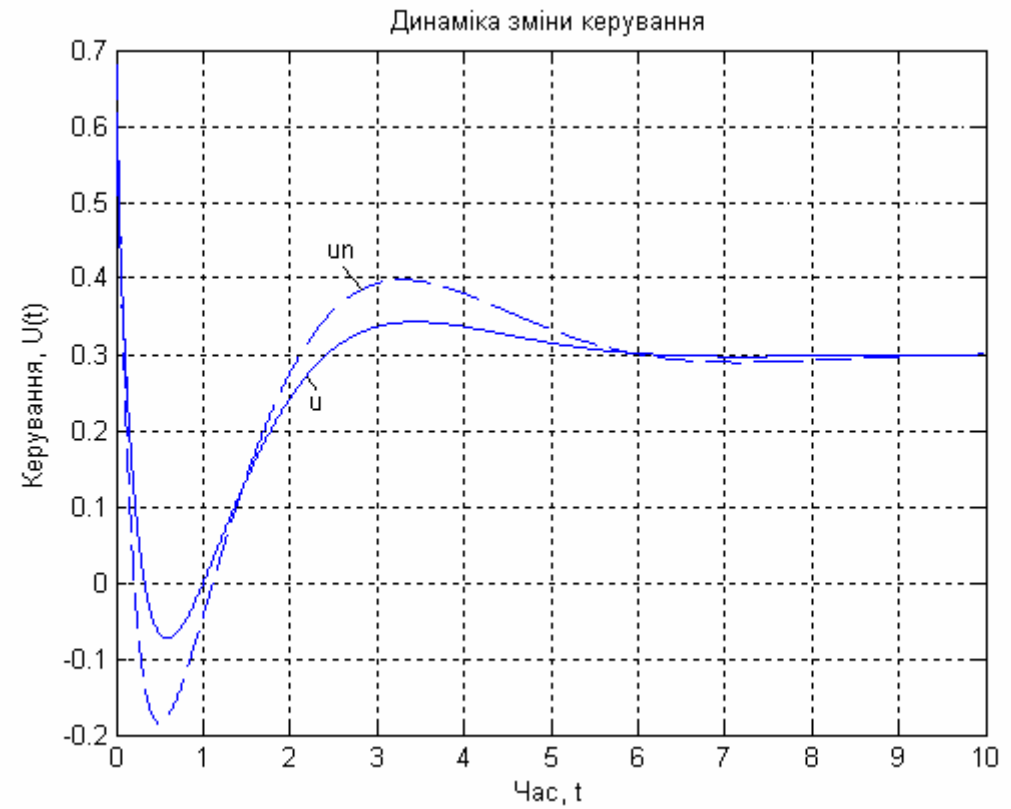
Власні значення замкненої системи E_p :

$E_p =$

-5.1231
 -0.5714 + 0.8059i
 -0.5714 - 0.8059i



а)



б)

Рис. 1.6. Динаміка руху змінних стану а) і керування б)

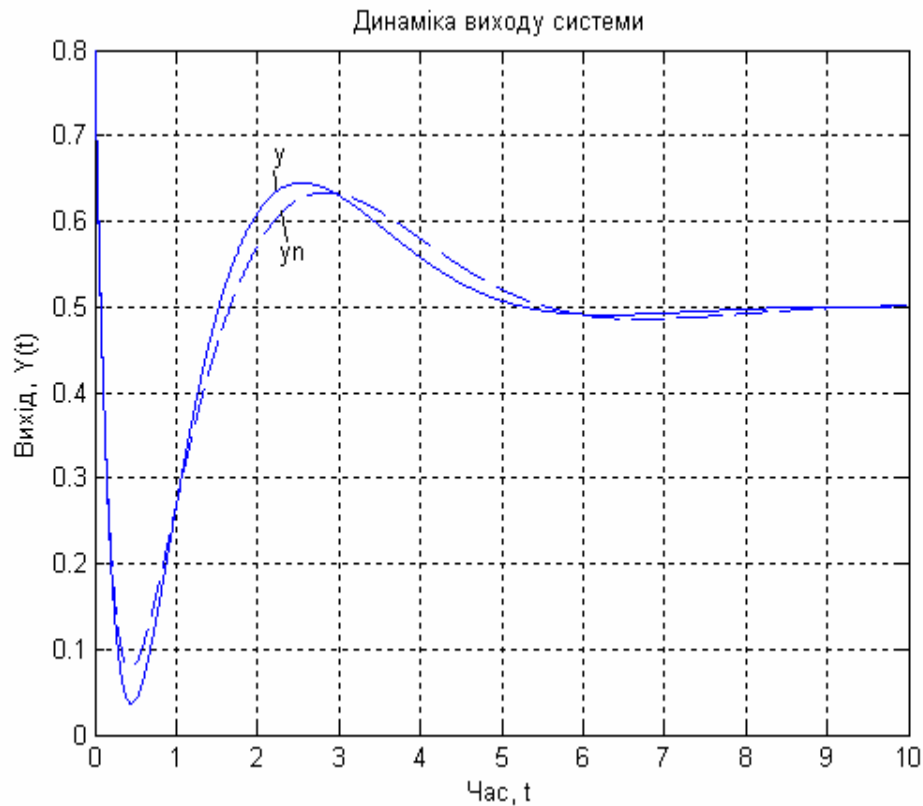


Рис. 1.7. Динаміка виходу системи

Досліджували якість отриманих регуляторів для звичайної системи та системи з урахуванням зв'язку між станом і керуванням (на графіках позначається штриховою кривою, а у програмі та формулах з індексом “n”) наступних значеннях параметрів:

$$X(0) \equiv X_n(0) = [-0,1 \quad 0,1 \quad 0,3]^T;$$

$$Xd \equiv Xd_n = [0,4 \quad 0,2 \quad 0,6]^T; Ud \equiv Ud_n = 0,3; Q = 1; R = 1/3;$$

$N = 0,2$. Коефіцієнт підсилення K отриманий за допомогою розв'язку алгебраїчного рівняння Ріккати. Виходи оптимального

регулятора, показані на рис. 1.7, змінні стану показані на рис. 1.6 а). Оптимальне керування представлено на рис. 1.6 б). Як видно з графіків динаміки виходу системи з урахуванням зв'язку між станом і керуванням, дещо менше перерегулювання ніж для звичайної системи.

1.3. Проектування оптимального лінійно-квадратичного регулятора з інтегральною складовою

Відмічено, що задачі синтезу оптимального в квадратичному значенні закону керування для лінійної системи, мають розв'язки в вигляді лінійних регуляторів. Такі регулятори дозволяють звести до нуля з плином часу вплив на вихід об'єкту, при ненульових початкових умов, чи короткострокових імпульсних впливів. Але у випадку постійних чи повільно змінюючих входних впливів, такі регулятори не можуть забезпечити вимоги рівності нулю відхилень регулюючих величин від заданих значень. Для того, щоб вони задовольняли такій вимозі, закон керування повинен мати не одну, а дві складові, одна з котрих залежить від вектора стану, а друга – від інтеграла вектора стану.

Тому, бажано переформулювати постановку задачі синтезу таким чином, щоб у керуванні з'явилась інтегральна складова, яка знижує похибки керування.

Одним способом є безпосереднє введення похідної сигналу керування в критерій якості:

$$J = \frac{1}{2} X(t_f)^T S X(t_f) + \frac{1}{2} \int_0^{t_f} \left[X^T Q X + \left(\frac{dU}{dt} \right)^T R \frac{dU}{dt} + 2 X^T N \frac{dU}{dt} \right] dt \quad (1.24)$$

При цьому необхідно продиференціювати рівняння динаміки:

$$\frac{d^2 X}{dt^2} = A \frac{dX}{dt} + B \frac{dU}{dt} \quad (1.25)$$

Після заміни змінних

$$V = \frac{dU}{dt}; \quad W_1 = X; \quad W_2 = \frac{dX}{dt}; \quad W = [W_1 \quad \vdots \quad W_2]^T \quad (1.26)$$

перейдемо до рівняння:

$$\frac{dW}{dt} = A_1 W + B_1 V, \quad (1.27)$$

$$\text{де } A_1 = \begin{bmatrix} 0 & \vdots & I \\ \cdots & \vdots & \cdots \\ 0 & \vdots & A \end{bmatrix}; \quad B_1 = \begin{bmatrix} 0 \\ \cdots \\ B \end{bmatrix}$$

та критерію якості

$$J = \frac{1}{2} [W^T S_1 W]_{t=t_f} + \frac{1}{2} \int_0^{t_f} [W^T Q_1 W + V^T R V + 2W^T N V] dt, \quad (1.28)$$

$$\text{де } Q_1 = \begin{bmatrix} Q & \vdots & 0 \\ \cdots & \vdots & \cdots \\ 0 & \vdots & 0 \end{bmatrix}; \quad S_1 = \begin{bmatrix} S & \vdots & 0 \\ \cdots & \vdots & \cdots \\ 0 & \vdots & 0 \end{bmatrix}.$$

Вираз для оптимального закону керування набуде вигляду:

$$V = -KW = -[K_1 \quad \vdots \quad K_2] \begin{bmatrix} W_1 \\ \cdots \\ W_2 \end{bmatrix} \quad (1.29)$$

або через вихідні змінні

$$U = -K_2 X - \int_0^\infty \left(K_1 - \frac{dK_2}{dt} \right) X dt, \quad (1.30)$$

що складає лінійний регулятор з інтегральною складовою.

В усталеному стані K_1 та K_2 не залежать від часу і оптимальний регулятор прийме вигляд:

$$U = -K_2 X - K_1 \int_0^\infty X dt, \quad (1.31)$$

тобто лінійний регулятор з інтегральною складовою.

Приклад

Розглянемо приклад для системи з одним входом і одним виходом третього порядку, що описується рівняннями у просторі стану:

$$\frac{dX}{dt} = A(X - Xd) + B(U - Ud), \quad X(0) = [-0,1 \quad 0,1 \quad 0,5]^T,$$

$$\text{де } A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -6 & -4 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix},$$

і квадратичний критерій якості

$$J = \int_0^\infty \left[(X - Xd)^T Q (X - Xd) + \left(\frac{dU}{dt} \right)^T R \frac{dU}{dt} \right] dt,$$

де $Xd = [0,1 \quad 0,2 \quad 0,3]^T$; $Ud = 0,3$; $Q = I$; $R = 1/2$.

Наведемо приклад програми для розв'язку цієї задачі з назвою "Lin_Reg_Int".

```
function Lin_reg_Int
disp('матриця А:')
A = [0 1 0; 0 0 1; -1 -6 -4] % введення матриці А
disp('матриця В:')
B = [0; 0; 2] % введення матриці В
% Параметри критерія якості:
disp('матриця Q:')
```

```

Q = 1/20000*eye(3) % Формування одиничної матриці
disp('Матриця R:')
R = 1/2 % Введення матриці R
disp('Матриця N:')
N = [] % Введення матриці N
disp('Вектор початкових умов x0:')
x0 = [0.1 0.1 0.1]' % Завдання вектора початкових
умов
disp('Вектор заданих значень відхилень xd:')
xd = [0.1 0.2 0.3]' % Завдання вектора заданих
значень відхилень
disp('Вектор заданих значень відхилень
керування Ud:')
ud = 0.3 % Завдання відхилень значень керування
disp('Час спостереження tf:')
tf = 30 % Завдання часу спостереження
disp('Довжина кроку dt:')
dt = 0.01 % Завдання довжини кроку
disp('Кількість кроків n:')
n = tf/dt
disp('Матриця A1:')
A1 = [zeros(3) eye(3); zeros(3) A] % Формування
матриці A1
disp('Матриця B1:')
B1 = [zeros(3,1); B] % Формування матриці B1
disp('Матриця Q1:')
Q1 = [Q zeros(3); zeros(3) zeros(3)] % Формування
матриці Q1
[K P E] = lqr(A1,B1,Q1,R,N); % Синтез регулятора
disp('Матриця зворотнього з язку:'); K
disp('Розв язок рівняння Ріккати:'); P
disp('Власні значення замкненої системи:'); E
% Визначення розмірності задачі
SA = size(A); SA = SA(1); SB = size(B); SB =
SB(2); SK = size(K); SK = SK(2);
% Формування векторів x, u та v
x = zeros(SA,n); u = zeros(SB,n-1); v =
zeros(SA,n);
% Формування початкового стану
x(:,1) = x0;

```

```

% Відокремлення пропорційної та інтегральної
частин матриці K
for i=SK/2:SK,
    K2(i) = K(i);
end
for i=1:SK/2,
    K1(i) = K(i);
    K2(1) = [];
end
% Знаходження оптимального керування u та змінних
стану x
for i=1:n-1,
    v(:,i) = v(:,i)+K1*x(:,i)*dt;
    u(:,i) = ud-K2*(x(:,i)-xd)+v(i);
    x(:,i+1) = (A*(x(:,i)-xd)+B*(u(:,i)-
ud))*dt+x(:,i);
end
% Побудова динаміки змінних стану
plot(0:dt:tf-dt,x), grid
title('Динаміка руху змінних стану');
xlabel('Час, t');
ylabel('Змінні стану, x(t)');
legend('x_1', 'x_2', 'x_3')
% Побудова вектора керування
figure(2)
plot(dt:dt:tf-dt,u), grid
title('Динаміка зміни керування'); xlabel('Час,
t'); ylabel('Керування, U(t)'); legend('u')

```

Результат виконання програми

Матриця A:

```

A =
    0    1    0
    0    0    1
   -1   -6   -4

```

Матриця B:

```

B =

```

0
0
2

Матриця Q:

Q =

1.0e-004 *

0.5000	0	0
0	0.5000	0
0	0	0.5000

Матриця R:

R =

0.5000

Матриця N:

N =

[]

Вектор початкових умов x0:

x0 =

0.1000
0.1000
0.1000

Вектор заданих значень відхилень xd:

xd =

0.1000
0.2000
0.3000

Вектор заданих значень відхилень керування Ud:

ud =

0.3000

Час спостереження tf:

tf =

30

Довжина кроку dt:

dt =

0.0100

Кількість кроків n:

n =

3000

Матриця A1:

A1 =

0	0	0	1	0	0
0	0	0	0	1	0
0	0	0	0	0	1
0	0	0	0	1	0
0	0	0	0	0	1
0	0	0	-1	-6	-4

Матриця B1:

B1 =

0

0
0
0
0
2

Матриця Q1:

Q1 =

1.0e-004 *

0	0.5000	0	0	0
0	0	0.5000	0	0
0	0	0	0.5000	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Матриця зворотного зв'язку:

K =

0.0100	-0.0000	-0.0000	0.0576
0.0382	0.0095		

Розв'язок рівняння Ріккати:

P =

1.0e+007 *

0.0000	-0.0000	0.0000	0.0000
0.0000	0.0000		

-0.0000	0.2368	1.8843	-0.2368	-
1.8843	-0.0000			
0.0000	1.8843	0.0542	-1.8843	-
0.0542	-0.0000			
0.0000	-0.2368	-1.8843		0.2368
1.8843	0.0000			
0.0000	-1.8843	-0.0542		1.8843
0.0542	0.0000			
0.0000	-0.0000	-0.0000		0.0000
0.0000	0.0000			

Власні значення замкненої системи:

E =

-1.9053 + 1.2837i
-1.9053 - 1.2837i
-0.1884
-0.0201
-0.0000
-0.0000

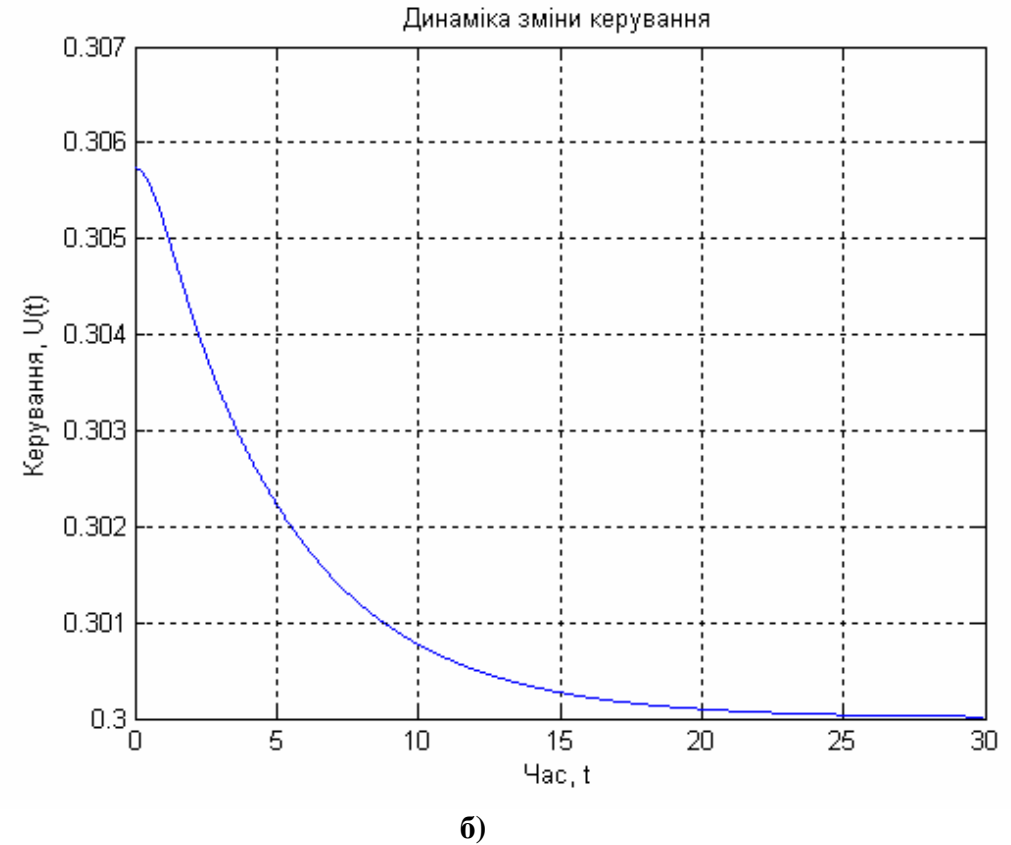
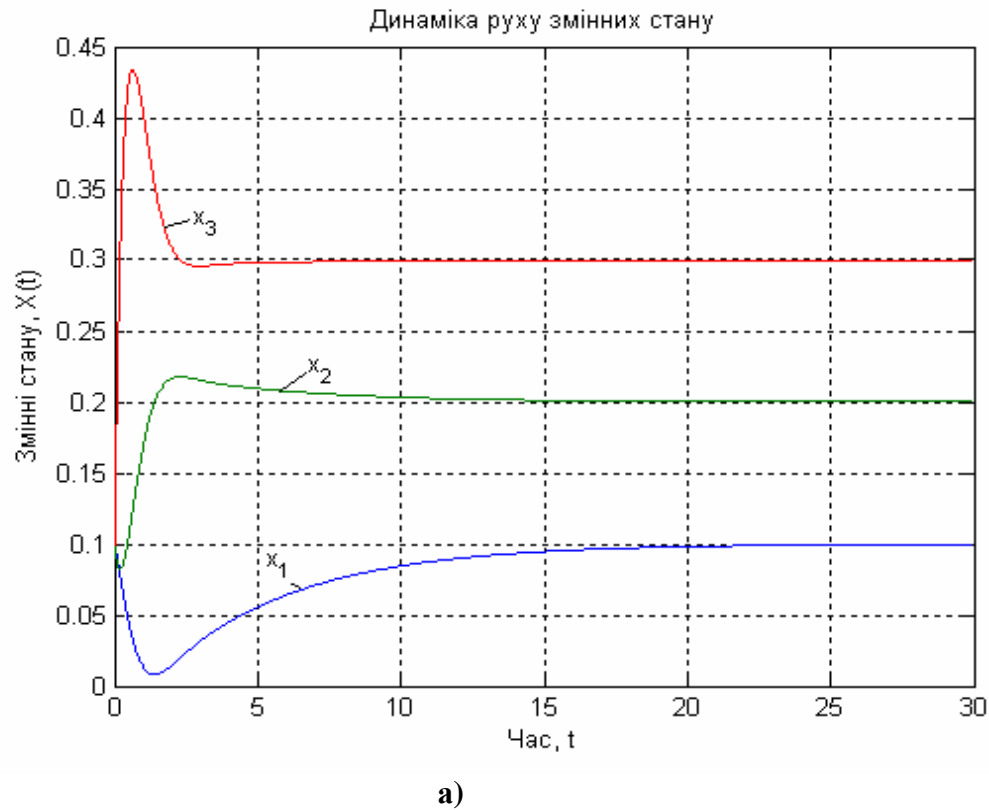


Рис. 1.8. Динаміка руху змінних стану а) і керування б)

Отримано лінійний регулятор з інтегральною складовою для наступних значеннях параметрів: $X(0) = [-0,1 \ 0,1 \ 0,5]^T$; $X_d = [0,1 \ 0,2 \ 0,3]^T$; $U_d = 0,3$; $Q = I$; $R = 1/2$; $N = 0$. Коефіцієнт підсилення K отриманий за допомогою розв'язку алгебраїчного рівняння Ріккати. Виходи оптимального регулятора, показані на рис. 1.8 а). Оптимальне керування представлено на рис. 1.8 б).

Якщо порівняти лінійний оптимальний регулятор з інтегральною складовою з звичайним ПІ – регулятором, то результати моделювання свідчать про значні переваги оптимального. Звичайні регулятори характеризуються значним перерегулюванням, в той же час оптимальна система практично не зазнає впливу збурень.

1.4. Проектування оптимального дискретного лінійно-квадратичного регулятора

Розглянемо дискретну систему, яка характеризується рівнянням:

$$X[n+1] = AX[n] + BU[n], \quad X[0] = X_0, \quad n = 0, 1, \dots, n_f, \quad (1.32)$$

де A і B можуть бути функціями k .

Функція якості

$$J = \frac{1}{2} X[n_f]^T SX[n_f] + \frac{1}{2} \sum_{n=0}^{n_f-1} (X[n]^T QX[n] + U[n]^T RU[n]), \quad (1.33)$$

де вагові матриці Q та R можуть бути функціями n .

Далі запишемо гамільтоніан

$$H[n] = \frac{1}{2} X[n]^T QX[n] + \frac{1}{2} U[n]^T RU[n] + \lambda[n+1]^T (AX[n] + BU[n]) \quad (1.34)$$

Спряжене векторне рівняння запишеться у вигляді:

$$\lambda[n] = QX[n] + A^T \lambda[n+1] \quad (1.35)$$

Видно, що $\lambda[n+1]$ не можна виразити через $\lambda[n]$, якщо A^{-1} не існує, так як A - матриця переходів стану, то вона завжди обернена.

Оскільки кінцевий стан не визначено, умова трансверсальності визначається з

$$\left[\lambda[n_f]^T - \frac{\partial G[n_f]}{\partial X[n_f]} \right]^T \omega[n_f] = 0 \quad (1.36)$$

$$\text{як } \lambda[n_f] = SX[n_f], \text{ маємо } \frac{\partial H}{\partial U[n]} = RU[n] + B^T \lambda[n+1] = 0.$$

Таким чином, для отримання оптимального лінійного керування за замкненим контуром, необхідно розв'язати лінійні різниці рівняння:

$$\begin{aligned} X[n+1] &= AX[n] - BR^{-1}B^T \lambda[n+1], \quad X[n_0] = X_0 \\ \lambda[n] &= QX[n] + A^T \lambda[n+1], \quad \lambda[n_f] = SX[n_f] \end{aligned} \quad (1.37)$$

Припустимо, що розв'язок цих рівнянь має вигляд:

$$\lambda[n] = P[n]X[n]. \quad (1.38)$$

Для виключення λ підставимо (1.38) в (1.37). Це дасть

$$\begin{aligned} X[n+1] &= AX[n] - BR^{-1}B^T P[n+1]X[n+1], \\ P[n]X[n] &= QX[n] + A^T P[n+1]X[n+1]. \end{aligned} \quad (1.39)$$

Розв'язавши ці рівняння відносно $X[n+1]$ та виключаємо цю змінну з останніх двох співвідношень, отримаємо:

$$P[n]X[n] = QX[n] + A^T P[n+1] (I + BR^{-1}B^T P[n+1])^{-1} AX[n], \quad (1.40)$$

що задовольняє для довільного $X[n]$, якщо тільки

$$\begin{aligned} P[n] &= Q + A^T P[n+1] (I + BR^{-1}B^T P[n+1])^{-1} A = \\ &= Q + A^T (P[n+1]^{-1} + BR^{-1}B^T)^{-1} A, \end{aligned} \quad (1.41)$$

за умови на кінцевому кроці

$$P[n_f] = S. \quad (1.42)$$

При розв'язанні матричних рівнянь Рікати в зворотному часі у напрямку від $n = n_f$ до $n = 0$, будуть отримані деякі функції «підсилення», які побудовані після попереднього розрахунку та які використовуються у фізичній системі за мірою їх отримання в реальному часі.

Рівняння по замкненому контуру записується у вигляді:

$$U[n] = -R^{-1}B^T A^{-T} (P[n] - Q)X[n] = -R^{-1}B^T (P[n+1] + BR^{-1}B^T)^T AX[n], \quad (1.43)$$

що досить схоже з методом отримання оптимального керування по замкненому контуру, для неперервного лінійного регулятора. Більша частина зауважень, зроблених відносно лінійного регулятора, може бути виконана і тут.

Скористаємось рівнянням

$$\begin{aligned} X[n+1] &= AX[n] - BR^{-1}B^T P[n+1]X[n+1] \Rightarrow \\ \Rightarrow X[n+1] &= (I + BR^{-1}B^T P[n+1])^{-1} AX[n]. \end{aligned} \quad (1.44)$$

Тоді оптимальне керування має вигляд:

$$\begin{aligned} U[n] &= -R^{-1}B^T P[n+1]X[n+1] = -R^{-1}B^T P[n+1](I + BR^{-1}B^T P[n+1])^{-1} AX[n] = \\ &= -R^{-1}(I + BR^{-1}B^T P[n+1])^{-1} B^T P[n+1]AX[n] = \\ &= -(R + B^T P[n+1]B)^{-1} B^T P[n+1]AX[n], \end{aligned} \quad (1.45)$$

або можна представити у вигляді

$$U[n] = -KX[n], \quad (1.46)$$

де K – коефіцієнт підсилення, що визначається:

$$K = (R + B^T P[n+1]B)^{-1} B^T P[n+1]A. \quad (1.47)$$

Можна також показати, що дискретне матричне рівняння Рікати має вигляд:

$$\begin{aligned} P[n] &= A^T P[n+1]A - P[n+1] - \\ &- A^T P[n+1]B (B^T P[n+1]B + R)^{-1} B^T P[n+1]A + Q, \quad P[n_f] = S. \end{aligned} \quad (1.48)$$

Вводячи більш загальний, ніж (1.33), критерій якості

$$J = \frac{1}{2} X[n_f]^T S X[n_f] + \frac{1}{2} \sum_{n=0}^{n_f-1} (X[n]^T Q X[n] + U[n]^T R U[n] + 2X[n]^T N U[n]), \quad (1.49)$$

в якому враховується взаємозв'язки між керуваннями та станом, можна показати, що оптимальний регулятор визначається, як і раніше, виразом (1.46) з коефіцієнтом підсилення

$$K = (B^T P[n+1]B + R)^{-1} (B^T P[n+1]A + N^T) \quad (1.50)$$

причому $P[n]$ визначається рівнянням Рікати:

$$\begin{aligned} P[n] &= A^T P[n+1]A - P[n+1] - \\ &- (A^T P[n+1]B + N) (B^T P[n+1]B + R)^{-1} (B^T P[n+1]A + N^T) + Q, \quad P[n_f] = S. \end{aligned} \quad (1.51)$$

Це більш загальний вираз для оптимального закону зворотного зв'язку, буде використано для синтезу нелінійних регуляторів.

Якщо $n_f \rightarrow \infty$ рівняння (1.51) набуває вигляду:

$$A^T P A - P - (A^T P B + N) (B^T P B + R)^{-1} (B^T P A + N^T) + Q = 0, \quad (1.52)$$

а критерій якості

$$J = \frac{1}{2} \sum_{n=0}^{\infty} (X[n]^T Q X[n] + U[n]^T R U[n] + 2X[n]^T N U[n]) \quad (1.53)$$

Вимоги, що до існування оптимального дискретного лінійно-квадратичного регулятора, такі ж, як і до неперервного.

Приклади

Розглянемо приклад для системи з одним входом і одним виходом третього порядку, що описується рівняннями у просторі стану:

$$X[n+1] = A(X[n] - Xd) + B(U[n] - Ud),$$

$$X[0] = [0, 1, 0, 3]^T,$$

$$\text{де } A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -5 & -10 & -8,5 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix},$$

і квадратичний критерій якості

$$J = \sum_{n=1}^{\infty} ((X[n] - Xd)^T Q (X[n] - Xd) + (U[n] - Ud)^T R (U[n] - Ud))$$

$$\text{де } Xd = [-0,4 \quad -0,2 \quad 0,6]^T; \quad Ud = 0,5; \quad Q = I; \quad R = 1/4; \\ T = 0,2.$$

Наведемо приклад програми для розв'язку цієї задачі з назвою "Discrete_Lin_Reg".

```
function Discrete_Lin_Reg
disp('Матриця A:')
A = [0 1 0; 0 0 1; -5 -10 -8.5] % Введення
матриці A
disp('Матриця B:')
B = [0; 0; 2] % Введення матриці B
% Параметри критерія якості:
```

```
disp('Матриця Q:')
Q = eye(3) % Формування одиничної матриці
disp('Матриця R:')
R = 1/4 % Введення матриці R
disp('Матриця N:')
N = [] % Введення матриці N
disp('Вектор початкових умов x0:')
x0 = [0.1 0.2 0.3]' % Завдання вектора початкових
умов
disp('Вектор заданих значень відхилень Xd:')
xd = [-0.4 -0.2 0.6]' % Завдання вектора заданих
значень відхилень
disp('Вектор заданих значень відхилень
керування Ud:')
ud = 0.5 % Завдання відхилень значень керування
disp('Час спостереження tf:')
tf = 10 % Завдання часу спостереження
disp('Період квантування T:')
T = 0.2 % Завдання періоду квантування
disp('Кількість кроків n:')
n = tf/T
[K P E] = dlqr(A,B,Q,R,N); % Синтез регулятора
disp('Матриця зворотного зв'язку K:'); K
disp('Розв'язок рівняння Ріккати P:'); P
disp('Власні значення замкненої системи E:'); E
% Визначення розмірності задачі
SA = size(A); SA = SA(1); SB = size(B); SB =
SB(2);
% Формування векторів x u
x = zeros(SA,n); u = zeros(SB,n-1);
% Формування початкового стану
x(:,1) = x0;
% Знаходження оптимального керування u та змінних
стану x
for i=1:n-1,
    u(:,i) = ud-K*(x(:,i)-xd);
    x(:,i+1) = (A*(x(:,i)-xd)+B*(u(:,i)-
ud))*T+x(:,i);
end
% Побудова динаміки змінних стану
stem(0:T:tf-T,x'), grid
```



```

title('Динаміка руху змінних стану');
xlabel('Час, nT');
ylabel('Змінні стану, X(nT)');
legend('x_1', 'x_2', 'x_3')
% Побудова вектора керування
figure(2)
stem(T:T:tf-T,u); grid on
title('Динаміка зміни керування'); xlabel('Час,
nT'); ylabel('Керування, U(nT)'); legend('u')

```

Результат виконання програми

Матриця A:

```

A =
     0     1.0000     0
     0     0     1.0000
    -5.0000   -10.0000   -8.5000

```

Матриця B:

```

B =
     0
     0
     2

```

Матриця Q:

```

Q =
     1     0     0
     0     1     0
     0     0     1

```

Матриця R:

```

R =
     0.2500

```

Матриця N:

```

N =
     []

```

Вектор початкових умов x0:

```

x0 =
     0.1000
     0.2000
     0.3000

```

Вектор заданих значень відхилень xd:

```

xd =
    -0.4000
    -0.2000
     0.6000

```

Вектор заданих значень відхилень керування ud:

```

ud =
     0.5000

```

Час спостереження tf:

```

tf =
     10

```

Період квантування T:

$T =$

0.2000

Кількість кроків n :

$n =$

50

Матриця зворотного зв'язку K :

$K =$

-2.4840 -4.8427 -3.9037

Розв'язок рівняння Ріккати P :

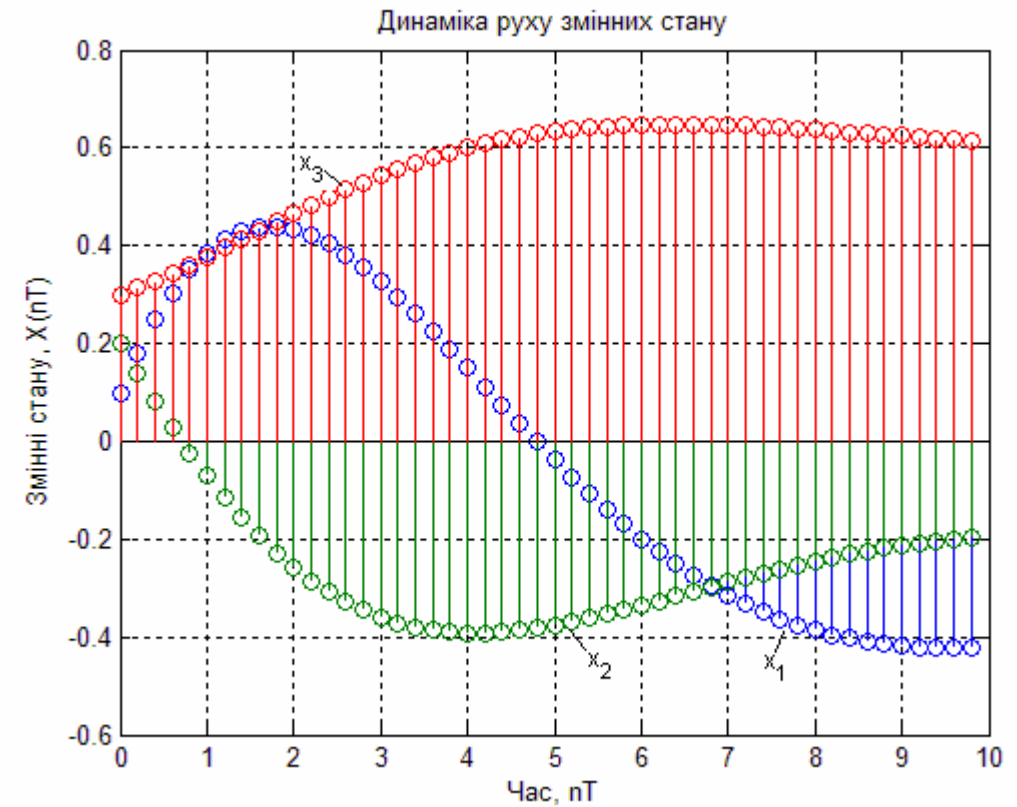
$P =$

2.5525 3.0267 2.4398
3.0267 8.8384 6.2166
2.4398 6.2166 9.6807

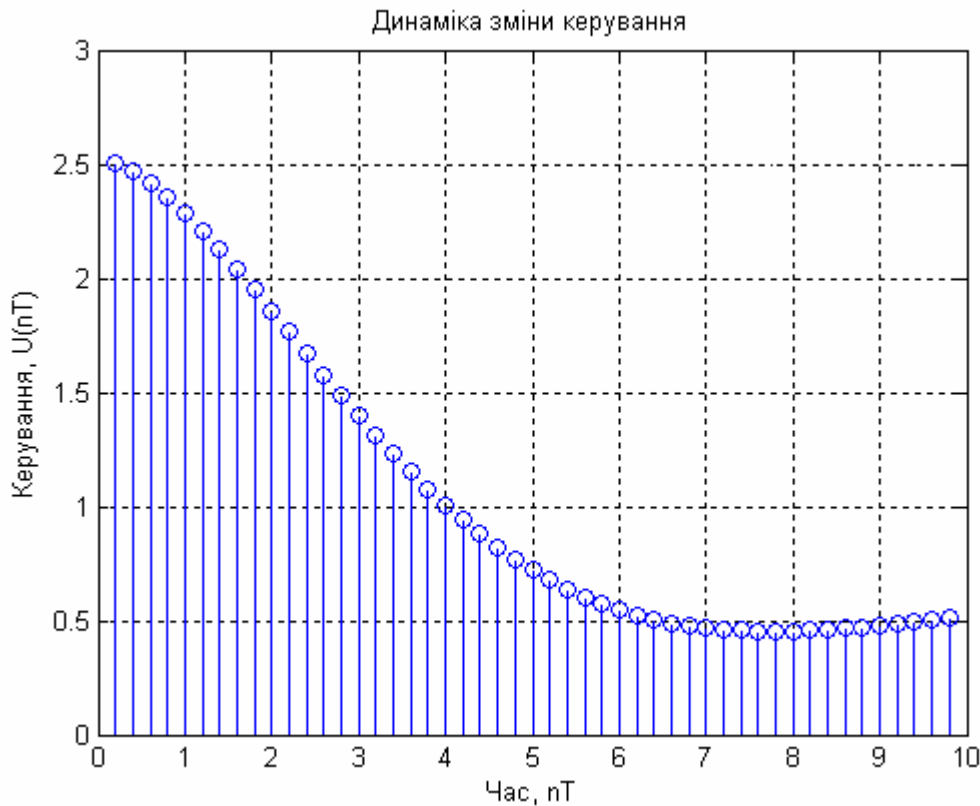
Власні значення замкненої системи E :

$E =$

-0.1337
-0.2794 + 0.4022i
-0.2794 - 0.4022i



а)



б)

Рис. 1.9. Динаміка зміни змінних стану а) і керування б)

Розраховали якість отриманого регулятора для наступних значеннях параметрів: $X[0] = [0,1 \ 0,2 \ 0,3]^T$;

$Xd = [-0,4 \ -0,2 \ 0,6]^T$; $Ud = 0,5$; $Q = I$; $R = 1/4$; $T = 0,2$.

Коефіцієнт підсилення K отриманий за допомогою розв'язку алгебраїчного рівняння Ріккаті. Виходи оптимального регулятора, показані на рис. 1.9 а). Оптимальне керування представлено на рис. 1.9 б).

Розглянемо приклад порівняння системи з одним входом і одним виходом третього порядку для звичайних систем, і для систем з врахуванням зв'язку між станом та керуванням. Система описується рівняннями у просторі стану:

$$X[n+1] = A(X[n] - Xd) + B(U[n] - Ud),$$

$$X[0] = [-0,1 \ 0,2 \ 0,3]^T,$$

$$\text{де } A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -3,5 & -6 & -3,5 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix},$$

і квадратичним критерієм якості

$$J = \sum_{n=0}^{\infty} \left((X[n] - Xd)^T Q (X[n] - Xd) + (U[n] - Ud)^T R (U[n] - Ud) + 2(X[n] - Xd)^T N (U[n] - Ud) \right),$$

де $Xd = [0,2 \ 0,3 \ 0,1]^T$; $Ud = 0,4$; $Q = I$; $R = 2$;
 $N = [0,1 \ 0,25 \ 0,3]^T$; $T = 0,4$.

Наведемо приклад програми для розв'язку цієї задачі з назвою "Discrete_Lin_RegN".

```
function Discrete_Lin_RegN
disp('Матриця A:')
A = [0 1 0; 0 0 1; -3.5 -6 -3.5] % Введення
матриці A
disp('Матриця B:')
B = [0; 0; 2] % Введення матриці B
% Параметри критерія якості:
disp('Матриця Q:')
Q = eye(3) % Формування одиничної матриці
disp('Матриця R:')
R = 2 % Введення матриці R
```

```

disp('Матриця N:')
N = [0.1 0.25 0.3]' % Введення матриці N
disp('Вектор початкових умов x0:')
x0 = [-0.1 0.2 0.3]' % Завдання вектора
початкових умов
disp('Вектор заданих значень відхилень xd:')
xd = [0.2 0.3 0.1]' % Завдання вектора заданих
значень відхилень
disp('Вектор заданих значень відхилень
керування Ud:')
ud = 0.4 % Завдання відхилень значень керування
disp('Час спостереження tf:')
tf = 16 % Завдання часу спостереження
disp('Період квантування T:')
T = 0.4 % Завдання періоду квантування
disp('Кількість кроків n:')
n = tf/T
[K P E] = d1qr(A,B,Q,R); % Синтез регулятора
disp('Матриця зворотного зв'язку K:'); K
disp('Розв'язок рівняння Ріккати P:'); P
disp('Власні значення замкненої системи E:'); E
disp('Синтез нелінійного регулятора:')
[Kn Pn En] = d1qr(A,B,Q,R,N); % Синтез регулятора
з нелінійною складовою
disp('Матриця зворотного зв'язку Kn:'); Kn
disp('Розв'язок рівняння Ріккати Pn:'); Pn
disp('Власні значення замкненої системи En:'); En
% Визначення розмірності задачі
SA = size(A); SA = SA(1); SB = size(B); SB =
SB(2);
% Формування векторів x u
x = zeros(SA,n); u = zeros(SB,n-1); xn =
zeros(SA,n); un = zeros(SB,n-1);
% Формування початкового стану
x(:,1) = x0; xn(:,1) = x0;
% Знаходження оптимального керування u та змінних
стану x
for i=1:n-1,
    u(:,i) = ud-K*(x(:,i)-xd);
    x(:,i+1) = (A*(x(:,i)-xd)+B*(u(:,i)-
ud))*T+x(:,i);

```

```

    un(:,i) = ud-Kn*(xn(:,i)-xd);
    xn(:,i+1) = (A*(xn(:,i)-xd)+B*(un(:,i)-
ud))*T+xn(:,i);
end
% Побудова динаміки змінних стану
stem(0:T:tf-T,x'), grid
hold on
stem(0:T:tf-T,xn','--','marker','square')
title('Динаміка руху змінних стану');
xlabel('Час, nT');
ylabel('Змінні стану, X(nT)');
legend('x_1','x_2','x_3','xn_1','xn_2','xn_3')
hold off
% Побудова вектора керування
figure(2)
stem(T:tf-T,[u;un]'); grid on
title('Динаміка зміни керування'); xlabel('Час,
nT'); ylabel('Керування, U(nT)');
legend('u','un')

```

Результат виконання програми

Матриця A:

```

A =
     0     1.0000     0
     0     0     1.0000
    -3.5000    -6.0000    -3.5000

```

Матриця B:

```

B =
     0
     0
     2

```

Матриця Q:

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Матриця R:

$$R = 2$$

Матриця N:

$$N = \begin{bmatrix} 0.1000 \\ 0.2500 \\ 0.3000 \end{bmatrix}$$

Вектор початкових умов x_0 :

$$x_0 = \begin{bmatrix} -0.1000 \\ 0.2000 \\ 0.3000 \end{bmatrix}$$

Вектор заданих значень відхилень x_d :

$$x_d = \begin{bmatrix} 0.2000 \\ 0.3000 \\ 0.1000 \end{bmatrix}$$

Вектор заданих значень відхилень керування u_d :

$$u_d =$$

$$0.4000$$

Час спостереження tf :

$$tf =$$

$$16$$

Період квантування T :

$$T =$$

$$0.4000$$

Кількість кроків n :

$$n =$$

$$40$$

Матриця зворотного зв'язку K :

$$K =$$

$$\begin{bmatrix} -1.6960 & -2.7685 & -1.2876 \end{bmatrix}$$

Розв'язок рівняння Ріккати P :

$$P =$$

$$\begin{bmatrix} 6.9361 & 9.6899 & 4.5065 \\ 9.6899 & 22.4613 & 13.2474 \\ 4.5065 & 13.2474 & 15.7156 \end{bmatrix}$$

Власні значення замкненої системи E :

$$E =$$

$$\begin{bmatrix} -0.2466 + 0.4350j \\ -0.2466 - 0.4350j \end{bmatrix}$$

-0.4317

Синтез нелінійного регулятора:
Матриця зворотного зв'язку K_p :

$K_p =$

-1.6968 -2.7569 -1.2707

Розв'язок рівняння Ріккати P_p :

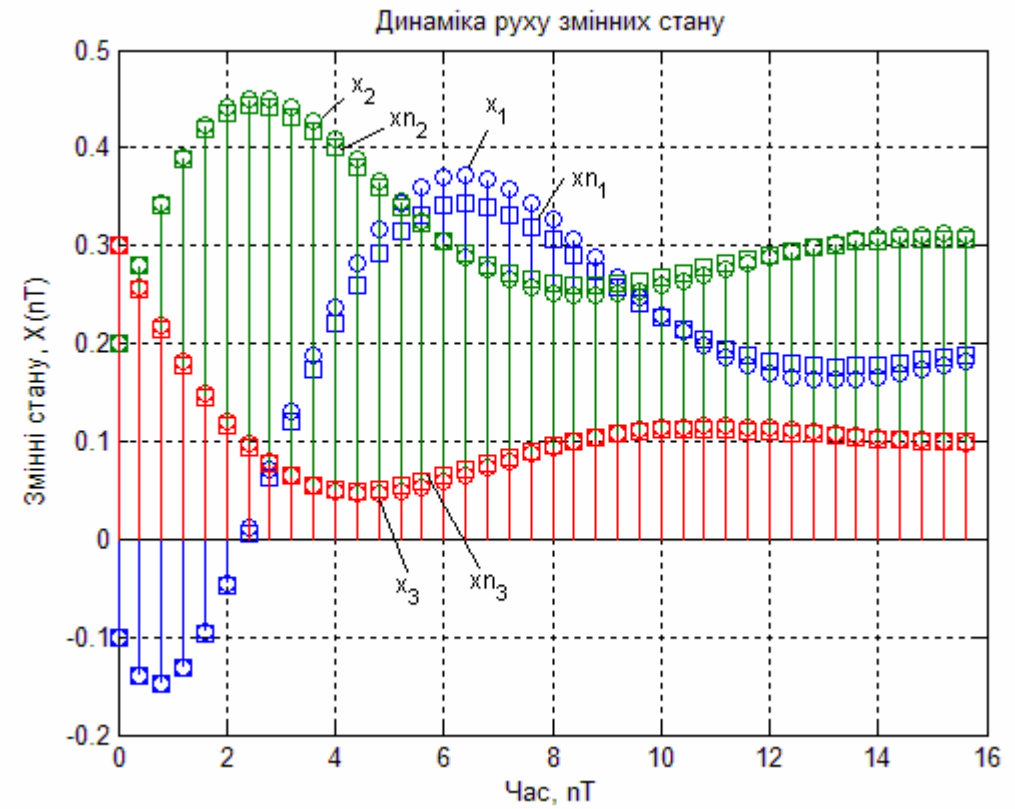
$P_p =$

7.2834 10.3623 5.0994
10.3623 23.7844 14.3153
5.0994 14.3153 16.4143

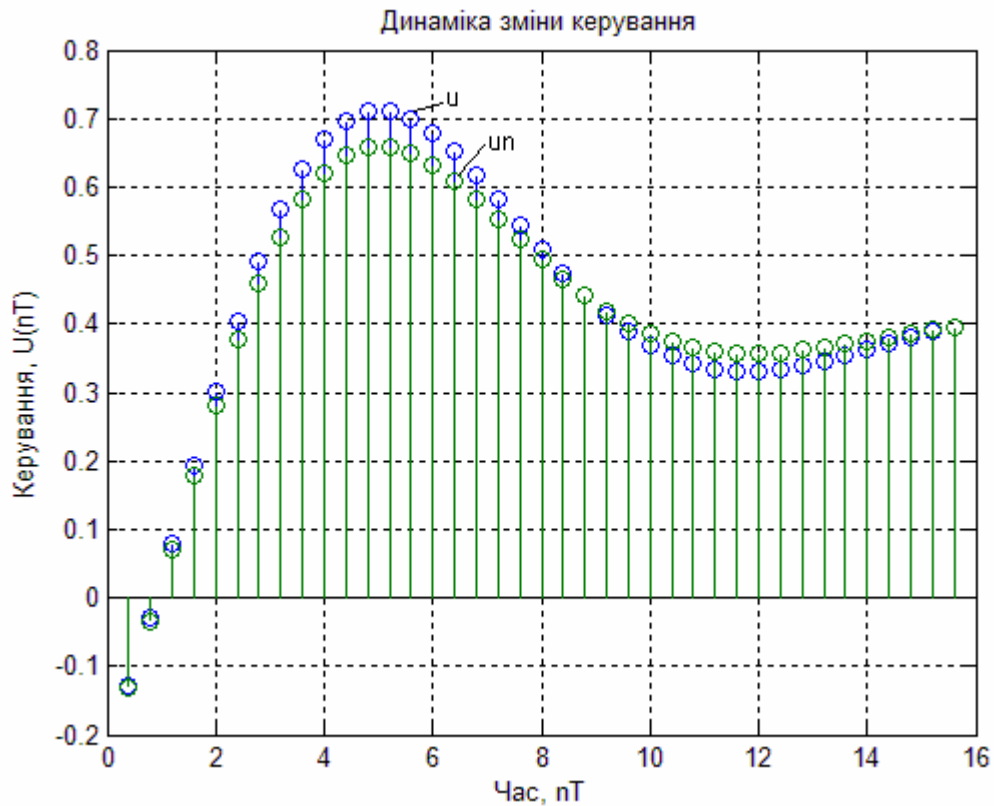
Власні значення замкненої системи E_p :

$E_p =$

-0.4065
-0.2761 + 0.4308i
-0.2761 - 0.4308i



а)



б)

Рис. 1.10. Динаміка руху змінних стану а) і керування б)

Якість отриманих регуляторів для звичайної системи та системи з урахуванням зв'язку між станом і керуванням (на графіках позначається штриховою кривою, а у програмі та формулах з індексом “ n ”) отримана при наступних значеннях параметрів:

$$X[0] \equiv X_n[0] = [-0,1 \quad 0,2 \quad 0,3]^T;$$

$$Xd \equiv Xd_n = [0,2 \quad 0,3 \quad 0,1]^T; \quad Ud \equiv Ud_n = 0,4; \quad Q = I; \quad R = 2;$$

$N = [0,1 \quad 0,25 \quad 0,3]^T$; $T = 0,4$. Коефіцієнт підсилення K отриманий за допомогою розв'язку алгебраїчного рівняння Ріккати. Виходи оптимального регулятора, показані на рис. 1.10 а), оптимальні керування на рис. 1.10 б). Як видно з графіків, якість дискретного регулятора з урахуванням зв'язку між станом і керуванням у порівнянні з звичайним регулятором краща, як і для неперервного регулятора.

1.5. Проектування оптимального дискретного лінійно-квадратичного регулятора за виходом системи

Для проектування дискретного лінійно-квадратичного регулятора по виходу системи справедливі всі попередні твердження, окрім того, що об'єкт керування описується рівняннями вигляду:

$$X[n+1] = AX[n] + BU[n], \quad X[0] = X_0, \quad n = 0, 1, \dots, n_f$$

$$Y[n] = CX[n] + DU[n]$$

(1.54)

та критерій якості

$$J = \frac{1}{2} X[n_f]^T S X[n_f] + \frac{1}{2} \sum_{n=0}^{n_f-1} (Y[n]^T Q Y[n] + U[n]^T R U[n] + 2Y[n]^T N U[n]).$$

(1.55)

Додатково розв'язується рівняння (1.23).

Приклади

Розглянемо прилад для системи з одним входом і одним виходом третього порядку, що описується рівняннями у просторі стану:

$$X[n+1] = A(X[n] - Xd) + B(U[n] - Ud), \quad X[0] = [0,1 \quad 0,4 \quad 0,3]^T,$$

$$Y[n] - Yd = C(X[n] - Xd) + D(U[n] - Ud),$$

$$\text{де } A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -5 & -6 & -4,5 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 0 \\ 1,5 \end{bmatrix}; C = [1 \ 0 \ 0]; D = 0,$$

і квадратичний критерій якості

$$J = \sum_{n=1}^{\infty} \left((Y[n] - Yd)^T Q (Y[n] - Yd) + (U[n] - Ud)^T R (U[n] - Ud) \right)$$

де $Xd = [0,4 \ -0,2 \ 0,6]^T$; $Ud = 0,4$; $Yd = 0,6$; $Q = 1$; $R = 2$;
 $T = 0,4$.

Наведемо приклад програми для розв'язку цієї задачі з назвою
"Discrete_Lin_Reg_Y".

```
function Discrete_Lin_Reg_Y
A = [0 1 0; 0 0 1; -5 -6 -4.5]; % Введення матриці A
B = [0 0 1.5]'; % Введення матриці B
C = [1 0 0]; % Введення матриці C
D = 0; % Введення матриці D
% Параметри критерія якості:
disp('Матриця Q:')
Q = 1 % Формування одиничної матриці
disp('Матриця R:')
R = 2 % Введення матриці R
disp('Вектор початкових умов x0:')
x0 = [0.1 0.4 0.3]' % Завдання вектора початкових умов
disp('Вектор заданих значень відхилень xd:')
xd = [0.4 -0.2 0.6]' % Завдання вектора заданих значень відхилень
disp('Вектор заданих значень відхилень керування Ud:')
ud = 0.4 % Завдання відхилень значень керування
disp('Вектор заданих значень відхилень вихідної величини Yd:')
```

```
yd = 0.6 % Завдання відхилень значень вихідної величини
disp('Час спостереження tf:')
tf = 14 % Завдання часу спостереження
disp('Період квантування T:')
T = 0.4 % Завдання періоду квантування
disp('Кількість кроків n:')
n = tf/T
disp('Модель системи у просторі станів:')
Sys = ss(A,B,C,D,T) %Створення моделі системи у просторі станів
[K P E] = lqr(Sys,Q,R); % Синтез регулятора
disp('Матриця зворотного зв'язку K:'); K
disp('Розв'язок рівняння Ріккати P:'); P
disp('Власні значення замкненої системи E:'); E
% Визначення розмірності задачі
SA = size(A); SA = SA(1); SB = size(B); SB = SB(2); SC = size(C); SC = SC(1);
% Формування векторів x u
x = zeros(SA,n); u = zeros(SB,n-1); y = zeros(SC,n-1);
% Формування початкового стану
x(:,1) = x0; xn(:,1) = x0;
% Знаходження оптимального керування u, змінних стану x та виходу системи y
for i=1:n-1,
    u(:,i) = ud-K*(x(:,i)-xd);
    x(:,i+1) = (A*(x(:,i)-xd)+B*(u(:,i)-ud))*T+x(:,i);
    y(:,i) = yd+C*(x(:,i)-xd);
end
% Побудова динаміки змінних стану
stem(0:T:tf-T,x'), grid
title('динаміка руху змінних стану');
xlabel('час, nT');
ylabel('Змінні стану, x(nT)');
legend('x_1','x_2','x_3','xn_1','xn_2','xn_3');
% Побудова виходу системи
figure(2)
stem(T:T:tf-T,y); grid on
```



```

title('Динаміка зміни виходу систми');
xlabel('Час, nT');
ylabel('Вихід системи, Y(nT)'); legend('y','yn');
% Побудова вектора керування
figure(3)
stem(T:T:tf-T,u); grid on
title('Динаміка зміни керування'); xlabel('Час,
nT'); ylabel('Керування, U(nT)');
legend('u','un');

```

Результат виконання програми

Матриця Q:

Q =
1

Матриця R:

R =
2

Вектор початкових умов x0:

x0 =
0.1000
0.4000
0.3000

Вектор заданих значень відхилень xd:

xd =
0.4000
-0.2000
0.6000

Вектор заданих значень відхилень керування Ud:

ud =
0.4000

Вектор заданих значень відхилень вихідної величини Yd:

yd =
0.6000

Час спостереження tf:

tf =
14

Період квантування T:

T =
0.4000

Кількість кроків n:

n =
35

Модель системи у просторі станів:

a =

	x1	x2	x3
x1	0	1	0
x2	0	0	1
x3	-5	-6	-4.5

$$b = \begin{matrix} & u1 \\ x1 & 0 \\ x2 & 0 \\ x3 & 1.5 \end{matrix}$$

$$c = \begin{matrix} & x1 & x2 & x3 \\ y1 & 1 & 0 & 0 \end{matrix}$$

$$d = \begin{matrix} & u1 \\ y1 & 0 \end{matrix}$$

Sampling time: 0.4
 Discrete-time model.
 Матриця зворотного зв'язку K:

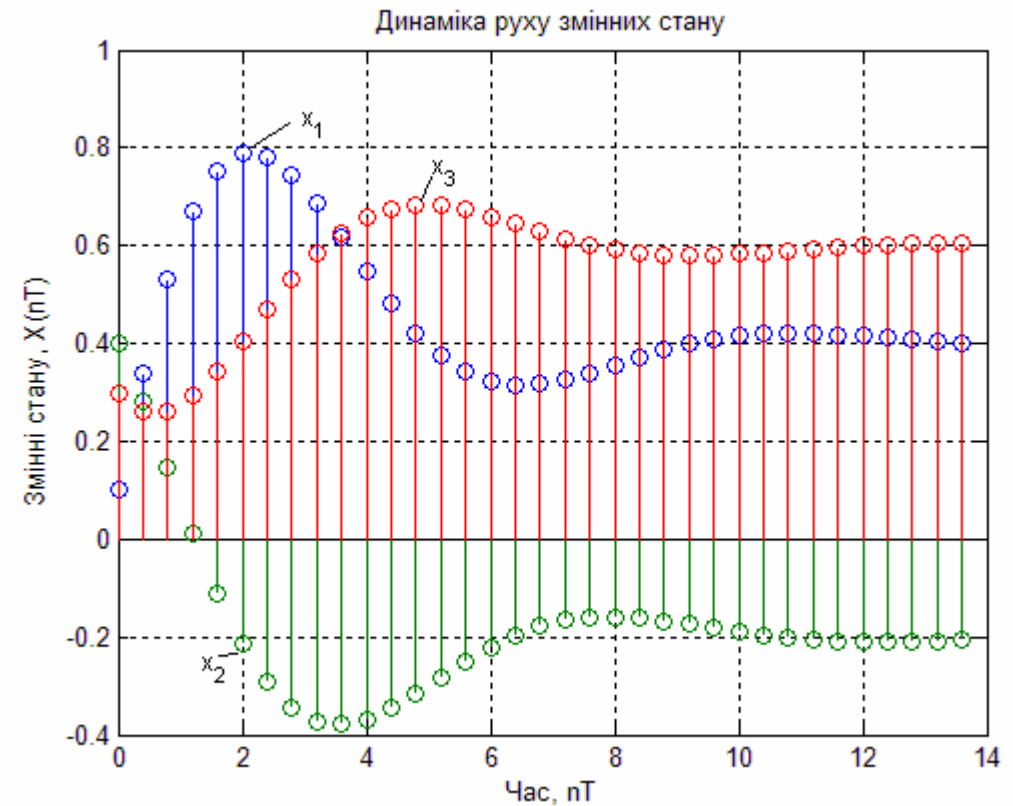
$$K = \begin{matrix} & & & \\ & -3.2158 & -3.4620 & -2.2527 \end{matrix}$$

Розв'язок рівняння Ріккати P:

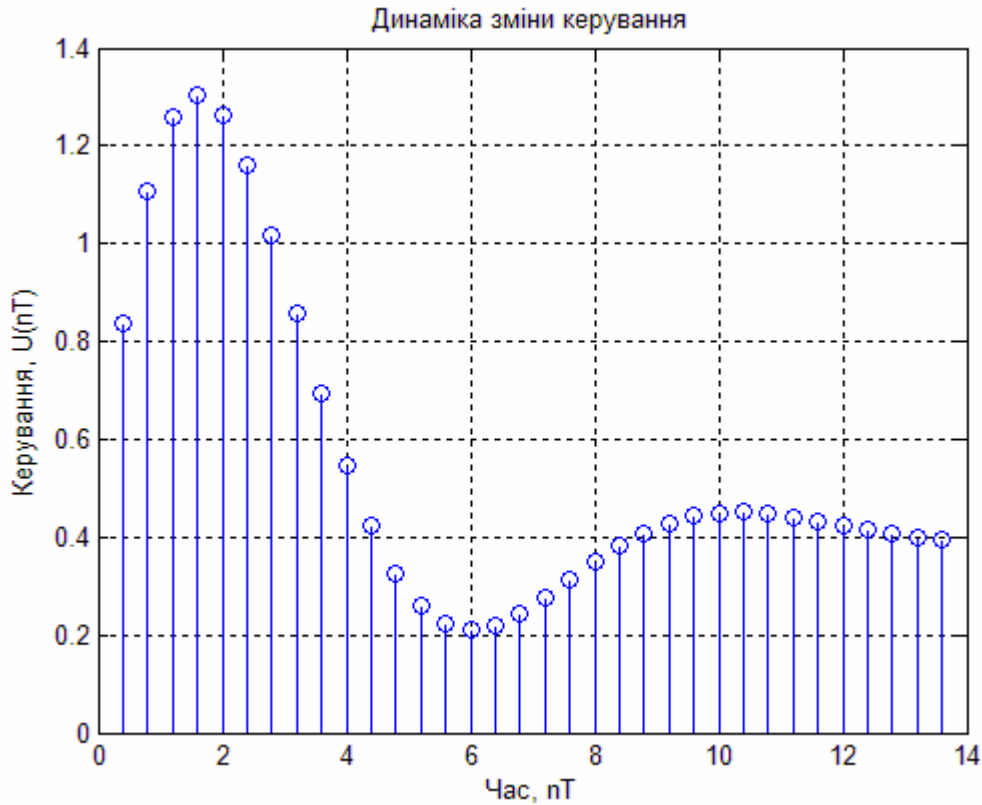
$$P = \begin{matrix} & & & \\ & 22.4389 & 23.0799 & 15.0181 \\ & 23.0799 & 38.0150 & 24.2676 \\ & 15.0181 & 24.2676 & 24.3293 \end{matrix}$$

Власні значення замкненої системи E:

$$E = \begin{matrix} & & & \\ & -0.4004 + 0.6248i & & \\ & -0.4004 - 0.6248i & & \\ & -0.3200 & & \end{matrix}$$



a)



б)

Рис. 1.11. Динаміка руху змінних стану а) і керування б)

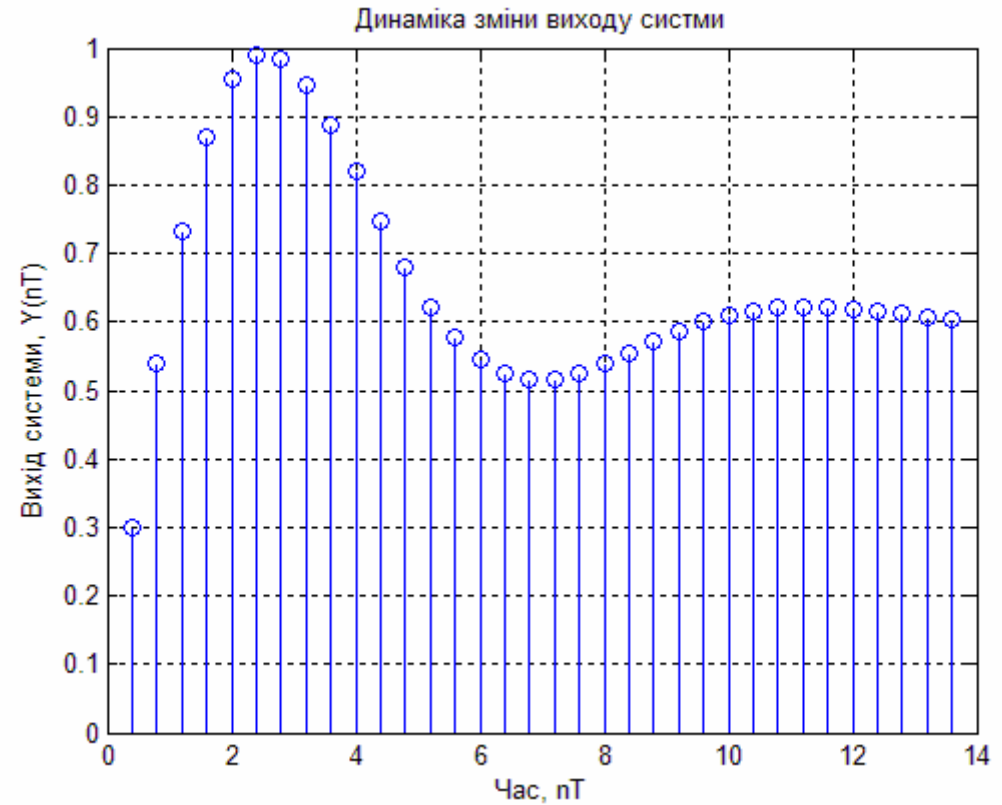


Рис. 1.12. Динаміка зміни виходу системи

Досліджували якість отриманого регулятора для наступних значеннях параметрів: $X[0] = [0,1 \ 0,4 \ 0,3]^T$; $Xd = [0,4 \ -0,2 \ 0,6]^T$; $Ud = 0,4$; $Yd = 0,6$; $Q = 1$; $R = 2$; $T = 0,4$. Вихід оптимального регулятора, показан на рис. 1.12, змінні стану показані на рис. 1.11 а). Оптимальне керування представлено на рис. 1.11 б).

Розглянемо приклад порівняння системи з одним входом і одним виходом третього порядку для звичайних систем, і для систем

з врахуванням зв'язку між станом та керуванням. Система описується рівняннями у просторі стану:

$$X[n+1] = A(X[n] - Xd) + B(U[n] - Ud), \quad X[0] = [0,1 \quad 0,4 \quad -0,3]^T$$

$$Y[n] - Yd = C(X[n] - Xd) + D(U[n] - Ud),$$

$$\text{де } A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -3 & -8 & -4,5 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 0 \\ 0,5 \end{bmatrix}; C = [1 \quad 0 \quad 0]; D = 0,$$

і квадратичний критерій якості

$$J = \sum_{n=1}^{\infty} \left((Y[n] - Yd)^T Q (Y[n] - Yd) + (U[n] - Ud)^T R (U[n] - Ud) + 2(Y[n] - Yd)^T N (U[n] - Ud) \right),$$

де $Xd = [2 \quad 4 \quad -2]^T$; $Ud = -20$; $Yd = 1$; $Q = 1$; $R = 2,5$; $N = 0,05$; $T = 0,4$.

Наведемо приклад програми для розв'язку цієї задачі з назвою "Discrete_Lin_RegN_Y".

```
function Discrete_Lin_RegN_Y
A = [0 1 0; 0 0 1; -3 -8 -4.5]; % Введення матриці A
B = [0 0 0.5]'; % Введення матриці B
C = [1 0 0]; % Введення матриці C
D = 0; % Введення матриці D
% Параметри критерія якості:
disp('Матриця Q:')
Q = 1 % Формування одиничної матриці
disp('Матриця R:')
R = 2.5 % Введення матриці R
disp('Матриця N:')
N = 0.05 % Введення матриці N
disp('Вектор початкових умов x0:')
```

```
x0 = [0.1 0.4 -0.3]' % Завдання вектора
початкових умов
disp('Вектор заданих значень відхилень Xd:')
xd = [2 4 -2]' % Завдання вектора заданих
значень відхилень
disp('Вектор заданих значень відхилень
керування Ud:')
ud = -20 % Завдання відхилень значень керування
disp('Вектор заданих значень відхилень вихідної
величини Yd:')
yd = 1 % Завдання відхилень значень вихідної
величини
disp('Час спостереження tf:')
tf = 20 % Завдання часу спостереження
disp('Період квантування T:')
T = 0.4 % Завдання періоду квантування
disp('Кількість кроків n:')
n = tf/T
disp('Модель системи у просторі станів:')
sys = ss(A,B,C,D,T) % Створення моделі системи у
просторі станів
disp('Синтез лінійного регулятора:')
[K P E] = lqry(sys,Q,R); % Синтез регулятора
disp('Матриця зворотного зв'язку K:'); K
disp('Розв'язок рівняння Ріккати P:'); P
disp('Власні значення замкненої системи E:'); E
disp('Синтез нелінійного регулятора:')
[Kn Pn En] = lqry(sys,R,N); % Синтез регулятора з
нелінійною складовою
disp('Матриця зворотного зв'язку Kn:'); Kn
disp('Розв'язок рівняння Ріккати Pn:'); Pn
disp('Власні значення замкненої системи En:'); En
% Визначення розмірності задачі
SA = size(A); SA = SA(1); SB = size(B); SB =
SB(2); SC = size(C); SC = SC(1);
% Формування векторів x u y
x = zeros(SA,n); u = zeros(SB,n-1); y =
zeros(SC,n-1); xn = zeros(SA,n);
un = zeros(SB,n-1); yn = zeros(SC,n-1);
% Формування початкового стану
x(:,1) = x0; xn(:,1) = x0;
```

```

% Знаходження оптимального керування u та змінних
стану x
for i=1:n-1,
    u(:,i) = ud-k*(x(:,i)-xd);
    x(:,i+1) = (A*(x(:,i)-xd)+B*(u(:,i)-
ud))*T+x(:,i);
    y(:,i) = yd+c*(x(:,i)-xd);
    un(:,i) = ud-Kn*(xn(:,i)-xd);
    xn(:,i+1) = (A*(xn(:,i)-xd)+B*(un(:,i)-
ud))*T+xn(:,i);
    yn(:,i) = yd+c*(xn(:,i)-xd);
end
% Побудова динаміки змінних стану
stem(0:T:tf-T,x'), grid
hold on
stem(0:T:tf-T,xn','--','marker','square')
title('динаміка руху змінних стану');
xlabel('час, nT');
ylabel('Змінні стану, X(nT)');
legend('x_1','x_2','x_3','xn_1','xn_2','xn_3')
hold off
% Побудова виходу системи
figure(2)
stem(T:T:tf-T,[y;yn]'); grid on
title('динаміка зміни виходу систми');
xlabel('час, nT');
ylabel('вихід системи, Y(nT)'); legend('y','yn')
% Побудова вектора керування
figure(3)
stem(T:T:tf-T,[u;un]'); grid on
title('динаміка зміни керування'); xlabel('час,
nT'); ylabel('керування, U(nT)');
legend('u','un')

```

Результат виконання програми

Матриця Q:

Q =

1

Матриця R:

R =

2.5000

Матриця N:

N =

0.0500

Вектор початкових умов x0:

x0 =

0.1000

0.4000

-0.3000

Вектор заданих значень відхилень xd:

xd =

2

4

-2

Вектор заданих значень відхилень керування Ud:

ud =

-20

Вектор заданих значень відхилень вихідної величини Yd:

yd =

1

Час спостереження t_f :

$t_f =$

20

Період квантування T :

$T =$

0.4000

Кількість кроків n :

$n =$

50

Модель системи у просторі станів:

$a =$

	x_1	x_2	x_3
x_1	0	1	0
x_2	0	0	1
x_3	-3	-8	-4.5

$b =$

	u_1
x_1	0
x_2	0
x_3	0.5

$c =$

	x_1	x_2	x_3
y_1	1	0	0

$d =$

u_1

y_1 0

Sampling time: 0.4

Discrete-time model.

Синтез лінійного регулятора:

Матриця зворотного зв'язку K :

$K =$

-5.8349 -15.0081 -6.6818

Розв'язок рівняння Ріккати P :

$P =$

88.5234	225.1219	100.2273
225.1219	639.1420	376.2221
100.2273	376.2221	353.4073

Власні значення замкненої системи E :

$E =$

-0.3334 + 0.2378i
 -0.3334 - 0.2378i
 -0.4924

Синтез нелінійного регулятора:

Матриця зворотного зв'язку K_n :

$K_n =$

-5.9049 -15.3919 -7.4541

Розв'язок рівняння Ріккати P_n :

$Pn =$

4.2715	4.6176	2.2362
4.6176	15.9050	8.8523
2.2362	8.8523	12.4169

Власні значення замкненої системи E_n :

$E_n =$

$-0.2415 + 0.3251i$
$-0.2415 - 0.3251i$
-0.2900

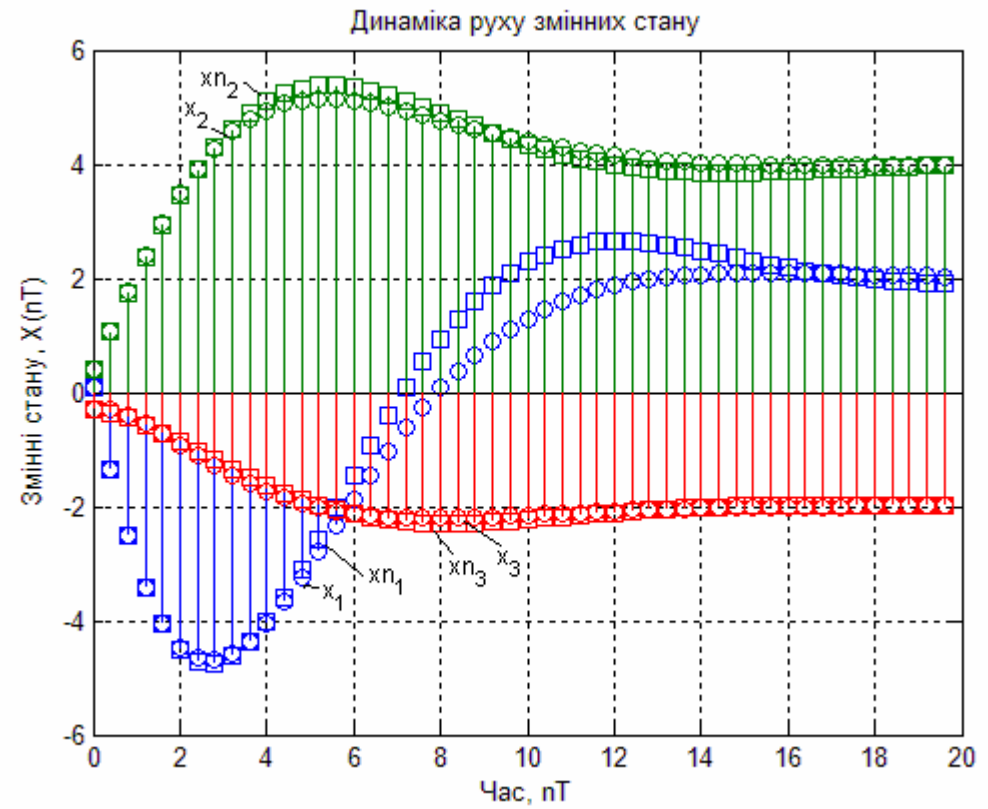
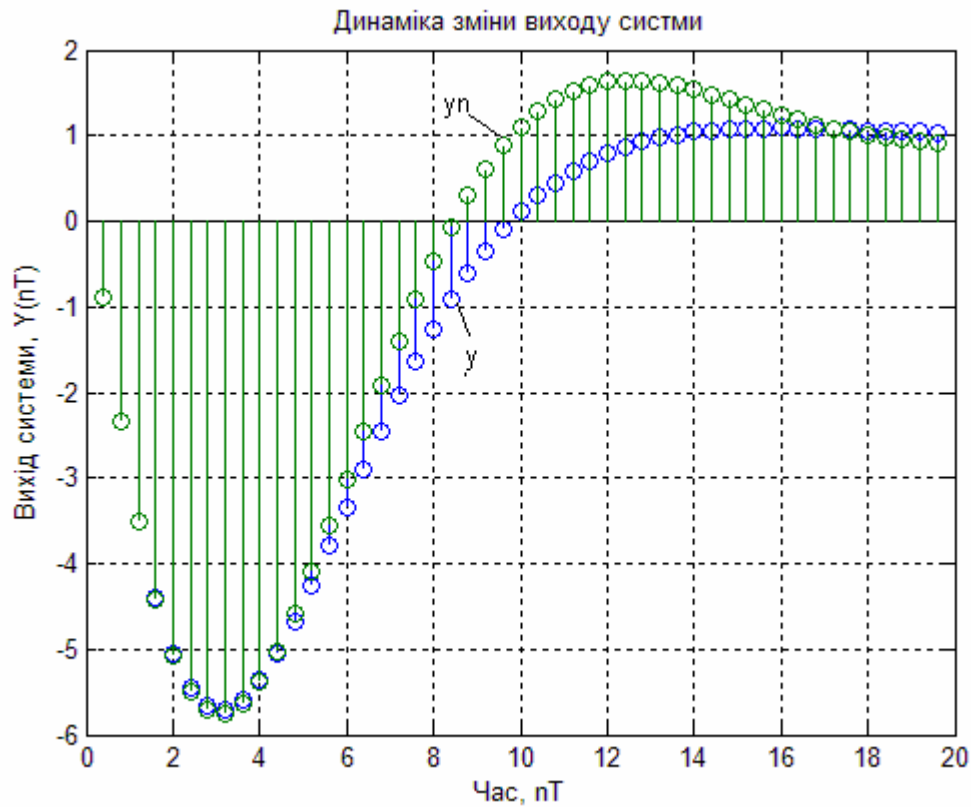
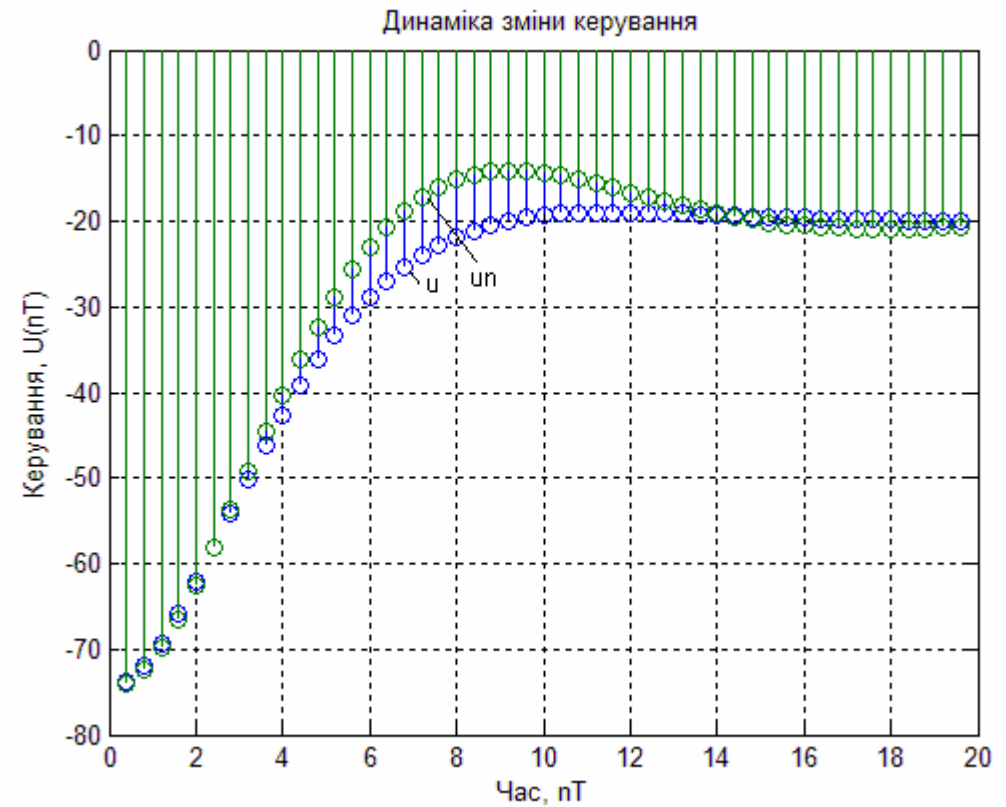


Рис. 1.13. Динаміка руху змінних стану



а)



б)

Рис. 1.14. Динаміка зміни виходу системи а) і керування б)

Якість отриманих регуляторів для звичайної системи та системи з урахуванням зв'язку між станом і керуванням (на графіках позначається штриховою кривою, а у програмі та формулах з індексом "n") отримана для наступних значеннях параметрів:

$$X[0] \equiv X_n[0] = [0,1 \quad 0,4 \quad -0,3]^T; \quad Xd \equiv Xd_n = [2 \quad 4 \quad -2]^T;$$

$$Ud \equiv Ud_n = -20; \quad Yd \equiv Yd_n = 1; \quad Q = 1; \quad R = 2,5; \quad N = 0,05;$$

$T = 0,4$. Коефіцієнт підсилення K отриманий за допомогою розв'язку алгебраїчного рівняння Ріккати. Виходи оптимального регулятора, показані на рис. 1.14 а). Оптимальне керування представлено на рис. 1.14 б). Змінні стану показані на рис. 1.13.

1.6. Проектування оптимального дискретного лінійно-квадратичного регулятора для неперервної системи

Проектування дискретного лінійно-квадратичного регулятора для неперервної системи, що описується рівнянням (1.2), функціоналу якості (1.20) та сигналом керування вигляду

$$U[n] = -K_d X[n]. \quad (1.56)$$

Матричний коефіцієнт підсилення K_d визначається з дискретизованої системи з періодом квантування T та розв'язання дискретного рівняння Ріккати (1.52), з урахуванням наступних визначень

$$\begin{aligned} \Phi(\tau) &= e^{A\tau}, \quad A_d = \Phi(T), \\ \Gamma(\tau) &= \int_0^\tau e^{A\eta} B d\eta, \quad B_d = \Gamma(T), \end{aligned} \quad (1.57)$$

дискретна модель системи матиме вигляд:

$$X[n+1] = A_d X[n] + B_d U[n], \quad (1.58)$$

та додатково до функціоналу якості, розв'язується наступне рівняння:

$$\begin{bmatrix} Q_d & N_d \\ N_d^T & R_d \end{bmatrix} = \int_0^T \begin{bmatrix} \Phi(\tau)^T & 0 \\ \Gamma(\tau)^T & I \end{bmatrix} \begin{bmatrix} Q & N \\ N^T & R \end{bmatrix} \begin{bmatrix} \Phi(\tau) & \Gamma(\tau) \\ 0 & I \end{bmatrix} d\tau. \quad (1.59)$$

Інтеграли обчислюються з використанням матричних показникових формул Ван Лона.

Вимоги, що до існування оптимального дискретного лінійно-квадратичного регулятора для неперервної системи, задовольняють

таким, що і для оптимального дискретного лінійно-квадратичного регулятора.

Приклади

Розглянемо приклад системи з одним входом і одним виходом третього порядку, що описується рівняннями у просторі стану:

$$\frac{dX}{dt} = A(X - Xd) + B(U - Ud), \quad X(0) = [-0,1 \quad 0,1 \quad 0,3]^T,$$

$$\text{де } A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -6 & -2 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix},$$

і квадратичний критерій якості

$$J = \int_0^\infty \left((X - Xd)^T Q (X - Xd) + (U - Ud)^T R (U - Ud) \right) dt,$$

де $Xd = [0,4 \quad 0,2 \quad 0,6]^T$; $Ud = 0,2$; $Q = I$; $R = 1/3$; $T = 0,1$.

Наведемо приклад програми для розв'язку цієї задачі з назвою "Discrete_Lin_Reg_For_Continuous_Plant".

```
function Discrete_Lin_Reg_For_Continuous_Plant
disp('Матриця A:')
A = [0 1 0; 0 0 1; -1 -6 -2] % введення матриці A
disp('Матриця B:')
B = [0; 0; 2] % введення матриці B
% Параметри критерія якості:
disp('Матриця Q:')
Q = eye(3) % формування одиничної матриці
disp('Матриця R:')
R = 1/3 % введення матриці R
disp('Вектор початкових умов x0:')
x0 = [-0.1 0.1 0.3]' % завдання вектора
початкових умов
disp('Вектор заданих значень відхилень Xd:')
```

```

xd = [0.4 0.2 0.6]' % Завдання вектора заданих
значень відхилень
disp('Вектор заданих значень відхилень
керування Ud:')
ud = 0.2 % Завдання відхилень значень керування
disp('Час спостереження tf:')
tf = 10 % Завдання часу спостереження
disp('Період квантування T:')
T = 0.1 % Завдання періоду квантування
disp('Кількість кроків n:')
n = tf/T
[K P E] = lqr(A,B,Q,R,T); % Синтез регулятора
disp('Матриця зворотнього зв'язку K:'); K
disp('Розв'язок рівняння Ріккати P:'); P
disp('Власні значення замкненої системи E:'); E
% Визначення розмірності задачі
SA = size(A); SA = SA(1); SB = size(B); SB =
SB(2);
% Формування векторів x u
x = zeros(SA,n); u = zeros(SB,n-1);
% Формування початкового стану
x(:,1) = x0;
% Знаходження оптимального керування u та змінних
стану x
for i=1:n-1,
    u(:,i) = ud-K*(x(:,i)-xd);
    x(:,i+1) = (A*(x(:,i)-xd)+B*(u(:,i)-
ud))*T+x(:,i);
end
% Подова динаміки змінних стану
plot(0:T:tf-T,x), grid
title('динаміка руху змінних стану');
xlabel('Час, t');
ylabel('Змінні стану, x(t)');
legend('x_1','x_2','x_3')
% Побудова вектора керування
figure(2)
h = stem(T:T:tf-T,u); grid on
title('динаміка зміни керування'); xlabel('Час,
nT'); ylabel('керування, U(nT)'); legend('u')

```

Результат виконання програми

Матриця A:

```

A =
     0     1     0
     0     0     1
    -1    -6    -2

```

Матриця B:

```

B =
     0
     0
     2

```

Матриця Q:

```

Q =
     1     0     0
     0     1     0
     0     0     1

```

Матриця R:

```

R =
     0.3333

```

Вектор початкових умов x0:

```

x0 =
    -0.1000
     0.1000
     0.3000

```

1.0942 0.9498 1.1099

Розв'язок рівняння Ріккати P:

P =

2.1462 1.2317 0.2180
1.2317 2.1973 0.2356
0.2180 0.2356 0.2218

Власні значення замкненої системи E:

E =

0.9464
0.8047 + 0.1246i
0.8047 - 0.1246i

Вектор заданих значень відхилень Xd:

xd =

0.4000
0.2000
0.6000

Вектор заданих значень відхилень керування Ud:

ud =

0.2000

Час спостереження tf:

tf =

10

Період квантування T:

T =

0.1000

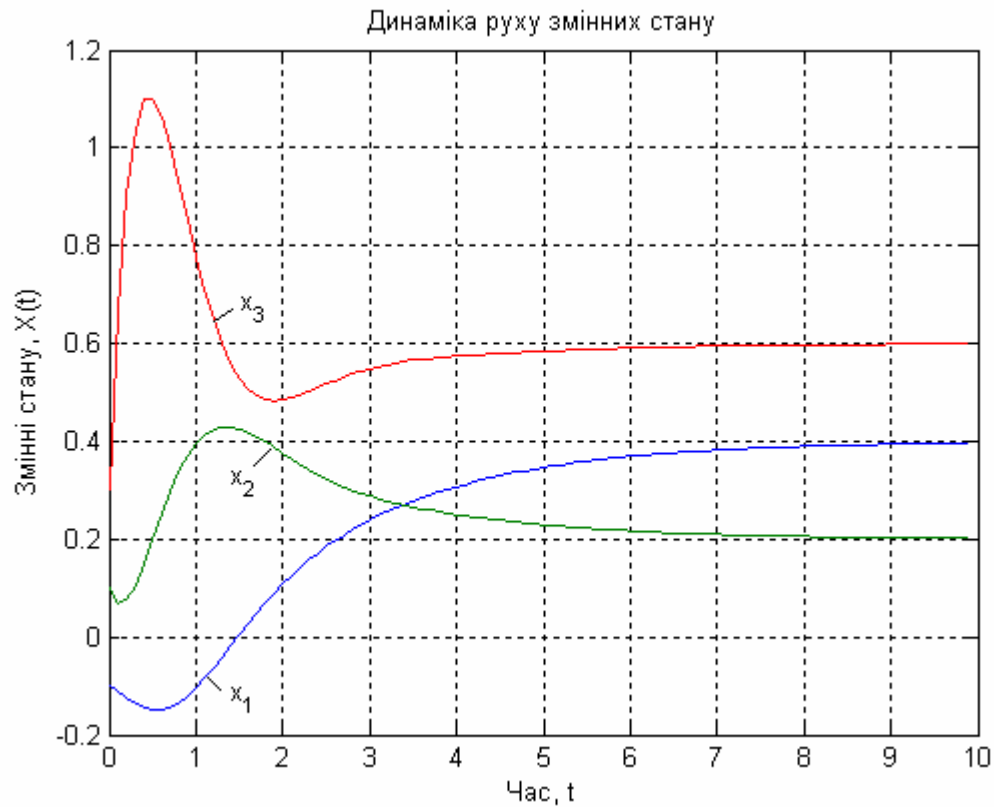
Кількість кроків n:

n =

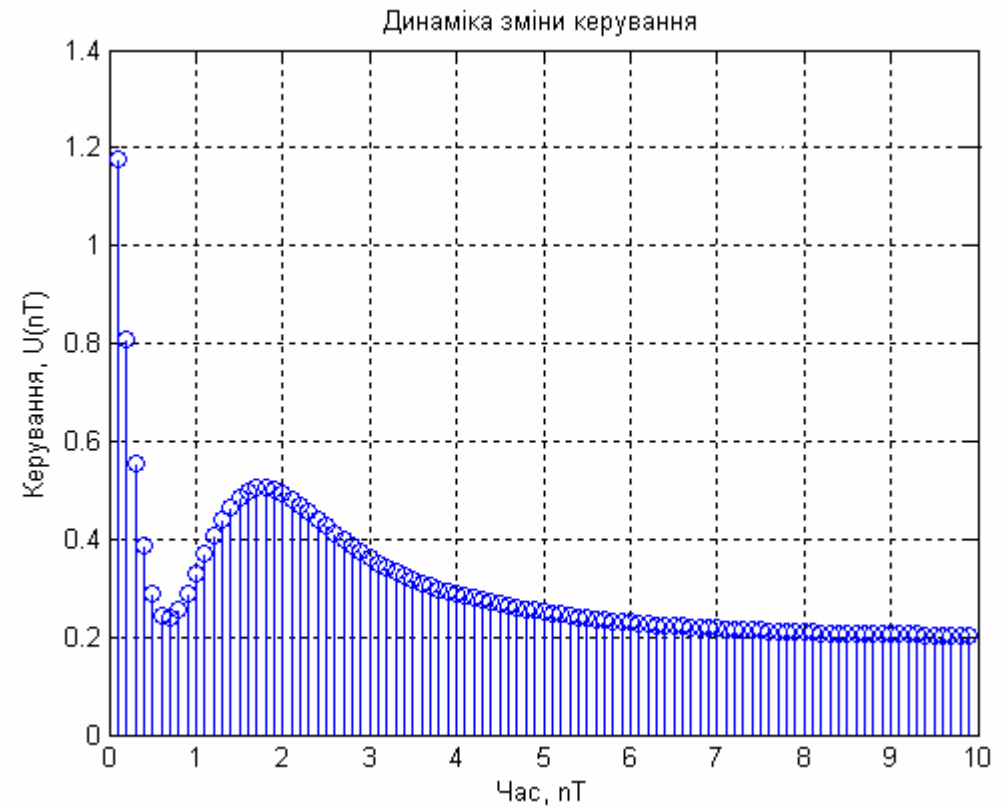
100

Матриця зворотнього зв'язку K:

K =



а)



б)

Рис. 1.15. Динаміка руху змінних стану а) і керування б)

Дослідимо якість отриманого регулятора для наступних значення параметрів: $X(0) = [-0,1 \ 0,1 \ 0,3]^T$; $X_d = [0,4 \ 0,2 \ 0,6]^T$; $U_d = 0,2$; $Q = I$; $R = 1/3$; $T = 0,2$. Коефіцієнт підсилення K отриманий за допомогою розв'язку алгебраїчного рівняння Ріккаті. Виходи оптимального регулятора, показані на рис. 1.15 а). Оптимальне керування представлено на рис. 1.15 б).

Розглянемо приклад порівняння системи з одним входом і одним виходом третього порядку для звичайних систем, і для систем з врахуванням зв'язку між станом та керуванням. Система описується рівняннями у просторі стану:

$$\frac{dX}{dt} = A(X - Xd) + B(U - Ud), \quad X(0) = [-0,1 \quad 0,1 \quad 0,3]^T,$$

$$\text{де } A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -6 & -3 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix},$$

і квадратичний критерій якості

$$J = \int_0^{\infty} ((X - Xd)^T Q (X - Xd) + (U - Ud)^T R (U - Ud) + 2(X - Xd)^T N (U - Ud)) dt,$$

$$\text{де } Xd = [0,4 \quad 0,2 \quad 0,6]^T; \quad Ud = 0,3; \quad Q = I; \quad R = 2;$$

$$N = [1,1 \quad 0,5 \quad 0,4]^T; \quad T = 0,1.$$

Наведемо приклад програми для розв'язку цієї задачі з назвою "Discrete_Lin_RegN_For_Continuous_Plant".

```
function Discrete_Lin_RegN_For_Continuous_Plant
disp('Матриця A:')
A = [0 1 0; 0 0 1; -2 -6 -3] % Введення матриці A
disp('Матриця B:')
B = [0 0 2]' % Введення матриці B
% Параметри критерію якості:
disp('Матриця Q:')
Q = eye(3) % Формування одиничної матриці
disp('Матриця R:')
R = 2 % Введення матриці R
disp('Матриця N:')
N = [1.1 0.5 0.4]' % Введення матриці N
disp('Вектор початкових умов x0:')
x0 = [-0.1 0.1 0.3]' % Завдання вектора
початкових умов
```

```
disp('Вектор заданих значень відхилень Xd:')
xd = [0.4 0.2 0.6]' % Завдання вектора заданих
значень відхилень
disp('Вектор заданих значень відхилень керування
Ud:')
ud = 0.3 % Завдання відхилень значень керування
disp('Час спостереження tf:')
tf = 10 % Завдання часу спостереження
disp('Період квантування T:')
T = 0.1 % Завдання періоду квантування
disp('Кількість кроків n:')
n = tf/T
disp('Синтез лінійного регулятора:')
[K P E] = lqrd(A,B,Q,R,T); % Синтез регулятора
disp('Матриця зворотного зв'язку K:'); K
disp('Розв'язок рівняння Ріккати P:'); P
disp('Власні значення замкненої системи E:'); E
disp('Синтез нелінійного регулятора:')
[Kn Pn En] = lqrd(A,B,Q,R,N,T); % Синтез
регулятора з нелінійною складовою
disp('Матриця зворотного зв'язку Kn:'); Kn
disp('Розв'язок рівняння Ріккати Pn:'); Pn
disp('Власні значення замкненої системи En:'); En
% Визначення розмірності задачі
SA = size(A); SA = SA(1); SB = size(B); SB =
SB(2);
% Формування векторів x u
x = zeros(SA,n); u = zeros(SB,n-1); xn =
zeros(SA,n); un = zeros(SB,n-1);
% Формування початкового стану
x(:,1) = x0; xn(:,1) = x0;
% Знаходження оптимального керування u та змінних
стану x
for i=1:n-1,
    u(:,i) = ud-K*(x(:,i)-xd);
    x(:,i+1) = (A*(x(:,i)-xd)+B*(u(:,i)-
ud))*T+x(:,i);
    un(:,i) = ud-Kn*(xn(:,i)-xd);
    xn(:,i+1) = (A*(xn(:,i)-xd)+B*(un(:,i)-
ud))*T+xn(:,i);
end
```

```

% Побудова динаміки змінних стану
plot(0:T:tf-T,x), grid
hold on
plot(0:T:tf-T,xn,'--')
title('Динаміка руху змінних стану');
xlabel('Час, t');
ylabel('Змінні стану, X(t)');
legend('x_1', 'x_2', 'x_3', 'xn_1', 'xn_2', 'xn_3')
hold off
% Побудова вектора керування
figure(2)
stem(T:T:tf-T,[u;un]'); grid on
title('Динаміка зміни керування'); xlabel('Час, nT'); ylabel('Керування, U(nT)');
legend('u', 'un')

```

Результат виконання програми

Матриця A:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -6 & -3 \end{bmatrix}$$

Матриця B:

$$B = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

Матриця Q:

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

Матриця R:

$$R = \begin{bmatrix} 2 \end{bmatrix}$$

Матриця N:

$$N = \begin{bmatrix} 1.1000 \\ 0.5000 \\ 0.4000 \end{bmatrix}$$

Вектор початкових умов x0:

$$x0 = \begin{bmatrix} -0.1000 \\ 0.1000 \\ 0.3000 \end{bmatrix}$$

Вектор заданих значень відхилень xd:

$$xd = \begin{bmatrix} 0.4000 \\ 0.2000 \\ 0.6000 \end{bmatrix}$$

Вектор заданих значень відхилень керування ud:

$$ud = \begin{bmatrix} 0.3000 \end{bmatrix}$$

Час спостереження t_f :

$t_f =$

10

Період квантування T :

$T =$

0.1000

Кількість кроків n :

$n =$

100

Синтез лінійного регулятора:

Матриця зворотного зв'язку K :

$K =$

0.1975 0.2183 0.2142

Розв'язок рівняння Ріккати P :

$P =$

2.0367 1.2647 0.2249
1.2647 2.1997 0.2812
0.2249 0.2812 0.2411

Власні значення замкненої системи E :

$E =$

0.9533
0.8482 + 0.1452i
0.8482 - 0.1452i

Синтез нелінійного регулятора:

Матриця зворотного зв'язку K_p :

$K_p =$

0.5551 0.3754 0.3425

Розв'язок рівняння Ріккати P_p :

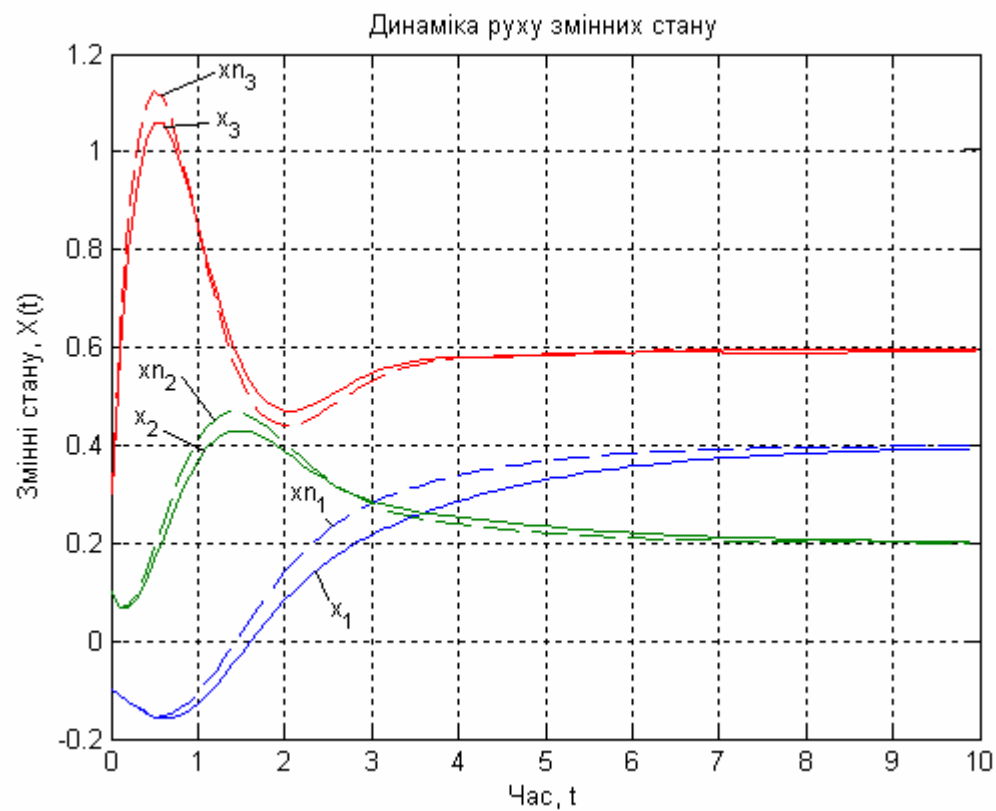
$P_p =$

1.3921 1.0443 0.0629
1.0443 2.1036 0.2210
0.0629 0.2210 0.1899

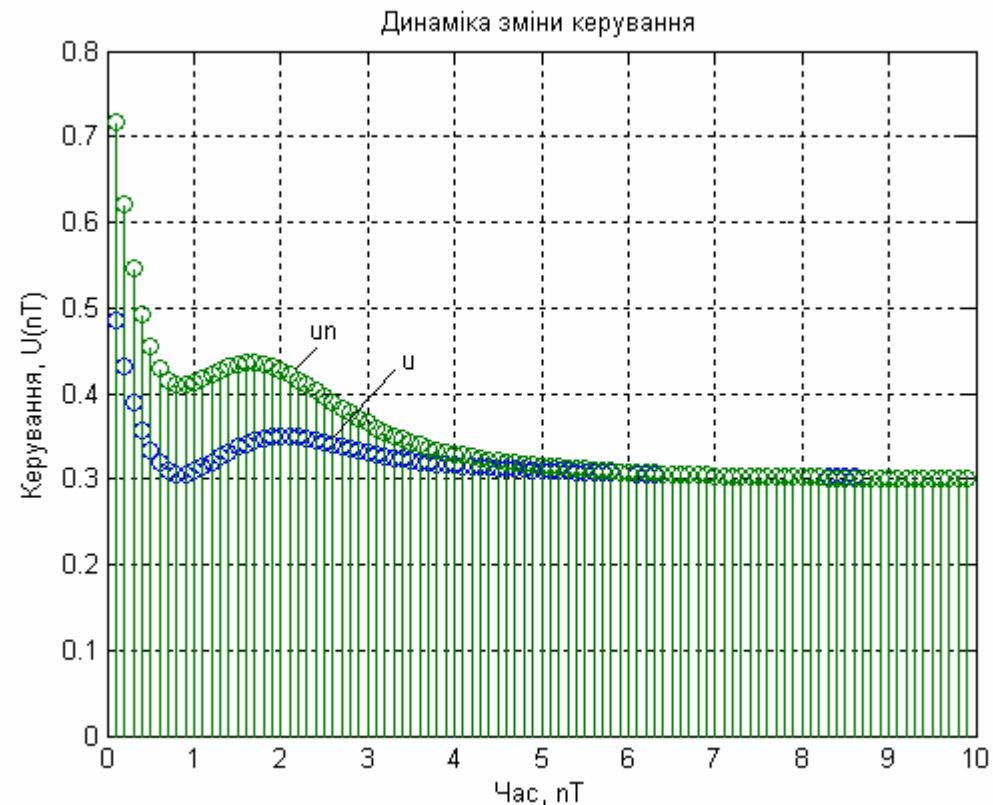
Власні значення замкненої системи E_p :

$E_p =$

0.8451 + 0.1334i
0.8451 - 0.1334i
0.9360



а)



б)

Рис. 1.16. Динаміка руху змінних стану а) і керування б)

Досліджували якість отриманих регуляторів для звичайної системи та системи з урахуванням зв'язку між станом і керуванням (на графіках позначається штриховою кривою, а у програмі та формулах з індексом "n") наступних значеннях параметрів:

$$X(0) \equiv X_n(0) = [-0,1 \quad 0,1 \quad 0,3]^T;$$

$$Xd \equiv Xd_n = [0,4 \quad 0,2 \quad 0,6]^T; \quad Ud \equiv Ud_n = 0,3; \quad Q = I; \quad R = 2;$$

$N = [1,1 \quad 0,5 \quad 0,4]^T$; $T = 0,1$. Виходи оптимального регулятора, показані на рис. 1.16 а). Оптимальне керування представлено на рис. 1.16 б).

2. Оптимальне оцінювання стану

В попередньому розділі основна увага приділялася алгоритмам оптимального керування; питання вимірювання стану, вплив випадкових збурень і завад не розглядали. В реальних технологічних процесах, як правило, не має можливості контролювати всі координати вектора стану; таким чином розмірність вектора виходів зазвичай менша розмірності вектора стану. Значна невизначеність вноситься похибками вимірювань, шумами та випадковими збуреннями, а також тим, що використовувані моделі не повною мірою адекватні об'єктам керування. Якщо враховувати всі перераховані джерела невизначеності та випадковості при проектуванні систем керування і використовувати звичайні алгоритми регулювання до зашумлених даних, то можливо отримати неефективну систему, що не забезпечує потрібну якість керування.

Часто, бажано мати можливість оцінювати стан системи на основі даних, які мають лише статистичний зв'язок з цим станом і, відповідно, не забезпечують його точного визначення. Необхідно знайти оцінку, що являється лінійною функцією від результатів спостереження у минулому та мінімізувати середньоквадратичну похибку оцінювання переданого сигналу. В загальному випадку, ця проблема називається **лінійним оцінюванням з мінімальною похибкою**, або **лінійною фільтрацією**.

2.1. Неперервний фільтр Калмана

Вихідну задачу оцінювання, опишемо наступним чином. Припустимо, що векторний сигнал $S(t)$ може бути представлений у вигляді:

$$S(t) = H(t)X(t), \quad (2.1)$$

де процес, що описує стан деякої системи, задовольняє рівнянню

$$\frac{dX}{dt} = F(t)X(t) + G(t)W(t), \quad (2.2)$$

а шум $W(t)$ має рівне нулю середнє значення та коваріаційну матрицю

$$E\{W(t)W^T(t)\} = Q(t)\delta(t-\tau). \quad (2.3)$$

Будемо вважати далі, що шум вимірювань $V(t)$ також має нульове середнє значення, а його коваріаційна матриця

$$E\{V(t)V^T(t)\} = R(t)\delta(t-\tau), \quad (2.4)$$

причому для простоти покладемо

$$E\{W(t)V^T(t)\} = 0. \quad (2.5)$$

Для спостерігаемого вектора запишемо

$$Z(t) = S(t) + V(t). \quad (2.6)$$

Нехай, вектор початкового стану $X(t_0) = X_0$ має рівне нулю середнє значення і коваріаційну матрицю

$$E\{X_0X_0^T\} = P(t_0), \quad (2.7)$$

при цьому

$$E\{V(t)X_0^T\} = E\{W(t)X_0^T\} = 0, \quad t \geq t_0. \quad (2.8)$$

Фільтр Калмана призначений для формування значення лінійної функції від $Z(t)$, позначеної через $\hat{X}(t)$, такої, щоб критерій якості

$$J = E \left\{ \left[X(t) - \hat{X}(t) \right]^T \left[X(t) - \hat{X}(t) \right] \right\} \quad (2.9)$$

мав мінімальне значення, а оцінка $\hat{X}(t)$ була б не зсунута для $X(t)$, $E\{\hat{X}(t)\} = E\{X(t)\}$.

Нехай, $\tilde{X}(t) = X(t) - \hat{X}(t)$ - похибка оцінювання в момент часу t , а $P(t) = E\{\tilde{X}(t)\tilde{X}^T(t)\}$ - дисперсійна матриця цієї похибки. Знайдемо функції $A(t, \tau)$ і $B(t, t_0)$, такі, щоб лінійна оцінка

$$\hat{X}(t) = B(t, t_0)\hat{X}(t_0) + \int_{t_0}^t A(t, \tau)Z(\tau)d\tau \quad (2.10)$$

забезпечувала мінімальне значення сліду $P(t)$ за умови $E\{\hat{X}(t)\} = 0$. Так, як

$$\tilde{X}(t) = X(t) - \hat{X}(t) = X(t) - B(t, t_0)\hat{X}(t_0) - \int_{t_0}^t A(t, \tau)Z(\tau)d\tau, \quad (2.11)$$

то

$$\begin{aligned} P(t) &= E\{\tilde{X}(t)\tilde{X}^T(t)\} = E\{X(t)X^T(t)\} - E\left\{ \int_{t_0}^t X(t)Z^T(\tau)A^T(t, \tau)d\tau \right\} - \\ &- E\left\{ \int_{t_0}^t A(t, \tau)Z(\tau)X^T(t)d\tau \right\} + E\left\{ \int_{t_0}^t \int_{t_0}^t A(t, \tau)Z(\tau)Z^T(\lambda)A^T(t, \lambda)d\tau d\lambda \right\} + \\ &+ E\{B(t, t_0)\hat{X}(t_0)\hat{X}^T(t_0)B^T(t, t_0)\} - \\ &- E\left\{ B(t, t_0)\hat{X}(t_0) \left[X^T(t) - \int_{t_0}^t Z^T(\tau)A^T(t, \tau)d\tau \right] \right\} - \\ &- E\left\{ \left[X(t) - \int_{t_0}^t A(t, \tau)Z(\tau)d\tau \right] \hat{X}^T(t_0)B^T(t, t_0) \right\}. \end{aligned} \quad (2.12)$$

Обчислимо математичне сподівання відповідних виразів, враховуючи, що $\hat{X}(t_0)$ не є випадковою величиною. Приводячи останні дві складові до вигляду правої частини (2.11) та враховуючи, що математичне сподівання (2.11) дорівнює нулю, отримаємо:

$$\begin{aligned} P(t) &= \text{cov}\left[\tilde{X}(t), \tilde{X}(t)\right] = \text{cov}\left[X(t), X(t)\right] - \\ &- \int_{t_0}^t \text{cov}\left[X(t), Z(\tau)\right]A^T(t, \tau)d\tau - \int_{t_0}^t A(t, \tau)\text{cov}\left[Z(\tau), X(t)\right]d\tau + \\ &+ \int_{t_0}^t \int_{t_0}^t A(t, \tau)\text{cov}\left[Z(\tau), Z(\lambda)\right]A^T(t, \lambda)d\tau d\lambda - B(t, t_0)\hat{X}(t_0)\hat{X}^T(t_0)B^T(t, t_0). \end{aligned} \quad (2.13)$$

Тепер скористаємося основною ідеєю варіаційного числення: покладемо в (2.13)

$$\begin{aligned}
A(t, \tau) &= \hat{A}(t, \tau) + \varepsilon \eta(t, \tau), \\
B(t, \tau) &= \hat{B}(t, \tau) + \varepsilon v(t, \tau), \\
X(t_0) &= \hat{X}_0(t_0) + \varepsilon \Gamma(t_0).
\end{aligned} \tag{2.14}$$

Далі необхідно розв'язати рівняння

$$\partial \text{cov}[\tilde{X}(t), \tilde{X}(t)] / \partial \varepsilon|_{\varepsilon=0} = 0 \tag{2.15}$$

відносно $\hat{A}(t, \tau)$ і $\hat{X}_0(t_0)$. Це рівняння повинно залишатися

справедливим при будь-яких функціях $v(t, \tau)$, $\eta(t, \tau)$ та $\Gamma(t_0)$. В результаті отримаємо:

$$\begin{aligned}
& - \int_{t_0}^t \eta(t, \tau) \text{cov}[Z(\tau), X(t)] d\tau + \int_{t_0}^t \int_{t_0}^t \eta(t, \tau) \text{cov}[Z(\tau), Z(\lambda)] \hat{A}^T(t, \lambda) d\tau d\lambda + \\
& + v(t, t_0) \hat{X}_0(t_0) \hat{X}_0^T(t_0) \hat{B}^T(t, t_0) + \hat{B}(t, t_0) [\hat{X}_0(t_0) \Gamma^T(t_0) + \Gamma(t_0) \hat{X}_0^T(t_0)] \times \\
& \times \hat{B}^T(t, t_0) - \int_{t_0}^t \text{cov}[X(t), Z(\tau)] \eta^T(t, \tau) d\tau + \\
& + \int_{t_0}^t \int_{t_0}^t \hat{A}(t, \tau) \text{cov}[Z(\tau), Z(\lambda)] \eta^T(t, \tau) d\tau d\lambda - \hat{B}(t, t_0) \hat{X}_0(t_0) \hat{X}_0^T(t_0) v^T(t, t_0) = 0.
\end{aligned} \tag{2.16}$$

Відмітимо, що останні три складові в (2.16) являють собою, просто результат транспонування перших трьох членів цього виразу. Відповідно, будь-яку з цієї пари сум можна покласти рівній нулю. В результаті маємо:

$$\begin{aligned}
0 &= \int_{t_0}^t \left\{ \text{cov}[X(t), Z(\lambda)] - \int_{t_0}^t \hat{A}(t, \tau) \text{cov}[Z(\tau), Z(\lambda)] d\tau \right\} \eta^T(t, \lambda) d\lambda + \\
& + \hat{B}(t, t_0) \hat{X}_0(t_0) \hat{X}_0^T(t_0) v^T(t, t_0)
\end{aligned} \tag{2.17}$$

для яких завгодно функцій $\eta(t, \lambda)$ і $v(t, t_0)$. Далі, щоб середня складова в (2.16) обернулася в нуль при довільній матриці $\Gamma(t)$, очевидно, необхідно мати $\hat{X}_0(t_0) = 0$. Таким чином, оптимальний фільтр характеризується одним з наступних еквівалентних рівностей:

$$\text{cov}[X(t), Z(\lambda)] = \int_{t_0}^t \hat{A}(t, \tau) \text{cov}[Z(\tau), Z(\lambda)] d\tau, \quad t_0 < \lambda < t, \tag{2.18}$$

$$\text{cov}\{\tilde{X}(t), Z(\lambda)\} = 0, \quad \text{cov}\{\tilde{X}(t), \tilde{X}(t)\} = 0.$$

(2.19)

кожне з яких складає зміст твердження леми про ортогональне проєкціювання.

Тепер, зведемо матричне інтегральне рівняння Вінера-Хопфа (2.18) до диференційного рівняння. Це можна зробити шляхом обчислення похідної за часом від обох частин рівняння (2.18). Виконуючи спочатку диференціювання лівої частини, отримаємо:

$$\begin{aligned}
\frac{\partial}{\partial t} \{ \text{cov}[X(t), Z(\lambda)] \} &= \text{cov}[X(t), Z(\lambda)] = \text{cov}[F(t)X(t) + G(t)W(t), Z(\lambda)] = \\
& = F(t) \text{cov}[X(t), Z(\lambda)] + G(t) \text{cov}[W(t), Z(\lambda)].
\end{aligned} \tag{2.20}$$

З (2.18) очевидно, що змінна λ приймає значення з обмеженого інтервалу $t_0 < \lambda < t$. Тому, так як завжди $t > \lambda$, $\text{cov}[W(t), Z(t)] = 0$. Це значить, що кореляція між значенням на вході в момент часу t і значенням процесу на виході

при будь-якому $\lambda < t$ відсутнє. В результаті рівність (2.20) приймає вигляд:

$$\frac{\partial}{\partial t} \left\{ \text{cov} [X(t), Z(\lambda)] \right\} = F(t) \text{cov} [X(t), Z(\lambda)]. \quad (2.21)$$

Виконаємо тепер диференціювання за часом правої частини рівняння (2.18):

$$\begin{aligned} \frac{\partial}{\partial t} \left\{ \int_{t_0}^t \hat{A}(t, \tau) \text{cov} [Z(t), Z(\lambda)] d\tau \right\} = \\ = \hat{A}(t, t) \text{cov} [Z(t), Z(\lambda)] + \int_{t_0}^t \hat{A}(t, \tau) \text{cov} [Z(t), Z(\lambda)] d\tau. \end{aligned} \quad (2.22)$$

Прирівнюючи результати диференціювання, отримаємо:

$$\begin{aligned} F(t) \text{cov} [X(t), Z(\lambda)] = \\ = \hat{A}(t, t) \text{cov} [Z(t), Z(\lambda)] + \int_{t_0}^t \hat{A}(t, \tau) \text{cov} [Z(t), Z(\lambda)] d\tau. \end{aligned} \quad (2.23)$$

Відзначимо далі, що справедливі рівності

$$\text{cov} [X(t), Z(\lambda)] = E \left\{ X(t) [H(\lambda) X(\lambda) + V(\lambda)]^T \right\} = \text{cov} [X(t), X(\lambda)] H^T(\lambda) \hat{A}(t, t) \text{cov} [Z(t), Z(\lambda)] = \hat{A}(t, t) H(t) \text{cov} [X(t), Z(\lambda)] + \hat{A}(t, t) R(t) \delta(t - \lambda), \quad (2.24)$$

$$\begin{aligned} \text{cov} [Z(\tau), Z(\lambda)] = E \left\{ [H(\tau) X(\tau) + V(\tau)] [H(\lambda) X(\lambda) + V(\lambda)]^T \right\} = \\ = H(\tau) \text{cov} [X(\tau), X(\lambda)] H^T(\tau) + R(\tau) \delta(\lambda - \tau), \end{aligned} \quad (2.25)$$

$$\begin{aligned} \text{cov} [X(t), X(\lambda)] = \varphi(t, \gamma) \text{cov} [X(\gamma), X(\gamma)] \varphi^T(t, \gamma) + \\ + \int_{t_0}^t \varphi(t, \gamma) G(\gamma) Q(\gamma) G^T(\gamma) \varphi^T(t, \gamma) d\gamma, \end{aligned} \quad (2.26)$$

з яких (2.24) та (2.25) слідує безпосередньо з визначення X і Z . Деякі складові (2.10) зручно записати трохи в іншому вигляді. Нехай,

$$\begin{aligned} \hat{X}(t) = \int_{t_0}^t \hat{A}(t, \tau) Z(\tau) d\tau, \\ \hat{A}(t, t) = K(t), K(t) = \Xi(t) H^T(t) R^{-1}(t). \end{aligned} \quad (2.27)$$

Щоб знайти диференціальне рівняння для оптимального фільтра, розглянемо більш детально рівність (2.23). Якщо врахувати (2.18), то ліву частину рівності можна записати наступним чином:

$$F(t) \text{cov} [X(t), Z(\lambda)] = F(t) \int_{t_0}^t \hat{A}(t, \tau) \text{cov} [Z(\tau), Z(\lambda)] d\tau. \quad (2.28)$$

Так, як $Z(t) = H(t) X(t) + V(t)$, то використовуючи це представлення в першій складовій правої частини рівності (2.23), отримаємо:

$$\text{cov} [X(t), Z(\lambda)] = \hat{A}(t, t) H(t) \text{cov} [X(t), Z(\lambda)] + \hat{A}(t, t) R(t) \delta(t - \lambda). \quad (2.29)$$

Так як $\delta(t - \lambda) = 0$ для $\lambda < t$. Таким чином, після підстановки двох останніх виразів в (2.23) з урахуванням (2.18), приходимо до рівності

$$\int_{t_0}^t \left\{ F(t) \hat{A}(t, \tau) - \frac{\partial \hat{A}(t, \tau)}{\partial t} - \hat{A}(t, t) H(t) \hat{A}(t, \tau) \right\} \text{cov} [Z(\tau), Z(\lambda)] d\tau = 0, \quad (2.30)$$

справедливого для всіх λ , таких, що $t_0 \leq \lambda < t$. Ця рівність має місце в тому випадку, якщо співмножник перед $\text{cov}[Z(\tau), Z(\lambda)]$ рівний нулю. Відповідно, імпульсний відгук оптимального фільтра повинен задовольняти наступному диференційному рівнянню:

$$F(t)\hat{A}(t, \tau) - \frac{\partial}{\partial t}\hat{A}(t, \tau) - \hat{A}(t, \tau)H(t)\hat{A}(t, \tau) = 0, t_0 \leq \tau < t. \quad (2.31)$$

Об'єднуючи (2.31) з (2.27) та з похідною за часом цього виразу, відразу ж отримаємо диференціальне рівняння, що описує структуру пристрою оптимального оцінювання

$$\frac{d\hat{X}}{dt} = F(t)\hat{X}(t) + K(t)[Z(t) - H(t)\hat{X}(t)]. \quad (2.32)$$

З (2.24) – (2.26) і визначень (2.13), (2.27) отримаємо $\Xi = P$ та дисперсійне рівняння:

$$\frac{dP}{dt} = F(t)P(t) + P(t)F^T(t) - P(t)H^T(t)R^{-1}(t)H(t)P(t) + G(t)Q(t)G^T(t), \quad (2.23)$$

які дозволяють знайти коваріаційну матрицю вектора похибки оцінювання. Очевидно, що в якості початкових умов можна прийняти

$$\hat{X}(t_0) = 0, \quad (2.34)$$

$$P(t_0) = E[\tilde{X}(t_0), \tilde{X}^T(t_0)] = \text{cov}[X(t_0), X^T(t_0)]. \quad (2.35)$$

Можна побудувати структурну схему оптимального фільтра, що показана на рис. 2.1.

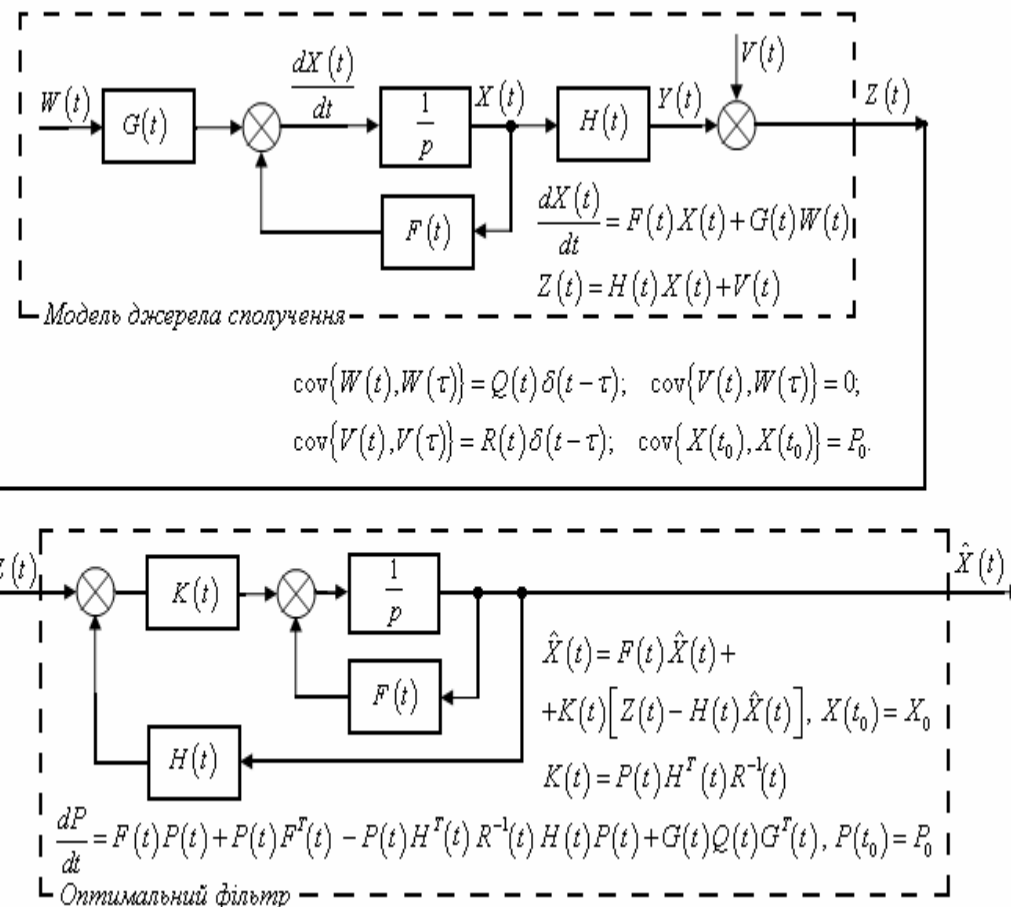


Рис. 2.1. Структурна схема моделі спостережуваного процесу і оптимального пристрою оцінювання стану

Більш конкретизуємо задачу синтезу фільтра Калмана для оцінювання стану в системах керування

$$\frac{dX}{dt} = AX + BU + GW - \text{рівняння стану},$$

$$Y_v = CX + DU + HW + V - \text{рівняння вимірювання},$$

(2.36)

де U – відомі входи та збурення за входами W і за вимірюванням V , що являються “білим” шумом і задовольняють

$$E(W) = E(V) = 0, \quad E(WW^T) = Q, \quad E(VV^T) = R, \quad E(WV^T) = N. \quad (2.37)$$

Розраховується оцінка вектора стану \hat{X} , що мінімізує коваріацію похибки

$$P = \lim_{t \rightarrow \infty} E\left(\{X - \hat{X}\}\{X - \hat{X}\}^T\right). \quad (2.38)$$

Оптимальним вирішенням є фільтр Калмана, що описується рівняннями:

$$\begin{aligned} \frac{d\hat{X}}{dt} &= A\hat{X} + BU + L(Y_v - C\hat{X} - DU), \\ \begin{bmatrix} \hat{X} \\ \hat{Y} \end{bmatrix} &= \begin{bmatrix} C \\ I \end{bmatrix} \hat{X} + \begin{bmatrix} D \\ 0 \end{bmatrix} U, \end{aligned} \quad (2.39)$$

де матриця коефіцієнтів зворотних зв'язків L , визначається на основі матричного алгебраїчного рівняння Ріккати. Цей оцінювач використовує відомі входи U та вимірювання Y_v розраховує оцінки стану \hat{X} та вектора виходу \hat{Y} , структурна схема якого представлена на рис. 2.2. Звернімо увагу на те, що \hat{Y} оцінює істинний вихід системи

$$Y = CX + DU + HW. \quad (2.40)$$

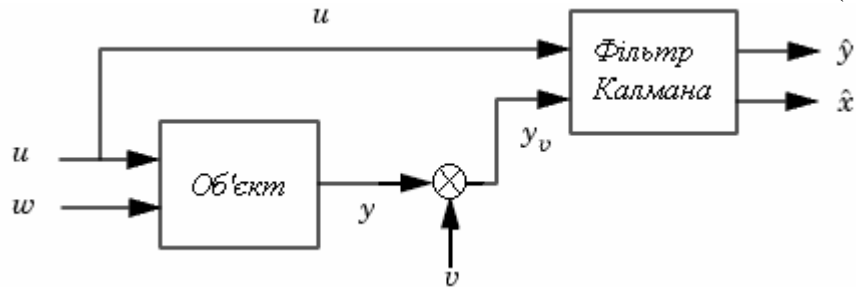


Рис. 2.2. Оцінювач Калмана.

Вимоги, що до існування фільтра Калмана:

- пара матриць $[C, A]$ має бути повністю спостережувана;
- мають виконуватися нерівності $\bar{R} > 0, \bar{Q} - \bar{N}\bar{R}^{-1}\bar{N}^T \geq 0$;
- пара матриць $[A - \bar{N}\bar{R}^{-1}C, \bar{Q} - \bar{N}\bar{R}^{-1}\bar{N}^T]$ не може мати спостережувані моди з власними значеннями на дійсній осі.

Тут

$$\bar{Q} = GQG^T; \bar{R} = R + HN + N^T H^T + HQH^T; \bar{N} = G(QH^T + N).$$

Приклад

Розглянемо прилад для системи представленої на рис. 2.3 з одним входом і одним виходом третього порядку, що описується рівняннями у просторі стану:

$$\begin{aligned} \frac{dX}{dt} &= AX + BU + GW, \\ Y_v &= CX + DU + HW + V, \end{aligned}$$

та рівняння фільтра Калмана:

$$\begin{aligned} \frac{d\hat{X}}{dt} &= A\hat{X} + BU + L(Y_v - C\hat{X} - DU), \\ \begin{bmatrix} \hat{Y} \\ \hat{X} \end{bmatrix} &= \begin{bmatrix} C \\ I \end{bmatrix} \hat{X} + \begin{bmatrix} D \\ 0 \end{bmatrix} U, \end{aligned}$$

$$\text{де } A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -25 & -10 & -3,5 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}; \quad G = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}; \quad C = [0 \quad 0 \quad 1];$$

$$D = 0; \quad H = 0,05,$$

$$\text{вхідний сигнал: } u(t) = \cos(\pi t), \quad t \in [0;5],$$

$$X(0) = \hat{X}(0) = [-0,5 \quad 0 \quad 0,5]^T,$$

та коваріаційні матриці:

$$E(WW^T) = Q = 2, \quad E(VV^T) = R = \frac{1}{2}, \quad E(W) = E(V) = 0,$$

$$E(WV^T) = N = 0.$$

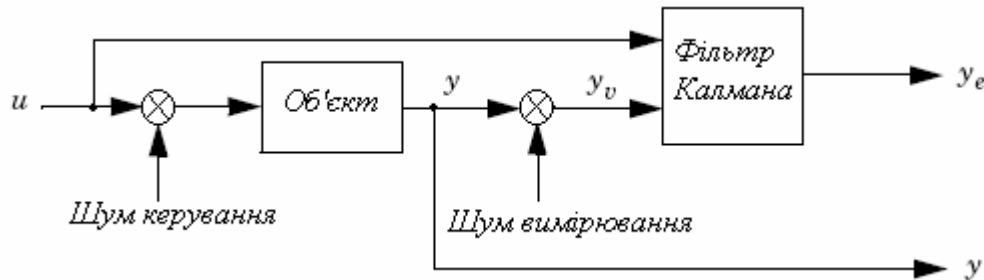


Рис. 2.3. Структурна схема системи

Розглянемо більш детально створення вище наведеної структурної схеми.

На входи фільтру Калмана *kamf* подаються сигнали U і Y_v , а з виходу знімаються $Y_e \equiv \hat{Y}$ і $X_e \equiv \hat{X}$ подаються на вхід оцінювача стану, як це показано на рис. 2.4.



Рис. 2.4. Фільтр Калмана

Для приєднання об'єкту до фільтру Калмана, використовується паралельне з'єднання, що показано на рис. 2.5.

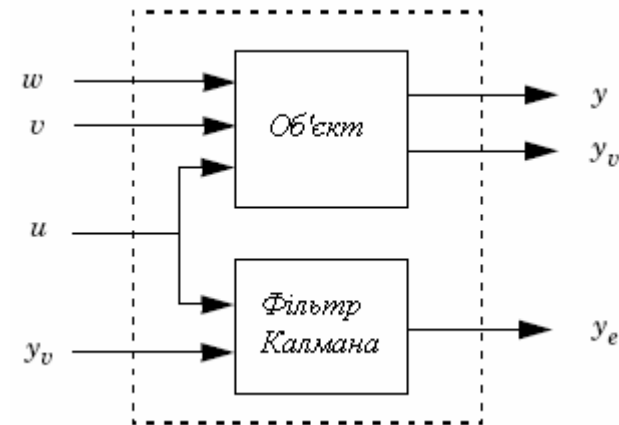


Рис. 2.5. Паралельне з'єднання об'єкта і фільтру Калмана

Наведемо приклад програми для розв'язку цієї задачі з назвою "Continuous_Kalman_Filter".

```
function Continuous_Kalman_Filter
A = [0 1 0; 0 0 1; -25 -10 -3.5]; % Введення
матриці A
B = [0 0 1]'; % Введення матриці B
G = [0 0 1]'; % Введення матриці G
C = [0 0 1]; % Введення матриці C
D = 0; % Введення матриці D
H = 0.05; % Введення матриці H
disp('Матриця Q:')
Q = 2 % Введення матриці Q
disp('Матриця R:')
R = 1/2 % Введення матриці R
disp('Матриця N:')
N = [] % Введення матриці N
disp('Вектор початкових умов x0:')
x0 = [-0.5 0 0.5]' % завдання вектора початкових
умов
disp('Вектор початкових умов x_e0:')
x_e0 = [-0.5 0 0.5]' % завдання вектора
початкових умов
disp('Час спостереження tf:')
tf = 5 % завдання часу спостереження
disp('Довжина кроку dt:')
```

```

dt = 0.05 % Завдання довжини кроку
% Формуємо дискретну модель ТОК
disp('Модель системи у просторі стану:')
Plant = ss(A,[B G],C,[D H], 'inputname',{ 'u'
'w' }, 'outputname', 'y')
% Проектування дискретного фільтра Калмана
[kest,L,P] = kalman(Plant,Q,R,N);
disp('Коефіцієнт підсилення фільтра:'); L
disp('Розв'язок рівнянь Рікатті:'); P
disp('Неперервна модель фільтра Калмана:'); kest
% Формування моделі процесу
P = ss(A,[B G
D*B],[C;C],[zeros(size(C));C], 'inputname',...
{'u' 'w' 'v'}, 'outputname',{'y' 'yv'});
sys = parallel(P,kest,1,1,[],[]); % Паралельне
з'єднання ТОК і фільтра
SimModel = feedback(sys,1,4,2,1); % Приєднуємо
вихід ТОК (#4) до входу фільтра (#2)
SimModel = SimModel([1 3],[1 2 3]); % Видалення
yv з I/O списку
% Моделювання поведінки фільтра
t = [0:dt:tf]'; % Час спостереження
u = cos(pi*t); % формування вхідного сигналу
n = length(t); % Визначення довжини вектора
w = sqrt(Q)*randn(n,1); % Шум керування
v = sqrt(R)*randn(n,1); % Шум вимірювання
[out,t,x] = lsim(SimModel,[w,v,u],t,[x0;x_e0]); %
Розрахунок виходів системи
y = out(:,1); % Вихід системи без фільтрації
ye = out(:,2); % Вихід системи після фільтрації
yv = y + v; % Зважений вихід системи
% Формування векторів x, e_e
for i = 1:length(2*A),
    x_e(:,i) = X(:,i+length(2*A));
    x(:,i) = X(:,i);
end
% Побудова динаміки змінних стану
figure(1)
plot(t,x), grid on
hold on, plot(t,x_e,'--'), hold off

```

```

title('динаміка руху змінних стану');
xlabel('Час, t');
ylabel('Змінні стану, X(t) та X_e(t)');
legend('x_1','x_2','x_3','x_e_1','x_e_2','x_e_3')
% Побудова графіку вихода системи
figure(2)
plot(t,y,'-',t,ye,'-',t,yv,'-'), grid on
xlabel('Час, t'), ylabel('Вихід системи: y, y_e,
y_v'),
title('динаміка зміни виходу системи'),
legend('y - без фільтрації','y_e - після
фільтрації','y_v - зважений',1),
% Побудова графіку похибки
figure(3)
plot(t,y-yv,'-',t,y-ye,'-'), grid on
title('динаміка зміни похибки'); xlabel('Час,
t');
ylabel('похибки: Y(t)-Y_v(t), Y(t)-Y_e(t)');
legend('Y_v - вимірювання','Y_e - фільтрації')
% Побудова вхідного сигналу
figure(4)
plot(t,u), grid
title('динаміка зміни вхідного сигналу');
xlabel('Час, t');
ylabel('Вхідний сигнал, U(t)'); legend('u -
вхідний сигнал');
% Обчислення коваріацій похибок
MeasErr = y-yv;
MeasErrCov =
sum(MeasErr.*MeasErr)/length(MeasErr);
EstErr = y-ye;
EstErrCov = sum(EstErr.*EstErr)/length(EstErr);
disp('Коваріація похибки перед фільтрацією
(похибка вимірювання):')
MeasErrCov
disp('Коваріація похибки після фільтрації
(похибка оцінювання):')
EstErrCov

```

Результат виконання програми


```

0.0077  0.0021  -0.0456
0.0021  0.0456  0.0000
-0.0456 0.0000  0.5069

```

```

x1_e    1    0    0
x2_e    0    1    0
x3_e    0    0    1

```

d =

```

      u  y
y_e   0  0
x1_e  0  0
x2_e  0  0
x3_e  0  0

```

Input groups:

Name	Channels
KnownInput	1
Measurement	2

Output groups:

Name	Channels
OutputEstimate	1
StateEstimate	2,3,4

Continuous-time model.

Коваріація похибки перед фільтрацією (похибка вимірювання):

MeasErrCov =

0.4108

Коваріація похибки після фільтрації (похибка оцінювання):

EstErrCov =

0.0304

Неперервна модель фільтра Калмана:

a =

	x1_e	x2_e	x3_e
x1_e	0	1	0.09021
x2_e	0	0	1
x3_e	-25	-10	-4.702

b =

	u	y
x1_e	0	-0.09021
x2_e	0	4.483e-018
x3_e	1	1.202

c =

	x1_e	x2_e	x3_e
y_e	0	0	1

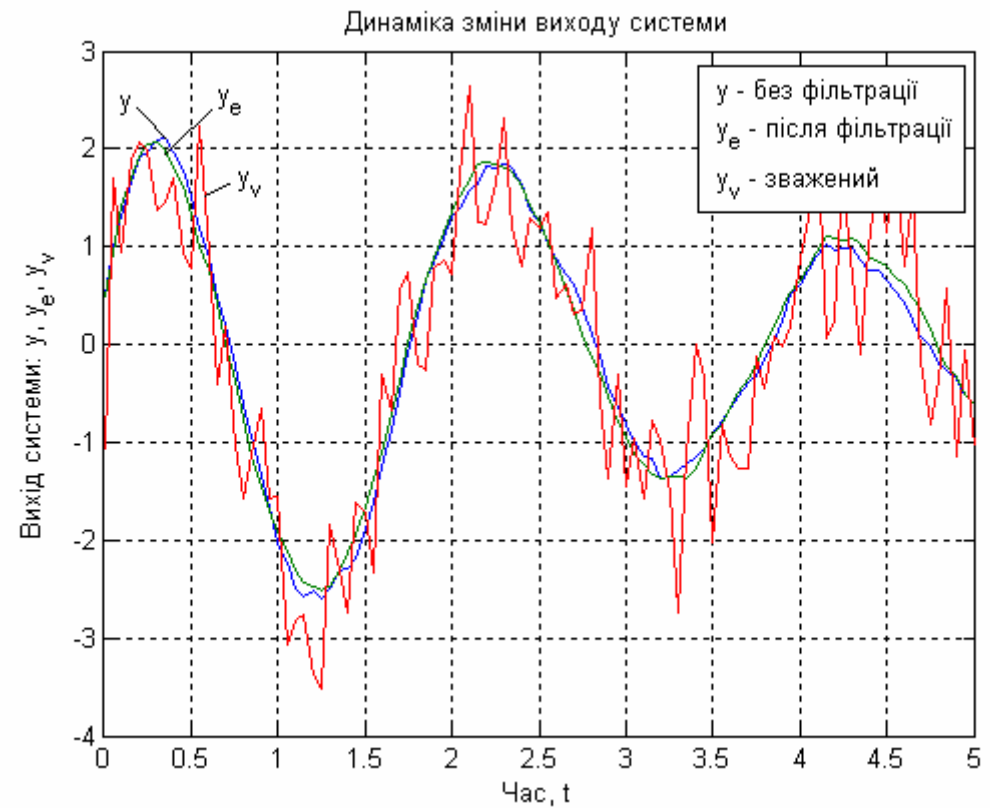
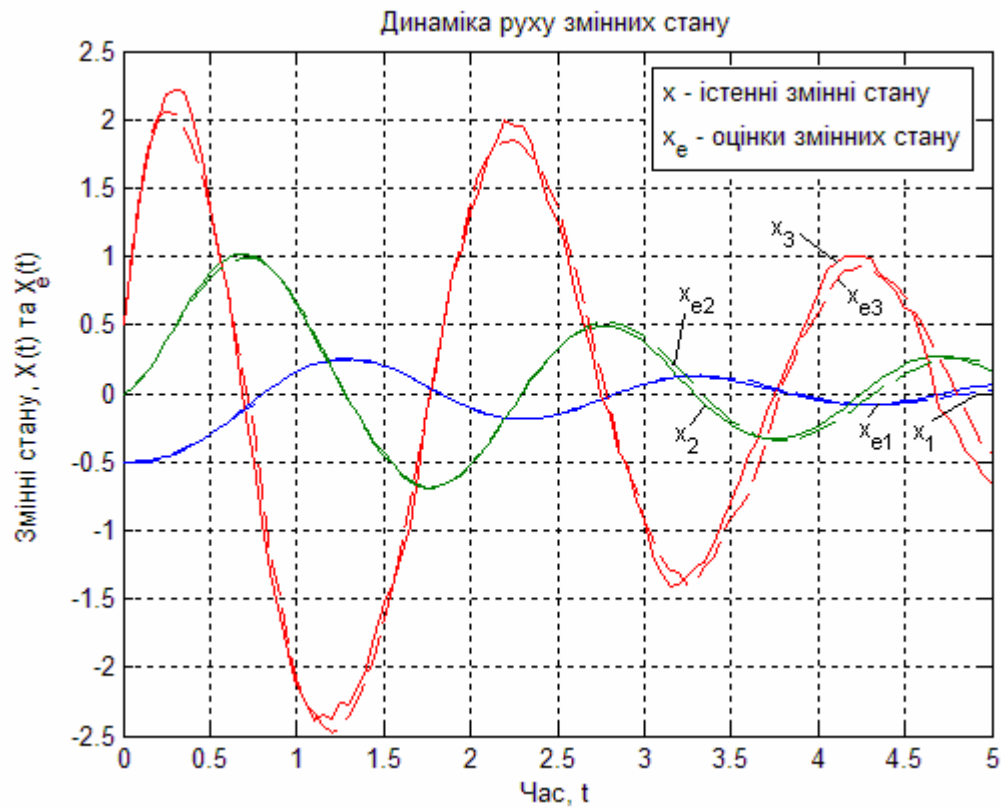
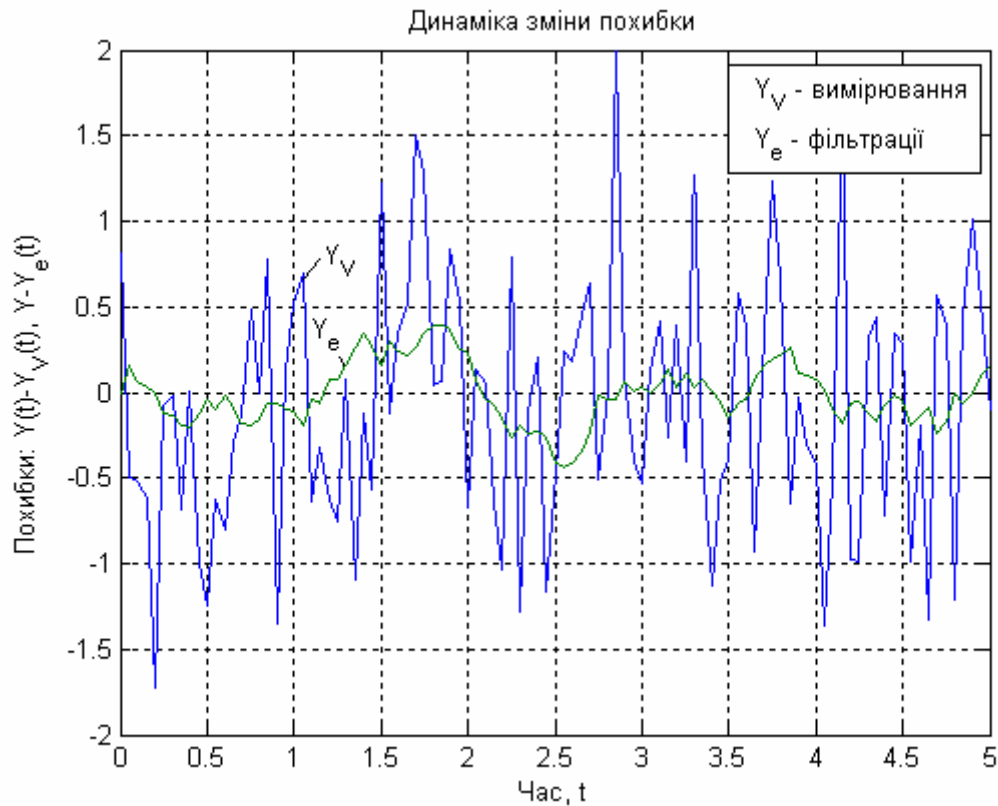
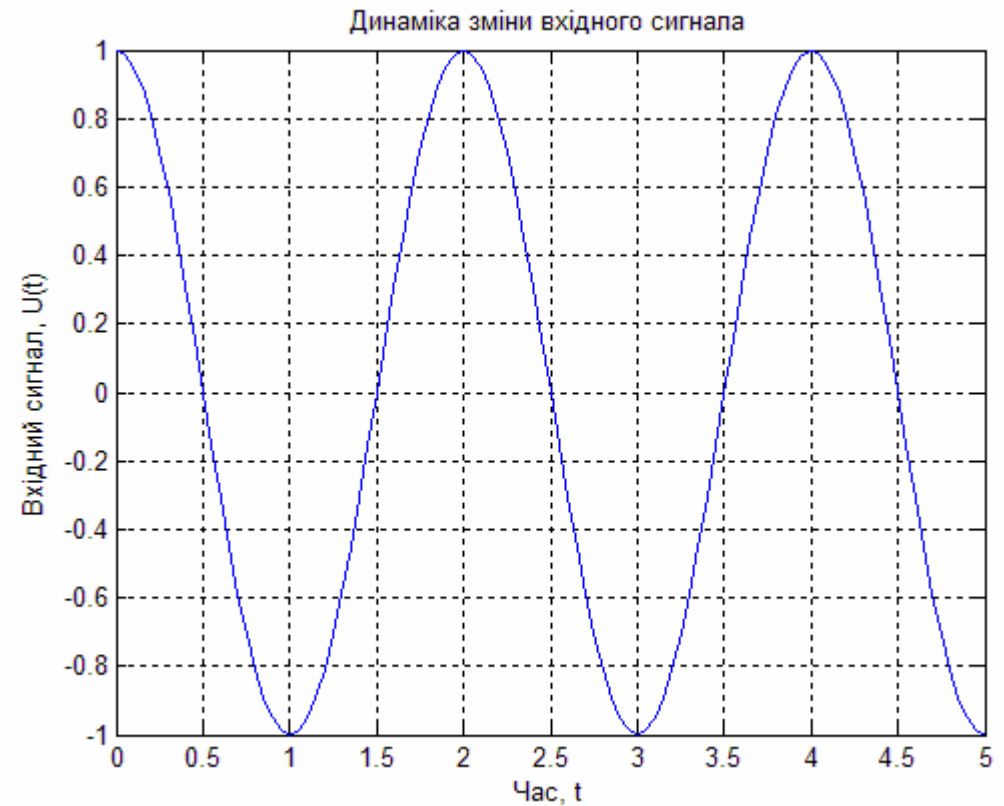


Рис. 2.6. Динаміка руху змінних стану а) і виходу системи б)



а)



б)

Рис. 2.7. Динаміка зміни похибки а) і вхідного сигналу б)

Дослідимо якість отриманого фільтра Калмана для наступних значеннях параметрів: $X(0) = \hat{X}(0) = [-0,5 \ 0 \ 0,5]^T$;

$$E(WW^T) = Q = 2; \quad E(VV^T) = R = \frac{1}{2}; \quad E(W) = E(V) = 0;$$

$E(WV^T) = N = 0$. Коефіцієнт підсилення фільтра L отриманий за допомогою розв'язку алгебраїчного рівняння Ріккати (дисперсійного рівняння). Виходи фільтру Калмана, показані на рис. 2.6 а), де X - без фільтрації; X_e - після фільтрації (штрихова крива), на рис. 2.6 б)

– показані сигнали: без фільтрації; після фільтрації та зважений, на рис. 2.7 а) – показано дві похибки: вимірювання; фільтрації, на рис. 2.7 б) – показано сигнал, що подається на вхід системи. Якість отриманого фільтра суттєво залежить від дисперсійних матриць Q, R, N . Відмітимо, що оцінки збігаються до істинних значень достатньо швидко, не дивлячись на відносно високий рівень завад і неточність завдання початкових оцінок.

2.2. Дискретний фільтр Калмана

Розглянемо задачу алгоритму оцінювання для дискретного часу, формулюючи лінійну не зсунуту оцінку з мінімальною дисперсією. Запишемо векторне різницеве рівняння:

$$X[n+1] = \varphi[n+1, n]X[n] + \Gamma[n]W[n] \quad (2.41)$$

і рівняння для спостережуваних даних:

$$Z[n] = H[n]X[n] + V[n], \quad (2.42)$$

де

$$E\{W[n]\} = E\{V[n]\} = 0, \quad E\{X[n_0]\} = \bar{X}[n_0], \quad \text{var}\{X[n_0]\} = P[n_0],$$

$$E\{W[n]W^T[j]\} = Q[n]\delta[n-j], \quad E\{V[n]V^T[j]\} = R[n]\delta[n-j]$$

і

$$\text{cov}\{V[n_0], X[n_0]\} = \text{cov}\{W[n], X[n_0]\} = \text{cov}\{V[n], W[j]\} = 0.$$

Як і раніше, припустимо, що $\text{cov}\{V[n], W[j]\} = 0$, введено для спрощення задачі. Тут δ символ Кронекера.

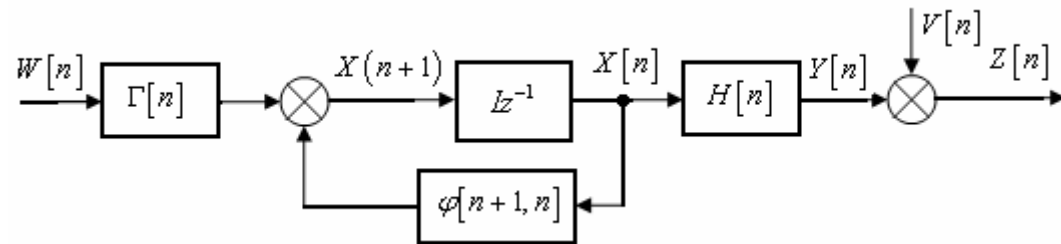


Рис. 2.6. Модель спостережуваного процесу в задачі оптимальної фільтрації за дискретного часу

На рис. 2.6 представлена структурна схема пристрою, реалізуючого введення моделі для вектора стану і спостережуваного процесу. Перша задача, яку потрібно розв'язати, постає в тому, щоб показати, що середнє значення та коваріаційна матриця процесу $X[n]$ задовольняє співвідношенням

$$E\{X[n+1]\} = \bar{X}[n+1] = \varphi[n+1, n]\bar{X}[n], \quad (2.43)$$

$$\begin{aligned} \text{var}\{X[n+1]\} &= P[n+1] = E\left\{[X[n+1] - \bar{X}[n+1]][X[n+1] - \bar{X}[n+1]]^T\right\} = \\ &= \text{var}\{\varphi[n+1, n][X[n] - \bar{X}[n]] + \Gamma[n]W[n]\} = \\ &= \varphi[n+1, n]P[n]\varphi^T[n+1, n] + \Gamma[n]Q[n]\Gamma^T[n], \end{aligned} \quad (2.44)$$

де вектор $\bar{X}[n_0]$ і матриця $P[n_0]$ задані в якості початкових умов.

Неважко показати, що

$$E\{X[n+1] | X[n]\} = \varphi[n+1, n]X[n], \quad (2.45)$$

$$\text{var}\{X[n+1] | X[n]\} = \Gamma[n]Q[n]\Gamma^T[n]. \quad (2.46)$$

Перехід від дискретного часу до неперервного часу для даного процесу, може бути реалізованим звичайним граничним переходом

$$\frac{dX(t)}{dt} = \lim_{T \rightarrow 0} \frac{X(t=nT) - X(t+T=(n+1)T)}{T}.$$

Також легко показати, що при такому переході різницеве рівняння для коваріаційної матриці за дискретного часу:

$$P[n+1] = \varphi[n+1, n]P[n]\varphi^T[n+1, n] + \Gamma[n]Q[n]\Gamma^T[n]$$

перехід в диференційне рівняння за неперервного часу

$$\frac{dP(t)}{dt} = A(t)P(t) + P(t)A^T(t) + B(t)Q(t)B^T(t),$$

а рівняння для вектора стану системи

$$X[n+1] = \varphi[n+1, n]X[n] + \Gamma[n]W[n]$$

перехід в диференційне рівняння

$$\frac{dX(t)}{dt} = A(t)X(t) + B(t)W(t),$$

де $\text{cov}\{W(t)\} = Q(t)/T$ так, що при $T \rightarrow 0$ коваріаційна функція процесу $W(t)$ містить дельта-функцію, а

$$A(t) = \lim_{T \rightarrow 0} \frac{1 - \varphi(t+T, t)}{T}, \quad B(t) = \lim_{T \rightarrow 0} \frac{\Gamma(t)}{T}, \quad (2.47)$$

$$\text{cov}[W(t_1), W(t_2)] = Q(t_1)\delta(t_1 - t_2), \quad Q(t) = \lim_{\substack{T \rightarrow 0 \\ nT \rightarrow t}} TQ(nT). \quad (2.48)$$

Інші вирази, що відносяться до неперервного часу для середніх і коваріаційних матриць, також можуть бути отримані з відповідних виразів для дискретного часу.

Знайдемо тепер функцію $\hat{X}[n+1|n]$, що має наступні властивості: вона є лінійною відносно послідовності даних $Z[n] = \{Z[n_0+1], \dots, Z[n]\}$ та початкового значення $\bar{X}(t_0)$; вона є умовно і без умовно незсуненою статистичною у тому випадку, якщо

$$E\{\hat{X}[n+1|n] | Z[n]\} = E\{X[n+1] | Z[n]\} \quad i$$

$E\{\hat{X}[n+1|n]\} = E\{X[n+1]\}$; вона забезпечує мінімальне значення для $\text{var}\{\tilde{X}[n+1|n] | Z[n]\}$ в класі всіх таких самих лінійних функцій від пари $(Z[n], \bar{X}[n_0])$; тут

$$\tilde{X}[n+1, n] = X[n+1] - \hat{X}[n+1|n] \quad (2.49)$$

називається похибкою передбачення на один крок. Функція $\hat{X}[n+1|n]$ описує структуру оптимального лінійного екстраполятора на один крок; вона буде використана надалі для виводу рівнянь для оптимального лінійного фільтру.

З метою отримання рекурентної формули для $\hat{X}[n+1|n]$, скористаймося теоремою про ортогональне проєціювання та зв'язаними з нею результатами. Лінійна оцінка вектора \bar{X} з мінімальною дисперсією похибки, визначена на лінійному просторі спостережень Z , може бути знайдена як ортогональна проєкція X на Z , тобто $\hat{X} = E\{X | Z\}$. Будемо використовувати символ \hat{E} замість E у зв'язку з тим, щоб підкреслити, що лінійна оцінка з мінімальною похибкою оцінки в загальному випадку не співпадає з умовним середнім значенням. Якщо всі розглянуті процеси є гауссовими, то $\hat{E}\{X | Z\} = E\{X | Z\}$. В негауссовому випадку оцінка, що отримана таким способом, є лише кращою лінійною оцінкою і не є обов'язково кращою в більш широкому класі оцінок.

Нехай ортогональна послідовність $\{\alpha_i\}^{m_i=1}$ утворює базис для Z . Тоді маємо

$$\hat{E}\{X | Z\} = \sum_{i=1}^m E\{X \alpha_i^T\} E\{\alpha_i \alpha_i^T\}^{-1} \alpha_i \quad (2.50)$$

З цього результату також випливає, якщо вектор α^* ортогональний Z , то $E\{Z^T \alpha^*\} = 0$, де $\alpha^* = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$, то

$$\hat{E}\{X | Z, \alpha^*\} = \hat{E}\{X | Z\} + \hat{E}\{X | \alpha^*\}. \quad (2.51)$$

При виведенні рекурентної формули для $\hat{X}[n+1|n]$ за індукцією покладається, що значення $\hat{X}[n|n-1]$ відоме та $\hat{X}[n+1|n]$ є функцією від $\hat{X}[n|n-1]$ і $Z[n]$. Щоб скористатися співвідношенням (2.51), необхідно знайти вектор α^* , ортогональний до $Z[n-1]$, так як в загальному випадку вектор $Z[n]$ не ортогональний до $Z[n-1]$. Незавжно показати, що оновлюючий вектор

$$\tilde{Z}[n] = Z[n] - H[n] \hat{X}[n|n-1] \quad (2.52)$$

ортогональний до $Z[n-1]$. Можна також показати, що лінійний векторний простір, утворений спостереженнями $\{Z[n], Z[n-1]\}$ та $\{\tilde{Z}[n], Z[n-1]\}$, ідентичні.

Внаслідок, з (2.51) маємо:

$$\begin{aligned} \hat{X}[n+1|n] &= \hat{E}\{X[n+1] | Z[n]\} = \hat{E}\{X[n+1] | \{\tilde{Z}[n], Z[n-1]\}\} = \\ &= \hat{E}\{X[n+1] | Z[n-1]\} + \hat{E}\{X[n+1] | \tilde{Z}[n]\}. \end{aligned} \quad (2.53)$$

На основі рівняння (2.41) та введених припущеннях властивостей шумів $W[n]$ і $V[n]$ можна записати:

$$\hat{E}\{X[n+1] | Z[n-1]\} = \varphi[n+1|n] \hat{X}[n|n-1]. \quad (2.54)$$

Можна також показати, що $\hat{X}[n+1|n] = \varphi[n+1|n] \hat{E}\{X[n] | Z[n]\}$; цей результат буде

використанам надалі. Таким чином, перший доданок правої частини рівності (2.53) можна представити як функцію від $X[n|n-1]$.

Розглянемо зараз другий доданок правої частини (2.53). З рівності (2.50) випливає, що

$$\hat{E}\{X[n+1] | \tilde{Z}[n]\} = E\{X[n+1] \tilde{Z}^T[n]\} \left[E\{\tilde{Z}[n] \tilde{Z}^T[n]\} \right]^{-1} \tilde{Z}[n] \quad (2.55)$$

Помічаємо, що $Z[n] = H[n] X[n|n-1] + V[n]$, отримаємо

$$\begin{aligned} E\{X[n+1] \tilde{Z}^T[n]\} &= E\{\left[\varphi[n+1|n] X[n] + \Gamma[n] W[n] \right] \times \\ &\times \left[H[n] \tilde{X}[n|n-1] + V[n] \right]^T\} = \varphi[n+1|n] E\{X[n] \tilde{X}^T[n|n-1]\} H^T[n] + \\ &+ \varphi[n+1|n] E\{X[n] V^T[n]\} + \Gamma[n] E\{W[n] \tilde{X}^T[n|n-1]\} H^T[n] + \\ &+ \Gamma[n] E\{W[n] V^T[n]\} = \varphi[n+1|n] E\{X[n] \tilde{X}^T[n|n-1]\} H^T[n], \end{aligned} \quad (2.56)$$

де всі останні складові рівні нулю, внаслідок введених припущень властивостей шумів. У відповідності з (2.49) маємо:

$$\begin{aligned} E\{X[n] \tilde{X}^T[n|n-1]\} &= E\{\hat{X}[n|n-1] \tilde{X}^T[n|n-1]\} + \\ &+ E\{\tilde{X}[n|n-1] \tilde{X}^T[n|n-1]\} = E\{\tilde{X}[n|n-1] \tilde{X}^T[n|n-1]\}, \end{aligned} \quad (2.57)$$

де доданок $E\{\hat{X}[n|n-1] \tilde{X}^T[n|n-1]\}$ дорівнює нулю на підставах теореми про ортогональне проєкціювання. Якщо ввести позначення $\sum[n|n-1] = E\{\tilde{X}[n|n-1] \tilde{X}^T[n|n-1]\}$, то з (2.56) і (2.57) випливає, що

$$E\{X[n+1] \tilde{Z}^T[n]\} = \varphi[n+1|n] \sum[n|n-1] H^T[n]. \quad (2.58)$$

Аналогічно, незважно показати, що

$$E\{\tilde{Z}[n]\tilde{Z}^T[n]\} = H[n]\sum[n|n-1]H^T[n] + R[n]. \quad (2.59)$$

Підставляючи (2.58) і (2.59) у (2.55), отримаємо

$$\begin{aligned} \hat{E}\{X[n+1]|\tilde{Z}[n]\} &= \varphi[n+1|n]\sum[n|n-1]H^T[n] \times \\ &\times [H[n]\sum[n|n-1]H^T[n] + R[n]]^{-1}\tilde{Z}[n]. \end{aligned} \quad (2.60)$$

Нарешті, підстановка (2.54) і (2.60) у (2.53) дозволить записати рекурентну формулу для лінійної оцінки з мінімальною дисперсією наступним чином:

$$\begin{aligned} \hat{X}[n+1|n] &= \varphi[n+1|n]\hat{X}[n|n-1] + \varphi[n+1|n]\sum[n|n-1]H^T[n] \times \\ &\times [H[n]\sum[n|n-1]H^T[n] + R[n]]^{-1}[Z[n] - H[n]\hat{X}[n|n-1]]. \end{aligned} \quad (2.61)$$

Часто (2.61) записують в іншій формі:

$$\hat{X}[n+1|n] = \varphi[n+1|n]\hat{X}[n|n-1] + \mathbf{K}[n][Z[n] - H[n]\hat{X}[n|n-1]] \quad (2.62)$$

де коефіцієнт підсилення Калмана $\mathbf{K}[n]$ визначається співвідношенням:

$$\mathbf{K}[n] = \varphi[n+1|n]\sum[n|n-1]H^T[n][H[n]\sum[n|n-1]H^T[n] + R[n]]^{-1}. \quad (2.63)$$

З (2.63) випливає, що при обчисленні коефіцієнта підсилення Калмана необхідно знаходити коваріаційну матрицю $\sum[n|n-1]$, для якої, як зараз буде показано, можливо отримати рекурентне співвідношення. Легко показати, що вектор похибок $\tilde{X}[n+1|n]$ задовольняє наступному рекурентному співвідношенню:

$$\tilde{X}[n+1|n] = \varphi[n+1|n]\tilde{X}[n|n-1] + \Gamma[n]W[n] - \mathbf{K}[n]\tilde{Z}[n]. \quad (2.64)$$

Як видно, що вектор $W[n]$ незалежний, а отже, і ортогональний як до $\tilde{X}[n|n-1]$, так і до $\tilde{Z}[n]$. Нагадуємо, що

$$\tilde{Z}[n] = H[n]\tilde{X}[n|n-1] + V[n],$$

Помітимо також, що

$$E\{\tilde{Z}[n]\tilde{X}^T[n|n-1]\} = H[n]\sum[n|n-1].$$

Тоді з визначення $\sum[n|n-1]$ і рівності (2.59), (2.62) та шойно записаного співвідношення отримаємо наступне рекурентне співвідношення для матриці:

$$\begin{aligned} \sum[n+1|n] &= \varphi[n+1|n]\sum[n|n-1]\varphi^T[n+1|n] + \Gamma[n]Q[n]\Gamma^T[n] + \\ &+ \mathbf{K}[n][H[n]\sum[n|n-1]H[n] + R[n]]\mathbf{K}^T[n] - \\ &- \varphi[n+1|n]\sum[n|n-1]H^T[n]\mathbf{K}^T[n] - \mathbf{K}[n]H[n]\sum[n|n-1]\varphi^T[n+1|n]. \end{aligned} \quad (2.65)$$

Існує декілька інших способів запису отриманого рекурентного співвідношення. Наприклад, підстановка коефіцієнта $\mathbf{K}[n]$, визначеного (2.63), у вираз (2.65) після декількох алгебраїчних перетворень призводить до рівняння:

$$\begin{aligned} \sum[n+1|n] &= \varphi[n+1|n]\sum[n|n-1]\varphi^T[n+1|n] - \\ &- \varphi[n+1|n]\sum[n|n-1]H^T[n][H[n]\sum[n|n-1]H^T[n] + R[n]]^{-1} \times \\ &\times H[n]\sum[n|n-1]\varphi^T[n+1|n] + \Gamma[n]Q[n]\Gamma^T[n], \end{aligned} \quad (2.66)$$

яке відомо як рівняння Ріккати.

Співвідношення (2.62) являє собою рекурсивний запис, що описує лінійний екстраполятор на один крок з мінімальною дисперсією похибки, для застосування якого потрібно знати коваріаційну матрицю цієї мінімальної похибки. Для цієї коваріаційної матриці отримані також рекурентні співвідношення (2.65) і (2.66). Відмітимо, що не одне з наведених співвідношень для

матриці $\sum[n+1|n]$ не потребує знання $Z[n]$; відповідно, матриці $\sum[n+1|n]$ можуть бути знайдені до проведення спостереження. Оцінка оптимального прогнозу, повинна обчислюватися в процесі спостереження, оскільки вона явним чином залежить від результатів спостереження. Виведення рекурентних співвідношень для $\hat{X}[n+1|n]$ і $\sum[n+1|n]$ закінчимо завданням початкових умов для цих рівнянь:

$$\hat{X}[n_0+1|n_0] = \varphi[n_0+1, n_0] \bar{X}[n_0], \quad (2.67)$$

$$\sum[n_0+1|n_0] = \varphi[n_0+1, n_0] P[n_0] \varphi^T[n_0+1, n_0] + \Gamma[n_0] Q[n_0] \Gamma^T[n_0] \quad (2.68)$$

Часто, але не завжди, замість прогнозування на один крок $\hat{X}[n+1|n]$ вважають за краще отримати оцінку $\hat{X}[n|n] = \hat{X}[n] = \hat{E}\{X[n]|Z[n]\}$. Раніше вже відмічалось, що ці величини зв'язані співвідношенням:

$$\hat{X}[n+1|n] = \varphi[n+1, n] \hat{X}[n]. \quad (2.69)$$

Підставляючи праву частину рівності (2.69) в (2.61) та множачи обидві частини отриманої рівності на $\varphi[n+1, n]^{-1} = \varphi[n, n+1]$, отримаємо

$$\hat{X}[n] = \varphi[n, n-1] \hat{X}[n-1] - K[n] [Z[n] - H[n] \varphi[n, n-1] \hat{X}[n-1]] \quad (2.70)$$

з початковою умовою $\hat{X}[n_0] = \bar{X}[n_0]$, де

$$K[n] = \varphi[n, n+1] \mathbf{K}[n]. \quad (2.71)$$

Структура відповідного оптимального фільтра зображена на рис. 2.7. Використовуючи такий самий підхід, що був використаний при отриманні $\hat{X}[n+1|n]$, не важко показати, що $\hat{X}[n]$ є лінійною

функцією від пари $[Z[n], X_0]$, що забезпечує мінімальне значення матриці $E\left\{\left[X[n] - \hat{X}[n]\right]\left[X[n] - \hat{X}[n]\right]^T\right\}$.

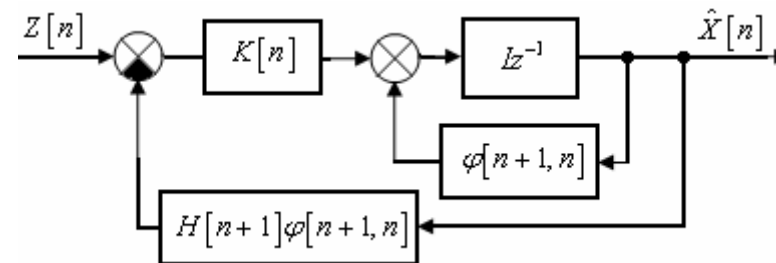


Рис. 2.7. Оптимальний фільтр за дискретного часу

Більш конкретизуємо задачу синтезу дискретного фільтра Калмана для оцінювання стану в системах керування. Задана дискретна модель об'єкта керування

$$X[n+1] = AX[n] + BU[n] + GW[n] - \text{рівняння стану},$$

$$Y_v[n] = CX[n] + DU[n] + HW[n] + V[n] - \text{рівняння вимірювання}, \quad (2.72)$$

де U – відомі входи та збурення за входами W і за вимірюванням V , що являються “білим” шумом і задовольняють

$$E(W[n]W^T[n]) = Q, \quad E(V[n]V^T[n]) = R, \quad E(W[n]V^T[n]) = N. \quad (2.73)$$

Розраховується оцінка вектора стану $\hat{X}[n]$, що мінімізує коваріацію похибки

$$P = \lim_{n \rightarrow \infty} E(e[n|n-1], e^T[n|n-1]), \quad e[n|n-1] = \hat{X}[n] - \hat{X}[n|n-1],$$

$$Z = \lim_{n \rightarrow \infty} E(e[n|n], e^T[n|n]), \quad e[n|n] = \hat{X}[n] - \hat{X}[n|n]. \quad (2.74)$$

Фільтр Калмана має вигляд:

$$\hat{X}[n+1|n] = A\hat{X}[n|n-1] + BU[n] + L(Y_V[n] - C\hat{X}[n|n-1] - DU[n]),$$

$$\begin{bmatrix} \hat{Y}[n|n] \\ \hat{X}[n|n] \end{bmatrix} = \begin{bmatrix} C(I-MC) \\ I-MC \end{bmatrix} \hat{X}[n|n-1] + \begin{bmatrix} (I-MC)D & CM \\ -MD & M \end{bmatrix} \begin{bmatrix} U[n] \\ Y_V[n] \end{bmatrix},$$

(2.75)

де матриця коефіцієнтів зворотних зв'язків L , і нова матриця зворотних зв'язків M визначається на основі матричного алгебраїчного рівняння Ріккати. Цей оцінювач розраховує оцінки стану $\hat{X}[n|n]$ та вектора виходу $\hat{Y}[n|n]$, використовуючи вимірювання виходу $Y_V[n]$, структурна схема якого представлена на рис.2.2. Нова матриця коефіцієнтів підсилення M використовується для нової модифікації $\hat{X}[n|n-1]$ з використанням нового вимірювання $Y_V[n]$.

$$\hat{X}[n|n] = \hat{X}[n|n-1] + M \left(\underbrace{Y_V[n] - C\hat{X}[n|n-1] - DU[n]}_{\text{нова модифікація}} \right).$$

(2.76)

Вимоги, що до існування фільтра Калмана:

- пара матриць $[C, A]$ має бути повністю спостережувана;
- мають виконуватися нерівності $\bar{R} > 0, \bar{Q} - \bar{N}\bar{R}^{-1}\bar{N}^T \geq 0$;
- пара матриць $[A - \bar{N}\bar{R}^{-1}C, \bar{Q} - \bar{N}\bar{R}^{-1}\bar{N}^T]$ не може мати спостережувані моди з власними значеннями за одиничним колом.

Тут

$$\bar{Q} = GQG^T; \bar{R} = R + HN + N^T H^T + HQH^T; \bar{N} = G(QH^T + N).$$

Приклад

Розглянемо прилад для системи представленої на рис. 2.3 з одним входом і одним виходом третього порядку, що описується рівняннями у просторі стану:

$$X[n+1] = AX[n] + BU[n] + GW[n],$$

$$Y_V[n] = CX[n] + DU[n] + HW[n] + V[n],$$

та рівняння фільтру Калмана:

$$\hat{X}[n+1] = A\hat{X}[n] + BU[n] + L(Y_V - C\hat{X}[n] - DU[n]),$$

$$\begin{bmatrix} \hat{Y}[n] \\ \hat{X}[n] \end{bmatrix} = \begin{bmatrix} C \\ I \end{bmatrix} \hat{X}[n] + \begin{bmatrix} D \\ 0 \end{bmatrix} U[n],$$

де $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0,01 & -0,02 & -0,05 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}; G = \begin{bmatrix} 0 \\ 0 \\ 0,05 \end{bmatrix};$

$C = [1 \ 0 \ 0]; D = 0; H = 0,05; T = 0,05,$

вхідний сигнал: $u(nT) = \sin(2\pi nT), \quad n \in [0; 2],$

$X(0) = \hat{X}(0) = [-0,5 \ 0 \ 0,5]^T,$

та коваріаційні матриці:

$E(WW^T) = Q = 1, \quad E(VV^T) = R = \frac{1}{2}, \quad E(W) = E(V) = 0,$

$E(WV^T) = N = 0.$

Наведемо приклад програми для розв'язку цієї задачі з назвою "Discrete_Kalman_Filter".

```
function Discrete_kalman_Filter
A = [0 1 0; 0 0 1; -0.01 -0.02 -0.05]; % Введення матриці A
B = [0 0 1]'; % Введення матриці B
G = [0 0 0.05]'; % Введення матриці G
C = [1 0 0]; % Введення матриці C
D = 0; % Введення матриці D
```

```

H = 0.05; % Введення матриці H
disp('матриця Q:')
Q = 1 % Введення матриці Q
disp('матриця R:')
R = 1/2 % Введення матриці R
disp('матриця N:')
N = [] % Введення матриці N
disp('Вектор початкових умов x0:')
x0 = [-0.5 0 0.5]' % Завдання вектора початкових
умов
disp('Вектор початкових умов x_e0:')
x_e0 = [-0.5 0 0.5]' % Завдання вектора
початкових умов
disp('Час спостереження nTf:')
nTf = 2 % Завдання часу спостереження
disp('Період квантування T:')
T = 0.05 % Завдання періода квантування
% Формуємо дискретну модель ТОК (встановлюємо -1
щоб відмітити, що модель
% дискретна)
disp('Модель системи у просторі стану:')
Plant = ss(A,[B G],C,[D H],T,'inputname',{ 'u'
'w'},'outputname','y')
% Проектування дискретного фільтра Калмана
[kest,L,P,M,Z] = kalman(Plant,Q,R,N);
disp('Коефіцієнт підсилення фільтра:'); L
disp('Розв'язок рівнянь Рікатті:'); P
disp('Іноваційний коефіцієнт підсилення:'); M
disp('Апостеріорна оцінка:'); Z
disp('Дискретна модель фільтра Калмана:'); kest
% Формування моделі процесу
P = ss(A,[B G
D*B],[C;C],[zeros(size(C));C],T,'inputname',...
{'u' 'w' 'v'},'outputname',{'y' 'yv'});
sys = parallel(P,kest,1,1,[],[]); % Паралельне
з'єднання ТОК і фільтра
SimModel = feedback(sys,1,4,2,1); % Приєднуємо
вихід ТОК (#4) до входу фільтра (#2)
SimModel = SimModel([1 3],[1 2 3]); % Видалення
yv з I/O списку
% Моделювання поведінки фільтра

```

```

t = [0:T:nTf]'; % Час спостереження
u = cos(pi*t); % Формування вхідного сигналу
n = length(t); % Визначення довжини вектора
w = sqrt(Q)*randn(n,1); % Шум керування
v = sqrt(R)*randn(n,1); % Шум вимірювання
[out,t,X] = lsim(SimModel,[w,v,u],t,[x0;x_e0]); %
Розрахунок виходів системи
% Моделювання поведінки фільтра
t = [0:T:nTf]'; % Час спостереження
u = sin(2*pi*t); % Формування вхідного сигналу
n = length(t); % Визначення довжини вектора
w = sqrt(Q)*randn(n,1); % Шум керування
v = sqrt(R)*randn(n,1); % Шум вимірювання
[out,x] = lsim(SimModel,[w,v,u]); % Розрахунок
виходів системи
y = out(:,1); % Вихід системи без фільтрації
y_e = out(:,2); % Вихід системи після фільтрації
yv = y + v; % Зважений вихід системи
% Формування векторів x, e_e
for i = 1:length(2*A),
    x_e(:,i) = x(:,i+length(2*A));
    x(:,i) = x(:,i);
end
% Побудова динаміки змінних стану
figure(1)
stem(t,x), grid on
hold on, stem(t,x_e,'--'), hold off
title('динаміка руху змінних стану');
xlabel('Час, nT');
ylabel('Змінні стану, X(nT) та x_e(nT)');
legend('x_1','x_2','x_3','x_e_1','x_e_2','x_e_3')
% Побудова графіку вихода системи[y',y_e',yv']
figure(2)
stem(t,[y,y_e,yv]), grid on
xlabel('Час, nT'), ylabel('вихід системи: y, y_e,
y_v'),
title('динаміка зміни виходу системи'),
legend('y - без фільтрації','y_e - після
фільтрації','y_v - зважений',1),
% Побудова графіку похибки
figure(3)

```

```

stem(t,[y-yv,y-y_e]), grid on
title('Динаміка зміни похибки'); xlabel('Час,
nT');
ylabel('Похибки: Y(nT)-Y_V(nT), Y-Y_e(nT)');
legend('Y_V - вимірювання', 'Y_e - фільтрації')
% Побудова вхідного сигналу
figure(4)
stem(t,u), grid
title('Динаміка зміни вхідного сигналу');
xlabel('Час, nT');
ylabel('Вхідний сигнал, U(nT)'); legend('u -
вхідний сигнал');
% Обчислення коваріацій похибок
MeasErr = y-yv;
MeasErrCov =
sum(MeasErr.*MeasErr)/length(MeasErr);
EstErr = y-y_e;
EstErrCov = sum(EstErr.*EstErr)/length(EstErr);
disp('Коваріація похибки перед фільтрацією
(похибка вимірювання):')
MeasErrCov
disp('Коваріація похибки після фільтрації
(похибка оцінювання):')
EstErrCov

```

Результат виконання програми

Матриця Q:

Q =

1

Матриця R:

R =

0.5000

Матриця N:

N =
[]

Вектор початкових умов x0:

x0 =
-0.5000
0
0.5000

Вектор початкових умов x_e0:

x_e0 =
-0.5000
0
0.5000

Час спостереження nTf:

nTf =
2

Період квантування T:

T =
0.0500

Модель системи у просторі стану:

a =

	x1	x2	x3
x1	0	1	0
x2	0	0	1
x3	-0.01	-0.02	-0.05

b =

	u	w

```

x1    0    0
x2    0    0
x3    1  0.05

```

```

c =
      x1  x2  x3
y    1   0   0

```

```

d =
      u    w
y    0  0.05

```

Sampling time: 0.05
Discrete-time model.
Коефіцієнт підсилення фільтра:

```

L =
-0.0002
-0.0001
 0.0049

```

Розв'язок рівнянь Рікатті:

```

P =
 0.0025  -0.0001  -0.0000
-0.0001   0.0025  -0.0001
-0.0000  -0.0001   0.0025

```

Іноваційний коефіцієнт підсилення:

```

M =
 0.0049
-0.0002

```

-0.0001

Апостеріорна оцінка:

```

Z =
 0.0025  -0.0001  -0.0000
-0.0001   0.0025  -0.0001
-0.0000  -0.0001   0.0025

```

Дискретна модель фільтра Калмана:

```

a =
      x1_e    x2_e    x3_e
x1_e  0.000241     1     0
x2_e  8.317e-005     0     1
x3_e -0.01491    -0.02  -0.05

```

```

b =
      u    y
x1_e  0  -0.000241
x2_e  0  -8.317e-005
x3_e  1   0.00491

```

```

c =
      x1_e    x2_e    x3_e
y_e  0.9951     0     0
x1_e  0.9951     0     0
x2_e  0.000241     1     0
x3_e  8.317e-005     0     1

```

```

d =
      u    y
y_e  0  0.004941
x1_e  0  0.004941
x2_e  0  -0.000241
x3_e  0  -8.317e-005

```

Input groups:

Name	Channels
KnownInput	1
Measurement	2

Output groups:

Name	Channels
OutputEstimate	1
StateEstimate	2,3,4

Sampling time: 0.05

Discrete-time model.

Коваріація похибки перед фільтрацією (похибка вимірювання):

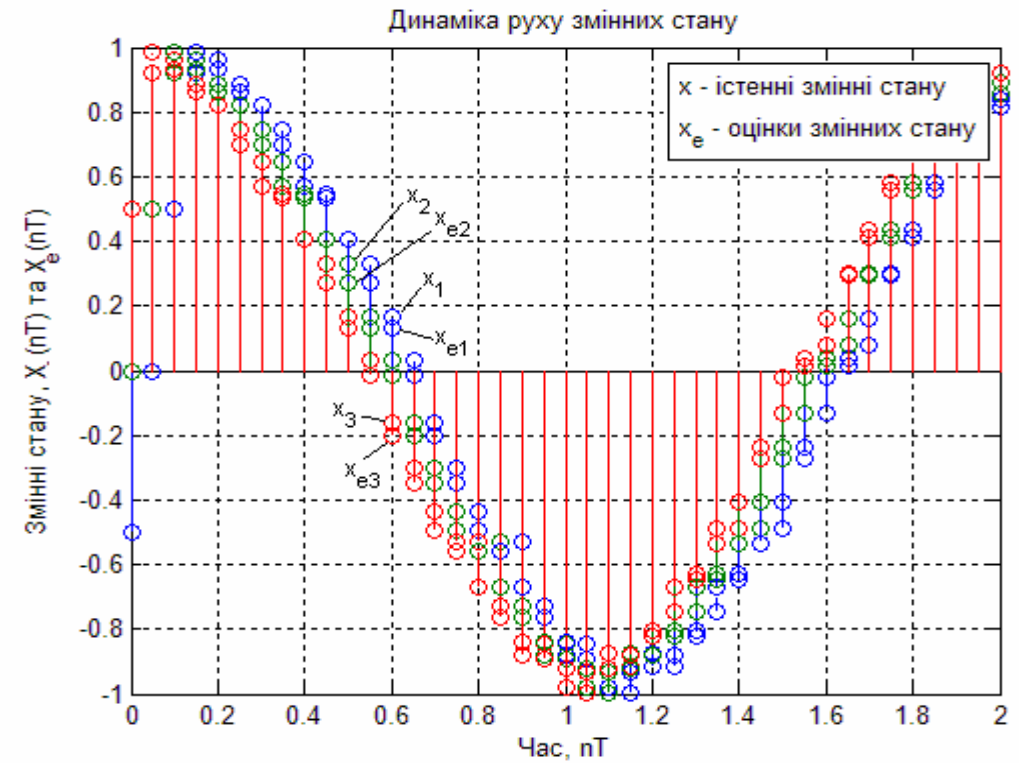
MeasErrCov =

0.3590

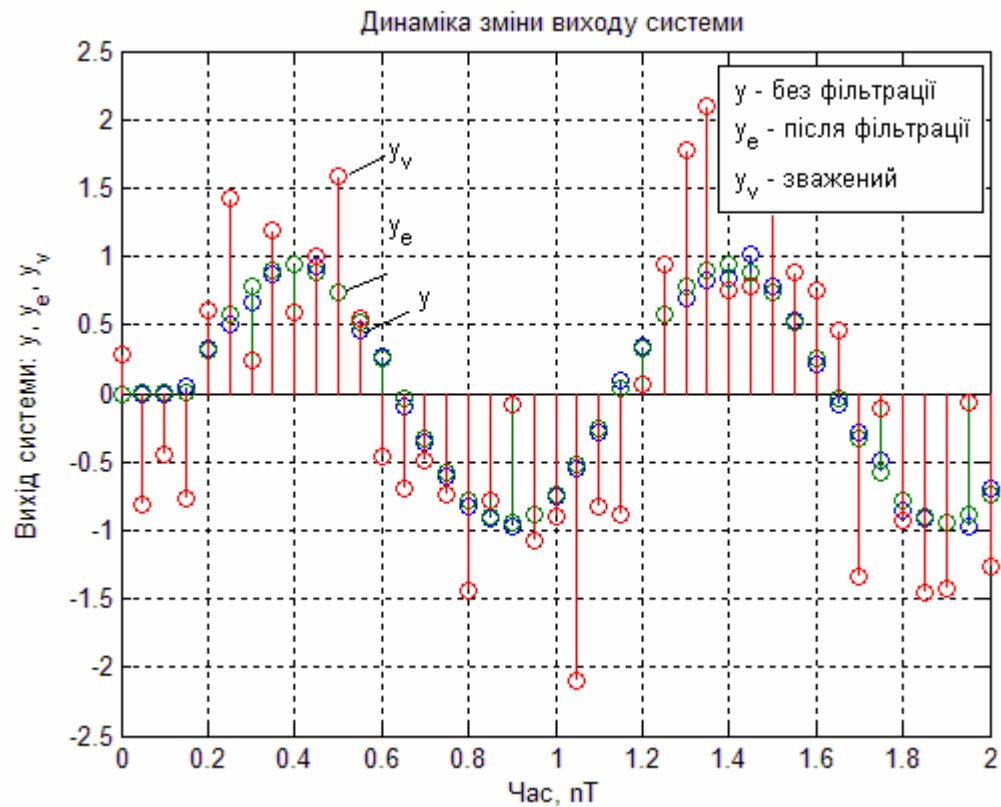
Коваріація похибки після фільтрації (похибка оцінювання):

EstErrCov =

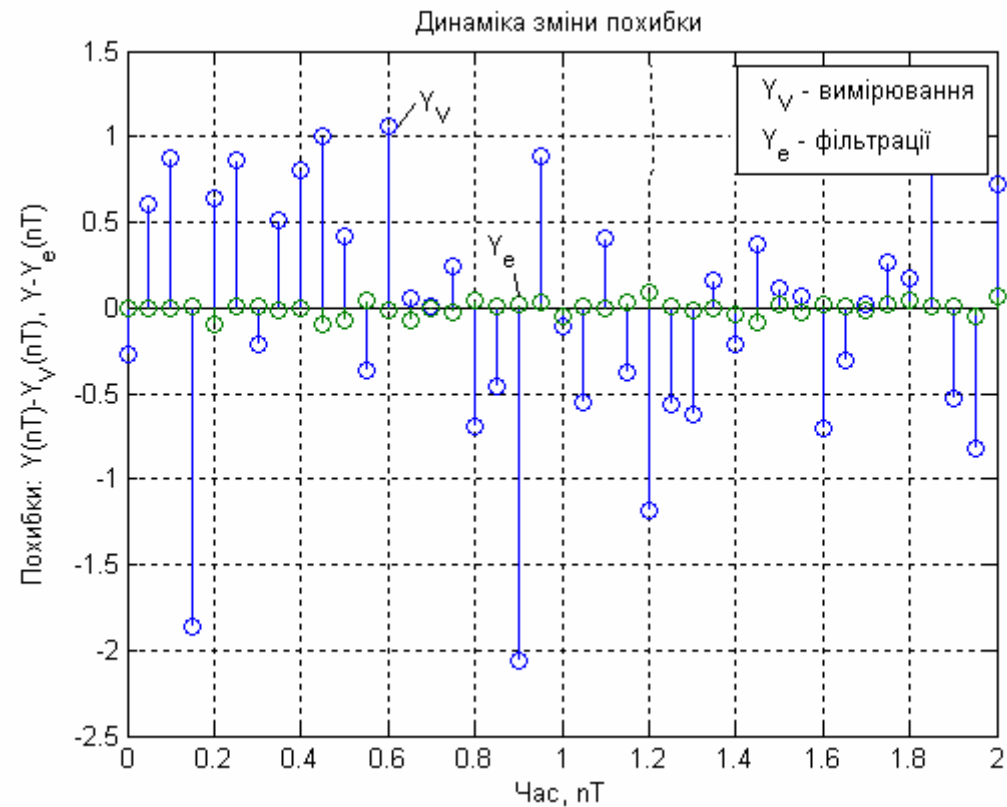
0.0022



а)

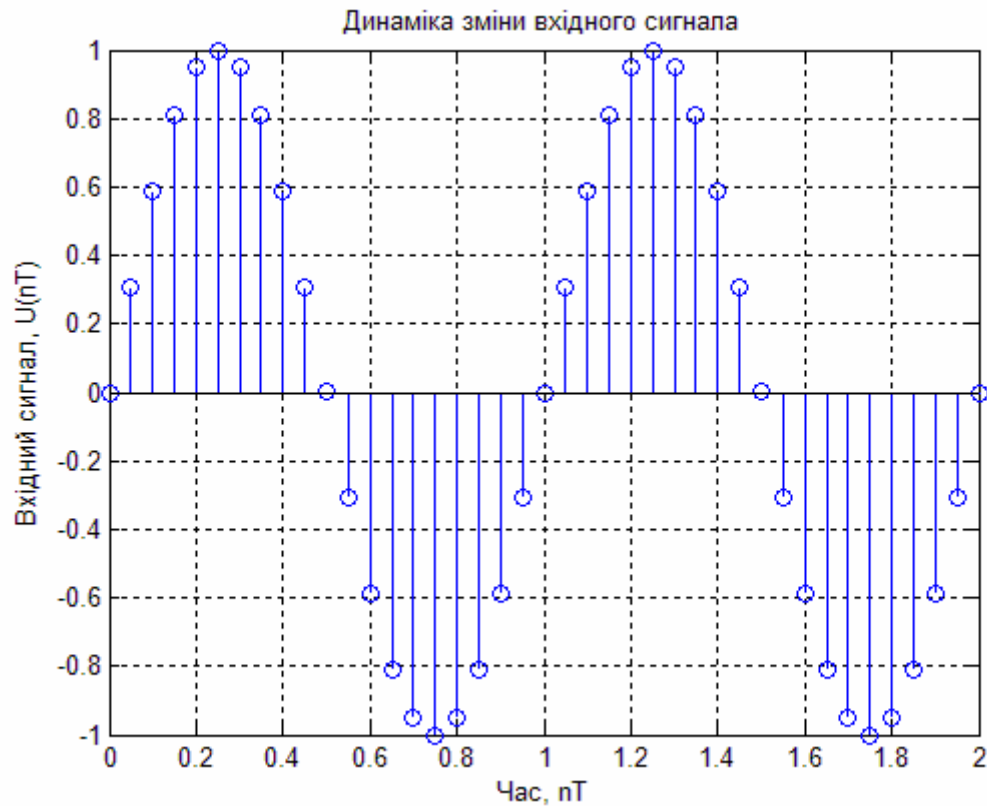


б)



а)

Рис. 2.8. Динаміка руху змінних стану а) і виходу системи б)



б)

Рис. 2.9. Динаміка зміни похибки а) і вхідного сигналу б)

Досліджували якість дискретного фільтра Калмана для наступних значень параметрів: $X(0) = \hat{X}(0) = [-0,5 \ 0 \ 0,5]^T$; $E(WW^T) = Q = 1$; $E(VV^T) = R = \frac{1}{2}$; $E(W) = E(V) = 0$; $E(WV^T) = N = 0$; $T = 0,05$. Коефіцієнт підсилення фільтра L , отриманий за допомогою розв'язку дискретного алгебраїчного рівняння Ріккати (дисперсійного рівняння). Виходи фільтра Калмана,

показані на рис. 2.8 а), де X - без фільтрації; X_e - після фільтрації (штрихова крива), на рис. 2.8 б) – показані сигнали: без фільтрації; після фільтрації та зважений, на рис. 2.9 а) – показано дві похибки: вимірювання; фільтрації, на рис. 2.9 б) – показан сигнал, що подається на вхід системи. Якість отриманого фільтра суттєво залежить від дисперсійних матриць Q, R, N . Відмітимо, що оцінки збігаються до істинних значень достатньо швидко, не дивлячись на відносно високий рівень завад і неточність завдання початкових оцінок.

2.3. Дискретний фільтр Калмана для неперервної системи

Для проектування дискретного фільтра Калмана для неперервної системи, справедливі всі попередні твердження, а об'єкт керування описується рівняннями вигляду:

$$\frac{dX}{dt} = AX + BU + GW - \text{рівняння стану}, \quad (2.77)$$

$$Y = CX + DU + V - \text{рівняння вимірювання},$$

де W – шум процесу; V – шум вимірювання.

Задовольняють наступні вимоги:

$$E(W) = E(V) = 0; \quad E(WW^T) = Q_n; \quad E(VV^T) = R_n; \quad E(WV^T) = 0. \quad (2.78)$$

Неперервна система дискретизується з періодом квантування T , і неперервні матриці коваріації Q_n та R_n замінюються їх дискретними еквівалентами

$$Q_d = \int_0^T e^{A\tau} G Q_n G^T e^{A^T \tau} d\tau, \quad (2.79)$$

$$R_d = R_n / T. \quad (2.80)$$

Інтеграли обчислюються з використанням матричних показникових формул Ван Лона.

Вимоги, що до існування дискретного фільтру Калмана для неперервної системи такі самі, що і для дискретного фільтру Калмана.

Приклад

Розглянемо прилад для системи представленої на рис. 2.3 з одним входом і одним виходом третього порядку, що описується рівняннями у просторі стану:

$$\frac{dX}{dt} = AX + BU + GW,$$

$$Y_v = CX + DU + V,$$

та рівняння фільтру Калмана:

$$\hat{X}[n+1] = A\hat{X}[n] + BU[n] + L(Y_v - C\hat{X}[n] - DU[n]),$$

$$\begin{bmatrix} \hat{Y}[n] \\ \hat{X}[n] \end{bmatrix} = \begin{bmatrix} C \\ I \end{bmatrix} \hat{X}[n] + \begin{bmatrix} D \\ 0 \end{bmatrix} U[n],$$

$$\text{де } A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -5 & -3 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 0 \\ 5,2 \end{bmatrix}; G = \begin{bmatrix} 0 \\ 0 \\ 0,65 \end{bmatrix}; C = [1 \ 0 \ 0];$$

$$D = 0; T = 0,08,$$

$$\text{вхідний сигнал: } u(t) = \sin\left(\frac{\pi}{2}t\right), \quad t \in [0;4],$$

$$X(0) = \hat{X}(0) = [-0,5 \ 0 \ 0,5]^T,$$

та коваріаційні матриці:

$$E(WW^T) = Q = 2, \quad E(VV^T) = R = \frac{1}{3}, \quad E(W) = E(V) = 0,$$

$$E(WV^T) = N = 0.$$

Наведемо приклад програми для розв'язку цієї задачі з назвою "Discrete_Kalman_Filter_For_Continuous_Plant".

```
function
Discrete_Kalman_Filter_For_Continuous_Plant
A = [0 1 0; 0 0 1; -2 -5 -3]; % Введення матриці
A
B = [0 0 5.2]'; % Введення матриці B
C = [1 0 0]; % Введення матриці C
G = [0 0 0.65]'; % Введення матриці G
D = 0; % Введення матриці D
disp('Матриця Q:')
Q = 2 % Введення матриці Q
disp('Матриця R:')
R = 1/3 % Введення матриці R
disp('Вектор початкових умов x0:')
x0 = [-0.5 0 0.5]' % Завдання вектора початкових
умов
disp('Вектор початкових умов x_e0:')
x_e0 = [-0.5 0 0.5]' % Завдання вектора
початкових умов
disp('Час спостереження ntf:')
tf = 4 % Завдання часу спостереження
disp('Період квантування T:')
T = 0.08 % Завдання періода квантування
% Формуємо модель ТОК
disp('Модель системи у просторі стану:')
Plant = ss(A,[B G],C,D,'inputname',{'u'
'w'},'outputname','y')
% Проектування дискретного фільтру Калмана для
неперервної системи
[kest,L,P,M,Z] = kalmd(Plant,Q,R,T);
disp('Коефіцієнт підсилення фільтра:'); L
disp('Розв'язок рівнянь Рікатті:'); P
disp('Іноваційний коефіцієнт підсилення:'); M
disp('Апостеріорна оцінка:'); Z
disp('Дискретна модель фільтру Калмана для
неперервної системи:'); kest
% Формування моделі процесу
P = c2d(ss(A,[B G
D*B],[C;C],[zeros(size(C));C],'inputname',...
```

```

{'u' 'w' 'v'}, 'outputname', {'y' 'yv'}), T);
sys = parallel(P, kest, 1, 1, [], []); % Паралельне
з'єднання ТОК і фільтра
SimModel = feedback(sys, 1, 4, 2, 1); % Приєднуємо
вихід ТОК (#4) до входу фільтра (#2)
SimModel = SimModel([1 3], [1 2 3]); % Видалення
ув з I/O списку
% Моделювання поведінки фільтра
t = [0:T:tf]'; % Час спостереження
u = cos(2*pi*t); % Формування вхідного сигналу
n = length(t); % Визначення довжини вектора
w = sqrt(Q)*randn(n,1); % Шум керування
v = sqrt(R)*randn(n,1); % Шум вимірювання
[out, t, X] = lsim(SimModel, [w, v, u], t, [x0; x_e0]); %
Розрахунок виходів системи
% Моделювання поведінки фільтра
t = [0:T:tf]'; % Час спостереження
u = sin(pi/2*t); % Формування вхідного сигналу
n = length(t); % Визначення довжини вектора
w = sqrt(Q)*randn(n,1); % Шум керування
v = sqrt(R)*randn(n,1); % Шум вимірювання
[out, x] = lsim(SimModel, [w, v, u]); % Розрахунок
виходів системи
y = out(:,1); % Вихід системи без фільтрації
y_e = out(:,2); % Вихід системи після фільтрації
yv = y + v; % Зважений вихід системи
% Формування векторів x, e_e
for i = 1:length(2*A),
    x_e(:,i) = X(:,i+length(2*A));
    x(:,i) = X(:,i);
end
% Побудова динаміки змінних стану
figure(1)
plot(t, x), grid on
hold on, stem(t, x_e), hold off
title('Динаміка руху змінних стану');
xlabel('Час, t');
ylabel('Змінні стану, X(t) та X_e(nT)');
legend('x_1', 'x_2', 'x_3', 'x_e_1', 'x_e_2', 'x_e_3')
% Побудова графіку вихода системи
figure(2)

```

```

plot(t, [y, yv]), grid on
hold on, stem(t, y_e), hold off
xlabel('Час, t, nT'), ylabel('Вихід системи: y,
y_e, y_v'),
title('Динаміка зміни виходу системи'),
legend('y - без фільтрації', 'y_v - зважений', 'y_e
- після фільтрації', 1),
% Побудова графіку похибок
figure(3)
stem(t, [y-yv, y-y_e]), grid on
title('Динаміка зміни похибки'); xlabel('Час,
nT');
ylabel('Похибки: Y(nT)-Y_V(nT), Y(nT)-Y_e(nT)');
legend('Y_V - вимірювання', 'Y_e - фільтрації')
% Побудова вхідного сигналу
figure(4)
plot(t, u), grid
title('Динаміка зміни вхідного сигналу');
xlabel('Час, t');
ylabel('Вхідний сигнал, U(t)'); legend('u -
вхідний сигнал');
% Обчислення коваріацій похибок
MeasErr = y-yv;
MeasErrCov =
sum(MeasErr.*MeasErr)/length(MeasErr);
EstErr = y-y_e;
EstErrCov = sum(EstErr.*EstErr)/length(EstErr);
disp('Коваріація похибки перед фільтрацією
(похибка вимірювання):')
MeasErrCov
disp('Коваріація похибки після фільтрації
(похибка оцінювання):')
EstErrCov

```

Результат виконання програми

Матриця Q:

Q =

2

Матриця R:

R =

0.3333

Вектор початкових умов x0:

x0 =

-0.5000
0
0.5000

Вектор початкових умов x_e0:

x_e0 =

-0.5000
0
0.5000

Час спостереження nTf:

tf =

4

Період квантування T:

T =

0.0800

Модель системи у просторі стану:

a =

	x1	x2	x3
x1	0	1	0
x2	0	0	1
x3	-2	-5	-3

b =

	u	w
x1	0	0
x2	0	0
x3	5.2	0.65

c =

	x1	x2	x3
y	1	0	0

d =

	u	w
y	0	0

Continuous-time model.

Коефіцієнт підсилення фільтра:

L =

0.0100
0.0000
-0.0073

Розв'язок рівнянь Рікатті:

P =

0.0419	0.0026	-0.0310
0.0026	0.0312	0.0000
-0.0310	0.0000	0.1610

Іноваційний коефіцієнт підсилення:

M =

0.0100
 0.0006
 -0.0074

Апостеріорна оцінка:

Z =

0.0415	0.0026	-0.0307
0.0026	0.0312	0.0000
-0.0307	0.0000	0.1608

Дискретна модель фільтра Калмана для неперервної системи:

a =

	x1_e	x2_e	x3_e
x1_e	0.9899	0.0796	0.002951
x2_e	-0.005935	0.9851	0.07074
x3_e	-0.1342	-0.3596	0.7729

b =

	u	y
x1_e	0.0004177	0.009984
x2_e	0.01534	3.3e-005
x3_e	0.3679	-0.007327

c =

	x1_e	x2_e	x3_e
y_e	0.99	0	0
x1_e	0.99	0	0
x2_e	-0.0006223	1	0
x3_e	0.007368	0	1

d =

	u	y
y_e	0	0.009957
x1_e	0	0.009957
x2_e	0	0.0006223
x3_e	0	-0.007368

Input groups:

Name	Channels
KnownInput	1
Measurement	2

Output groups:

Name	Channels
OutputEstimate	1
StateEstimate	2,3,4

Sampling time: 0.08

Discrete-time model.

Коваріація похибки перед фільтрацією (похибка вимірювання):

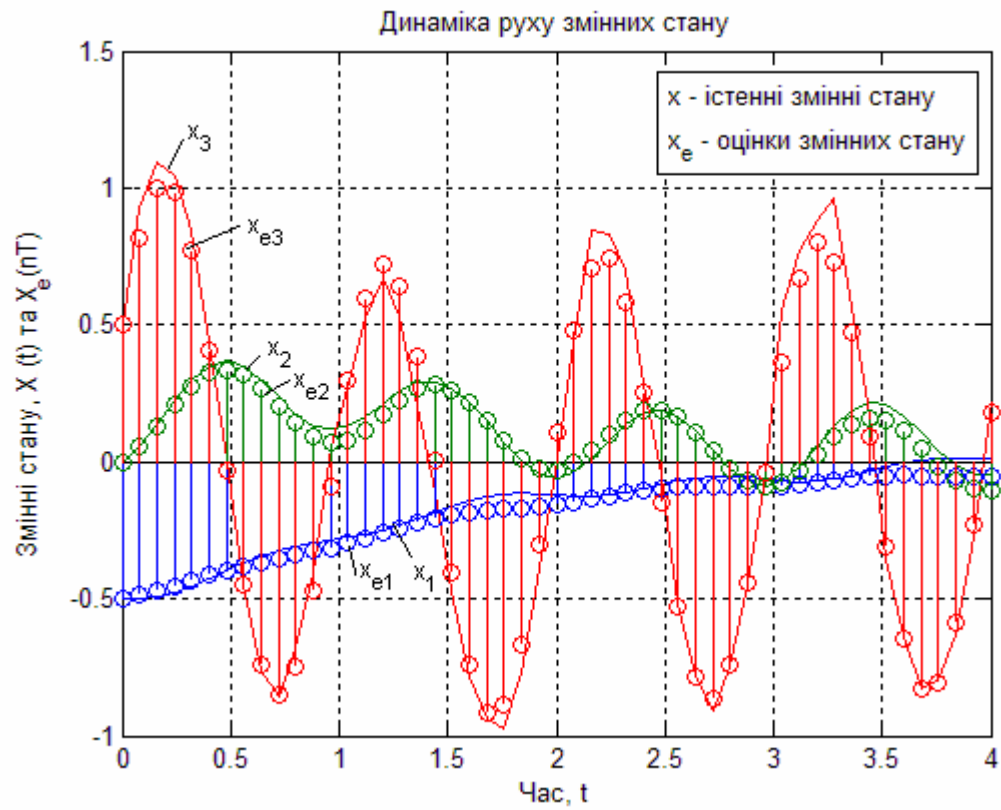
MeasErrCov =

0.3331

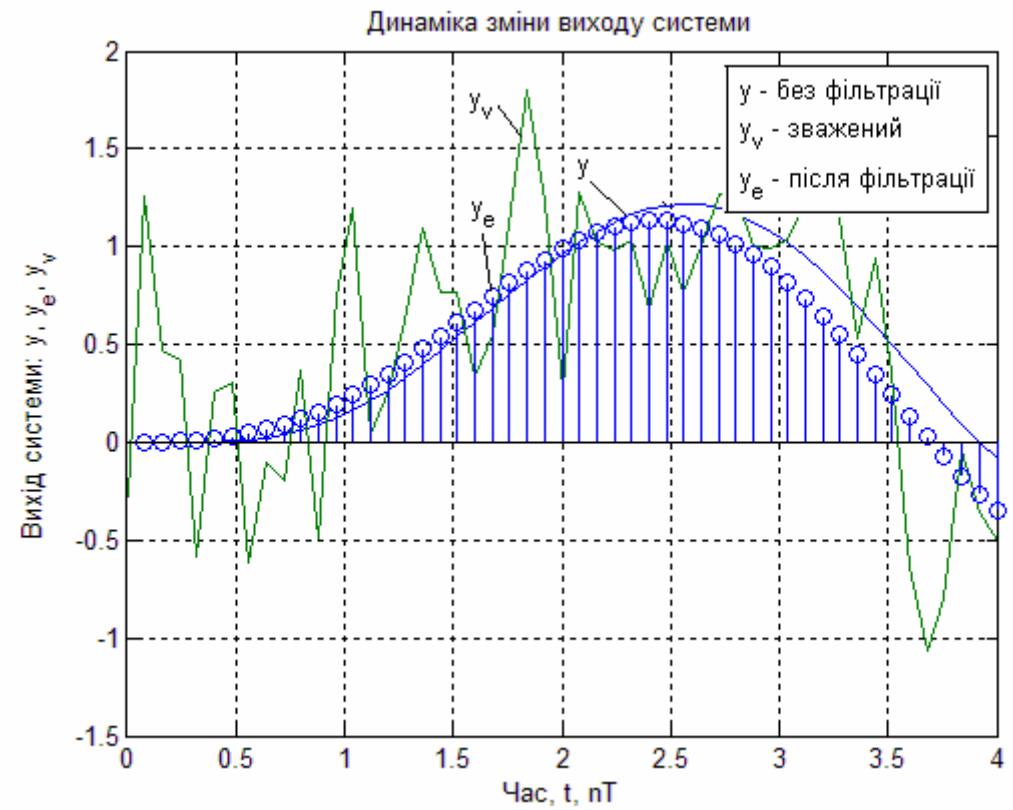
Коваріація похибки після фільтрації (похибка оцінювання):

EstErrCov =

0.0084

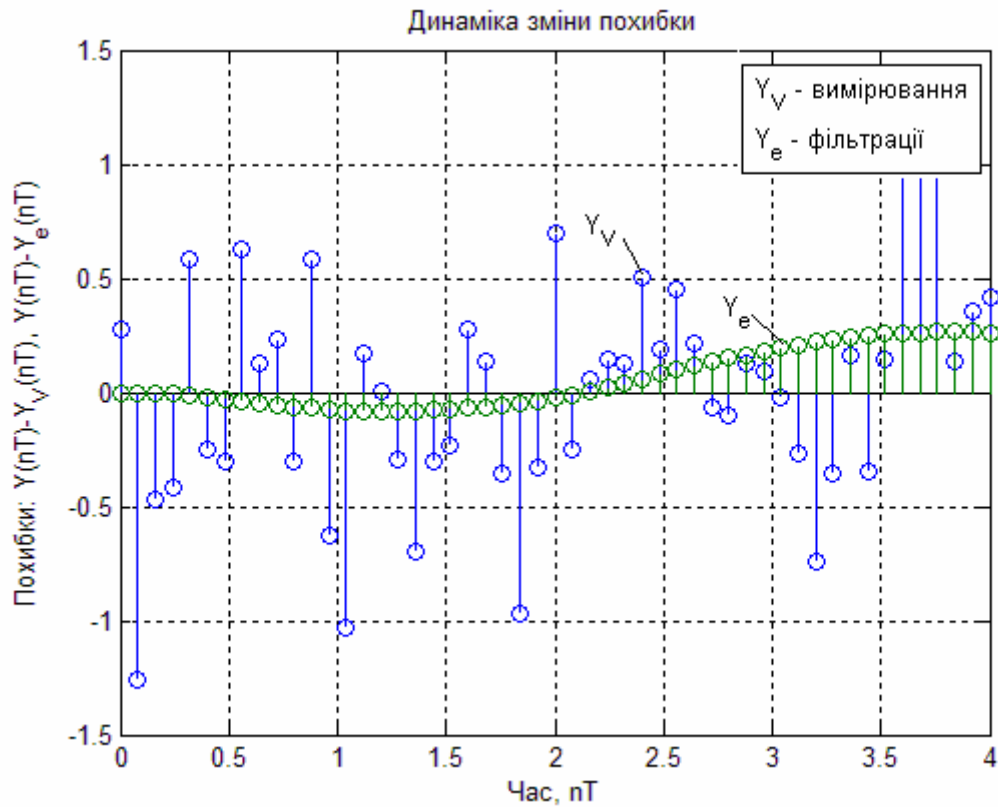


а)

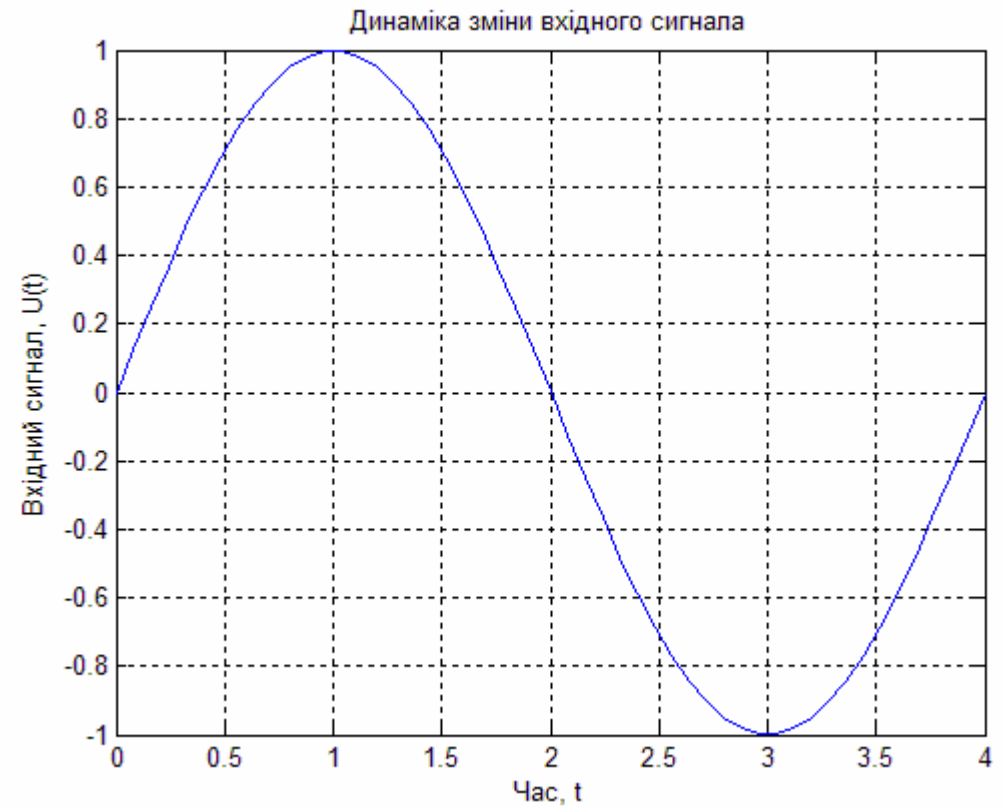


б)

Рис. 2.10. Динаміка руху змінних стану а) і виходу системи б)



а)



б)

Рис. 2.11. Динаміка зміни похибки а) і вхідного сигналу б)

Дослідили якість дискретного фільтра Калмана для неперервної системи, при наступних значеннях параметрів:
 $X(0) = \hat{X}(0) = [-0,5 \ 0 \ 0,5]^T$; $E(WW^T) = Q = 2$;
 $E(VV^T) = R = \frac{1}{3}$; $E(W) = E(V) = 0$; $E(WV^T) = N = 0$;
 $T = 0,08$. Коефіцієнт підсилення фільтра L отриманий за допомогою розв'язку алгебраїчного рівняння Ріккати (дисперсійного

рівняння). Виходи фільтру Калмана, показані на рис. 2.10 а), де X - без фільтрації; X_e - після фільтрації (штрихова крива), на рис. 2.10 б) – показані сигнали: без фільтрації; після фільтрації та зважений, на рис. 2.11 а) – показано дві похибки: вимірювання; фільтрації, на рис. 2.11 б) – показан сигнал, що подається на вхід системи.

3. Проектування лінійно-квадратичного гауссіана за заданими коефіцієнтами фільтру Калмана

Як правило, фізичні системи керування, які повинні реалізуватися, важко розглядати як детерміновані, оскільки спостереженням доступні лише спотворені шумом деякі змінні їх стану. Таким чином, більш реалістичними моделями таких систем в задачах керування є моделі, що містять випадкові процеси. При таких моделях потрібно приймати рішення чи визначати керування на основі результатів вимірів, отриманих у минулому. В цьому випадку обидві операції – оцінювання станів і керування – можуть бути включені в єдиний процес вибору рішення. Частинним випадком цієї загальної задачі є лінійна квадратична гауссовська (ЛКГ) задача.

Оптимальне стохастичне керування реалізується у вигляді послідовно з'єднаних оптимального фільтра, що оцінює стан об'єкту керування, і оптимального детермінованого регулятора, що використовує отримані оцінки замість істинних станів. Цей результат, відомий як принцип розділення чи ЛКГ – задача, може застосовуватись до достатньо широкого класу лінійних систем.

Слід відмітити, що основний сенс (властивість) розділення полягає в тому, що при його виконанні задача з випадковим шумом в сигналі вимірювань, тобто є задачі з частково спостережуваними змінними стану, зводиться до задач з повністю спостережуваними змінними стану, роль яких в цьому випадку виконують оцінки стану.

У випадку виконання ЛКГ – умов може бути здійснене, як іноді говорять, аналітичне конструювання оптимальних регуляторів. Особливість ЛКГ – задач керування полягає в тому, що вони достатньо добре описують реальні задачі, лише при розробці задач

оптимізації, коли потрібно підтримувати регульовані змінні поблизу їх заданих значень. В таких задачах якість регулювання часто характеризується квадратом відхилення змінних стану від завдання, а у зв'язку з малим значенням цих відхилень динаміка процесу звичайно задовільно описується лінійними моделями.

У більш складних нелінійних системах ЛКГ – умови не виконуються. Оскільки перехід від лінійного гауссовського випадку до загального випадку з нелійними моделями динаміки і негауссівськими корельованими випадковими збуреннями викликає значні труднощі, на практиці замість них розглядаються задачі субоптимального керування.

Для проектування ЛКГ за заданими коефіцієнтами фільтру Калмана для неперервної та дискретної системи, справедливі всі попередні твердження про регулятори та фільтри Калмана.

При неперервному часі ЛКГ розраховує сигнал керування

$$U = -K\hat{X}, \quad (3.1)$$

де \hat{X} оцінка стану за Калманом. Рівняння регулятора мають вигляд:

$$\frac{d\hat{X}}{dt} = [A - LC - (B - LD)K] \hat{X} + LY_v, \quad (3.2)$$

де Y_v є входом фільтру Калмана (рівняння вимірювання).

Структурна схема такої системи наведена на рис. 3.1.

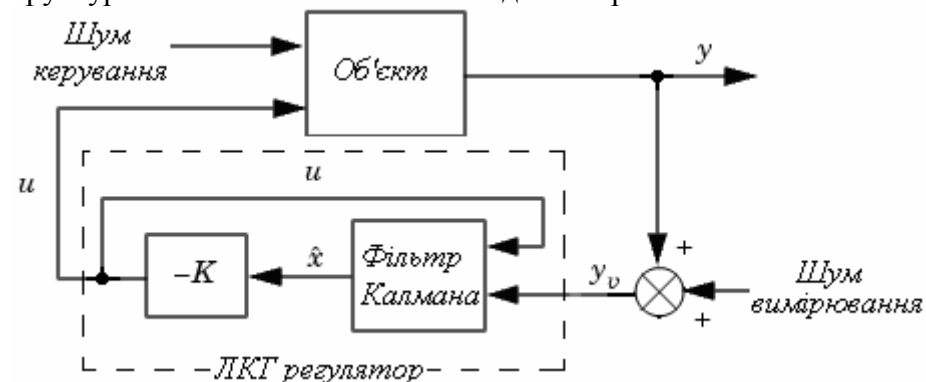


Рис. 3.1. Структурна схема системи

При дискретному часі для проектування ЛКГ використовуються будь-які оцінки вектора стану $\hat{X}[n|n-1]$, $\hat{X}[n]$, що базуються на вимірюваннях $Y_V[n-1]$ та оцінки поточного стану $\hat{X}[n|n]$, яка базується на $Y_V[n]$, та розраховується керування вигляду:

$$U[n] = -K\hat{X}[n|n-1], \quad (3.3)$$

та за оцінкою поточного стану

$$U[n] = -K\hat{X}[n|n], \quad (3.4)$$

є причиною коли $(I - KMD)$ обернене (див. фільтр Калмана). Крім того, практичне виконання поточного регулятора повинно враховувати час обробки інформації для обчислення $U[n]$ одного вимірювання $Y_V[n]$, що стало доступним (це сновить затримку зворотного зв'язку).

Для проектування ЛКГ регулятора використовуються наступні функції:

- Неперервний регулятор для неперервної системи: *lqr* або *lqry* та *kalman*;
- Дискретний регулятор для дискретної системи: *dqr* або *lqry* та *kalman*;
- Дискретний регулятор для неперервної системи: *lqrd* та *kalmd*.

Оцінювач, що має додаткові входи об'єкта U_d та Y_V наведений на рис. 3.2.

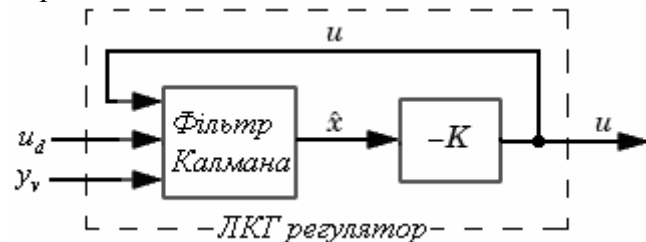


Рис. 3.2. Структурна схема з оцінювачем, що має додаткові входи

Приклад

Розглянемо систему шостого порядку з трьома входами і трьома виходами, що описується рівняннями у просторі стану:

$$\frac{dX}{dt} = A(X - Xd) + B(U - Ud) + GW,$$

$$Y_V = C(X - Xd) + D(X - Xd) + HW + V,$$

та рівняння регулятора

$$\frac{d\hat{X}}{dt} = [A - LC - (B - LD)K](\hat{X} - \hat{X}d) + LY_V,$$

$$\begin{bmatrix} \hat{X} - \hat{X}d \\ \hat{Y} - \hat{Y}d \end{bmatrix} = \begin{bmatrix} C \\ I \end{bmatrix} (\hat{X} - \hat{X}d) + \begin{bmatrix} D \\ 0 \end{bmatrix} (U - Ud),$$

де $A = \begin{bmatrix} -0,51 & 0 & 1,3 & 0 & 0 & 2,1 \\ 1,8 & -0,25 & -1,3 & -0,1 & 0 & 0 \\ 0 & 0 & -0,16 & -0,1 & 0 & -0,2 \\ 0 & -0,35 & 0 & -0,65 & 0 & 0 \\ 0 & 0,21 & -0,61 & 0 & -0,35 & 0 \\ -0,1 & 0,581 & 0,04 & 0,02 & 0,05 & -0,14 \end{bmatrix};$

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0,2 & 0 \\ 1,5 & 0 & 0 \\ 0 & -1,1 & 0 \\ 0 & 0 & 1 \end{bmatrix};$$

$$G = \begin{bmatrix} 0,1 & 0 & 0 \\ 0,35 & 0 & 0 \\ 0 & 0,21 & 0 \\ 0,51 & 0 & 0 \\ 0 & 0,28 & 0 \\ 0 & 0 & 0,1 \end{bmatrix};$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix};$$

$$D = \begin{bmatrix} 0,01 & 0 & 0 \\ 0 & 0 & 0,2 \\ 0 & 0 & 0,85 \end{bmatrix};$$

$$H = \begin{bmatrix} 0,1 & 0 & 0,8 \\ 0,1 & 0,51 & 1,21 \\ 0,2 & 0 & 0,81 \end{bmatrix},$$

і квадратичний критерій якості

$$J = E \left\{ \int_0^{\infty} \left((Y - Yd)^T Q (Y - Yd) + (U - Ud)^T R (U - Ud) + 2(Y - Yd)^T N (U - Ud) \right) dt \right\},$$

де $X(0) = \hat{X}(0) = [-0,6 \ 0 \ -0,2 \ 0,2 \ 0,4 \ 0,6]^T$;

$Xd = \hat{X}d = [-0,2 \ -0,6 \ -0,4 \ 0 \ 0,2 \ 0,4]^T$;

$Yd = \hat{Y}d = [1 \ 0,6 \ 0]^T$; $Ud = [-0,4 \ 0,2 \ 0]^T$.

Та коваріаційні матриці:

$$E(WW^T) = Q = \begin{bmatrix} 1,5 & 0 & 0 \\ 0 & 1,3 & 0 \\ 0 & 0 & 0,085 \end{bmatrix};$$

$$E(VV^T) = R = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1,5 & 0 \\ 0 & 0 & 0,9 \end{bmatrix}; \quad E(W) = E(V) = 0;$$

$$E(WV^T) = N = \begin{bmatrix} 0,01 & 0,08 & 0,02 \\ 0,021 & 0,05 & 0,075 \\ 0,04 & 0,07 & 0,08 \end{bmatrix}.$$

Наведемо приклад програми для розв'язку цієї задачі з назвою "LQG_Reg".

```
function LQG_Reg
A = [-0.51 0 1.3 0 0 -2.1
      1.8 -0.25 -1.3 -0.1 0 0
      0 0 -0.16 -0.1 0 -0.2
      0 -0.35 0 -0.65 0 0
      0 0.21 -0.61 0 -0.35 0
      -0.1 0.581 0.04 0.02 0.05 -0.14]; % Введення
матриці А
B = [1 0 0
      1 0 0
      0 0.2 0
      1.5 0 0
      0 -1.1 0
      0 0 1]; % Введення матриці В
G = [0.1 0 0
      0.35 0 0
      0 0.21 0
      0.51 0 0
      0 0.28 0
      0 0 0.1]; % Введення матриці G
C = [1 0 0 0 0 0
      0 1 1 0 0 0
```

```

D = [0 0 0 1 1 1]; % Введення матриці C
D = [0.1 0 0
     0 0 0.2
     0 0.1 0]; % Введення матриці D
H = [0.1 0 0.8
     0.1 0.51 1.21
     0.2 0 0.81]; % Введення матриці H
disp('Матриця Q:')
Q = [1.5 0 0
     0 1.3 0
     0 0 0.085] % Введення матриці Q
disp('Матриця R:')
R = [2 0 0
     0 1.5 0
     0 0 0.9] % Введення матриці R
disp('Матриця N:')
N = [0.01 0.08 0.02
     0.021 0.05 0.075
     0.04 0.07 0.08] % Введення матриці N
disp('Вектор початкових умов x0:')
x0 = [-0.6 0 -0.2 0.2 0.4 0.6]' % Завдання
вектора початкових умов
disp('Вектор заданих значень відхилень xd:')
xd = [-0.2 -0.6 -0.4 0 0.2 0.4]' % Завдання
вектора заданих значень відхилень
disp('Вектор заданих значень відхилень
керування Ud:')
ud = [-0.4 0.2 0]' % Завдання відхилень значень
керування
disp('Вектор заданих значень відхилень вихідної
величини Yd:')
yd = [1 0.5 0]' % Завдання відхилень значень
вихідної величини
disp('Час спостереження tf:')
tf = 18 % Завдання часу спостереження
disp('Довжина кроку dt:')
dt = 0.001 % Завдання довжини кроку
disp('Кількість кроків n:')
n = tf/dt
% Формуємо дискретну модель ТОК
disp('Модель системи у просторі стану:')

```

```

Plant = ss(A,[B G],C,[D H],'inputname',...
{'u1' 'u2' 'u3' 'w1' 'w2'
'w3'},'outputname',{'y1' 'y2' 'y3'})
% Проектування неперервного фільтра Калманаї
disp('Синтез фільтра Калмана:');
[kest,L,P] = kalman(Plant,Q,R,N);
disp('Коефіцієнт підсилення фільтра:'); L
disp('Розв'язок рівнянь Рікати:'); P
disp('Неперервна модель фільтра Калмана:'); kest
% Проектування регулятора
disp('Синтез регулятора:');
[K P E] = lqrg(ss(A,B,C,D),Q,R,N);
disp('Матриця зворотнього зв'язку K:'); K
disp('Розв'язок рівняння Рікати P:'); P
disp('Власні значення замкненої системи E:'); E
disp('Модель ЛКГ регулятора:'); rlqg =
lqgreg(kest,K)
% Визначення розмірності задачі
SA = size(A); SA = SA(1); SB = size(B); SB =
SB(2); SC = size(C); SC = SC(1);
% Формування векторів x_e, u, y_e
x = zeros(SA,n); x_e = zeros(SA,n); u =
zeros(SB,n-1); y_e = zeros(SC,n-1);
yv = y_e;
% Формування початкового стану
x(:,1) = x0; x_e(:,1) = x0;
% Знаходження оптимального керування u, змінних
стану x та виходу системи y
for i=1:n-1,
    v(:,i) = sqrtm(Q)*randn(SC,1); % Шум
вимірювання
    w(:,i) = sqrtm(R)*randn(SC,1); % Шум
керування
    u(:,i) = ud-K*(x_e(:,i)-xd);
    x(:,i+1) = (A*(x(:,i)-xd)+B*(u(:,i)-
ud)+G*w(:,i))*dt+x(:,i);
    yv(:,i) = C*(x(:,i)-xd)+D*(u(:,i)-
ud)+H*w(:,i)+v(:,i);
    x_e(:,i+1) = ((A-L*C-(B-L*D)*K)*(x_e(:,i)-
xd)+L*yv(:,i))*dt+x_e(:,i);
    y_e(:,i) = yd-C*(x_e(:,i)-xd)+D*(u(:,i)-ud);

```

```

end
% Побудова динаміки змінних стану
figure(1)
plot(0:dt:tf-dt,x), grid
hold on, plot(0:dt:tf-dt,x_e,'--'), hold off
title('Динаміка руху змінних стану');
xlabel('Час, t');
ylabel('Змінні стану, X(t) та X_e(t)');
legend('x_1', 'x_2', 'x_3', 'x_4', 'x_5', 'x_6', 'x_e_1',
'x_e_2', 'x_e_3', 'x_e_4', ...
'x_e_5', 'x_e_6')
% Побудова виходу системи
figure(2)
plot(dt:dt:tf-dt,y_e), grid
title('Динаміка виходу системи'); xlabel('Час,
t'); ylabel('Вихід, Y_e(t)');
legend('y_e_1', 'y_e_2', 'y_e_3')
figure(3)
plot(dt:dt:tf-dt,yv), grid
title('Динаміка виходу системи без фільтрації');
xlabel('Час, t');
ylabel('Вихід, Y_v(t)');
legend('y_1', 'y_2', 'y_3')
% Побудова вектора керування
figure(4)
plot(dt:dt:tf-dt,u), grid
title('Динаміка зміни керування'); xlabel('Час,
t'); ylabel('Керування, U(t)');
legend('u_1', 'u_2', 'u_3')

```

Результат виконання програми

Матриця Q:

Q =

```

1.5000    0    0
0    1.3000    0
0    0    0.0850

```

Матриця R:

R =

```

2.0000    0    0
0    1.5000    0
0    0    0.9000

```

Матриця N:

N =

```

0.0100    0.0800    0.0200
0.0210    0.0500    0.0750
0.0400    0.0700    0.0800

```

Вектор початкових умов x0:

x0 =

```

-0.6000
0
-0.2000
0.2000
0.4000
0.6000

```

Вектор заданих значень відхилень xd:

xd =

```

-0.2000
-0.6000
-0.4000
0
0.2000
0.4000

```

Вектор заданих значень відхилень керування ud :

$$ud = \begin{pmatrix} -0.4000 \\ 0.2000 \\ 0 \end{pmatrix}$$

Вектор заданих значень відхилень вихідної величини Yd :

$$yd = \begin{pmatrix} 1.0000 \\ 0.5000 \\ 0 \end{pmatrix}$$

Час спостереження tf :

$$tf = 18$$

Довжина кроку dt :

$$dt = 1.0000e-003$$

Кількість кроків n :

$$n = 18000$$

Модель системи у просторі стану:

$$a =$$

$$\begin{array}{rcccccc} & x1 & x2 & x3 & x4 & x5 & x6 \\ x1 & -0.51 & 0 & 1.3 & 0 & 0 & -2.1 \\ x2 & 1.8 & -0.25 & -1.3 & -0.1 & 0 & 0 \\ x3 & 0 & 0 & -0.16 & -0.1 & 0 & -0.2 \\ x4 & 0 & -0.35 & 0 & -0.65 & 0 & 0 \\ x5 & 0 & 0.21 & -0.61 & 0 & -0.35 & 0 \\ x6 & -0.1 & 0.581 & 0.04 & 0.02 & 0.05 & -0.14 \end{array}$$

$b =$

$$\begin{array}{rcccccc} & u1 & u2 & u3 & w1 & w2 & w3 \\ x1 & 1 & 0 & 0 & 0.1 & 0 & 0 \\ x2 & 1 & 0 & 0 & 0.35 & 0 & 0 \\ x3 & 0 & 0.2 & 0 & 0 & 0.21 & 0 \\ x4 & 1.5 & 0 & 0 & 0.51 & 0 & 0 \\ x5 & 0 & -1.1 & 0 & 0 & 0.28 & 0 \\ x6 & 0 & 0 & 1 & 0 & 0 & 0.1 \end{array}$$

$c =$

$$\begin{array}{rcccccc} & x1 & x2 & x3 & x4 & x5 & x6 \\ y1 & 1 & 0 & 0 & 0 & 0 & 0 \\ y2 & 0 & 1 & 1 & 0 & 0 & 0 \\ y3 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}$$

$d =$

$$\begin{array}{rcccccc} & u1 & u2 & u3 & w1 & w2 & w3 \\ y1 & 0.1 & 0 & 0 & 0.1 & 0 & 0.8 \\ y2 & 0 & 0 & 0.2 & 0.1 & 0.51 & 1.21 \\ y3 & 0 & 0.1 & 0 & 0.2 & 0 & 0.81 \end{array}$$

Continuous-time model.

Синтез фільтра Калмана:

Коефіцієнт підсилення фільтра:

$L =$

$$\begin{pmatrix} 0.6630 & 0.4299 & -0.6232 \\ 0.2420 & 0.8222 & -0.0073 \\ 0.1100 & 0.1070 & -0.1158 \\ 0.1302 & -0.0324 & 0.1663 \end{pmatrix}$$

-0.0845	0.0901	0.1338
-0.2975	0.0677	0.3440

Розв'язок рівнянь Рікатті:

P =

1.3794	0.6415	0.2328	0.2200	-
0.1416	-0.5672			
0.6415	1.8302	-0.0392	-0.1255	
0.0330	0.2826			
0.2328	-0.0392	0.1133	0.0190	-
0.0087	-0.1046			
0.2200	-0.1255	0.0190	0.3115	-
0.0525	-0.2167			
-0.1416	0.0330	-0.0087	-0.0525	
0.1033	0.0959			
-0.5672	0.2826	-0.1046	-0.2167	
0.0959	0.4693			

Неперервна модель фільтра Калмана:

a =

	x1_e	x2_e	x3_e	x4_e
x5_e	x6_e			
x1_e	-1.173	-0.4299	0.8701	0.6232
0.6232	-1.477			
x2_e	1.558	-1.072	-2.122	-0.09269
0.007309	0.007309			
x3_e	-0.11	-0.107	-0.267	0.01576
0.1158	-0.08424			
x4_e	-0.1302	-0.3176	0.03244	-0.8163
-0.1663	-0.1663			
x5_e	0.08453	0.1199	-0.7001	-0.1338
-0.4838	-0.1338			
x6_e	0.1975	0.5133	-0.02775	-0.324
-0.294	-0.484			

b =

	y1	y2	u1	y3	u2	u3
x1_e	0.9337	0.06232	-0.08598			
0.663	0.4299	-0.6232				
x2_e	0.9758	0.0007309	-0.1644			
0.242	0.8222	-0.007309				
x3_e	-0.011	0.2116	-0.02139			
0.11	0.107	-0.1158				
x4_e	1.487	-0.01663	0.006487			
0.1302	-0.03244	0.1663				
x5_e	0.008453	-1.113	-0.01802			-
0.08453	0.0901	0.1338				
x6_e	0.02975	-0.0344	0.9865			-
0.2975	0.06775	0.344				

c =

	x1_e	x2_e	x3_e	x4_e	x5_e	x6_e
y1_e	1	0	0	0	0	0
y2_e	0	1	1	0	0	0
y3_e	0	0	0	1	1	1
x1_e	1	0	0	0	0	0
x2_e	0	1	0	0	0	0
x3_e	0	0	1	0	0	0
x4_e	0	0	0	1	0	0
x5_e	0	0	0	0	1	0
x6_e	0	0	0	0	0	1

d =

	u1	u2	u3	y1	y2	y3
y1_e	0.1	0	0	0	0	0
y2_e	0	0	0.2	0	0	0
y3_e	0	0.1	0	0	0	0
x1_e	0	0	0	0	0	0
x2_e	0	0	0	0	0	0
x3_e	0	0	0	0	0	0
x4_e	0	0	0	0	0	0
x5_e	0	0	0	0	0	0
x6_e	0	0	0	0	0	0

Input groups:

Name

Channels

KnownInput 1,2,3
 Measurement 4,5,6

Output groups:

Name Channels
 OutputEstimate 1,2,3
 StateEstimate 4,5,6,7,8,9

Continuous-time model.

Синтез регулятора:

Матриця зворотного з'язку K:

K =

0.8198	0.6049	0.1845	-0.0154	
0.0191	-0.3700			
0.1420	0.0852	0.3154	-0.0162	-
0.0209	-0.1691			
-0.8733	0.2699	-0.8799	0.1891	
0.1438	1.8676			

Розв'язок рівняння Ріккати P:

P =

1.1606	0.4539	0.3507	-0.0832	-
0.0537	-0.8700			
0.4539	0.8958	0.1763	-0.0987	-
0.0509	-0.0647			
0.3507	0.1763	2.1190	-0.1184	
0.0088	-1.1923			
-0.0832	-0.0987	-0.1184	0.0744	
0.0692	0.1053			

-0.0537	-0.0509	0.0088	0.0692
0.0993	0.0609		
-0.8700	-0.0647	-1.1923	0.1053
0.0609	1.7486		

Власні значення замкненої системи E:

E =

-1.2080 + 1.1868i
-1.2080 - 1.1868i
-1.5711
-0.6357
-0.3962 + 0.1867i
-0.3962 - 0.1867i

Модель ЛКГ регулятора:

a =

	x1_e	x2_e	x3_e	x4_e
x5_e	x6_e			
x1_e	-2.022	-0.9768	0.6025	0.6548
0.619	-0.9603			
x2_e	0.6143	-1.618	-2.447	-0.04652
0.01229	0.6756			
x3_e	-0.1497	-0.1126	-0.3505	0.02306
0.1235	-0.01258			
x4_e	-1.341	-1.217	-0.2309	-0.7949
-0.1961	0.3689			
x5_e	0.2199	0.2145	-0.3664	-0.1483
-0.5047	-0.2853			
x6_e	1.039	0.232	0.8456	-0.5106
-0.4371	-2.321			

b =

	y1	y2	y3
x1_e	0.663	0.4299	-0.6232
x2_e	0.242	0.8222	-0.007309
x3_e	0.11	0.107	-0.1158
x4_e	0.1302	-0.03244	0.1663
x5_e	-0.08453	0.0901	0.1338

$x6_e$ -0.2975 0.06775 0.344

$c =$

	$x1_e$	$x2_e$	$x3_e$	$x4_e$	
$x5_e$	$x6_e$				
$u1$	-0.8198	-0.6049	-0.1845	0.01544	-
0.01914	0.37				
$u2$	-0.142	-0.08522	-0.3154	0.01617	
0.02095	0.1691				
$u3$	0.8733	-0.2699	0.8799	-0.1891	-
0.1438	-1.868				

$d =$

	$y1$	$y2$	$y3$
$u1$	0	0	0
$u2$	0	0	0
$u3$	0	0	0

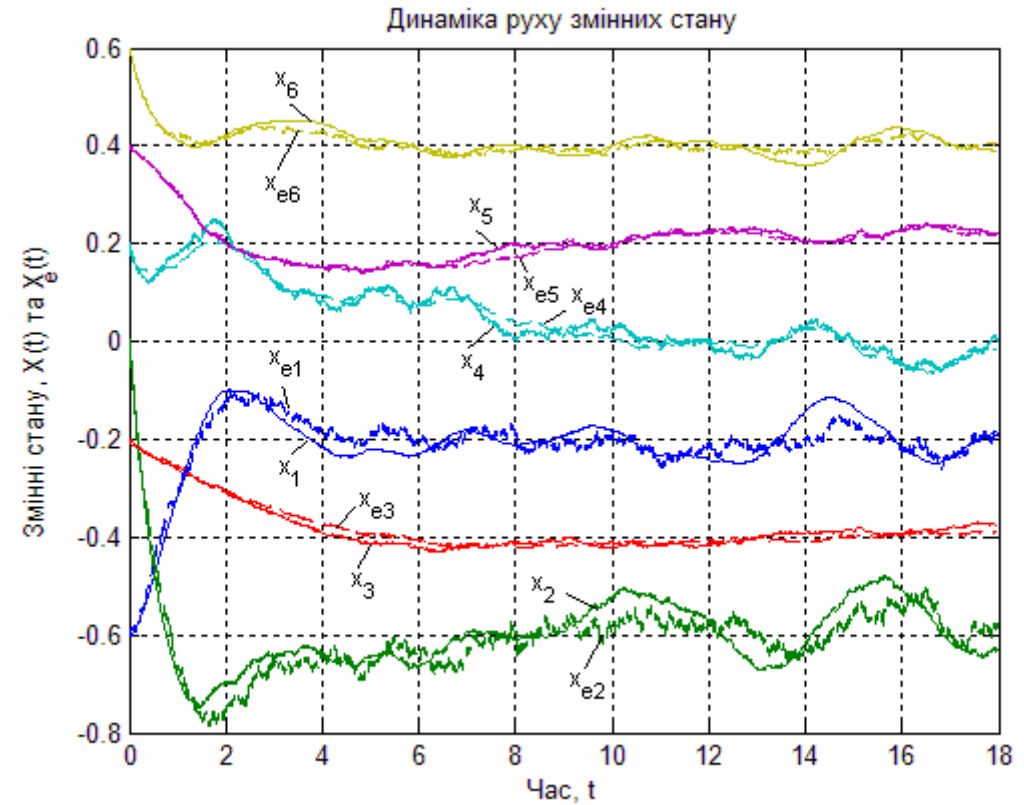
Input groups:

Name	Channels
Measurement	1,2,3

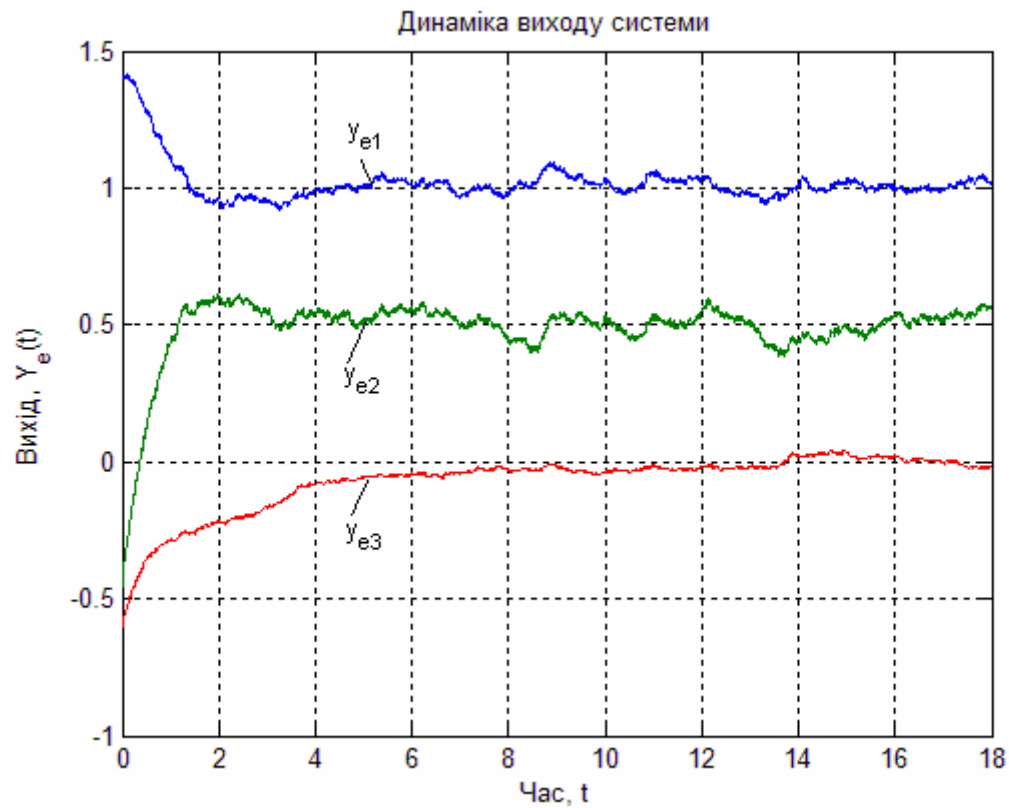
Output groups:

Name	Channels
Controls	1,2,3

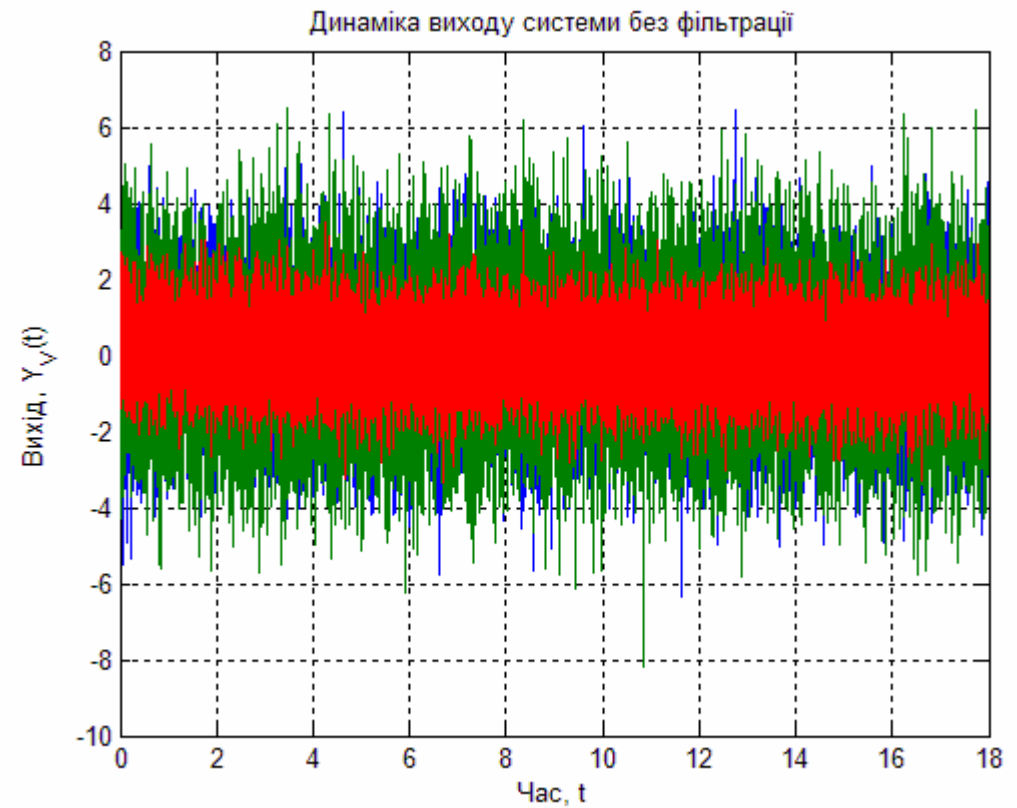
Continuous-time model.



a)

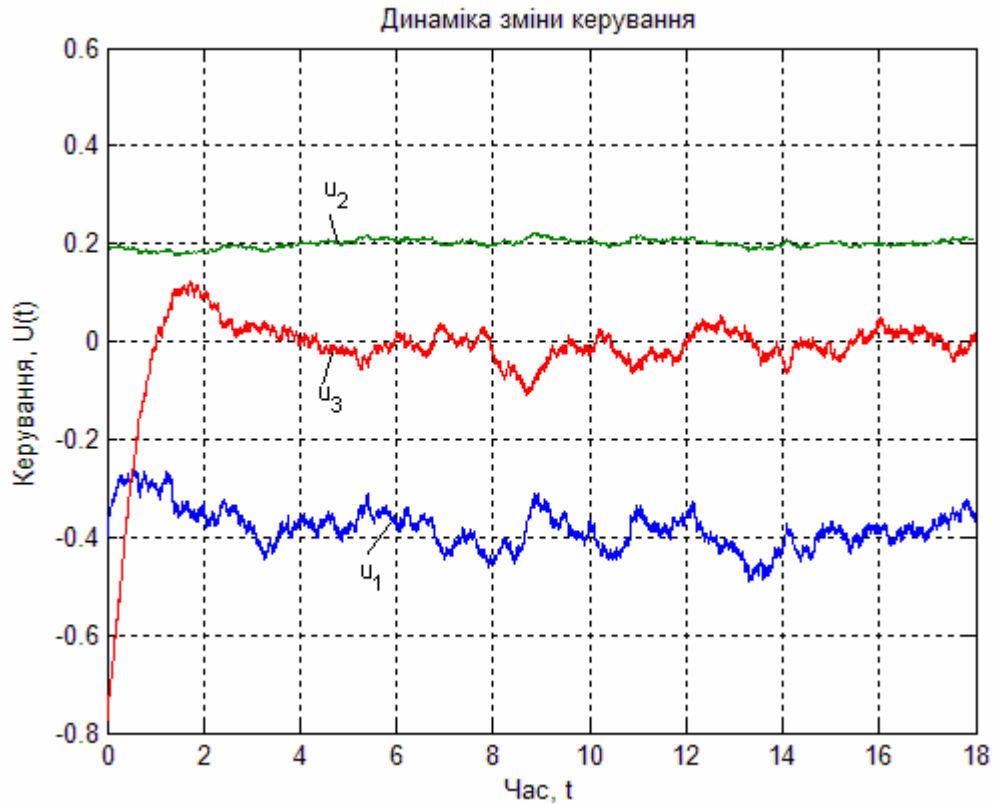


б)



а)

Рис. 3.3. Динаміка руху змінних стану а) і виходу системи б)



б)

Рис. 3.4. Динаміка зміни виходу системи без фільтрації а) і керування б)

Досліджували якість отриманого регулятора для звичайної системи, та системи з випадковими процесами (на графіках позначається штриховою кривою, а у програмі та формулах з індексом "e") з урахуванням зв'язку між станом і керуванням для наступних значеннях параметрів:

$$X(0) \equiv \hat{X}(0) = [-0,6 \quad 0 \quad -0,2 \quad 0,2 \quad 0,4 \quad 0,6]^T;$$

$$Xd \equiv \hat{X}d = [-0,2 \quad -0,6 \quad -0,4 \quad 0 \quad 0,2 \quad 0,4]^T;$$

$$Yd \equiv \hat{Y}d = [1 \quad 0,6 \quad 0]^T; \quad Ud = [-0,4 \quad 0,2 \quad 0]^T.$$

Виходи оптимального регулятора показані на рис. 3.3 а), і оцінка виходу системи представлена на рис. 3.3 б). Вихід системи без фільтрації показан на рис. 3.4 а), та оптимальне керування представлено на рис. 3.4 б). Регулятор показує досить високу якість, незважаючи на досить високий рівень завад.

4. Розміщення полюсів за допомогою зворотного зв'язку за станом

За допомогою зворотного зв'язку, за станом, можна забезпечити задане розташування полюсів передатної функції замкненої системи. Відповідний метод базується на використанні зворотного зв'язку по всім змінним стану

$$U = KX. \quad (4.1)$$

При цьому корені характеристичного рівняння розміщуються так, щоб перехідна характеристика системи задовольняла висунутим вимогам.

4.1. Задане розміщення полюсів за допомогою зворотного зв'язку за станом

Розглянемо систему, що описується рівняннями у просторі стану:

$$\frac{dX}{dt} = AX + BU. \quad (4.2)$$

Вважається, що пара матриць $[A, B]$ повністю керована. Задача ставиться наступним чином: знайти таку матрицю K розміром $r \times n$, щоб при керуванні

$$U = -KX \quad (4.3)$$

власні значення $A - BK$ розміщувались довільно в заданих точках на p – площині, або z – площини (якщо система дискретна).

Уявимо собі систему з одним входом

$$\frac{dX}{dt} = AX + B^*U \quad (4.4)$$

та визначимо матрицю B^* розмірністю $n \times 1$ як

$$B^* = BW, \quad (4.5)$$

де W має розмірність $r \times 1$. Матриця W повинна бути обрана так, щоб пара матриць $[A, B^*]$ була керована. Тоді за допомогою зворотного зв'язку

$$U = -K^*X \quad (4.6)$$

можна розмістити власні значення матриці $A - B^*K^*$ в тих самих точках, що й власні значення матриці $A - BK$. Відповідно, задача зводиться до синтезу зворотного зв'язку за станом з одним входом, що описується рівняннями (4.4). Якщо буде знайдена матриця зворотного зв'язку K^* , то K визначається виразом:

$$K = WK^* \quad (4.7)$$

оскільки $BK = B^*K^*$.

Очевидно, що матриця W не є єдиною. Вимагається тільки, щоб вона задовольняла умові керованості пари матриць $[A, BW]$. Матрицю коефіцієнтів зворотного зв'язку K^* для одновимірної задачі, можна знайти з розв'язку системи з n лінійних алгебраїчних рівнянь, з n невідомими відносно шуканих коефіцієнтів

$$\det(pI - A + BK^*) = q(p), \quad (4.8)$$

де $q(p)$ - бажаний характеристичний поліном замкненої системи

$$q(p) = \prod_{i=1}^n (p - p_i) \equiv p^n + \alpha_1 p^{n-1} + \dots + \alpha_n, \quad (4.9)$$

де p_i - бажані полюси замкненої системи.

Зовнішній зворотній зв'язок. Розглянений вище метод показує, що при синтезі систем за заданим розміщенням полюсів, систему з декількома входами, можна замінити на «еквівалентну» систему з одним входом. Дві системи вважаються «еквівалентними» тільки в тому смислі, що вони мають однакові власні значення.

Матриця W , використовується для перетворення системи з декількома входами в систему з одним входом, має одне обмеження: остання система повинна бути керованою. Матрицю W , можна обрати довільним чином, однак на практиці ми можемо накласти і інші корисні обмеження на вибір її елементів. Так як W множиться на вектор входу U , то зміст W полягає в тому, що її елементи надають різні ваги зворотнім зв'язкам від змінних стану до керуючих впливів. Наприклад, якщо вибрати $w_1 = 2w_2$, то це значить, що вплив зворотніх зв'язків на u_1 буде в 2 рази сильнішим, чим на u_2 .

Застосування цього алгоритму для синтезу оцінювача стану, зводиться до транспонування матриці A та підстановки C^T замість B . Реалізація в MATLAB виконується наступним чином

- $L = \text{place}(A', C', P)$.

Тут P – бажане розташування полюсів; L – матричний коефіцієнт підсилення оцінювача.

Приклад

Розглянемо систему з одним входом і одним виходом третього порядку, що описується рівняннями у просторі стану:

$$\frac{dX}{dt} = A(X - Xd) + B(U - Ud), \quad X(0) = [-0,8 \quad 0,6 \quad 0,2]^T$$

$$Y - Yd = C(X - Xd) + D(U - Ud),$$

$$\text{де } A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -6 & -4 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 0 \\ 2,5 \end{bmatrix}; \quad C = [0 \ 0 \ 1]; D = 0;$$

$$X_d = [20 \ 10 \ 5]^T; U_d = 20.$$

Бажане розташування полюсів $P = [-1 \ -0,5 \ -2,5]$.

Примітка. Побудова кореневого годографа або карти нулів та полюсів для системи, будується для кожного каналу окремо. Тут і в наступному прикладі рівняння виходу введено лише для побудови кореневого годографа та карти нулів та полюсів.

Наведемо приклад програми для розв'язку цієї задачі з назвою "Zad_Roz_Pol".

```
function Zad_Roz_Pol
A = [0 1 0; 0 0 1; -1 -6 -4]; % Введення матриці
A
B = [0; 0; 2.5]; % Введення матриці B
C = [0 0 1]; % Введення матриці C
disp('Бажане розташування полюсів P:')
P = [-1 -5 -2] % Завдання бажаного розташування
полюсів
disp('Вектор початкових умов x0:')
x0 = [0.8 -0.6 0.2]' % Завдання вектора
початкових умов
disp('Вектор заданих значень відхилень Xd:')
xd = [20 10 5]' % Завдання вектора заданих
значень відхилень
disp('Задане значення відхилення керування Ud:')
ud = 20 % задання значення відхилення керування
disp('час спостереження tf:')
tf = 10 % завдання часу спостереження
disp('довжина кроку dt:')
dt = 0.001 % завдання довжини кроку
disp('кількість кроків n:')
n = tf/dt
[K,prec,message] = place(A,B,P); % Визначення
матриці зворотнього зв'язку
```

```
disp('Матриця зворотнього з язку K:'); K
disp('число точних десяткових цифр у фактичних
полюсах замкненої системи прес:'); прес
disp('повідомлення, що попереджує проте, що
отриманий полюс відрізняється на 10 % від
даного message:'); message
disp('Модель системи у просторі стану:')
w1 = ss(A,B,C,0) % Формування моделі системи у
просторі стану
rlocus(w1), grid; % Побудова кореневого годографа
xlabel('Дійсна вісь'); ylabel('Уявна вісь');
title('Кореневий годограф системи до синтезу')
w2 = ss(A-B*K,B,C,0); % Формування моделі системи
з оптимальною ланкою зворотного % зв'язку
figure(2); rlocus(w2), grid; % Побудова
кореневого годографа системи з оптимальною %
ланкою зворотного зв'язку
xlabel('Дійсна вісь'); ylabel('Уявна вісь');
title('Кореневий годограф системи після синтезу')
% Визначення розмірності задачі
SA = size(A); SA = SA(1); SB = size(B); SB =
SB(2);
% Формування векторів x та u
x = zeros(SA,n); u = zeros(SB,n-1);
% Формування початкового стану
x(:,1) = x0;
% Знаходження оптимального керування u та змінних
стану x
for i=1:n-1,
    u(:,i) = ud-K*(x(:,i)-xd);
    x(:,i+1) = (A*(x(:,i)-xd)+B*(u(:,i)-
ud))*dt+x(:,i);
end
% Подова динаміки змінних стану
figure(3)
plot(0:dt:tf-dt,x), grid
title('динаміка руху змінних стану');
xlabel('Час, t');
ylabel('Змінні стану, X(t)');
legend('x_1','x_2','x_3')
% Побудова вектора керування
```

```

figure(4)
plot(dt:dt:tf-dt,u), grid
title('Динаміка зміни керування'); xlabel('час,
t'); ylabel('керування, U(t)'); legend('u')

```

Результат виконання програми

Бажані полюси системи P:

```

P =
    -1    -5    -2

```

Вектор початкових умов x0:

```

x0 =
    0.8000
   -0.6000
    0.2000

```

Вектор заданих значень відхилень xd:

```

xd =
    20
    10
     5

```

Задане значення відхилення керування ud:

```

ud =

```

```

20

```

Час спостереження tf:

```

tf =

```

```

10

```

Довжина кроку dt:

```

dt =

```

```

1.0000e-003

```

Кількість кроків n:

```

n =

```

```

10000

```

Матриця зворотного зв'язку K:

```

K =
    3.6000    4.4000    1.6000

```

число точних десяткових цифр у фактичних полюсах замкненої системи pres:

```

pres =

```

```

15

```

повідомлення, що попереджує проте, що отриманий полюс відрізняється на 10 % від заданого message:

```

message =

```

```

''

```

Модель системи у просторі стану:

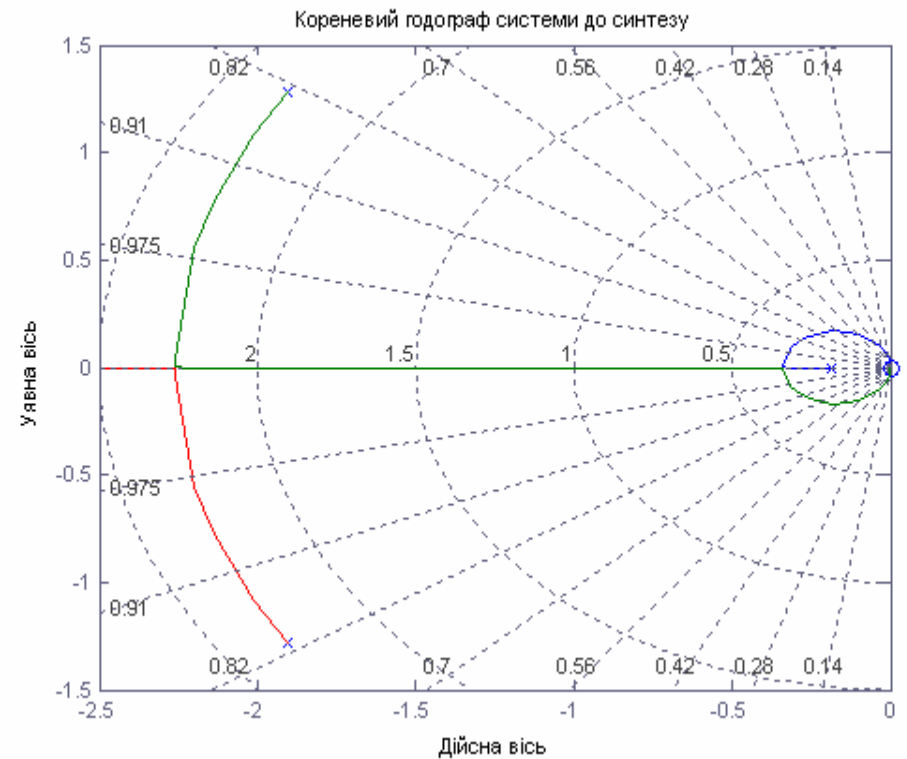
$$a = \begin{array}{c|ccc} & x1 & x2 & x3 \\ \hline x1 & 0 & 1 & 0 \\ x2 & 0 & 0 & 1 \\ x3 & -1 & -6 & -4 \end{array}$$

$$b = \begin{array}{c|c} & u1 \\ \hline x1 & 0 \\ x2 & 0 \\ x3 & 2.5 \end{array}$$

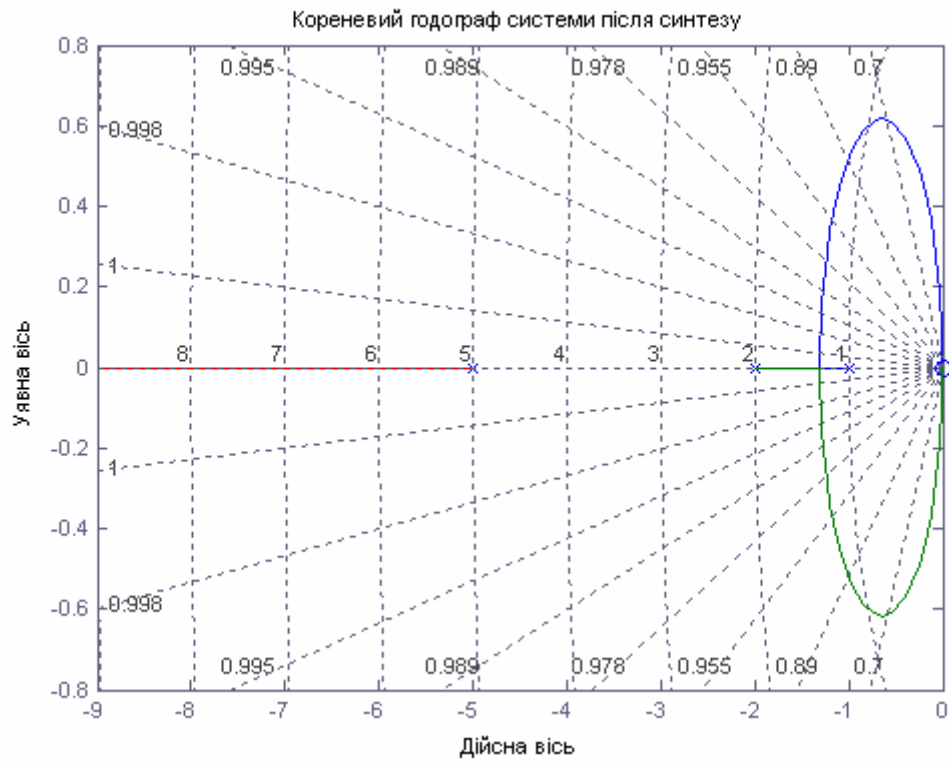
$$c = \begin{array}{c|ccc} & x1 & x2 & x3 \\ \hline y1 & 0 & 0 & 1 \end{array}$$

$$d = \begin{array}{c|c} & u1 \\ \hline y1 & 0 \end{array}$$

Continuous-time model.

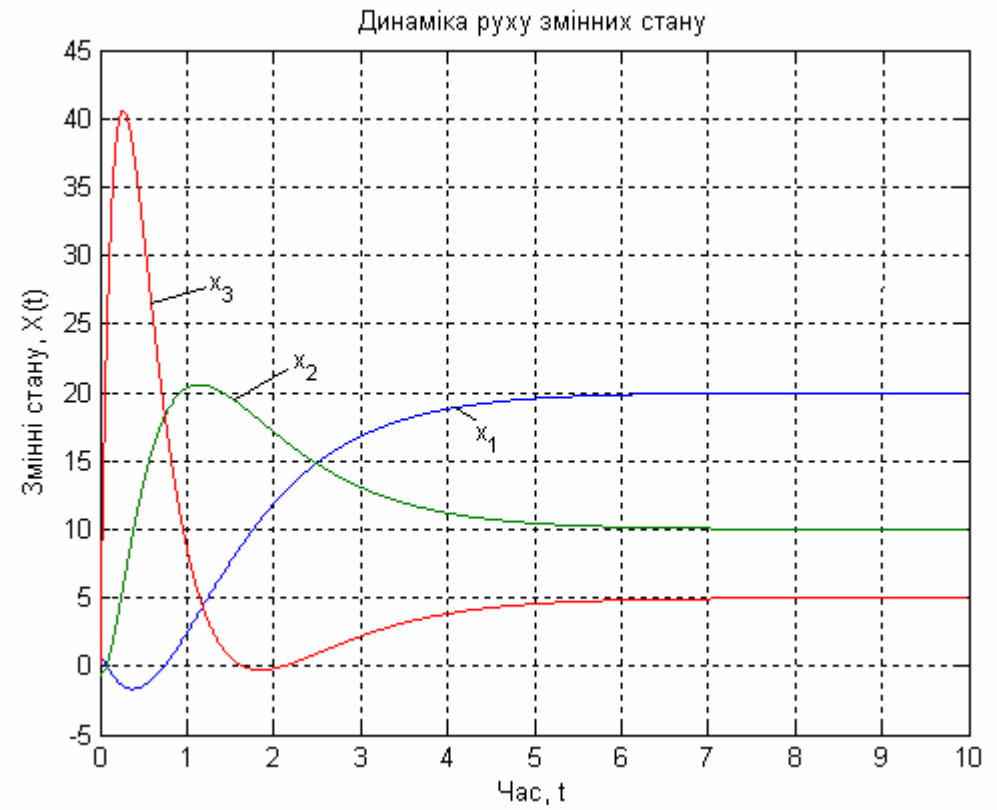


a)

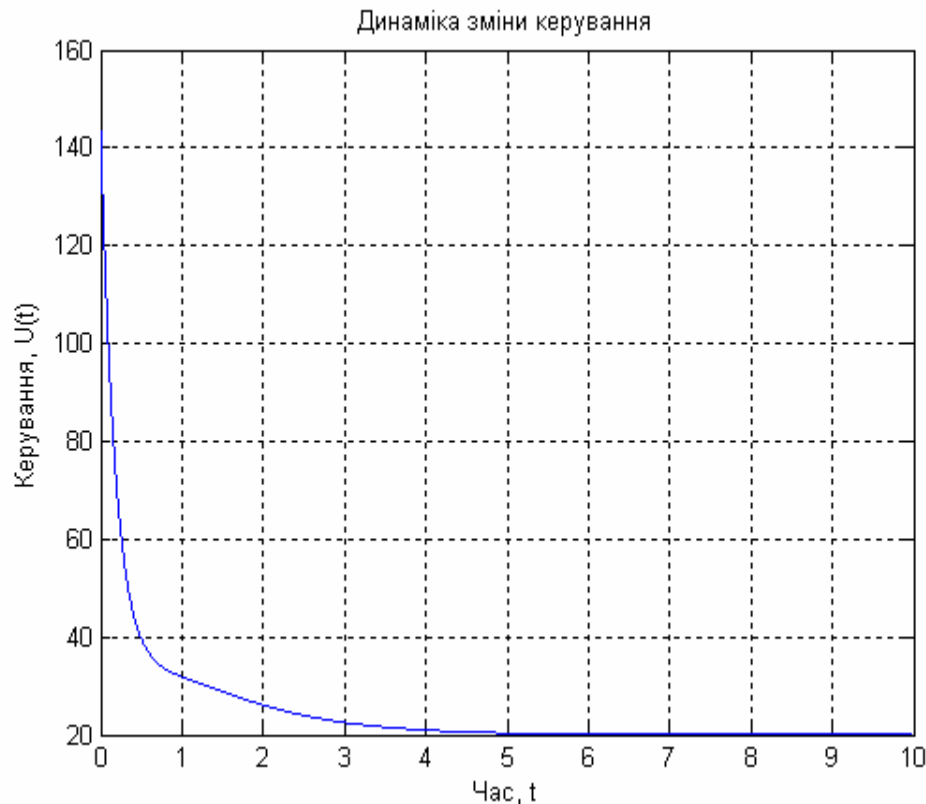


б)

Рис. 4.1. Кореневий годограф системи до а) і після синтезу б)



а)



б)

Рис. 4.2. Динаміка руху змінних стану а) і керування б)

Досліджували якість отриманого регулятора для наступних значеннях параметрів:

$$X(0) = [-0,8 \quad 0,6 \quad 0,2]^T; Xd = [20 \quad 10 \quad 5]^T; Ud = 20 \quad \text{та}$$

бажаного розташування полюсів $P = [-1 \quad -0,5 \quad -2,5]$.

Коефіцієнт підсилення K визначається за допомогою зворотного зв'язку за станом. Як видно з кореневого годографа до синтезу рис. 4.1 а), і після синтезу рис. 4.1 б) полюси замкненої системи повністю

задовольняють висунутим вимогам. Змінні стану показані на рис. 4.2 а). Оптимальне керування представлено на рис. 4.2 б).

Розглянемо дискретну систему з одним входом і одним виходом третього порядку, що описується рівняннями у просторі стану:

$$X[n+1] = A(X[n] - Xd) + B(U[n] - Ud), \quad X[0] = [0,1 \quad 0,2 \quad -0,3]^T$$

$$Y[n] - Yd = C(X[n] - Xd) + D(U[n] - Ud),$$

$$\text{де } A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0,1 & -0,6 & -0,8 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 0 \\ 0,5 \end{bmatrix}; C = [0 \quad 0 \quad 1]; D = 0;$$

$$Xd = [-0,3 \quad 0,2 \quad 0,4]^T; Ud = 1,2; T = 0,5.$$

Бажане розташування полюсів $P = [-0,5 \quad -0,1 \quad -0,2]$.

Наведемо приклад програми для розв'язку цієї задачі для дискретної системи з назвою "Zad_Roz_Pol_Discr".

```
function Zad_Roz_Pol_Discr
A = [0 1 0; 0 0 1; -0.1 -0.6 -0.8]; % Введення матриці A
B = [0; 0; 0.5]; % Введення матриці B
C = [0 0 1]; % Введення матриці C
disp('Бажане розташування полюсів P:')
P = [-0.5 -0.1 -0.2] % Завдання бажаного розташування полюсів
disp('Вектор початкових умов x0:')
x0 = [0.1 0.2 0.3]' % Завдання вектора початкових умов
disp('Вектор заданих значень відхилень xd:')
xd = [-0.3 0.2 0.4]' % Завдання вектора заданих значень відхилень
disp('Задане значення відхилення керування Ud:')
ud = 1.2 % Задання значення відхилення керування
disp('Час спостереження tf:')
tf = 25 % Завдання часу спостереження
```

```

disp('довжина кроку dt:')
T = 0.5 % Період квантування
disp('кількість кроків n:')
n = tf/T
[K,prec,message] = place(A,B,P); % Визначення
матриці зворотнього зв'язку
disp('Матриця зворотнього зв'язку K:'); K
disp('число точних десяткових цифр у фактичних
полюсах замкненої системи prec:'); prec
disp('повідомлення, що попереджує проте, що
отриманий полюс відрізняється на 10 % від
заданого message:'); message
disp('Модель системи у просторі стану:')
w1 = ss(A,B,C,0,T) % Формування моделі системи у
просторі стану
rlocus(w1), grid; % Побудова кореневого годографа
xlabel('дійсна вісь'); ylabel('уявна вісь');
title('Кореневий годограф системи до синтезу')
w2 = ss(A-B*K,B,C,0,T); % Формування моделі
системи з оптимальною
% ланкою зворотного зв'язку
figure(2); rlocus(w2), grid; % Побудова
кореневого годографа системи з
% оптимальною ланкою зворотного зв'язку
xlabel('дійсна вісь'); ylabel('уявна вісь');
title('Кореневий годограф системи після синтезу')
% Визначення розмірності задачі
SA = size(A); SA = SA(1); SB = size(B); SB =
SB(2);
% Формування векторів x та u
x = zeros(SA,n); u = zeros(SB,n-1);
% Формування початкового стану
x(:,1) = x0;
% Знаходження оптимального керування u та змінних
стану x
for i=1:n-1,
    u(:,i) = ud-K*(x(:,i)-xd);
    x(:,i+1) = (A*(x(:,i)-xd)+B*(u(:,i)-
ud))*T+x(:,i);
end
% Подова динаміки змінних стану

```

```

figure(3)
stem(0:T:tf-T,x'), grid
title('динаміка руху змінних стану');
xlabel('Час, t');
ylabel('Змінні стану, X(t)');
legend('x_1','x_2','x_3')
% Побудова вектора керування
figure(4)
stem(T:T:tf-T,u), grid
title('Динаміка зміни керування'); xlabel('Час,
t'); ylabel('Керування, U(t)'); legend('u')

```

Результат виконання програми

Бажане розташування полюсів P:

P =

-0.5000 -0.1000 -0.2000

Вектор початкових умов x0:

x0 =

*0.1000
0.2000
0.3000*

Вектор заданих значень відхилень xd:

xd =

*-0.3000
0.2000
0.4000*

Задане значення відхилення керування Ud:

ud =

1.2000

Час спостереження tf :

$tf =$

25

Довжина кроку dt :

$T =$

0.5000

Кількість кроків n :

$n =$

50

Матриця зворотнього з язку K :

$K =$

-0.1800 -0.8600 0.0000

число точних десяткових цифр у фактичних полюсах замкненої системи $prec$:

$prec =$

15

повідомлення, що попереджує проте, що отриманий полюс відрізняється на 10 % від заданого $message$:

$message =$

''

Модель системи у просторі стану:

$a =$

	$x1$	$x2$	$x3$
$x1$	0	1	0
$x2$	0	0	1
$x3$	-0.1	-0.6	-0.8

$b =$

	$u1$
$x1$	0
$x2$	0
$x3$	0.5

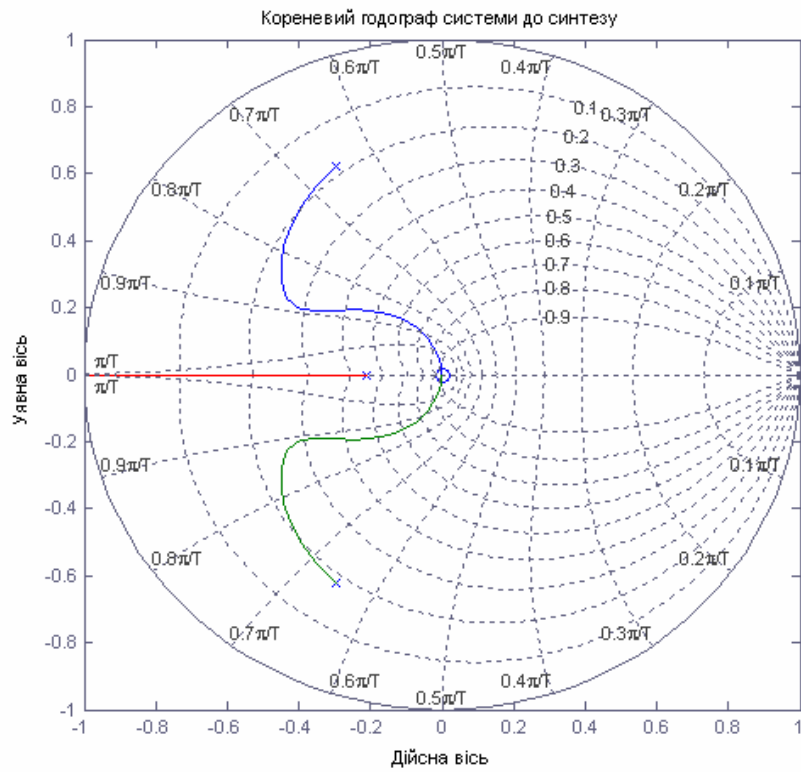
$c =$

	$x1$	$x2$	$x3$
$y1$	0	0	1

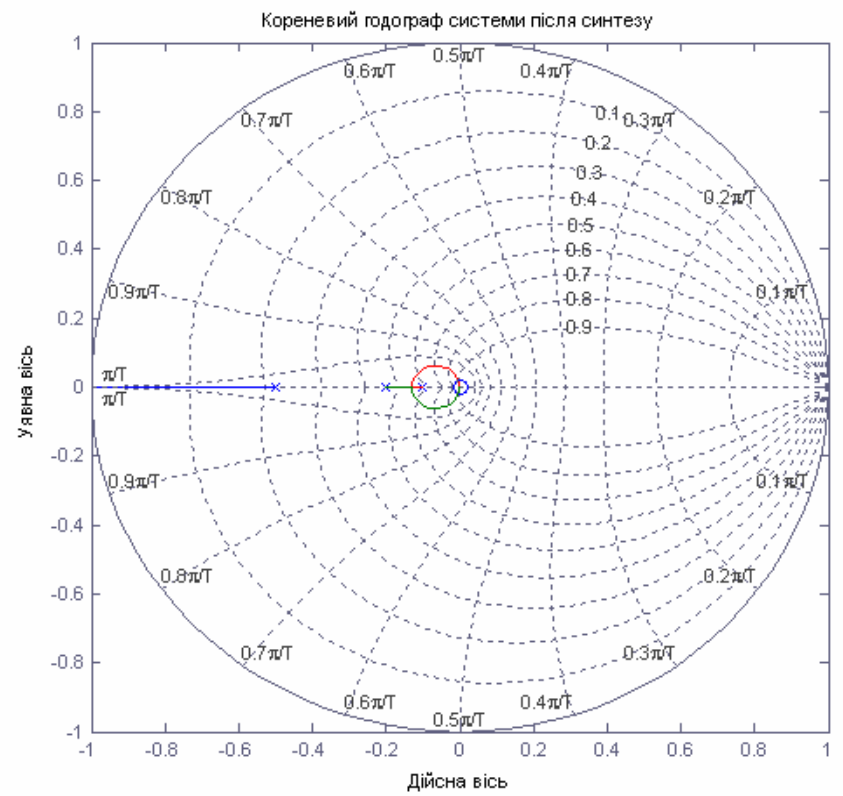
$d =$

	$u1$
$y1$	0

Sampling time: 0.5
Discrete-time model.

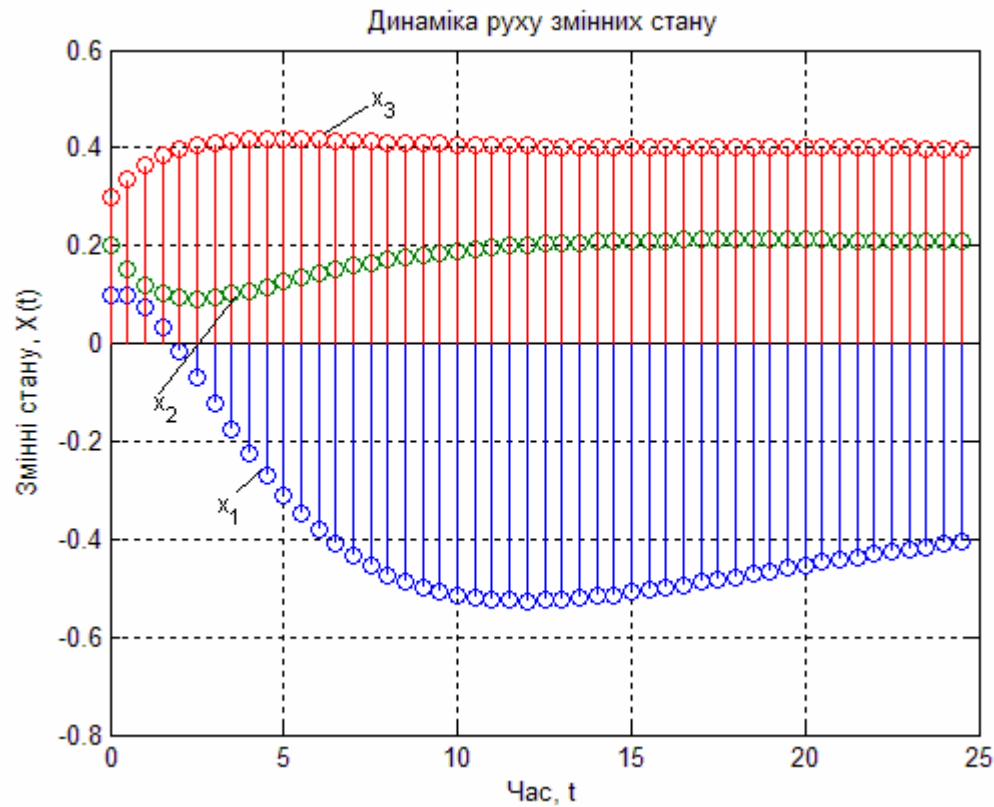


а)

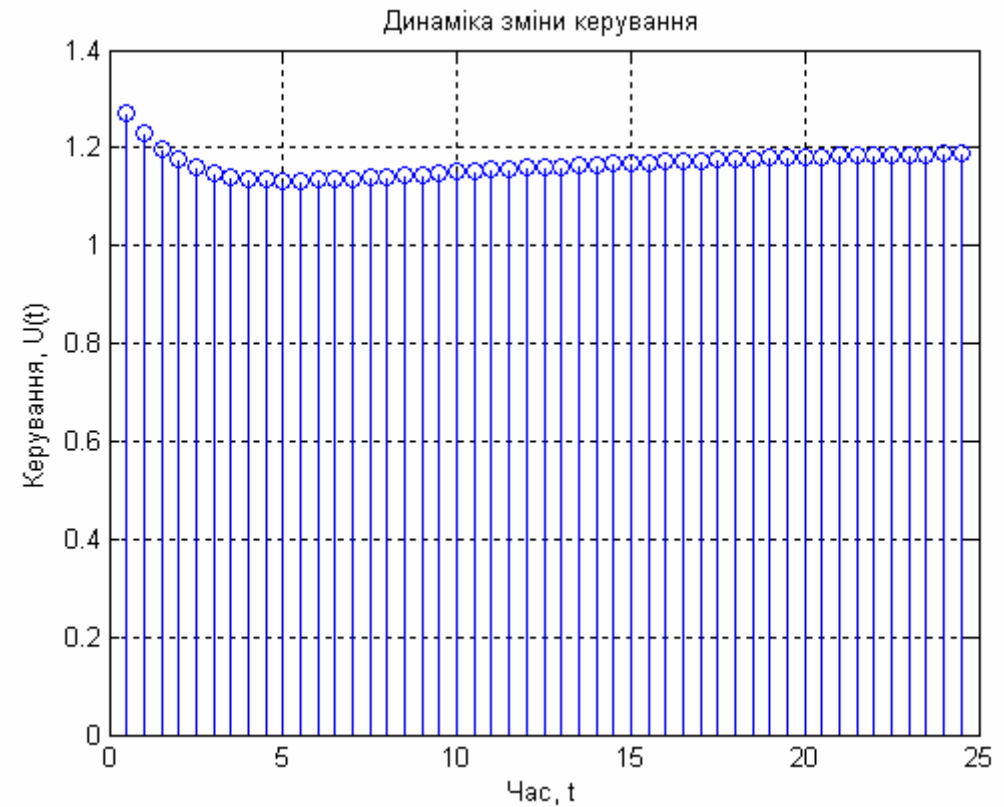


б)

Рис. 4.3. Кореневий годограф системи до а) і після синтезу б)



а)



б)

Рис. 4.4. Динаміка руху змінних стану а) і керування б)

Досліджували якість дискретного регулятора, для наступних значень параметрів: $X[0] = [0,1 \ 0,2 \ -0,3]^T$; $X_d = [-0,3 \ 0,2 \ 0,4]^T$; $U_d = 1,2$; $T = 0,5$ та бажаного розташування полюсів $P = [-0,5 \ -0,1 \ -0,2]$. Коефіцієнт підсилення K визначається за допомогою зворотного зв'язку за станом. Полюси замкненої системи повністю задовольняють висуnutим вимогам, як видно з кореневого годографа до синтезу рис.

4.3 а) і після синтезу рис. 4.3 б). Змінні стану показані на рис. 4.4 а), а оптимальне керування представлено на рис. 4.4 б).

4.2. Застосування формули Аккермана

Для системи з одним входом і одним виходом матрицю коефіцієнтів зворотного зв'язку стану $K = [k_1 \ k_2 \ \dots \ k_n]$, за допомогою якої формується керуючий сигнал $u = -Kx$, можна визначити за допомогою формули Аккермана.

За бажаними значеннями полюсів p_i задаємо бажаний характеристичний поліном замкненої системи

$$q(p) = \prod_{i=1}^n (p - p_i) \equiv p^n + \alpha_1 p^{n-1} + \dots + \alpha_n \quad (4.10)$$

то матриця коефіцієнтів зворотного зв'язку, за станом, визначається виразом:

$$K = [0 \ 0 \ \dots \ 1] P_c^{-1} q(A), \quad (4.11)$$

де $q(A) = A^n + \alpha_1 A^{n-1} + \dots + \alpha_{n-1} A + \alpha_n I$, а

$P_c = [B \ AB \ A^2 B \ \dots \ A^{n-1} B]$ – матриця керованості.

Застосування цього алгоритму для синтезу оцінювача стану зводиться до транспонування матриці A та підстановки C^T замість B . Реалізація в MATLAB виконується наступним чином:

- $L = \text{acker}(A', C', P)$.

Тут P – бажане розташування полюсів; L – матричний коефіцієнт підсилення оцінювача.

Обмеження зворотного зв'язку за станом

Є дві причини, за якими зворотній зв'язок за станом недоцільно використовувати. Перша полягає в тому, що цей зворотній зв'язок за своєю дією еквівалентний ПД чи ПД – регулятором, які мають нескінченну смугу пропускання, тоді як реальні елементи і регулятори завжди мають кінцеву смугу пропускання. Друга причина в тому, що звичайно просто неможливо виміряти всі змінні стану і використати їх в якості сигналів зворотного зв'язку. Реально вимірювати тільки деякі із змінних стану. Тому на практиці, регулятор часто використовує інформацію тільки про входні, вихідні сигнали системи і про обмежене число змінних стану.

Приклад

Розглянемо систему з одним входом і одним виходом третього порядку, що описується рівняннями у просторі стану:

$$\frac{dX}{dt} = A(X - Xd) + B(U - Ud), \quad X(0) = [-0,8 \ 0,6 \ 0,2]^T$$

$$Y = CX,$$

де $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -6 & -4 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix};$

$$C = [0 \ 0 \ 1]; \quad Xd = [0,2 \ 0,3 \ 0,4]^T; \quad Ud = 2.$$

Бажане розташування полюсів $P = [-1 \ -0,5 \ -2,5]$.

Примітка. Побудова кореневого годографа або карти нулів та полюсів для системи, будується для кожного каналу окремо. Тут і в наступному прикладі рівняння виходу введено лише для побудови корневих годографів.

Наведемо приклад програми для розв'язку цієї задачі з назвою "Ackerman".

function Ackerman

A = [0 1 0; 0 0 1; -1 -6 -4]; % Введення матриці А

```

B = [0; 0; 2]; % Введення матриці B
C = [0 0 1]; % Введення матриці C
disp('Бажані полюси системи P:')
P = [-1 -0.5 -2.5] % Введення бажаних полюсів
системи
disp('Вектор початкових умов x0:')
x0 = [0.8 -0.6 0.2]' % Завдання вектора
початкових умов
disp('Вектор заданих значень відхилень xd:')
xd = [0.2 0.3 0.4]' % Завдання вектора заданих
значень відхилень
disp('Задане значення відхилення керування Ud:')
ud = 2 % Задання значення відхилення керування
disp('час спостереження tf:')
tf = 20 % Завдання часу спостереження
disp('довжина кроку dt:')
dt = 0.001 % Завдання довжини кроку
disp('кількість кроків n:')
n = tf/dt
disp('матриця зворотнього зв'язку K:')
K = acker(A,B,P) % Визначення матриці зворотнього
зв'язку за допомогою формули
% Аккермана
disp('Модель системи у просторі стану:')
w1 = ss(A,B,C,0) % формування моделі системи у
просторі стану
pzmap(w1), grid; % Побудова нулів та полюсів
системи
xlabel('дійсна вісь'); ylabel('Уявна вісь');
title('Карта нулів та полюсів системи до
синтезу')
w2 = ss(A-B*K,B,C,0); % формування моделі системи
з оптимальною ланкою зворотного зв'язку
figure(2); pzmap(w2), grid; % Побудова нулів та
полюсів системи з оптимальною
% ланкою зворотного зв'язку
xlabel('дійсна вісь'); ylabel('Уявна вісь');
title('Карта нулів та полюсів системи після
синтезу')
% Визначення розмірності задачі
SA = size(A); SA = SA(1);

```

```

% Формування векторів x та u
x = zeros(SA,n); u = zeros(1,n-1);
% Формування початкового стану
x(:,1) = x0;
% Знаходження оптимального керування u та змінних
стану x
for i=1:n-1,
    u(:,i) = ud-k*(x(:,i)-xd);
    x(:,i+1) = (A*(x(:,i)-xd)+B*(u(:,i)-
ud))*dt+x(:,i);
end
% Подова динаміки змінних стану
figure(3)
plot(0:dt:tf-dt,x), grid
title('динаміка руху змінних стану');
xlabel('час, t');
ylabel('Змінні стану, X(t)');
legend('x_1','x_2','x_3')
% Побудова вектора керування
figure(4)
plot(dt:dt:tf-dt,u), grid
title('динаміка зміни керування'); xlabel('час,
t'); ylabel('Керування, U(t)'); legend('u')

```

Результат виконання програми

Бажані полюси системи P:

P =

-1.0000 -0.5000 -2.5000

Вектор початкових умов x0:

x0 =

*0.8000
-0.6000
0.2000*

Вектор заданих значень відхилень xd:

$xd =$
0.2000
0.3000
0.4000

Задане значення відхилення керування Ud :

$ud =$
2

Час спостереження tf :

$tf =$
20

Довжина кроку dt :

$dt =$
 $1.0000e-003$

Кількість кроків n :

$n =$
20000

Матриця зворотного зв'язку K :

$K =$
0.1250 -0.8750 0

Модель системи у просторі стану:

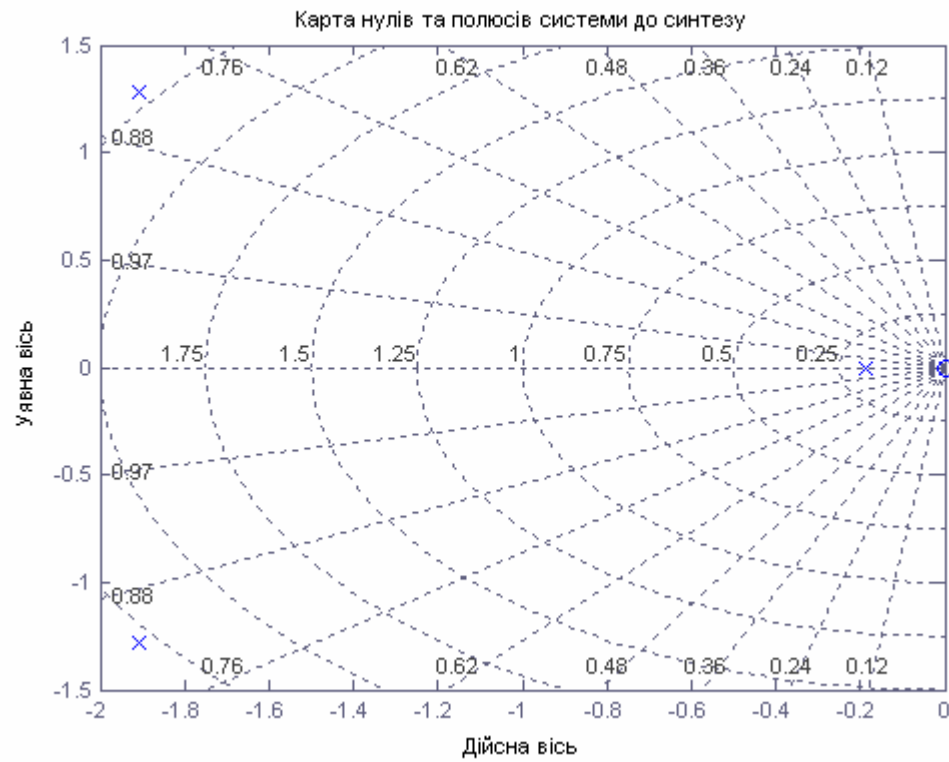
$a =$
 $x1$ $x2$ $x3$
 $x1$ 0 1 0
 $x2$ 0 0 1
 $x3$ -1 -6 -4

$b =$
 $u1$
 $x1$ 0
 $x2$ 0
 $x3$ 2

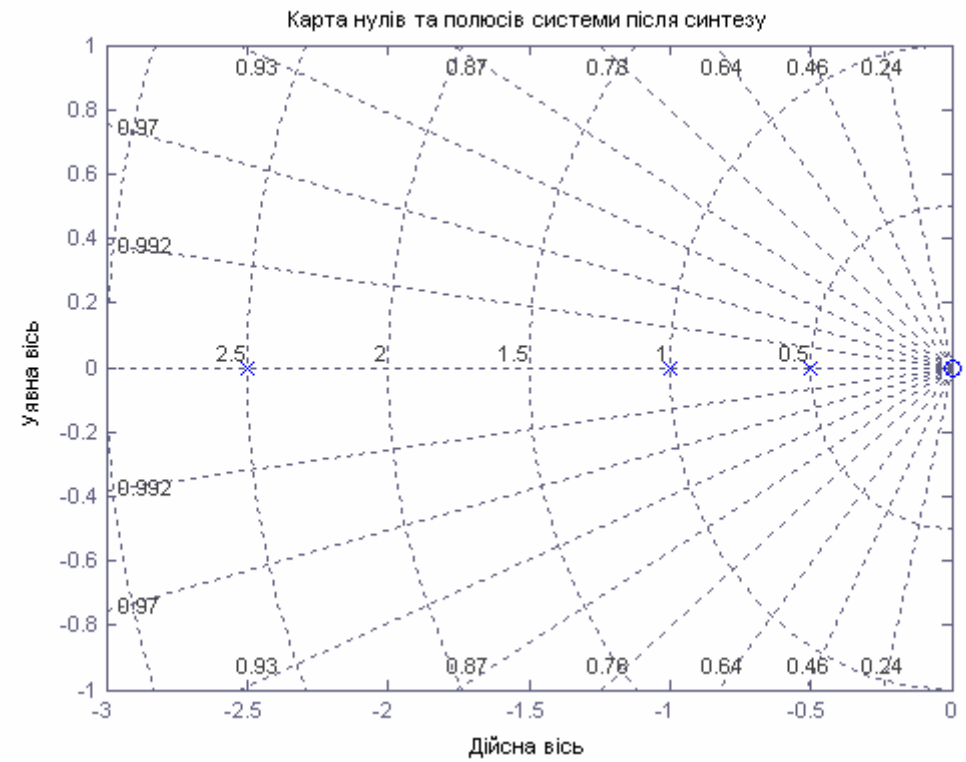
$c =$
 $x1$ $x2$ $x3$
 $y1$ 0 0 1

$d =$
 $u1$
 $y1$ 0

Continuous-time model.

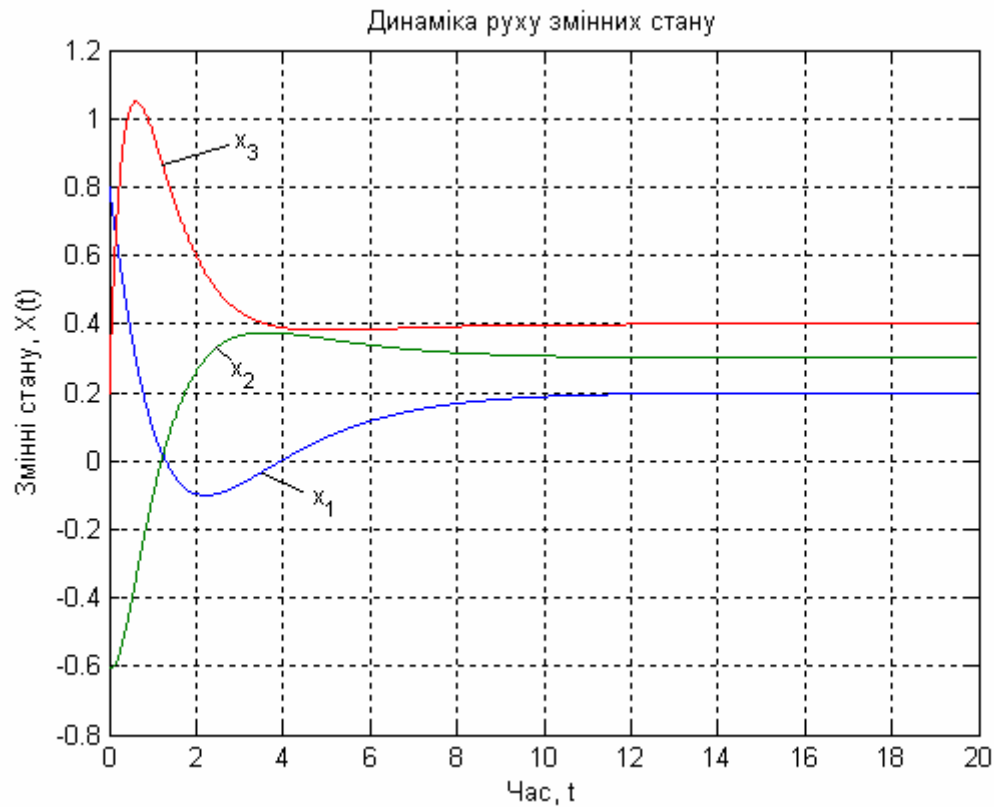


а)

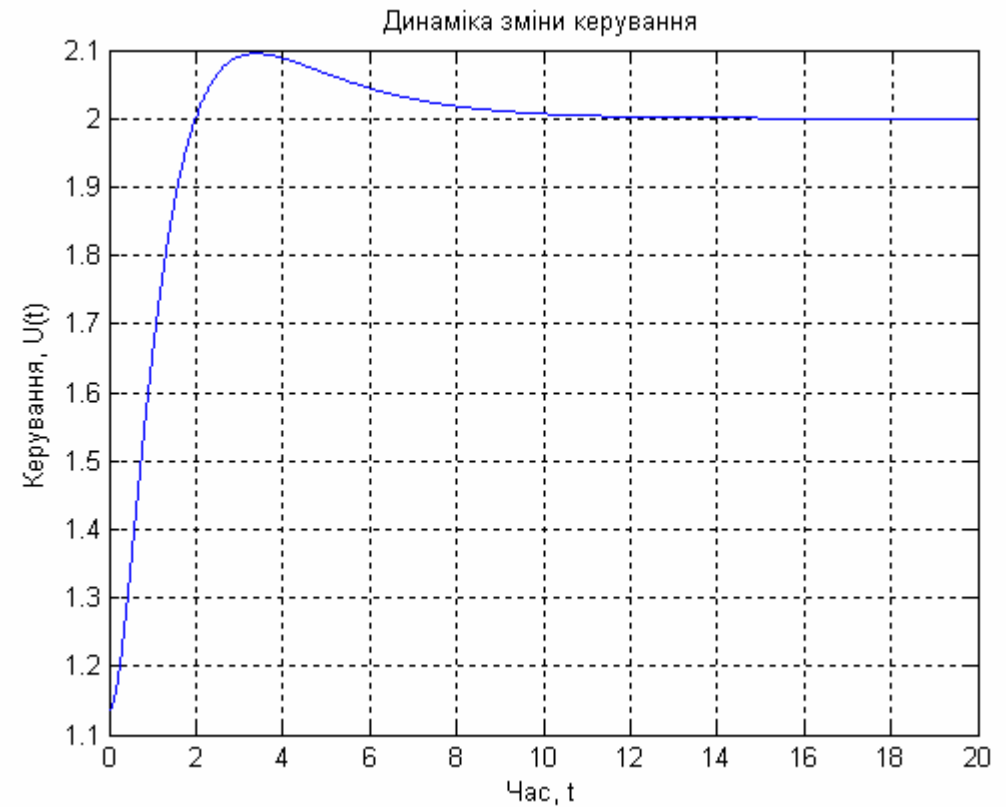


б)

Рис. 4.5. Карта нулів та полюсів системи до а) і після синтезу б)



а)



б)

Рис. 4.6. Динаміка руху змінних стану а) і керування б)

Якість регулятора досліджували для таких значень параметрів: $X(0) = [-0,8 \ 0,6 \ 0,2]^T$; $X_d = [0,2 \ 0,3 \ 0,4]^T$; $U_d = 2$ та бажаного розташування полюсів $P = [-1 \ -0,5 \ -2,5]$. Коефіцієнт підсилення K визначається з формули Аккермана. Полюси замкненої системи представлені на рис. 4.5 а) до синтезу, і на рис. 4.5 б) після синтезу і задовольняють висунутим вимогам. Змінні

стану показані на рис. 4.6 а). Оптимальне керування представлено на рис. 4.6 б).

Розглянемо дискретну систему з одним входом і одним виходом третього порядку, що описується рівняннями у просторі стану:

$$X[n+1] = A(X[n] - Xd) + B(U[n] - Ud), \quad X[0] = [0, 1 \quad -0,1 \quad 0,2]^T$$

$$Y[n] - Yd = C(X[n] - Xd) + D(U[n] - Ud),$$

$$\text{де } A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0,1 & -0,5 & -0,4 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}; C = [0 \quad 0 \quad 1]; D = 0;$$

$$Xd = [0,4 \quad -0,4 \quad 0,2]^T; Ud = 2; T = 0,4.$$

Бажане розташування полюсів $P = [-0,3 \quad -0,5 \quad -0,4]$.

Наведемо приклад програми для розв'язку цієї задачі з назвою "Ackerman_Discr".

```
function Ackerman_Discr
A = [0 1 0; 0 0 1; -0.1 -0.5 -0.4]; % Введення матриці A
B = [0; 0; 2]; % Введення матриці B
C = [0 0 1]; % Введення матриці C
disp('Бажані полюси системи P:')
P = [-0.3 -0.5 -0.4] % Введення бажаних полюсів системи
disp('Вектор початкових умов x0:')
x0 = [0.1 -0.1 0.2]' % Завдання вектора початкових умов
disp('Вектор заданих значень відхилень xd:')
xd = [-0.4 0.4 0.2]' % Завдання вектора заданих значень відхилень
disp('Задане значення відхилення керування Ud:')
ud = 2 % Задання значення відхилення керування
disp('час спостереження tf:')
tf = 20 % Завдання часу спостереження
```

```
disp('довжина кроку dt:')
T = 0.4 % Завдання довжини кроку
disp('кількість кроків n:')
n = tf/T
disp('Матриця зворотнього з язку K:')
K = acker(A,B,P) % Визначення матриці зворотнього зв'язку за допомогою формули Аккермана
disp('Модель системи у просторі стану:')
w1 = ss(A,B,C,0,T) % Формування моделі системи у просторі стану
pzmap(w1), grid; % Побудова нулів та полюсів системи
xlabel('дійсна вісь'); ylabel('уявна вісь');
title('Карта нулів та полюсів системи до синтезу')
w2 = ss(A-B*K,B,C,0,T); % Формування моделі системи з оптимальною ланкою зворотного зв'язку
figure(2); pzmap(w2), grid; % Побудова нулів та полюсів системи з оптимальною ланкою зворотного зв'язку
xlabel('дійсна вісь'); ylabel('уявна вісь');
title('Карта нулів та полюсів системи після синтезу')
% Визначення розмірності задачі
SA = size(A); SA = SA(1);
% Формування векторів x та u
x = zeros(SA,n); u = zeros(1,n-1);
% Формування початкового стану
x(:,1) = x0;
% Знаходження оптимального керування u та змінних стану x
for i=1:n-1,
    u(:,i) = ud-K*(x(:,i)-xd);
    x(:,i+1) = (A*(x(:,i)-xd)+B*(u(:,i)-ud))*T+x(:,i);
end
% Подова динаміки змінних стану
figure(3)
stem(0:T:tf-T,x'), grid
```

```

title('Динаміка руху змінних стану');
xlabel('Час, t'); ylabel('Змінні стану, x(t)');
legend('x_1', 'x_2', 'x_3')
% Побудова вектора керування
figure(4)
stem(T:T:tf-T,u), grid
title('Динаміка зміни керування'); xlabel('Час,
t'); ylabel('Керування, U(t)'); legend('u')

```

Результат виконання програми

Бажані полюси системи P:

```

P =
    -0.3000    -0.5000    -0.4000

```

Вектор початкових умов x0:

```

x0 =
    0.1000
   -0.1000
    0.2000

```

Вектор заданих значень відхилень xd:

```

xd =
   -0.4000
    0.4000
    0.2000

```

Задане значення відхилення керування Ud:

```

ud =
    2

```

Час спостереження tf:

```

tf =
    20

```

Довжина кроку dt:

```

T =
    0.4000

```

Кількість кроків n:

```

n =
    50

```

Матриця зворотнього зв'язку K:

```

K =
   -0.0200   -0.0150    0.4000

```

Модель системи у просторі стану:

```

a =
      x1      x2      x3
x1      0      1      0
x2      0      0      1
x3  -0.1  -0.5  -0.4

```

```

b =
      u1
x1      0
x2      0
x3      2

```

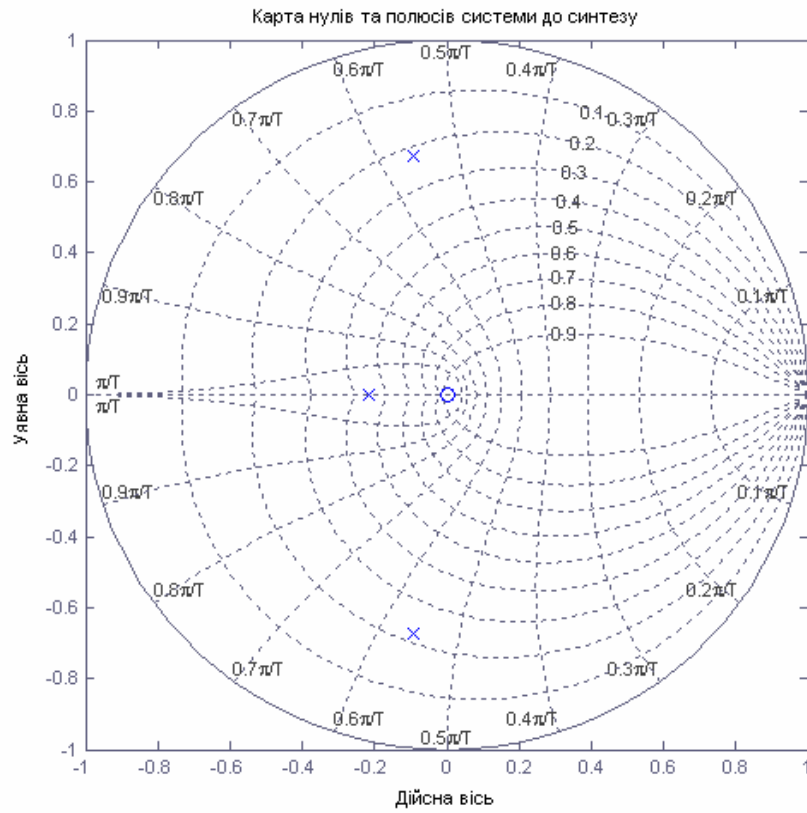
```

c =
      x1      x2      x3
y1      0      0      1

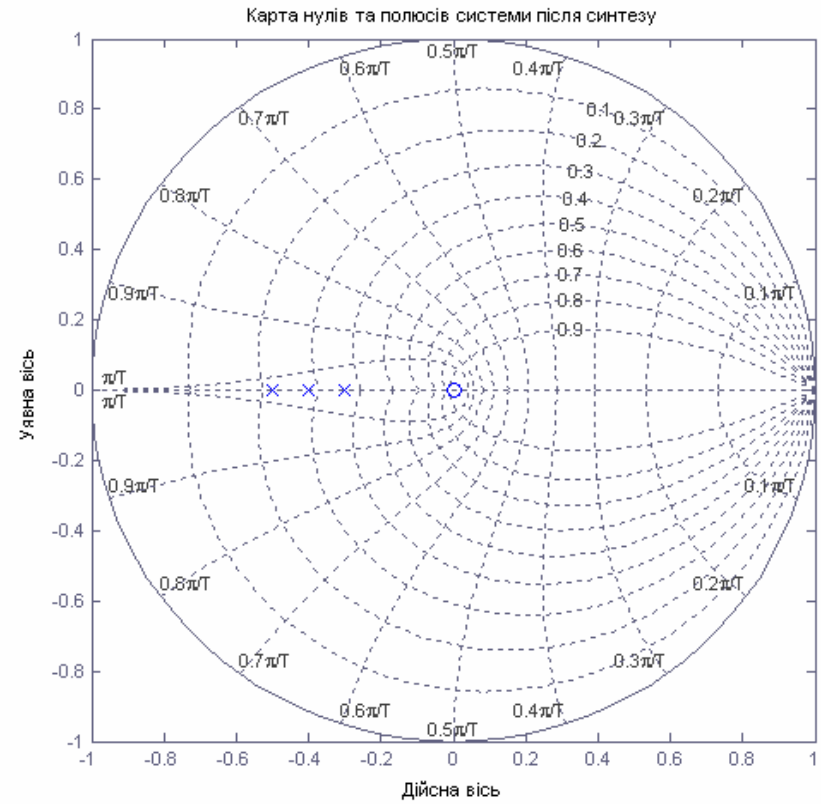
```

$$d = \begin{matrix} u1 \\ y1 \end{matrix} \begin{matrix} 0 \\ 0 \end{matrix}$$

Sampling time: 0.4
Discrete-time model.

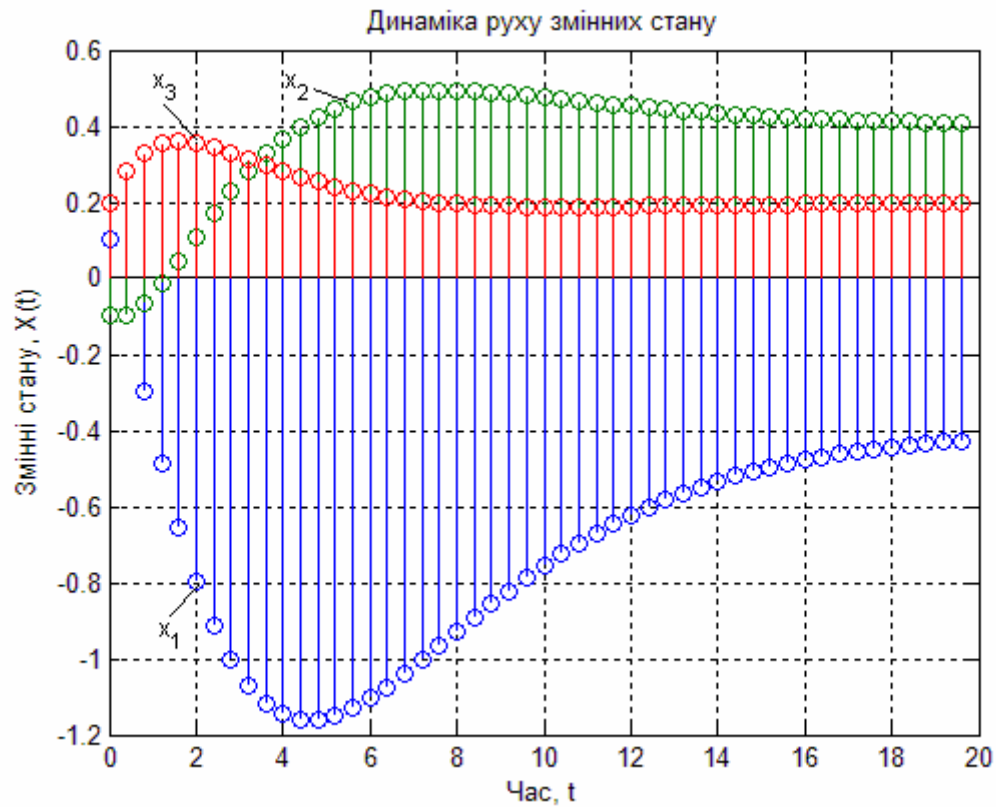


а)

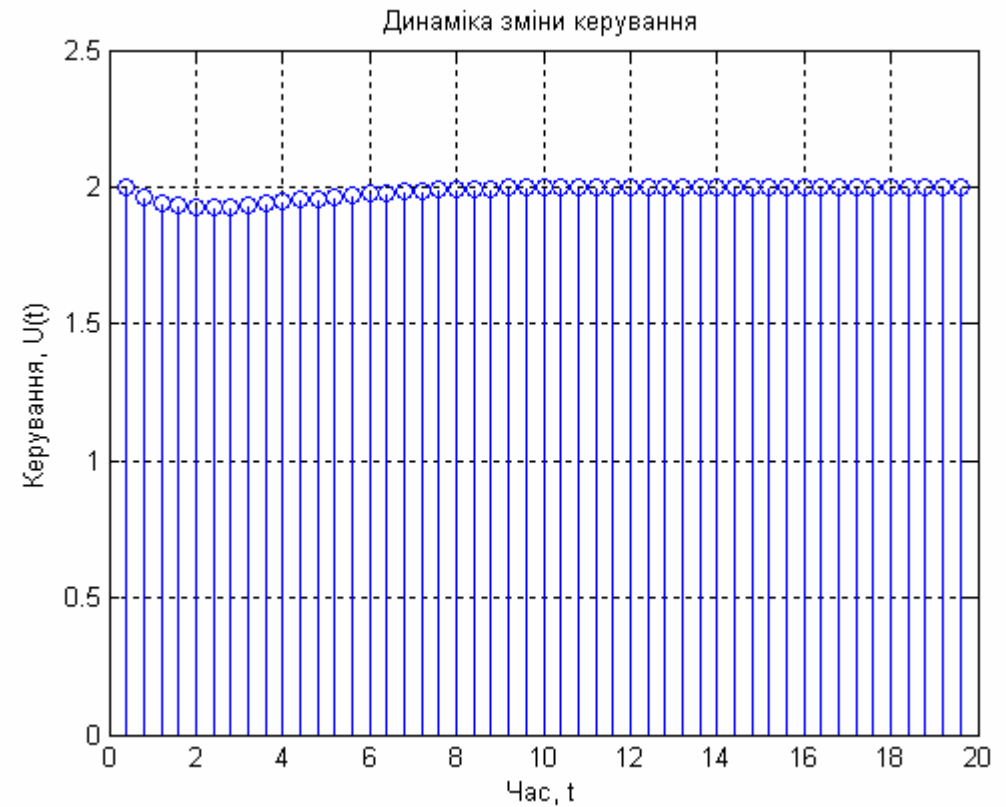


б)

Рис. 4.7. Карта нулів та полюсів системи до а) і після синтезу б)



а)



б)

Рис. 4.8. Динаміка руху змінних стану а) і керування б)

Досліджували якість дискретного регулятора для наступних значеннях параметрів: $X[0] = [0,1 \quad -0,1 \quad 0,2]^T$; $X_d = [0,4 \quad -0,4 \quad 0,2]^T$; $U_d = 2$; $T = 0,4$ та бажаного розташування полюсів $P = [-0,3 \quad -0,5 \quad -0,4]$. Коефіцієнт підсилення K визначається з формули Аккермана.

5. Нелінійні системи

Інструментальний пакет **Signal Constraint** (обмеження сигналу), надає в розпорядження користувача графічний інтерфейс для настройки параметрів динамічних об'єктів, що забезпечують бажану якість перехідних процесів. Як засіб для досягнення вказаної мети приймається оптимізаційний підхід, що забезпечує мінімізацію функції штрафу за порушення динамічних обмежень. За допомогою даного інструменту, можна настроювати параметри нелінійної Simulink-моделі, в якій може бути заявлена будь-яка кількість змінних, включаючи скаляри, вектори і матриці. Особливу значущість має та обставина, що в процесі настройки можуть враховуватися невизначеності параметричного типу математичної моделі, що дозволяє синтезувати робастні закони керування.

Завдання динамічних обмежень здійснюється у візуальному режимі. На базі цих обмежень **Signal Constraint** автоматично генерує задачу кінцево вимірної оптимізації так, щоб точка екстремуму в просторі параметрів, що настроюються, відповідала виконанню всіх вимог, що пред'являються до якості процесу. Ця задача розв'язується із залученням спеціалізованої процедури квадратичного програмування з пакету Optimization Toolbox. Хід оптимізації контролюється на екрані за допомогою відображення графіка контрольованого процесу і поточних значень функції, що мінімізується. Після закінчення процесу його результат фіксується в робочому просторі MATLAB.

Постановка задачі про настройку параметрів

Розглянемо дві типові задачі, для вирішення яких раціонально використати даний інструментальний засіб.

1. Задача пошуку керування з фіксованою структурою, забезпечуючого виконання деяких вимог, що пред'являються до процесів в замкненій системі. Припустимо, що задана математична модель об'єкту у вигляді системи нелінійних диференціальних (або різницевих) рівнянь:

$$\frac{dX}{dt} = f(t, X, U, F), \quad (5.1)$$

де $t \in [0, T]$ – час; X – вектор стану об'єкту; U – вектор управління; F – збурення.

Разом з математичною моделлю об'єкту управління, введемо в розгляд рівняння регулятора у вигляді:

$$U = W(p, H)X, \quad (5.2)$$

де $W(p)$ – передатна матриця регулятора. Припустимо, що структура передатної матриці $W(p)$ задана, причому у складі вказаної структури виділений вектор H , компонентами якого є параметри регулятора, що підлягають визначенню.

Задамо деяку припустиму множину Ω_X , до якої повинен належати рух замкненої системи у будь-який момент часу. Очевидно, що за інших рівних умов, факт приналежності цій множині однозначно визначається вибором вектора параметрів H .

В даному випадку задача полягатиме в пошуку вектора H з деякої припустимої множини Ω_X , який забезпечує виконання умови

$$X(t, H) \in \Omega_X, \quad \forall t \in [0, T]. \quad (5.3)$$

2. Задача ідентифікації полягає в пошуку значень параметрів деякої динамічної системи, яка моделює рух реальної системи. Припустимо, що структура моделі задана з точністю до значення деяких параметрів

$$f(p, t, X, H) = 0, \quad (5.4)$$

де p – змінна Лапласа; t – час; X – вектор стану об'єкту; H – вектор параметрів, що підлягають визначенню. При постановці задачі ідентифікації, вибір вказаних параметрів повинен забезпечити максимальне наближення деякого руху моделі до відповідного руху реальної системи, представленого відомою функцією часу. Іншими словами, задача полягає в пошуку вектора H з деякої припустимої множини Ω_X , який забезпечує виконання умови:

$$X(t, H) \in \Omega_X, \quad \forall t \in [0, T],$$

де Ω_X – в околі модельованого руху реальної системи.

Дві вказані задачі можуть бути з'єднані загальною постановкою задачі про настройку параметрів вигляду

$$\gamma [X(t, H)] \rightarrow \min_{\gamma \in \Omega_X}, \quad (5.5)$$

де H – вектор параметрів, що настроюються; $X(t, H)$ – векторна функція, що представляє рух динамічної системи; Ω_X – безліч допустимих параметрів;

γ – скалярна величина, що характеризує міру порушення приналежності руху $X(t, H)$ бажаній множині Ω_X .

Signal Constraint представляє поставлену оптимізаційну задачу у вигляді:

$$\begin{cases} \min_{H, \gamma} \gamma, \\ g(H) - W\gamma \leq 0, \\ \underline{H} \leq H \leq \bar{H}, \end{cases} \quad (5.6)$$

де \underline{H}, \bar{H} – вектори верхніх і нижніх меж змінних, що настроюються, відповідно, вектор $g(H)$ формується з величин, що визначають порушення обмежень, в множині Ω_X ; вектор W – вектор вагів для вказаних величин. Простіше кажучи, **Signal Constraint** намагається мінімізувати максимальне (зважене) відхилення вхідного сигналу від заданого коридору.

При цьому **Signal Constraint** розбиває відрізок $[0, T]$ інтегрування із заданим кроком дискретизації. Далі, на кожному кроці оптимізації в кожній вузловій точці отриманої сітки формується компоненти вектора $g(H)$. Для пошуку екстремуму **Signal Constraint** використовує процедури з Optimization Toolbox.

3. Схема застосування блоку

1. Створити Simulink-модель.
2. Під'єднати **Signal Constraint** блок (або блоки) до сигналу (сигналам), на який накладатимуться обмеження.
3. Задати в робочому просторі MATLAB початкові значення параметрів, що настроюються.
4. Відкрити **Signal Constraint** блок в Simulink-моделі. При цьому відкриється вікно, в якому в візуальному режимі формуються припустимі "коридори" вигляду $x_{i-}(t) \leq x_i(t) \leq x_{i+}(t)$ для кожного контрольованого сигналу $x_i(t)$.
5. Сформувати вказані коридори шляхом розтягування, переміщення, розбиття або об'єднання граничних ламаних, використовуючи команди меню Edit, кнопки миші або "гарячі клавіші". Припустимі коридори можна зберігати і завантажувати з файлу.
6. Відкрити діалог Optimization Parameters за допомогою команди Parameters ... з меню Optimization. Вказати інтервал дискретизації (розробники рекомендують вибирати його в інтервалі між 1 і 2 відсотками від загального часу інтегрування). Вказати імена параметрів, що настроюються, розділяючи їх пробілами або комами. Там же вказати точність завдання змінних і обмежень, що настроюються.
7. Якщо є необхідність, відкрити діалог Uncertain Variables, вибравши команду Uncertainty меню Optimization для завдання припустимих меж параметрів невизначеності моделі. Вказані параметри повинні ініціалізувати в робочому просторі MATLAB.
8. Якщо є необхідність, зберегти всі установки блоку з використанням команди Save... меню File. Збережені установки можуть використовуватися надалі: їх завантаження здійснюється за допомогою команди Load... меню File.
9. Натискувати кнопку Start або вибрати команду Start з меню Optimization для запуску процесу оптимізації.
10. Спостерігати за розвитком процесу на екрані.

11. Після закінчення процесу оптимізації отримані значення параметрів фіксуються в робочому просторі MATLAB.

Нижче наведено основні настройки блоку Signal Constraint

Бажаний відгук

При оптимізації, бажаний відгук системи є альтернатива методу позиціонування обмеження для конкретизації бажаного відгуку системи, наприклад час зростання, час усталення, час промаху.

Щоб визначити характеристику відгуків кроків, вибирають **Goals** → **Desired Response** у вікні **Signal Constraint** або клікають правою кнопкою миші на білому просторі вікна і вибирають **Desired Response** з контекстного меню. Це показує діалогове вікно рис. 5.1. Вибирають **Specify step response characteristics**, щоб визначити характеристики відгуків кроків.

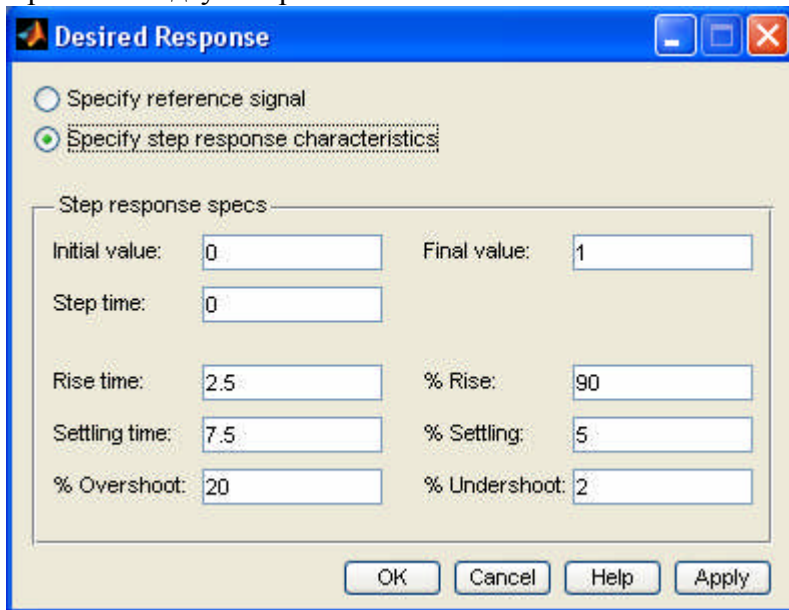


Рис. 5.1. Визначення характеристик відгуків кроків

Вибір відгуку кроку для детальної конкретизації:

- **Initial value: (Початкове значення)** – початковий рівень перед тим, як крок відбуватиметься.
- **Step time: (Час кроку)** – крок за часом.
- **Final value: (Завершальне значення)** – вхідний рівень після того, як крок відбувається.

Решта конкретизує відповіді характеристики сигналу. Кожна характеристика пояснюється нижче (рис. 5.2).

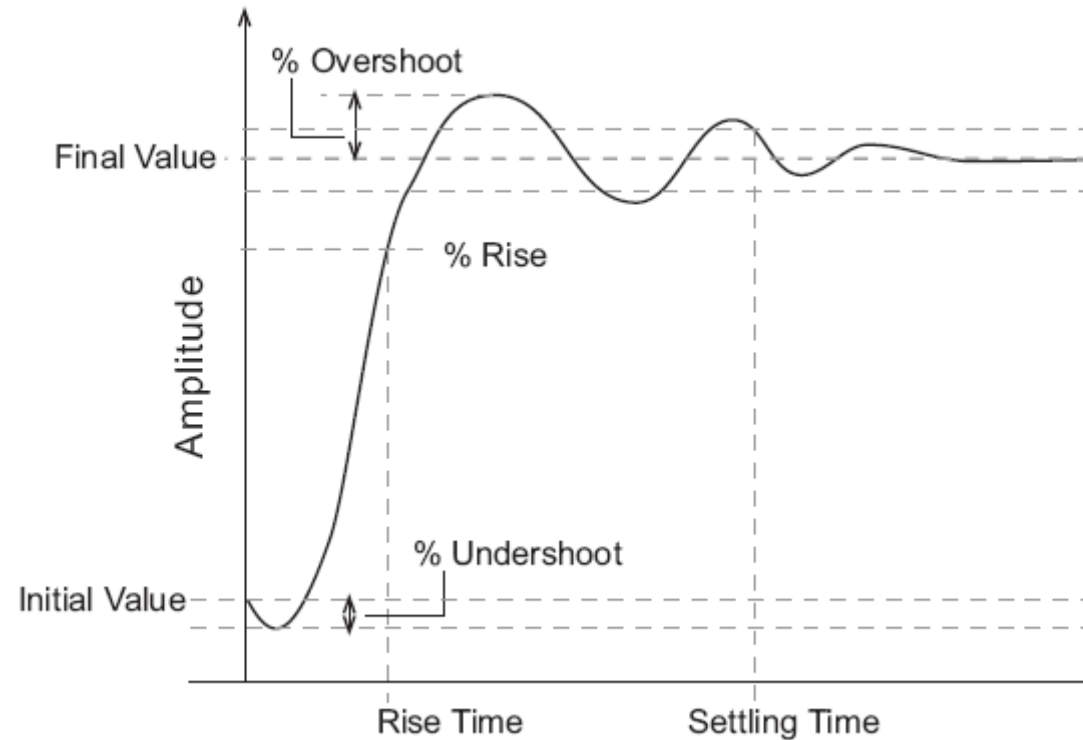


Рис. 5.2. Характеристики сигналу

- **Rise time (Час зростання)** – час, прийнятий, щоб сигнал відгуку досяг точно встановлений відсоток діапазону кроку. Діапазон кроку – різниця між кінцевим і початковим значенням.

- **% Rise (% Зростання)** – відсоток, використаний за час зростання.
- **Settling time (Час регулювання)** – час, висунутий до сигналу відгуку за який сигнал точно потрапляє в встановлену область поблизу кінцевого значення. Ця область регулювання визначена, як значення кінцевого кроку плюс або мінус точно встановлений відсоток кінцевого значення.
- **% Settling (% Регулювання)** – відсоток, використаний за час регулювання.
- **% Overshoot (% Перерегулювання)** – відсоток, який сигнал відгуку може перевищити кінцеве значення. Ця кількість точно встановлена, як відсоток діапазону кроку.
- **% Undershoot (% Відхилення від номіналу)** – Кількість, коли сигнал відгуку може недостатньо зростати з початкового значення. Ця кількість точно встановлена, як відсоток діапазону кроку.

Стеження довідкових Сигналів

Користувач може конкретизувати бажаний відгук, як ідеальний або довідкову *траєкторію*. Спершу, треба вибрати **Track reference signal** знизу Signal Constraint реакції зв'язку. Потім, креслять план довідкового сигналу в межах вісей.

Визначення довідкового сигналу

Креслення плану довідкового сигналу відбувається вибором **Goals** → **Desired Response** в **Signal Constraint** реакції зв'язку або клікаючи правою кнопкою в білому просторі вікна, вибираючи **Desired Response** з контекстного меню. Виберіть radio змінну кнопку **Specify reference signal**, щоб виділити установку довідкового сигналу, як показано на рис. 5.3.

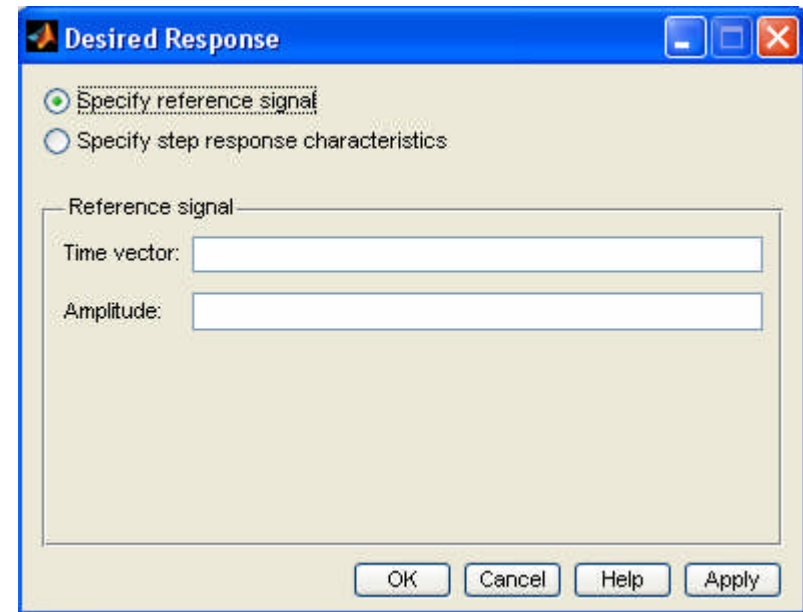


Рис. 5.3. Визначення довідкового сигналу

Довідковий сигнал визначають вхідними векторами, або змінними від робочої області, протягом часу і амплітуди сигналу, а потім, натискають **OK**.

Визначення параметрів, що настроюються в моделі

Перед запуском оптимізації, потрібно визначити параметри системи, які є настроюваними. Настроюючи ці параметри, Simulink Response Optimization робить сигнал відгуку відповідним накладеним обмеженням.

Simulink Response Optimization оптимізує сигнал відгуку моделі, змінюючи параметри таким чином, щоб сигнал відгуку лежав в межах сегментів, меж обмежень, або близько довідковому сигналу. Конкретизація параметрів настроювання відбувається вибором **Optimization** → **Tuned Parameters** у вікні Signal Constraint.

Примітка. Коли є більш ніж один блок Signal Constraint в моделі, потрібно конкретизувати настроєні параметри тільки в одному вікні, ці настроювальні параметри звернуться до всіх інших сигналів в межах моделі.

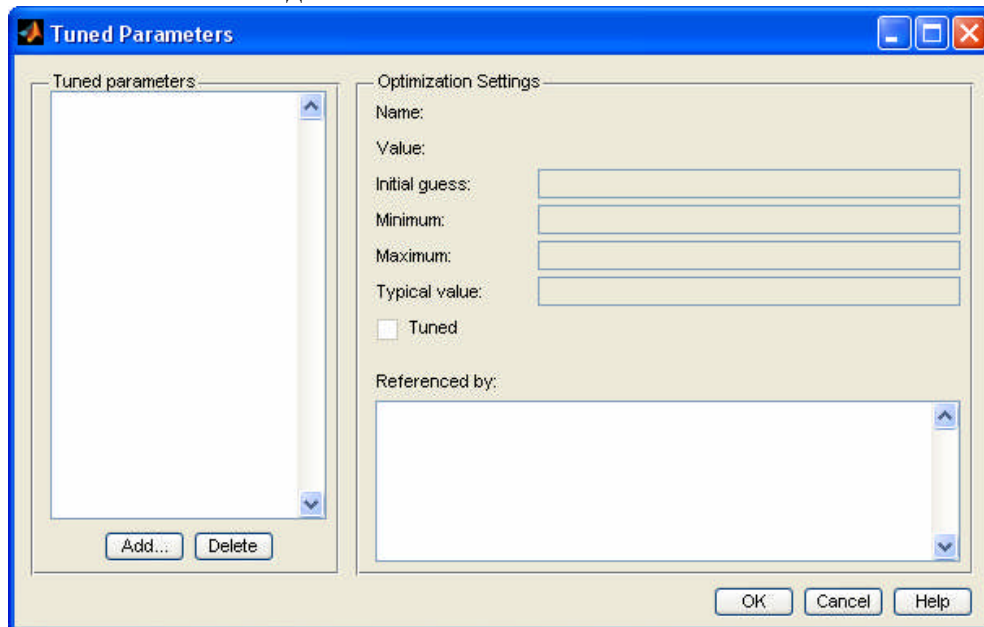


Рис. 5.4. Конкретизація параметрів настроювання

Після введення настроєних параметрів оптимізації, треба натиснути **ОК**, щоб зберегти зміни і вийти з діалогового вікна.

Включення невизначеності в значеннях параметрів

Simulink Response Optimization дозволяє об'єднувати невизначеність в проекті двома різними способами:

- **Пасивний метод:** Оптимізує сигнали, засновані на значеннях тільки номінальних параметрів.

- **Активний метод:** Оптимізує сигнали, засновані на номінальних значеннях параметрів та невизначених значеннях параметрів. Цей метод – більше трудомісткий.

Щоб конкретизувати невизначеність в параметрах, вибирають **Optimization** → **Uncertain Parameters** від вікна Signal Constraint як показано на рис. 5.5.

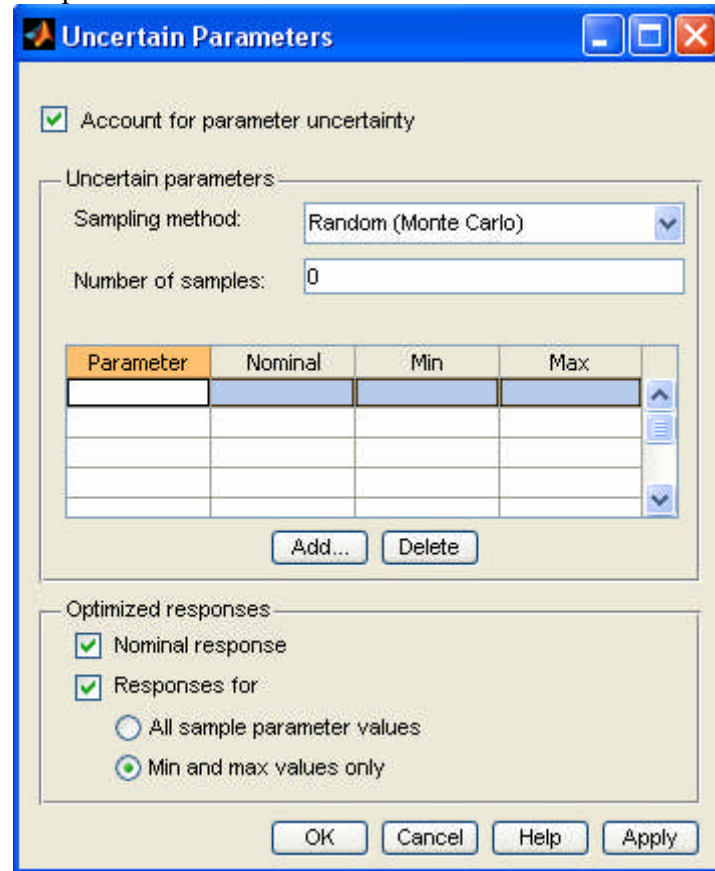


Рис. 5.5. Конкретизація невизначеності в параметрах

Додавання невизначених параметрів

Щоб додати новий невизначений параметр до **Uncertain parameters** складають список, і натискають кнопку **Add**. Це

призводить до показу діалогового вікна вибраних параметрів, лістинг, де всі зразкові параметри на теперішній час, доступні в робочій області MATLAB. (Якщо у параметри вже внесено до списку або настроєні параметри, або невизначені параметри, це не з'явиться в діалоговому вікні вибраних параметрів).

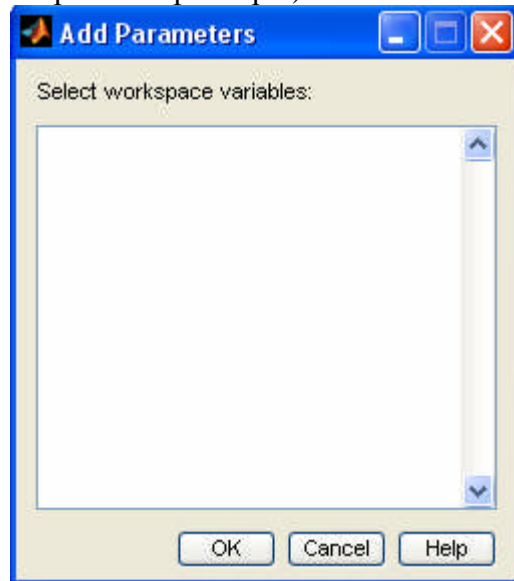


Рис. 5.6. Додавання невизначених параметрів

До параметрів, яким додається невизначеність, треба натиснути **OK**, щоб додати їх до списку невизначених параметрів. Щоб видалити параметр **Uncertain parameters**, треба натиснути **Delete**.

Заміна описів невизначених параметрів

Є два методи, виконання вибірки яких дозволяє розкрити невизначені параметри. Обидва залучають використання декількох значень типових параметрів в діапазоні невизначеності.

- **Random (Monte Carlo) (Випадковий (Монте-Карло))** оптимізує в деякій мірі, випадкові значення параметрів в межах діапазону невизначеності.

Коли більш ніж один параметр містить невизначеність, випадкові параметри комбінації вибрані в межах гіпер-прямокутника, визначеного мінімальними і максимальними значеннями всіх параметрів. Наприклад, для випадку двох невизначених параметрів a і b , з значеннями, що розташованими від 0 до 3 та з 1 до 2,5 відповідно, типові значення, представлені чорними точками, розсіпані безладно в межах прямокутника, як показано на рис. 5.7.

- **Grid (Сітка)** оптимізує в деякій мірі конкретизовані значення параметрів в межах діапазону невизначеності.

Коли більш ніж один параметр містить невизначеність, типові значення формують сітку комбінацій параметрів. Наприклад, у разі двох невизначених параметрів a і b , із значеннями [0 1 2 3] та [1 1,5 2 2,5], що представлені чорними точками, формують сітку комбінацій параметрів, показані зліва на рис. 5.7.

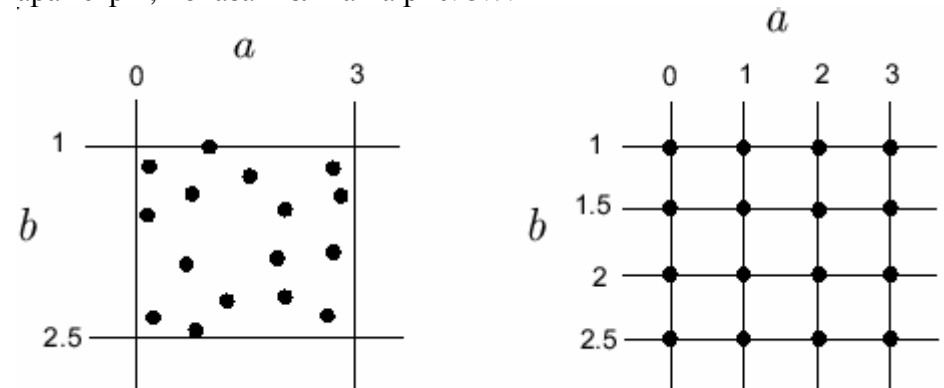


Рис. 5.7. Опис невизначених параметрів

Щоб збільшити швидкість обчислення, можна не використовувати весь зразок значення параметрів оптимізації. Щоб включати номінальний параметр значення в оптимізацію, вибирають **Nominal response**, а значення параметрів окрім номінального значення в оптимізації виберіть **Response for**, а потім виберіть будь-який **All simple parameter values** або **Min and max values only**. Цей вибір включає або комбінації всіх значень типових параметрів, або всі комбінації мінімальних і максимальних значень параметрів, відповідно. Тільки оптимізовані відгуки використовуються, для

регулювання настроювальних параметрів. Відгуки, засновані на інших значеннях типових параметрів, можливо, все ще кресляться на плані у вікні Signal Constraint.

Настроювання результатів оптимізації

Окремі опції оптимізації можуть бути змінені користувачем. Ці опції включають алгоритм оптимізації і похибки алгоритмів, що використовуються. Щоб встановити опції оптимізації, виберіть **Optimization** → **Optimization Options** в вікні **Signal Constraint**. Це відкриває діалогове вікно опцій рис. 5.8.

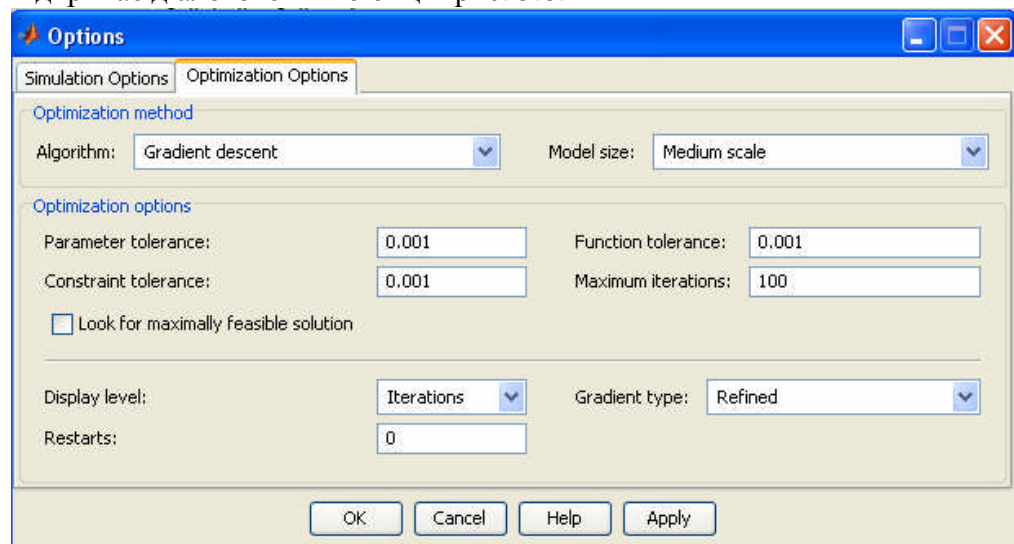
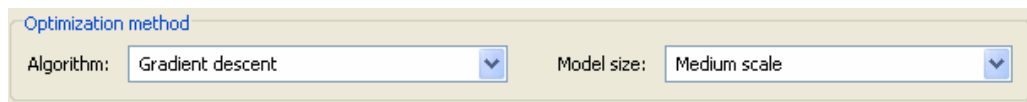


Рис. 5.8. Діалогове вікно опцій оптимізації

Примітка. Якщо оптимізація не дала результатів, то треба змінити (вдосконалити) опції.

Вибір методів оптимізації

Для змінення алгоритму так і для розмірності алгоритму використовується **Optimization method**.



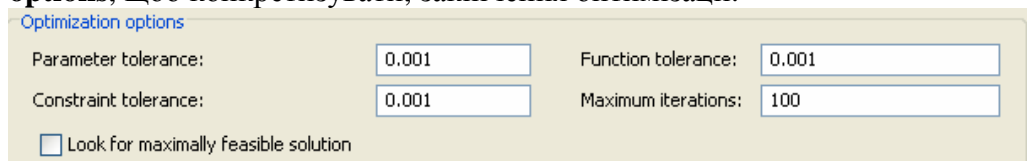
Для вибору алгоритму оптимізації існує три варіанти:

- **Gradient descent (градієнтний спуск)** – використовує функції fmincon з Optimization Toolbox, градієнтні методи великої та середньої розмірності.
- **Pattern search (Зразковий пошук)** – використовує функції з Genetic Algorithm and Direct Search Toolbox, прямий пошук та генетичні алгоритми.
- **Simplex search (Симплексний пошук)** – використовує функції fminsearch з Optimization Toolbox. Симплексний пошук є найкорисніший для простих задач оптимізації і для задач, що мають розриви похідних.

За умовчанням, параметр **Model size** визначений як **Medium scale (Середній масштаб)**. Коли модель дуже велика і **Gradient descent** вибраний як алгоритм оптимізації, користувач можете змінити **Model size** на **Large scale (Великий масштаб)** для збільшення швидкості обчислення.

Вибір опцій закінчення оптимізації

Користувач може використовувати панель **Optimization options**, щоб конкретизувати, закінчення оптимізації.



- **Parameter tolerance (Похибка обчислення параметра)** – при використуванні симплексного пошуку, алгоритм оптимізації закінчується, коли значення параметрів стають менше ніж зазначені.
- **Constraint tolerance (Похибка обчислення обмежень)** – цей параметр показує максимальне порушення обмежень і дозволяє успішне проведення оптимізації.

- **Function tolerance (похибка обчислення функції)** – оптимізація закінчується, коли наступне значення функції стає меншим ніж значення.
- **Maximum iterations (Максимальна кількість ітерацій)** – максимальне число дозволених ітерацій.
- **Look for maximally feasible solution (Пошук максимально припустимого розв'язку)** – якщо вибрати цей параметр, оптимізація продовжує після того, як було знайдено початковий розв'язок, поки не буде знайдено максимально припустимий, оптимальний розв'язок. Коли ця опція невібрана, оптимізація закінчується, як тільки буде знайдений розв'язок, який задовольняє обмеженням.

За допомогою варіювання цих параметрів, користувач може продовжувати оптимізацію для більш точного вирішення поставленої задачі.

Вибір додаткових опцій оптимізації

Знизу панелі **Optimization options** є група додаткових опцій оптимізації.

Display level: Gradient type:
 Restarts:

Додаткові опції оптимізації включають:

- **Display level (Рівень показу)** – ця опція конкретизує форму висновку, який з'являється у вікні Optimization Progress. **Iterations** – це відображення інформації після кожної ітерації; **Notify** – закриває відображення висновку; **Termination** – відображення тільки заключного висновку.
- **Restarts (Повторні запуски)** – в деяких випадках оптимізації Гесіан не може бути оновленим і оптимізація не зводить в жодну

точку. В цих випадках іноді корисно знову почати оптимізацію після зупинки, використовуючи кінцеву точку попередньої оптимізації як початкову точку для наступного пошуку, користувач може вказати кількість разів повторного запуску оптимізації.

- **Gradient type (Тип градієнта)** – при використуванні **Gradient descent** як алгоритму оптимізації, Simulink Response Optimization обчислює градієнт, методом кінцевих різниць. По замовчуванню метод для обчислення градієнту **Basic**. **Refined** – метод пропонує стійкий і менш вимогливий метод обчислення градієнту, ніж **Basic**, хоча іноді більш довше працює з певними моделями, як наприклад моделі SimPowerSystems. Якщо оптимізація не знаходить розв'язок, перед зміною solvers або межі параметра, треба змінити тип градієнта на вдосконаленого **Refined**.

Знизу панелі **Optimization options** є група додаткових опцій оптимізації для **Pattern search**.

Display level: Search method:
 Restarts: Limit search to first iterations

Розглянемо додаткові опції оптимізації, що стосуються лише **Pattern search**:

- **Search method (Метод пошуку)** – при використанні **Pattern search** як алгоритм оптимізації, Simulink Response Optimization використовується один з методів зразкового пошуку: **None** – відсутній; **Positive basis Np1** – використовується в шаблоні методу прямого пошуку, мінімальний і додатний базис з $N+1$ векторів; **Positive basis 2N** – використовується в шаблоні методу прямого пошуку, максимальний додатний базис з $2N$ векторів; **Genetic algorithm** – для пошуку застосовується генетичний алгоритм; **Latin hypercube** – застосовується один з під видів методу Монте-Карло, тобто використовується схема Монте-Карло з деякими обмеженнями, що визначають спосіб за допомогою якого розподіляються вхідні керовані змінні; **Nelder-Mead** – вибирається метод Нелдера-Міда.

- **Limit search to first (Пошук границі в початку)** – додатне ціле число, яке визначає номер ітерації зразкового пошуку, що виконує **Latin hypercube**.

Установка опцій моделювання

Щоб оптимізувати сигнали відповідної моделі використовують, Simulink Response Optimization моделювання рухів моделі. Користувач може встановити опції для даного моделювання, вибрав **Optimization** → **Simulation Options** в вікні **Signal Constraint**. Це відкриває діалогове вікно опцій (рис. 5.9).

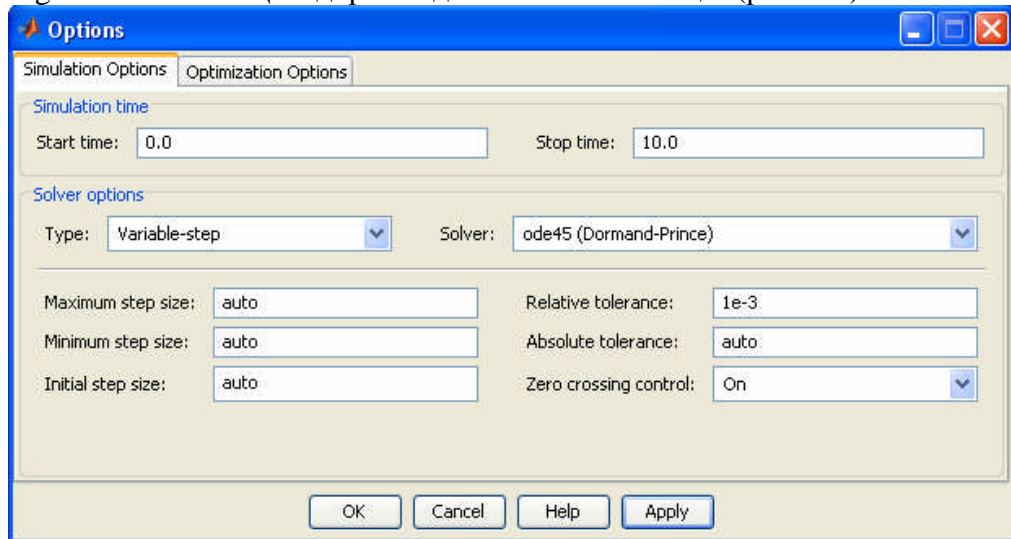
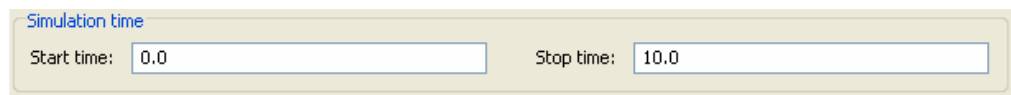


Рис. 5.9. Діалогове вікно опцій моделювання

Вибір часу моделювання

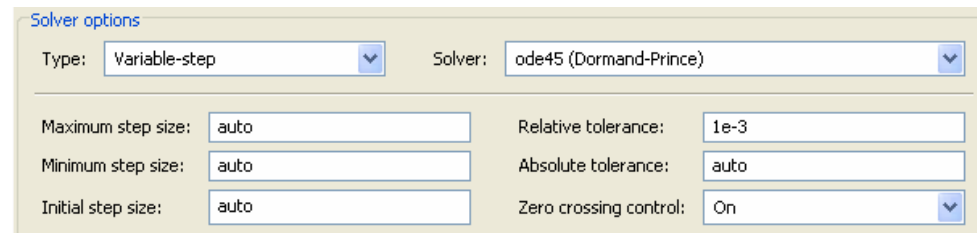


За умовчанням, тривалість **Start time (Час розгону)** і **Stop time (Час зупинки)** автоматично встановлені, як зразковий запуск і зупинка. Щоб конкретизувати альтернативний запуск і зупинку для

проекту оптимізації, потрібно їх в **Simulation time (Час моделювання)**.

Примітка. Час зупинки **Inf** викликає в Simulink Response Optimization ввести нескінченне керування, Simulink Response Optimization автоматично замінює це значення з найбільшою тимчасовою змінною в обмеженнях.

Вибір солвера



При запуску моделювання, Simulink розв'язує динамічну систему використовуючи один з solvers. Користувач може конкретизувати окремі опції solver, згідно з **Solver options** в діалоговому вікні опцій. Вид solver може бути із змінним кроком або з фіксованим кроком. Solvers із змінним кроком утримують похибку в межах конкретизованої за допомогою регулювання stepsize (розмір кроку). Solvers з фіксованим кроком використовує сталий розмір кроку.

Солвер із змінним кроком

Коли користувач вибирає солвер із змінним кроком, може бути вибрана будь-яка з процедура як Solver:

- **discrete (no continuous states)** – метод дискретний (не неперервний стан);
- **ode45 (Dormand-Prince)** – метод Дормана-Пренса;
- **ode23 (Bogacki-Shampine)** – метод Богадцького-Шампена;
- **ode113 (Adams)** – метод Адамса;
- **ode15s (stiff/NDF)** – жорсткий метод NDF;
- **ode23s (stiff/Mod. Rosenbrock)** – жорсткий метод Розенброка;

- **ode23t (Mod. stiff/Trapezoidal)** – жорсткий метод трапецій;
- **ode23tb (stiff/TR-BDF2)** – жорсткий метод TR-BDF2.

Додаткові опції солвера із змінним кроком

- **Maximum step size (Максимальний розмір кроку)** – найбільший розмір кроку, що може використовуватися протягом моделювання.
- **Minimum step size (Мінімальний розмір кроку)** – найменший розмір кроку, що може використовуватися протягом моделювання.
- **Initial step size (Початковий розмір кроку)** – початковий розмір кроку, щоб почати моделювання.
- **Relative tolerance (Відносна похибка)** – максимально припустима відносна похибка на будь-якому кроці моделювання.
- **Absolute tolerance (Абсолютна похибка)** – максимально припустима абсолютна похибка на будь-якому кроці моделювання.
- **Zero crossing control (Контроль нульового перетину)** – використовується для визначення різких змін у неперервних сигналах. Ця функція грає важливу роль: для керування скачками стану, для точного інтегрування неперервних сигналів.

За умовчанням, Simulink автоматично вибирає значення для цих опцій, користувач може встановити власні значення, для підвищення ефективності моделювання.

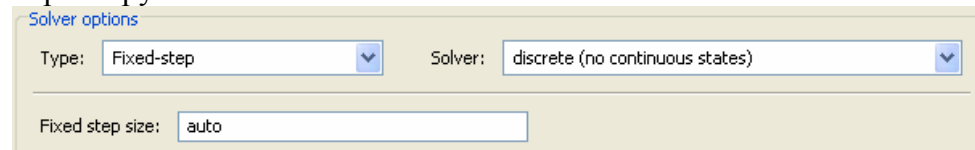
Солвер з фіксованим кроком

Коли користувач вибирає солвер з сталим кроком, може бути вибрана будь-яка з процедури як Solver:

- **discrete (no continuous states)** – метод дискретний (не неперервний стан);
- **ode5 (Dormand-Prince)** – метод Дормана-Пренса (п'ятого порядку);
- **ode4 (Runge-Kutta)** – метод Рунге-Кутта (четвертого порядку);

- **ode23 (Bogacki-Shampine)** – метод Богадцького-Шампена (третього порядку);
- **ode2 (Heun)** – метод Хойне (другого порядку);
- **ode1 (Euler)** – метод Ейлера (першого порядку).

Коли користувач вибирає solver з сталим кроком, то можливе встановлення фіксованого розміру кроку, що визначає stepsize і використовується солвером протягом моделювання, по замовчуванню Simulink автоматично вибирає значення цього параметру.



Приклади програм

Розглянемо нелінійну систему, що описується рівняннями:

$$\begin{cases} \frac{dx_1}{dt} = -x_1 - x_1^2 + u_1 \\ \frac{dx_2}{dt} = x_1^2 - x_2 - 2x_2^{1/2} + u_2 \end{cases}$$

з початковими умовами $x_1(0) = 0$; $x_2(0) = 0$, починаючи з точки

$$K^{(0)} = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}.$$

Знайти, матричну ланку зворотного зв'язку, побудувати перехідні процеси та оптимальні керування.

Для розв'язання поставленої задачі, спочатку ініціалізуємо в командному вікні MATLAB початкове значення ланки зворотного зв'язку

```
>> K = [-1 1; -1 1];
```

Створимо SIMULINK – модель (блок-схему) рис. 5.10.

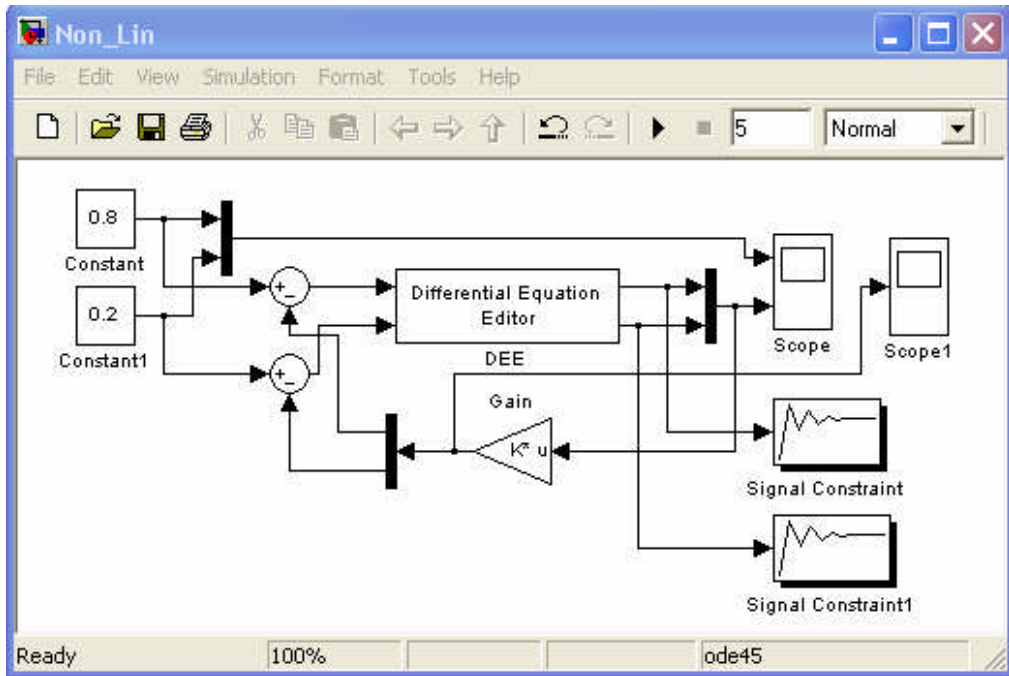


Рис. 5.10. SIMULINK – модель

Виконаємо налаштування моделі об'єкта на базі Differential Equation Editor (DEE) – редактор диференціальних рівнянь, для його виклику необхідно ввести команду в командному вікні MATLAB **>> dee** та виконати налаштування блока, як показано на рис. 5.11.

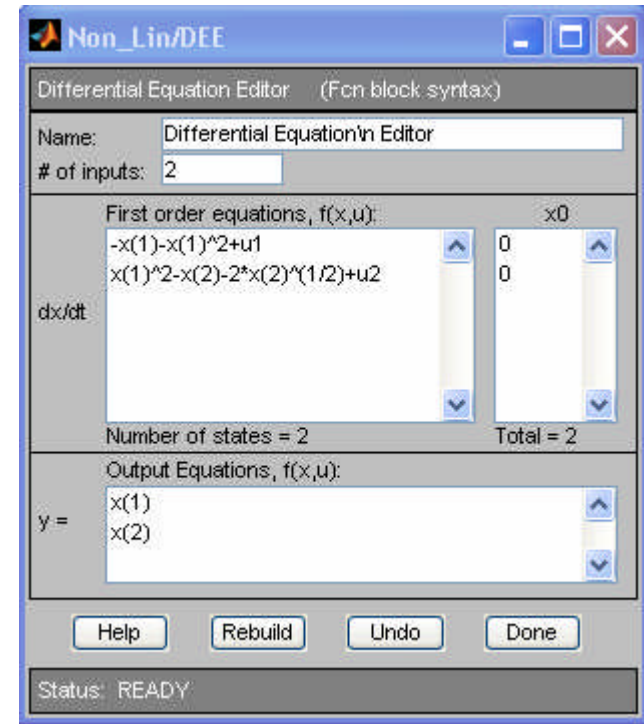


Рис. 5.11. Редактор диференціальних рівнянь

Для ініціалізації завдання системі регулювання використаємо блоки констант Constant та Constant1, які знаходяться в бібліотеці SIMULINK Sources і виконаємо відповідні налаштування як показано на рис. 5.12.

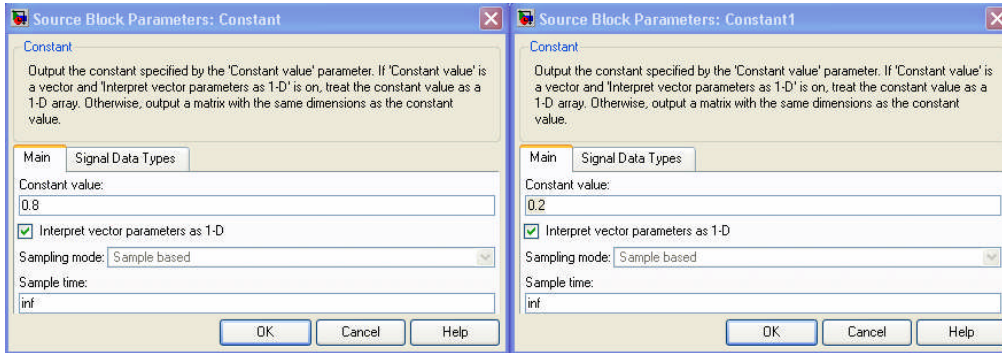


Рис. 5.12. Боки констант

Суматори знаходяться в бібліотеці SIMULINK Math Operations виконаємо відповідні настройки як показано на рис. 5.13.

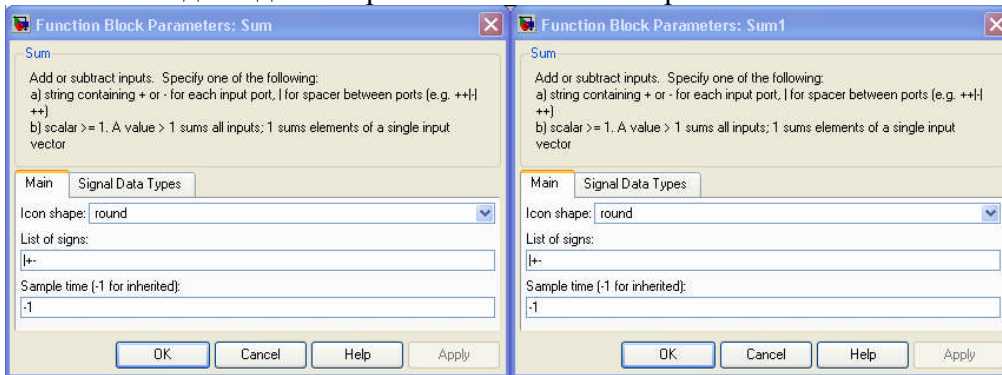


Рис. 5.13. Суматори

Ланка зворотного зв'язку знаходиться в бібліотеці SIMULINK Math Operations та настраюються як показано на рис. 5.14.

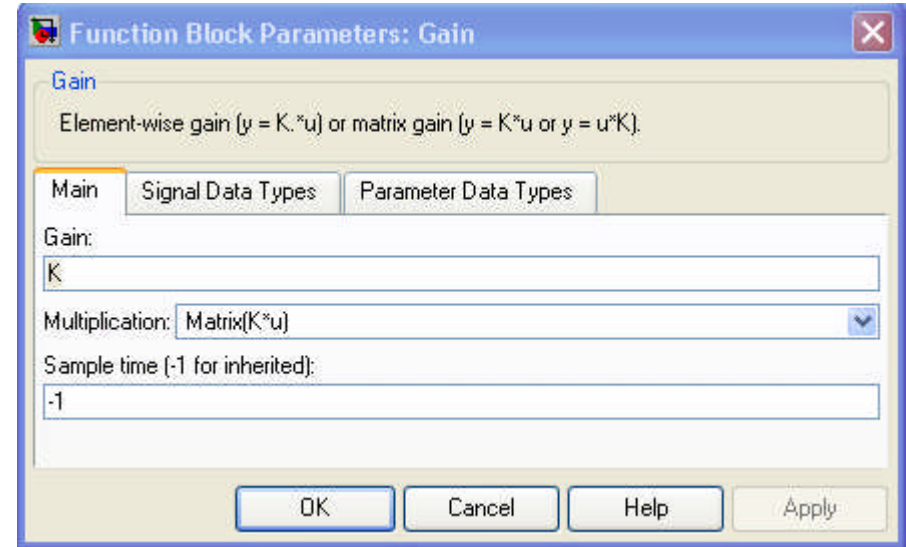


Рис. 5.14. Ланка зворотного зв'язку

З'єднувач та розгалужувачі сигналу знаходяться в бібліотеці SIMULINK Signal Routing та настраюються як показано на рис. 5.15.

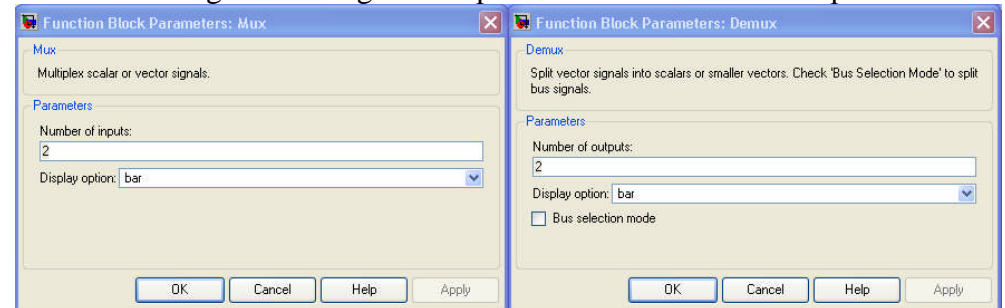


Рис. 5.15. Змішувачі та розгалужувачі сигналу

Блоки Signal Constraint знаходяться в бібліотеці SIMULINK Simulink Response Optimization та настраюються як показано на рис. 5.16.

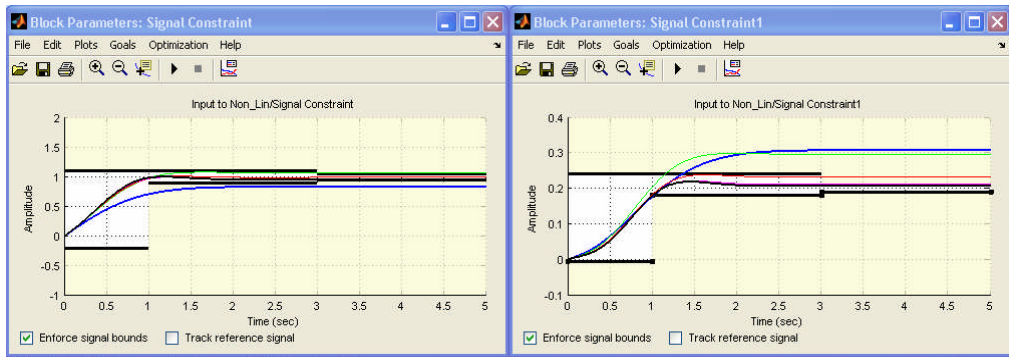


Рис. 5.16. Блоки Signal Constraint

Введення параметрів, що настраюються показано на рис. 5.17.

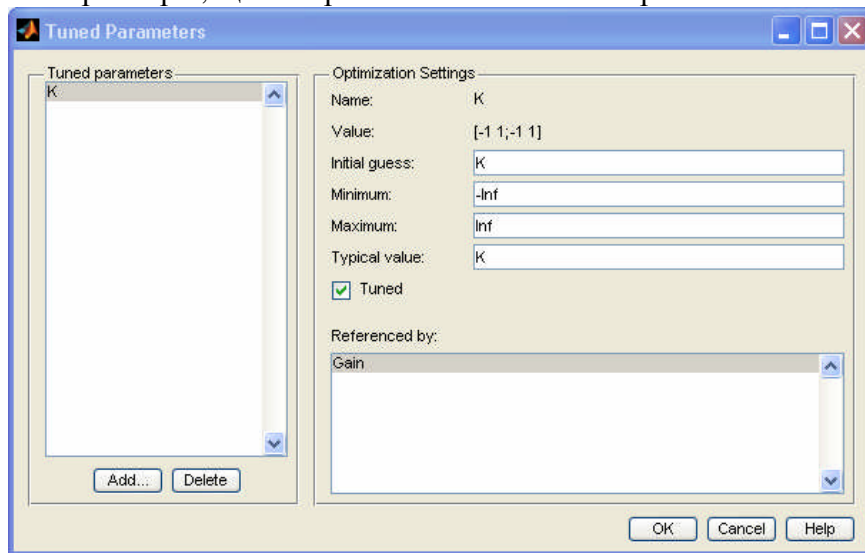


Рис. 5.17. Параметри, що настраюються

В процесі виконання моделювання в блоці Signal Constraint та Signal Constraint1 відображається кожна основна ітерація, а також відкривається нове вікно Optimization Progress, в якому відображається інформація про оптимізацію (залежно від параметрів настройки блока Signal Constraint) та з'являється зображення на осцилографах Scope та Scope1.

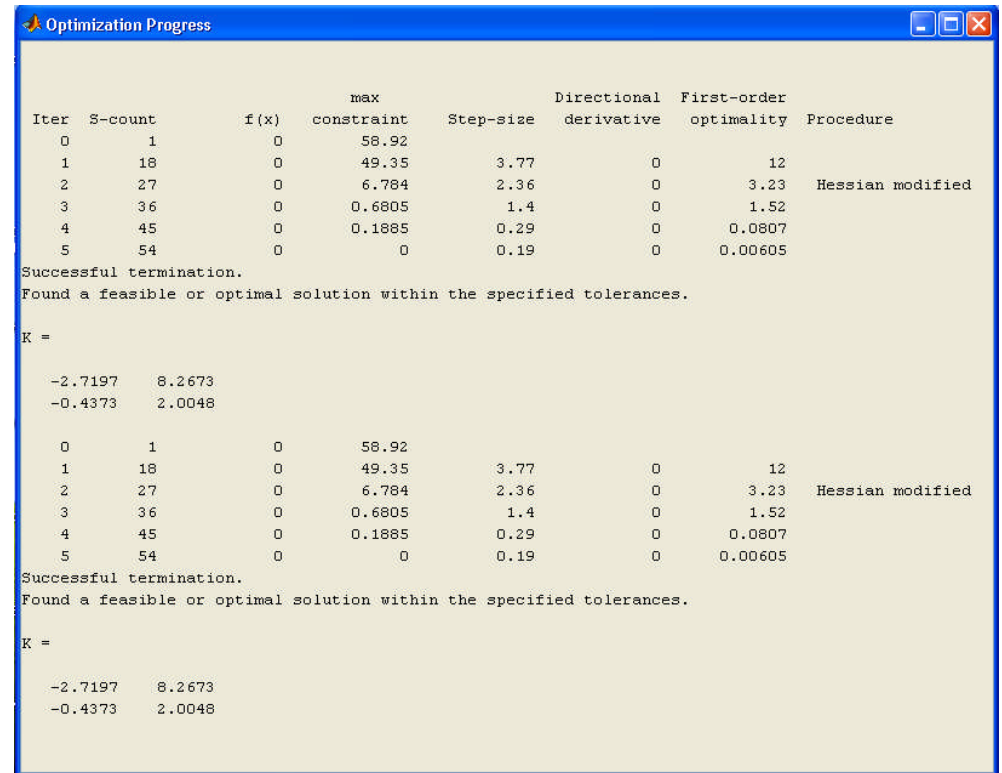


Рис. 5.18. Відображення інформації про оптимізацію

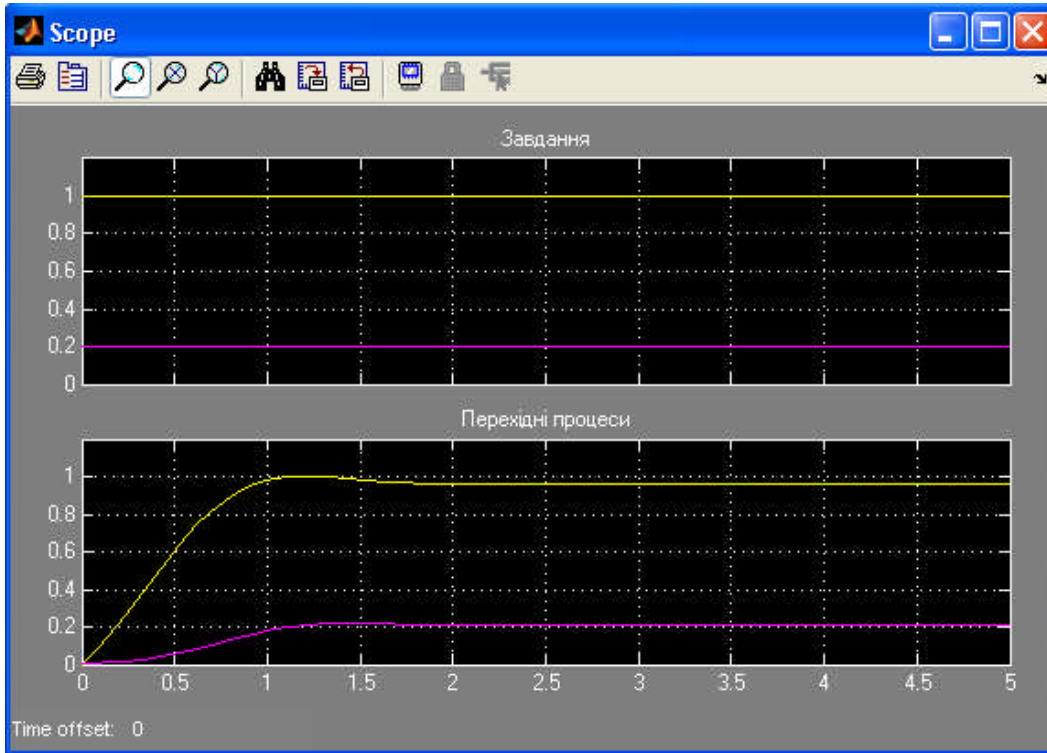


Рис. 5.19. Осцилограми завдання та перехідних процесів

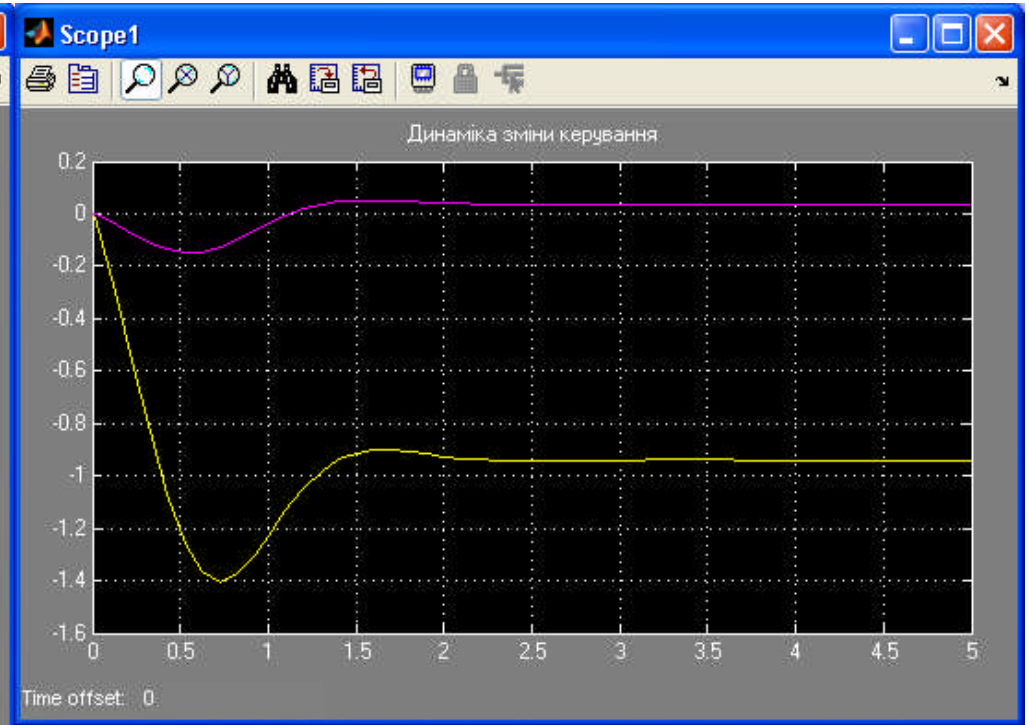


Рис. 5.20. Осцилограма керувань

По закінченню процесу оптимізації, оптимальні значення настроюваних параметрів зберігаються в робочому просторі MATLAB.

Як було зазначено вище **Signal Constraint** дозволяє враховувати в процесі настройки математичної моделі невизначеності параметричного типу. Розглянемо приклад системи, що описується рівняннями в просторі стану:

$$\frac{dX(t)}{dt} = AX(t-0,1) + BU(t),$$

$$Y(t-0,1) = CX(t-0,1),$$

де

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ a & b & c \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix},$$

де $a_{nom} = -1$; $a_{min} = -0,9$; $a_{max} = -1,1$;

$b_{nom} = 2$; $b_{min} = 1,8$; $b_{max} = 2,2$;

$c_{nom} = -4$; $c_{min} = -4,4$; $c_{max} = -3,6$.

Треба забезпечити робастне керування, тобто керування, що забезпечує виконання вимог, пред'явлених до замкненої системи за будь-яких значень параметрів a, b, c з зазначених діапазонів. В якості зворотного зв'язку, будемо використовувати ПІД – регулятор з вихідним сигналом вигляду:

$$u(t) = K \left(e(t) + \frac{1}{Ti} \int_0^t e(t) dt + Tv \frac{de(t)}{dt} \right) \equiv k_p e(t) + k_i \int_0^t e(t) dt + k_d \frac{de(t)}{dt}$$

де $k_p = 6$; $k_i = 40$; $k_d = 0,1$ – початкові значення параметрів настройки регулятора.

Регулятор повинен відпрацювати завдання у вигляді:

$$y_0(t) = \frac{1}{2} + \frac{1}{4}(1 + \text{sign}(t - 2,5)).$$

Для розв'язання поставленої задачі, спочатку ініціалізуємо в командному вікні MATLAB змінні математичної моделі

>> a = -1; b = 2; c = -4;

Та початкові значення параметрів настройки регулятора

>> kp= 6; ki = 40; kd = 0.1;

Створимо SIMULINK – модель (блок-схему) рис. 5.21.

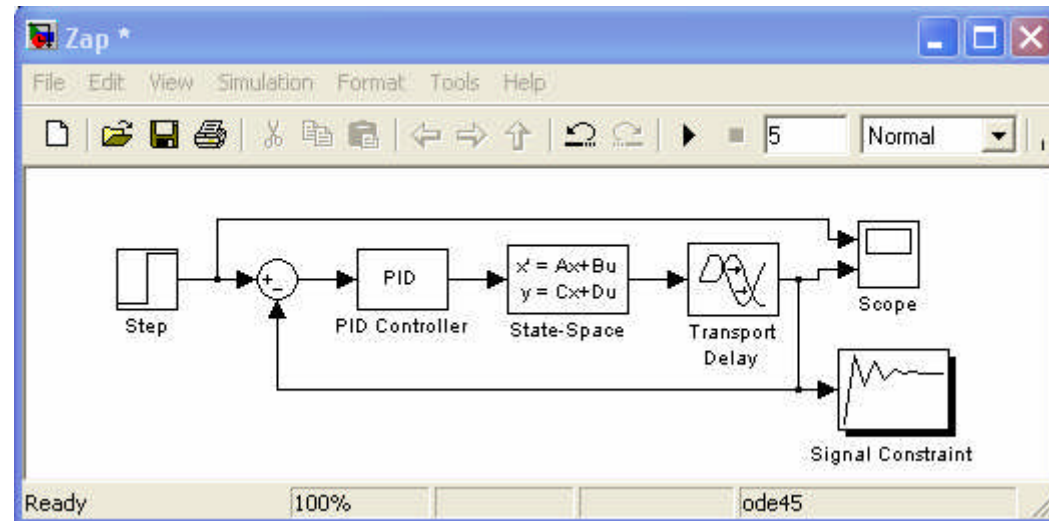


Рис. 5.21. SIMULINK – модель

Виконаємо настроювання моделі об'єкта на базі простору станів State-Space, що знаходиться в бібліотеці SIMULINK Continuous та виконаємо настройку блока, як показано на рис. 5.22.

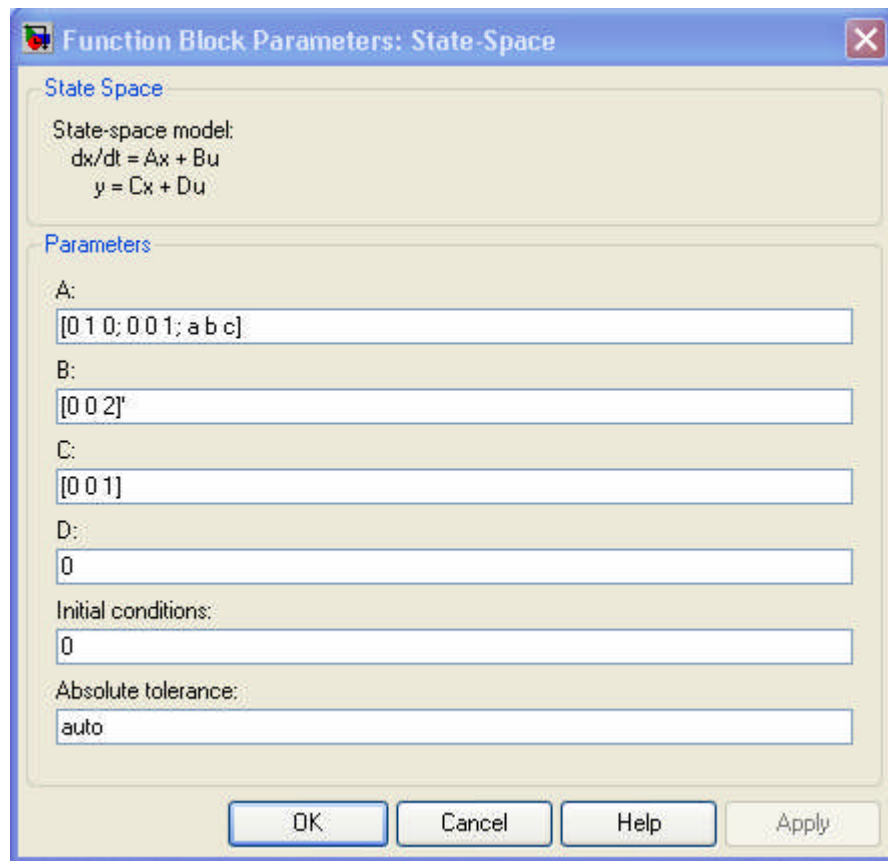


Рис. 5.22. Модель об'єкта на базі простору станів

Блок транспортного запізнення знаходиться в бібліотеці SIMULINK Continuous та виконаємо настройку блока, як показано на рис. 5.23.

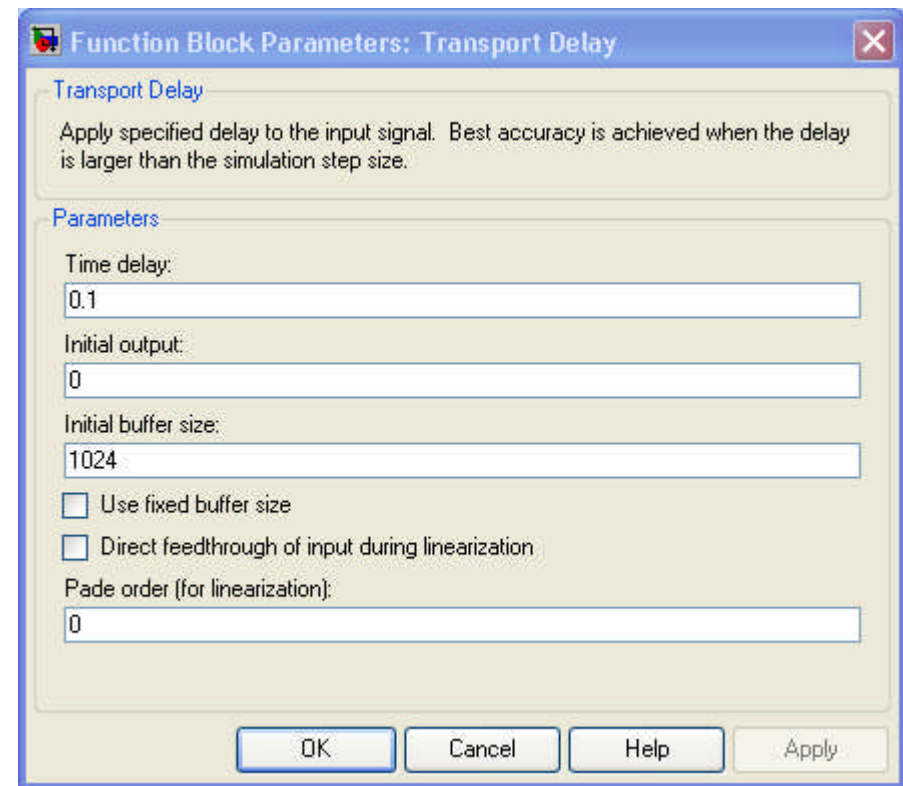


Рис. 5.23. Блок транспортного запізнення

Блок ПІД – регулятора знаходиться в бібліотеці SIMULINK Simulink Extras / Additional Linear та виконаємо настройку блока, як показано на рис. 5.24.

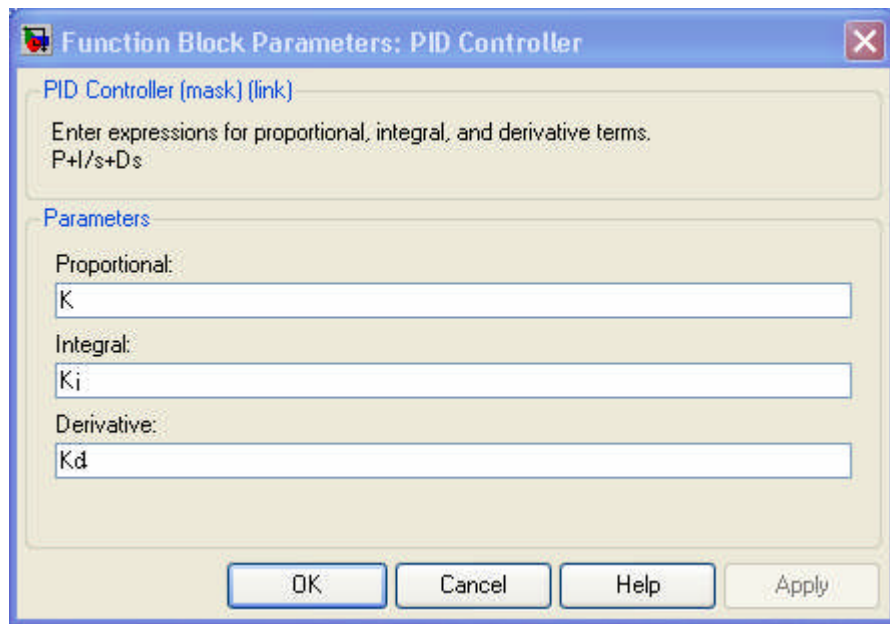


Рис. 5.24. Блок ПІД – регулятора

Суматор знаходиться в бібліотеці SIMULINK Math Operations, виконаємо відповідні настройки, як показано на рис. 5.25.

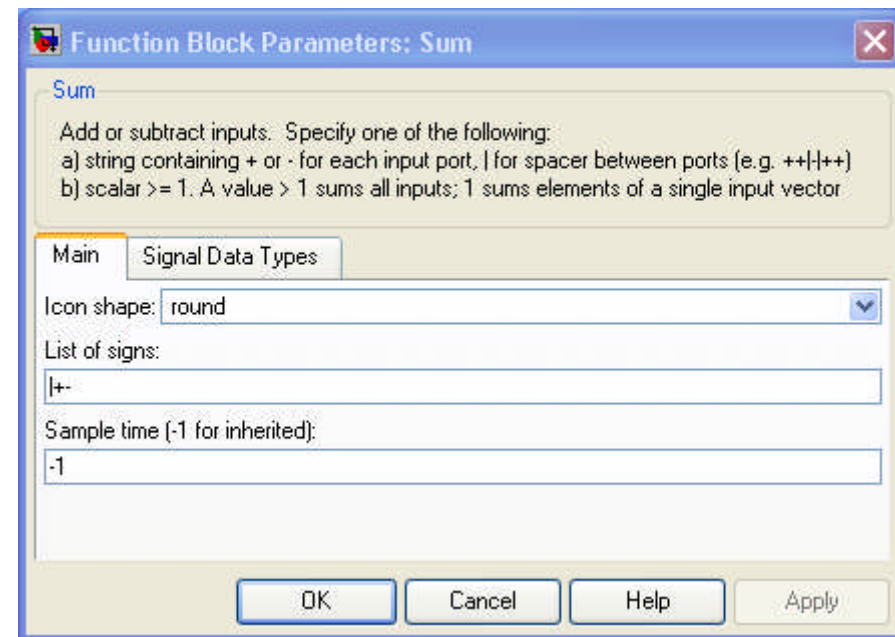


Рис. 5.25. Суматор

Для ініціалізації завдання в системі регулювання використаємо блок Step, що знаходиться в бібліотеці SIMULINK Sources і виконаємо відповідні настройки як показано на рис. 5.26.

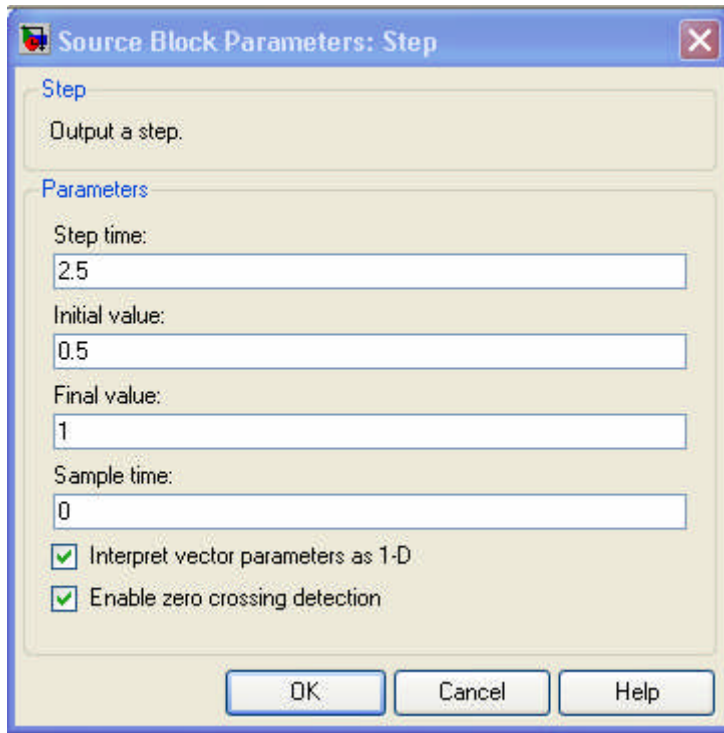


Рис. 5.26. Ініціалізація завдання в системі регулювання

Блок Signal Constraint знаходиться в бібліотеці SIMULINK Simulink Response Optimization та настроюється як показано на рис. 5.27.

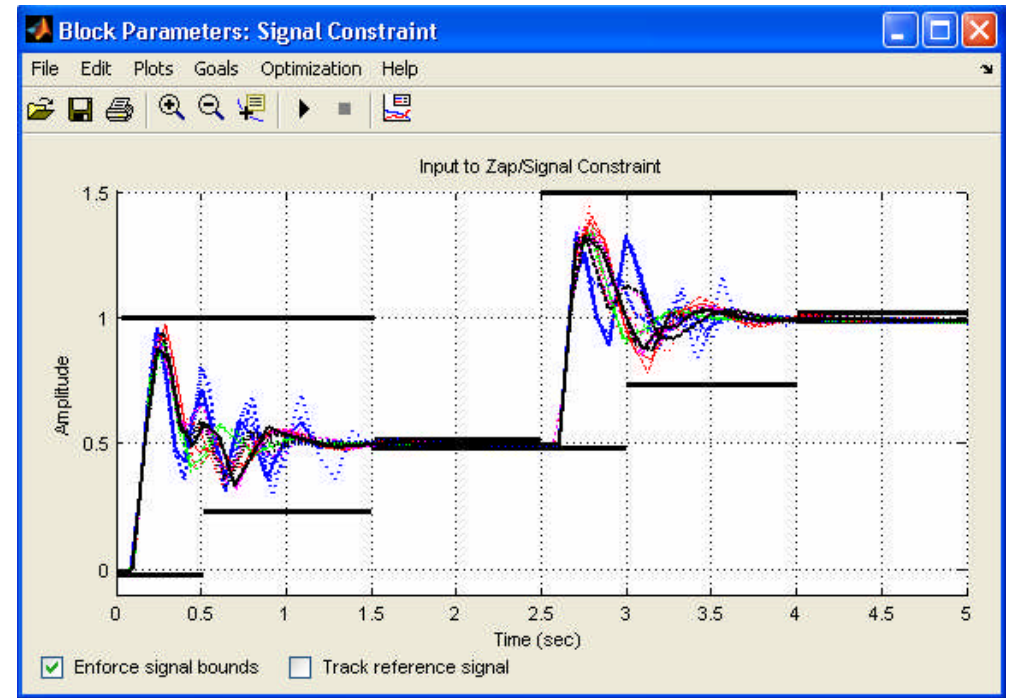


Рис. 5.27. Блок Signal Constraint

Введення параметрів, що настроюються показано на рис. 5.28.

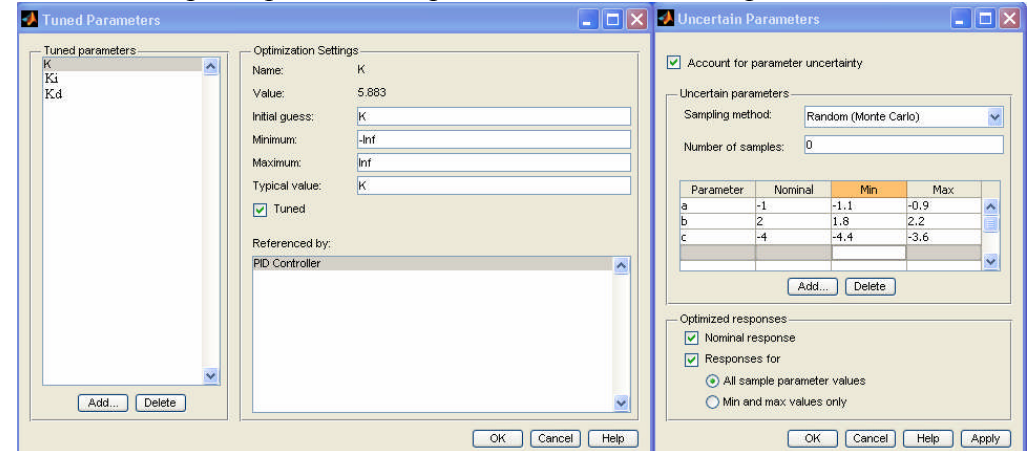


Рис. 5.28. Введення параметрів, що настроюються

В процесі виконання моделювання в блоці Signal Constraint відображається кожна основна ітерація, а також відкривається нове вікно Optimization Progress в якому відображається інформація про оптимізацію (залежно від параметрів настройки блока Signal Constraint) та з'являється зображення на осцилографах Scope.

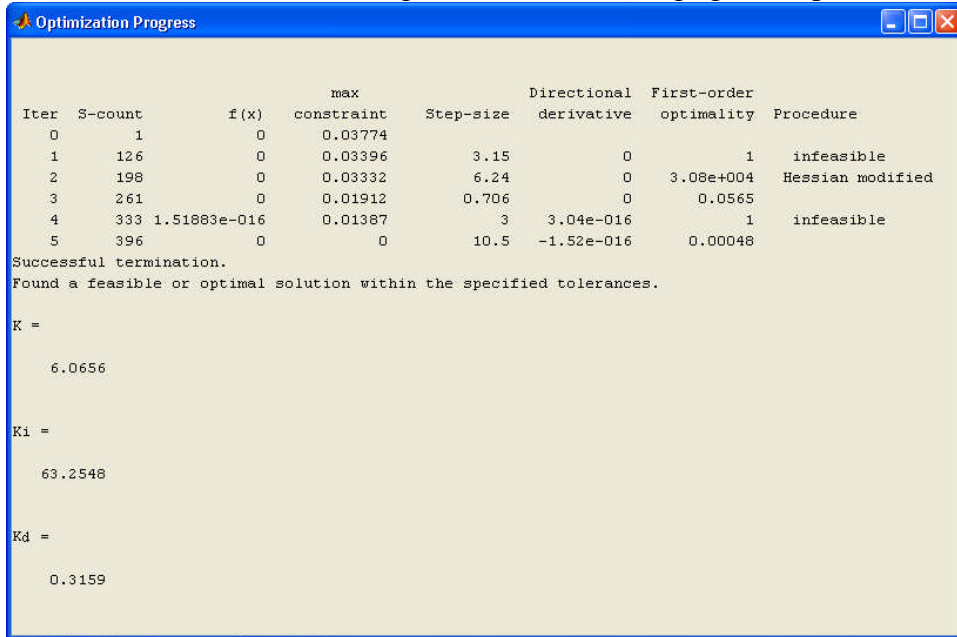


Рис. 5.29. Відображення інформації про оптимізацію

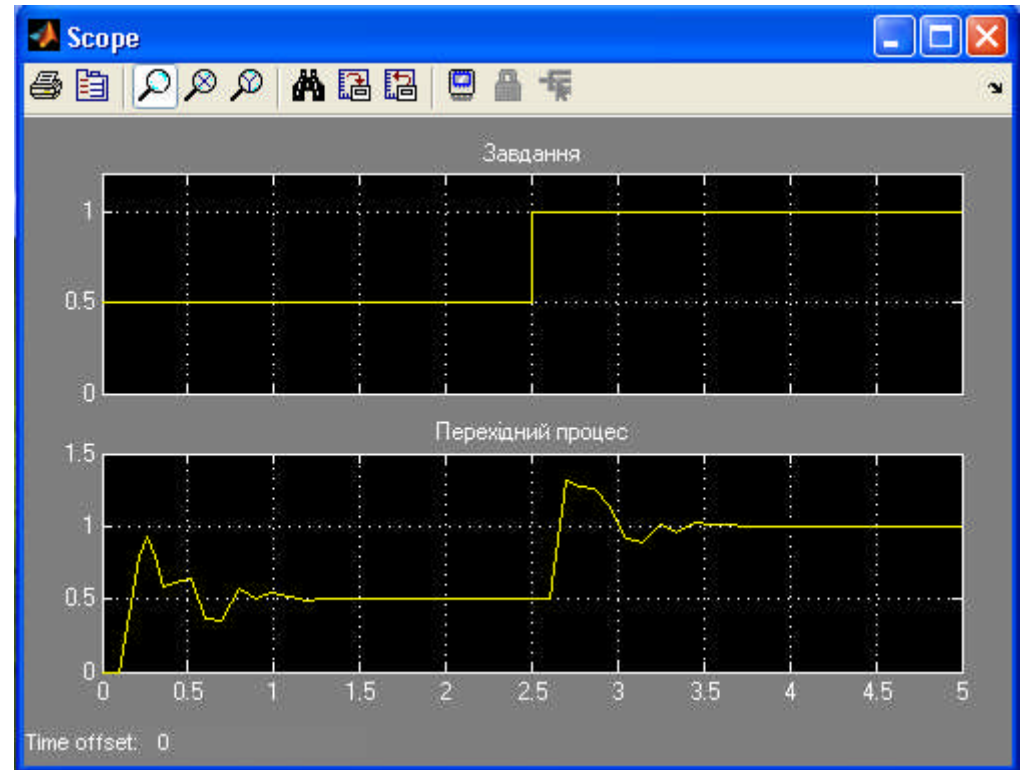


Рис. 5.30. Осцилограми завдання та перехідного процесу

По закінченню процесу оптимізації, оптимальні значення настроєваних параметрів зберігаються в робочому просторі MATLAB.

6. Розв'язання систем нелінійних рівнянь

Під розв'язанням деякої нелінійної системи рівнянь $F(X)$ розуміється знаходження такого розв'язку, коли кожне рівняння в даній нелінійній системі буде дорівнювати нулю. Таким чином, маємо n рівнянь та n невідомих і потрібно знайти такі $X \in \mathcal{R}^n$, що буде справедливе $F(X) = 0$, де

$$F(X) = \begin{bmatrix} F_1(X) \\ F_2(X) \\ \vdots \\ F_n(X) \end{bmatrix}. \quad (6.1)$$

Метод Ньютона-Гаусса

Одним з можливих підходів до розв'язання даної задачі полягає у використанні нелінійного методу найменших квадратів [3,11]. Оскільки ми припустили, що корені системи існують, то вона повинна мати і найменшу нев'язку і тому в даному випадку може бути ефективним метод Ньютона-Гаусса. При цьому на кожній ітерації необхідно вирішувати деяку лінійну задачу за допомогою методу найменших квадратів, згідно рівнянню (6.3), і визначати напрям пошуку.

Після позначення матриці Якобі для $F(X)$ розмірністю $m \times n$ через $J(X)$, вектора градієнта функції $F(X)$ через $G(X)$, матриці Гессе для $F(X)$ через $H(X)$ і матриці Гессе для кожної $F_i(X)$ через $H_i(X)$ одержимо:

$$G(X) = 2J(X)^T F(X), \quad (6.2)$$

$$H(X) = 2J(X)^T J(X) + 2Q(X),$$

де

$$Q(X) = \sum_{i=1}^m F_i(X) H_i(X).$$

Матриця $Q(X)$ має властивість, що коли нев'язка $\|F(X)\|$ прямує до нуля при прямуванні X_k до точки розв'язку, то $Q(X)$ прямує до нуля. Таким чином, при невеликих значеннях $\|F(X)\|$ в точці розв'язку, одним з найбільш ефективних методів є використання напрямку Ньютона-Гаусса як основи для процедури й оптимізації.

Відповідно до методу Ньютона-Гаусса напрямок пошуку D_k перебуває на кожній ітерації k так, що б розв'язок задачі на метод найменших квадратів буде

$$\min_{X \in \mathbb{R}^n} \|J(X_k) D_k - F(X_k)\|_2^2. \quad (6.3)$$

Отриманий відповідно до даного методу напрямок, є еквівалентом напрямку Ньютона, при ігноруванні члена $Q(X)$. Напрямок пошуку X_k може бути використано як складова стратегія лінійного пошуку, що забезпечує умову, що на кожній ітерації йде зменшення функції $F(X)$.

Метод Левенберга-Марквардта

В основу методу Левенберга-Марквардта покладений напрямок пошуку, який знаходиться при розв'язанні системи лінійних рівнянь:

$$\left(J(X_k)^T J(X_k) + \lambda_k \text{diag} \left(J(X_k)^T J(X_k) \right) \right) D_k = -J(X_k)^T F(X_k), \quad (6.4)$$

де скаляр λ_k задає як величину, так і напрямок параметра D_k . Коли λ_k дорівнює нулю, то напрямок D_k буде ідентично цьому ж параметру, що в методі Ньютона-Гаусса. У міру того, як λ_k прямує до нескінченності, то D_k прямує до вектора з нульовими компонентами й напрямку найшвидшого спуску. У цьому випадку передбачається, що для досить великих значень λ_k , $F(X_k + D_k) < F(X_k)$ залишається справедливим. Отже, член λ_k може бути контрольованим з метою забезпечення найшвидшого спуску, якщо буде потреба обліку членів другого порядку, які, у свою чергу, помітно обмежують ефективність методу Ньютона-Гаусса. Звідси випливає, що метод Левенберга-Марквардта, заснований на напрямку пошуку, що є сполученням напрямку Ньютона-Гаусса й найшвидшого спуску.

Метод ламаних довірчих областей

Інший підхід в розв'язанні лінійної системи рівнянь, полягає в знаходженні напрямку пошуку, а саме, згідно методу Ньютона потрібно знайти такий напрям пошуку D_k , що

$$J(X_k)D_k = -F(X_k)$$

$$X_{k+1} = X_k + D_k$$

де $J(X_k)$ Якоб'іан розмірністю $n \times n$

$$J(X_k) = \begin{bmatrix} \nabla F_1(X_k)^T \\ \nabla F_2(X_k)^T \\ \vdots \\ \nabla F_n(X_k)^T \end{bmatrix}.$$

У методі Ньютона можуть зустрічатися різні труднощі. Якщо $J(X_k)$ буде сингулярною, то крок в методі Ньютона не можна буде навіть визначити. Крім того, обчислювальні витрати для точного визначення кроку за Ньютоном D_k можуть бути надмірно великими. Якщо початкова точка була вибрана надмірно далеко від точки розв'язку, то метод Ньютона може взагалі не збігатися.

Застосування методики довірчих областей дозволить поліпшити стійкість розрахунків у випадках, коли початкова точка вибрана дуже далеко від точки розв'язку, і долати неприємності у разі сингулярної матриці $J(X_k)$. Для використання стратегії довірчих областей необхідна якась функція вигоди для того, що б визначити, що точка буде краща або гірше чим X_k . Можлива стратегія вибору може полягати в наступному:

$$\min_D f(D) = \frac{1}{2} F(X_k + D)^T F(X_k + D)$$

Але мінімум $f(D)$ не обов'язково повинен бути коренем $F(X)$.

Крок за Ньютоном D_k є коренем

$$M(X_k + D) = F(X_k) + J(X_k)D$$

і оскільки це також є мінімумом і для $m(D)$, де

$$\begin{aligned} \min_D m(D) &= \frac{1}{2} \|M(X_k + D)\|_2^2 = \frac{1}{2} \|F(X_k) + J(X_k)D\|_2^2 = \\ &= \frac{1}{2} F(X_k)^T F(X_k) + D^T J(X_k)^T F(X_k) + \frac{1}{2} D^T (J(X_k)^T J(X_k)) D \end{aligned} \quad (6.5)$$

Тоді $m(D)$ буде більш кращою стратегією вибору для функції вигоди, чим $f(D)$, і звідси під задача довірчої області полягатиме в наступному:

$$\min_D \left[\frac{1}{2} F(X_k)^T F(X_k) + D^T J(X_k)^T F(X_k) + \frac{1}{2} D^T (J(X_k)^T J(X_k)) D \right] \quad (6.6)$$

За умови, що $\|D\| \leq \Delta$. Така внутрішня під-задача може бути успішно вирішена за допомогою стратегії ламаних напрямів. Детальніше про метод довірчих областей можна знайти в роботах Кону [4] і Носеда [5].

Реалізація розв'язання нелінійних рівнянь

Реалізація даного методу розв'язання нелінійних рівнянь складається з трьох частин:

- Реалізація методу Ньютона-Гаусса;
- Реалізація методу Левенбрга-Марквардта;
- Реалізація методу ламаних довірчих областей.

Реалізація методу Ньютона-Гаусса

Метод Ньютона-Гаусса реалізований за допомогою стратегії поліноміального лінійного пошуку. При розв'язанні лінійної задачі методом найменших квадратів, є можливість уникнути посилення збурень при узгодженні параметрів рівнянь, шляхом використання QR - розкладання для $J(X_k)$ і застосування подібного розкладання для $F(X_k)$. Даний підхід відрізняється від

застосування інверсії явної матриці $J(X_k)^T J(X_k)$, де можливо виникнення несподіваних помилок.

Додаткові заходи по забезпеченню стійкості, також включені в даний метод. Дані заходи включають коригування алгоритму методу Левенберга-Марквардта у випадку, якщо довжина кроку стає меншою за якесь порогове значення (10^{-15} в даному виконанні) або коли число обумовленості $J(X_k)$ буде менше 10^{-10} . Під числом обумовленості в даному випадку розуміється відношення найбільшого сингулярного значення до як найменшого.

Реалізація методу Левенберга-Марквардта

Основна складність в реалізації методу Левенберга-Марквардта, полягає у виборі ефективної стратегії, для управління розміром λ_k для кожної ітерації таким чином, що б вона підходила для широкого спектру задач. Прийнятий в даному методі підхід реалізації, полягає в оцінці відносної нелінійності $f(X)$ на базі застосування лінійно прогнозованих сум квадратів $f_p(X_k)$ і кубічно інтерпольованої оцінки мінімуму $f_k(X_*)$. При такому підході розмір λ_k може бути визначений для кожної ітерації. Лінійно прогнозована сума квадратів розраховується як

$$f_p(X_k) = J(X_{k-1})D_{k-1} + F(X_{k-1}), \quad (6.7)$$

а член $f_k(X_*)$ визначається з кубічної інтерполяції точок $f(X_k)$ і $f(X_{k-1})$. З даної інтерполяції, так само розраховується і параметр довжини кроку α^* , що є оцінкою кроку у напрямку до мінімуму. Якщо $f_p(X_k)$ буде більше, ніж $f_k(X_*)$, то λ_k зменшується, а в протилежному випадку – збільшується. Обґрунтуванням для даного підходу служить той факт, що різниця між $f_p(X_k)$ і $f_k(X_*)$ є мірою ефективності методу Ньютона-Гаусса і ступеня лінійності задачі. В даному випадку необхідно провести вибір: використовувати напрям згідно напрямку для методу найшвидшого спуску або використовувати напрям згідно методу Ньютона-Гаусса. Формули для відповідного зменшення або збільшення параметра λ_k , які були отримані з аналізу великого числа тестових задач, приведені на рисунку (6.1).

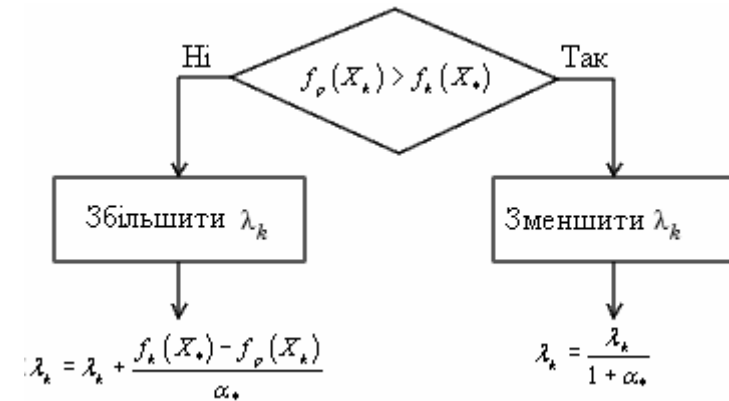


Рис. 6.1. Коригування λ_k

Далі, вслід за коригуванням λ_k , результат розв'язання рівняння (6.4) використовується для знаходження напрямку пошуку D_k . Потім у напрямі D_k довжина кроку приймається рівній одиниці з подальшим застосуванням процедури лінійного пошуку. Процедура лінійного пошуку, забезпечує виконання умови $f(X_{k+1}) > f(X_k)$ для кожної основної ітерації і таким чином, даний метод стає методом найшвидшого спуску.

Реалізація методу ламаних довірчих областей

Ключовою особливістю даного алгоритму є використання методики ламаних напрямів Пауелла для розрахунку D з мінімізацією згідно рівнянню (6.6). Детально метод приведений в роботі Пауелла [6].

Крок D будується на основі опуклої комбінації кроку Коші (крок уздовж напрямку найшвидшого спуску) і кроку Ньютона-Гауса для $F(X)$. Крок Коші визначається як

$$D_C = -\alpha J(X_k)^T F(X_k),$$

де α вибрано з мінімуму згідно рівнянню (6.5).

Крок Ньютона-Гауса знаходиться з рішення рівняння:

$$J(X_k)D_{GN} = -F(X_k)$$

Звідси вибраний крок буде

$$D = D_C + \lambda(D_{GN} - D_C),$$

де λ є найбільше значення на інтервалі $[0, 1]$ таке, що $\|D\| \leq \Delta$.

Якщо J_k є (приблизно) сингулярним, то D якраз буде напрямом Коші.

Алгоритм ламаних напрямів ефективніший, оскільки для нього на кожній ітерації потрібне тільки одне звернення до розрахунку кроку за методом Ньютона-Гауса. Крім того, він може бути і стійкішим за рахунок використання методу Ньютона-Гауса для лінійного пошуку.

Приклади програм

Розв'язок алгебраїчних рівнянь Ріккати

Система з одним входом і одним виходом описується рівнянням стану, в якому

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -4 & -9 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}.$$

Знайти розв'язок алгебраїчних рівнянь Ріккати за допомогою стандартних функцій MATLAB.

Знайдемо спочатку чисельний розв'язок алгебраїчних рівнянь Ріккати, за допомогою функції призначеної для розв'язання алгебраїчних і матричних рівнянь. Для цього запишемо процедуру (M – файл) та збережемо його як "Riccati_R"

```
function [F A B Q R N] = Riccati_R(P)
A = [0 1 0; 0 0 1; -2 -4 -9]; % Введення матриці A
B = [0 0 2]'; % Введення матриці B
Q = eye(3); % Введення матриці Q
R = 1/2; % Введення матриці R
```

```
N = [0 0 0]'; % Введення матриці N
F = A'*P+P*A-(P*B+N)*inv(R)*(B'*P+N')+Q;
```

Запишемо основну програму (M – файл)

```
P0 = eye(3) % Завдання початкової точки
% Параметри опцій оптимізації
options =
optimset('Display','iter','NonlEqnAlgorithm','lm');
% Виклик функції оптимізації
[P,fval,exitflag,output,jacobian] =
fsolve(@Riccati_R,P0,options)
% P – розв'язок рівняння
% fval – значення функції в точці розв'язку
% exitflag – флаг виходу алгоритма (інформація про стан алгоритма)
% output – інформація структура інформації про оптимізацію
% jacobian – матриця якобі
```

Результат виконання програми

P0 =

```
1 0 0
0 1 0
0 0 1
```

Directional Iteration derivative	Func-count Lambda	Residual	Step-size
0	10	655	
1	23	147.435	0.658
-47.3	6.01476e-025		
2	37	97.3999	1.22
-0.0171	217.785		

3	51	91.4365	8.54
-0.000308	218.541		
4	65	76.1504	1.6
-0.0103	22.9044		
5	79	47.0195	2.41
-0.015	8.80039		
6	93	18.772	1.4
-0.0539	2.57805		
7	107	5.48656	1.57
-0.00771	1.07159		
8	121	2.18087	1.53
-0.00139	0.417497		
9	135	0.292045	1.99
-0.0759	0.165134		
10	148	0.0176133	1.19
-6.26e-005	0.0530548		
11	162	0.000317052	1.23
-1.35e-006	0.024209		
12	175	1.4488e-006	1.05
-3.26e-010	0.0108653		
13	188	1.69155e-009	1.05
-6.19e-014	0.00529928		
14	201	4.95587e-013	1.01
-4.56e-018	0.00258234		
15	202	1.1434e-016	1
-9.79e-013	0.00128319		

1.0e-008 *

-0.9490	0.0733	0.2869
0.0733	-0.1653	0.0816
0.2869	0.0816	-0.1637

exitflag =

1

output =

```

iterations: 15
funcCount: 202
stepsize: 1
cgiterations: []
firstorderopt: []
algorithm: 'medium-scale: Levenberg-
Marquardt, line-search'
message: [1x147 char]

```

Optimization terminated: directional derivative
along
search direction less than TolFun and infinity-
norm of
gradient less than 10*(TolFun+TolX).

P =

2.2364	1.9915	0.1830
1.9915	4.4685	0.4343
0.1830	0.4343	0.0994

fval =

jacobian =

0	0	-3.4641	0	0
0	-3.4641	0	0	0
1.0000	0	-7.4742	0	0
0	0	-3.4641	0	0
0	0	1.0000	-9.7953	0
0	0	0	-3.4641	0

```

1.0000      0      0      0      0      -
3.4641    -7.4742      0      0      0      -
      0      1.0000      0      1.0000      0      -
7.4742      0    -7.4742      0      0      -
      0      0      1.0000      0      1.0000      -
9.7953      0      0      0    -7.4742      0
      0      0      0      0      1.0000      0
0    -9.7953      0      0    -3.4641      0
      0      0      0      0      0      1.0000
0      1.0000    -9.7953    -7.4742      0      0
      0      0      0      0      0      0
1.0000      0      1.0000    -19.5907

```

L =

```

-9.0083
-0.3935 + 0.4793i
-0.3935 - 0.4793i

```

G =

```

0.7321    1.7371    0.3977

```

report =

```

1.3592e-015

```

Тепер розв'яжемо алгебраїчні рівняння Ріккаті за допомогою стандартної функції. Програма (M – файл) наведений нижче

```

A = [0 1 0; 0 0 1; -2 -4 -9]; % Введення
матриці A
B = [0 0 2]'; % Введення матриці B
Q = eye(3); % Введення матриці Q
R = 1/2; % Введення матриці R
N = [0 0 0]'; % Введення матриці N
E = eye(3); % Введення матриці E (в нашому
випадку I)
[P L G report] = care(A,B,Q,R,N,E) %
Розв'язок рівнянь Ріккаті
% P - розв'язок
% L - власні значення замкненої системи
% G - матричний коефіцієнт підсилення
% report - RR норма Фробеніуса відносних
залишків

```

Результат виконання програми

P =

```

2.2364    1.9915    0.1830
1.9915    4.4685    0.4343
0.1830    0.4343    0.0994

```

Розв'язок нелінійної системи рівнянь

Розв'язати систему рівнянь

$$\begin{cases} x_1^4 - \cos x_1 e^{x_2} = 0 \\ 2x_2 - \sin x_2 e^{x_1} = 0 \end{cases}$$

Знайдемо спочатку чисельний розв'язок системи рівнянь, за допомогою функції призначеної для розв'язання алгебраїчних і матричних рівнянь. Для цього запишемо процедуру (M – файл) та збережемо його як "Eq"

```

function F = Eq(x)
F = [x(1)^4 - cos(x(1))*exp(x(2)); 2*x(2) -
sin(x(2))*exp(x(1))];

```

Запишемо основну програму (M – файл)

```

x0 = [1 -1]'; % Завдання початкової точки
% Параметри опцій оптимізації

```

```

options =
optimset('Display','iter','NonlEqnAlgorithm','gn'
);
% Виклик функції оптимізації
[x,fval,exitflag,output,jacobian] =
fsolve(@Eq,x0,options)
% x - розв'язок рівняння
% fval - значення функції в точці розв'язку
% exitflag - флаг виходу алгоритма (інформація
про стан алгоритма)
% output - інформація структура інформації про
оптимізацію
% jacobian - матриця якобі

```

Результат виконання програми

```

Directional
Iteration  Func-count  Residual  Step-size
derivative
    0           3      0.724549
    1          10      0.000355977      1.27
-0.0123
    2          16      4.44751e-007      1
-4.63e-007
    3          22      2.20634e-013      1
-2.85e-013
Optimization terminated: directional derivative
along
search direction less than TolFun and infinity-
norm of
gradient less than 10*(TolFun+TolX).

```

```

x =
    0.7739
   -0.6903

```

```
fval =
```

```

1.0e-006 *
    0.4318
   -0.1849

exitflag =
    1

output =
    iterations: 4
    funcCount: 22
    stepsize: 1.0009
    cgiterations: []
    firstorderopt: []
    algorithm: 'medium-scale: Gauss-Newton,
line-search'
    message: [1x147 char]

jacobian =
    2.2042   -0.3586
    1.3806    0.3282

Тепер отримаємо аналітичний розв'язок за допомогою функції
solve зпишемо М - файл
% Отримання розв'язку і знесення його в структуру
S
S = solve('x1^4-cos(x1)*exp(x2) = 0','2*x2-
sin(x2)*exp(x1) = 0');
% Вилучення інформації з структури S
x1 = S.x1

```

$$x2 = S.x2$$

Результат виконання програми

$$x1 =$$

$$.77386194903675472502436430883255$$

$$x2 =$$

$$-.69027767856405687546646609111260$$

Як можна побачити з вище отриманих розв'язків, що результати майже співпадають.

7. Розв'язання системи нелінійних диференційних рівнянь

Для розв'язання диференційних рівнянь та систем рівнянь, призначені спеціальні функції MATLAB, в обчислювальній математиці їх називають *солвери*. MATLAB має достатньо великий набір солверів, заснованих на різноманітних числових методах.

Для розв'язання задачі Коші в MATLAB існує сім солверів: ode45, ode23, ode113, ode15s, ode23s, ode23t, ode23tb.

Задача Коші для диференційного рівняння полягає в знаходженні функції, що задовольняє диференційному рівнянню довільного порядку

$$\frac{d^n y}{dt^n} = f\left(t, y, \frac{dy}{dt}, \dots, \frac{d^{n-1}y}{dt^{n-1}}\right)$$

та

початковим

умовам

$$y(t) \Big|_{t=t_0} = y_0, \frac{dy}{dt} \Big|_{t=t_0} = y_1, \dots, \frac{d^{n-1}y}{dt^{n-1}} \Big|_{t=t_0} = y_{n-1}.$$

Приклади програм

Розв'язок диференційного рівняння

Знайдемо розв'язок диференційного рівняння вигляду:

$$\frac{d^2 y}{dt^2} + 2 \frac{dy}{dt} + 10y = \cos t + e^{-3t},$$

$$y(t) \Big|_{t=0} = 0,05, \frac{dy}{dt} \Big|_{t=0} = 0,1.$$

Представимо дане диференційне рівняння у вигляді системи диференційних рівнянь першого порядку (приведемо до форми Коші)

$$\begin{cases} y_1' = y_2 \\ y_2' = -2y_2 - 10y_1 + \cos t + e^{-3t} \end{cases}, \quad \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} = \begin{bmatrix} 0,05 \\ 0,1 \end{bmatrix}.$$

Запишемо програму, що виконує порівняння аналітичного (символьного) розв'язку "Deff_Rivn"

```
function Deff_Rivn
% Початкові умови
Y0 = [0.05 0.1];
% Початковий та кінцевий момент часу
T = [0 20];
% Структура опцій солвера
options = odeset('RelTol',1e-4,'AbsTol',[1e-6 1e-6]);
% Виклик солвера
[T,Y] = ode45(@func,T,Y0,options);
% Аналітичний розв'язок
% Отримання розв'язку і знесення його в структуру
S
S = dsolve('Dy1 = y2','Dy2 = -2*y2-10*y1+cos(t)+exp(-3*t)',...
'y1(0) = 0.05','y2(0) = 0.1');
% Вилучення інформації з структури S
y1 = S.y1
y2 = S.y2
```

```

% Виведення графіку розв'язку вихідного
диференціального рівняння (чисельного
% розв'язку)
plot(T,Y(:,1),'o',T,Y(:,2),'o','color','blue')
hold on
% Виведення графіку розв'язку вихідного
диференціального рівняння (аналітичного
% (символьного) розв'язку)
ezplot(y1,[0 20]), ezplot(y2,[0 20])
% Виведення пояснень на графік
title('Розв'язок  $\{y\} \{prime\} + 2\{y\} \{prime\} + 10\{y\} \{prime\} = \cos t + \exp(-3t)$ ')
xlabel('\itt'), ylabel('\ity, \ity \prime '),
legend('- Чисельний розв'язок  $\{y\}$ ', '-
чисельний розв'язок  $\{y\} \{prime\}$ ', '-
'- Аналітичний розв'язок  $\{y\}$ ', '-
Аналітичний розв'язок  $\{y\} \{prime\}$ '), grid on
hold off

```

Результат виконання програми

```

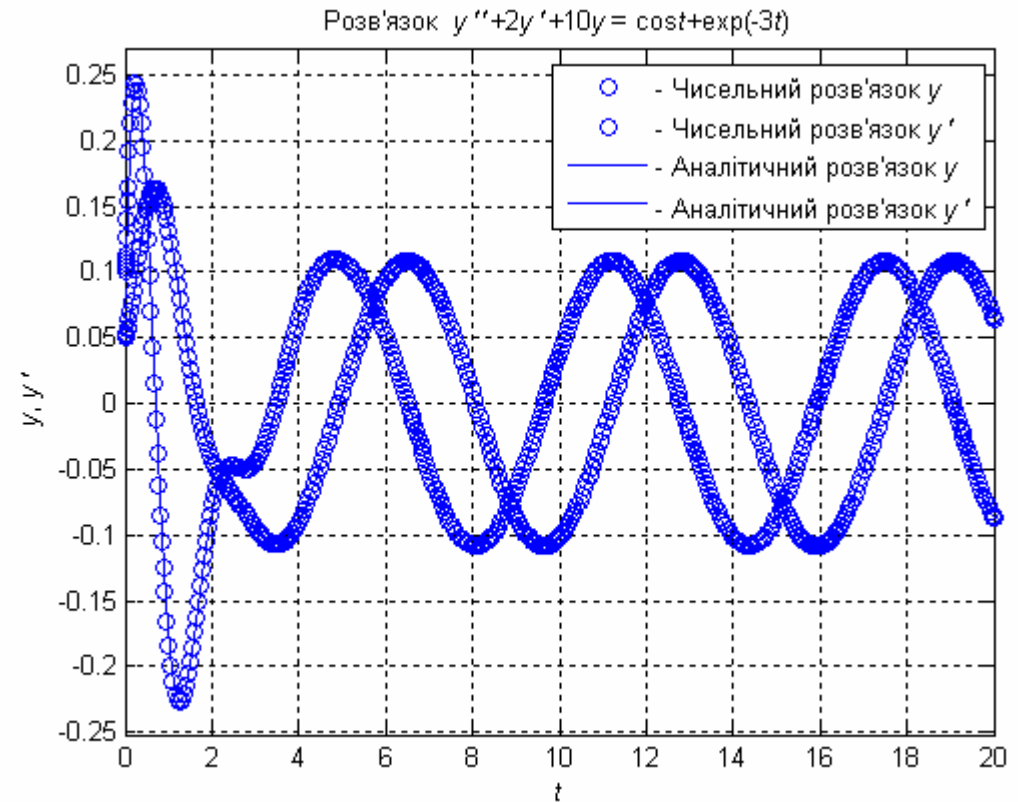
y1 =
257/4420*exp(-t)*sin(3*t)-
587/4420*exp(t)*cos(3*t)+9/85*cos(t)+2/85*sin(t)+
1/13*exp(-3*t)

```

```

y2 =
376/1105*exp(-t)*sin(3*t)+679/2210*exp(-
t)*cos(3*t)-9/85*sin(t)+2/85*cos(t)-3/13*exp(-
3*t)

```



Розв'язання рівнянь Ріккати

Розглянемо скалярну систему

$$\frac{dx(t)}{dt} = \frac{1}{2}x(t) + u(t), \quad x(t_0) = x_0$$

та функцію вартості

$$J = \frac{1}{2}sx^2(t_f) + \frac{1}{2} \int_{t_0}^{t_f} [2x^2(t) + u^2(t)] dt.$$

Рівняння Ріккати матиме вид

$$\frac{dp(t)}{dt} = p(t) + p(t)^2 - 2, \quad p(t_f) = s.$$

Розв'язок цього рівняння можна записати у вигляді:

$$p(t) = -0,5 + 1,5th(-1,5t + \xi_1),$$

або

$$p(t) = -0,5 + 1,5cth(-1,5t + \xi_2),$$

де ξ_1 і ξ_2 обираються так, що $p(t_f) = s$.

Наприклад, якщо $s = 0$, $t_f = 1$ то $\xi_1 = 1,845$ радіан, що дасть

$$p(t) = -0,5 + 1,5th(-1,5t + 1,845)$$

Тепер порівняємо отриманий розв'язок з аналітичним та чисельним розв'язками отримані за допомогою MATLAB. Наведемо приклад програми, що розв'язує дану задачу "Poriv_Ric".

```
function Poriv_Ric
% Введення функції, що була отримана аналітично
t = -1:0.1:5; % діапазон зміни аргументу
p = -0.5+1.5*tanh(-1.5*t+1.845);
% Отримання аналітичного (символьного) розв'язку
за допомогою MATLAB
y = dsolve('Dp = p+p^2-2', 'p(1) = 0', 't');
disp('y = ')
pretty(y)
% Отримання чисельного розв'язку
% Виклик солвера
[T P] = ode45('Riccati', [1 4], 0);
% Виведення графіку розв'язку рівняння
Ріккати (аналітичного (символьного))
ezplot(y)
hold on
% Виведення графіку розв'язку рівняння
Ріккати (аналітичного)
plot(t,p, '.', 'Color', 'red')
% Виведення графіку розв'язку рівняння
Ріккати (чисельного)
plot(T,P, 'o')
```

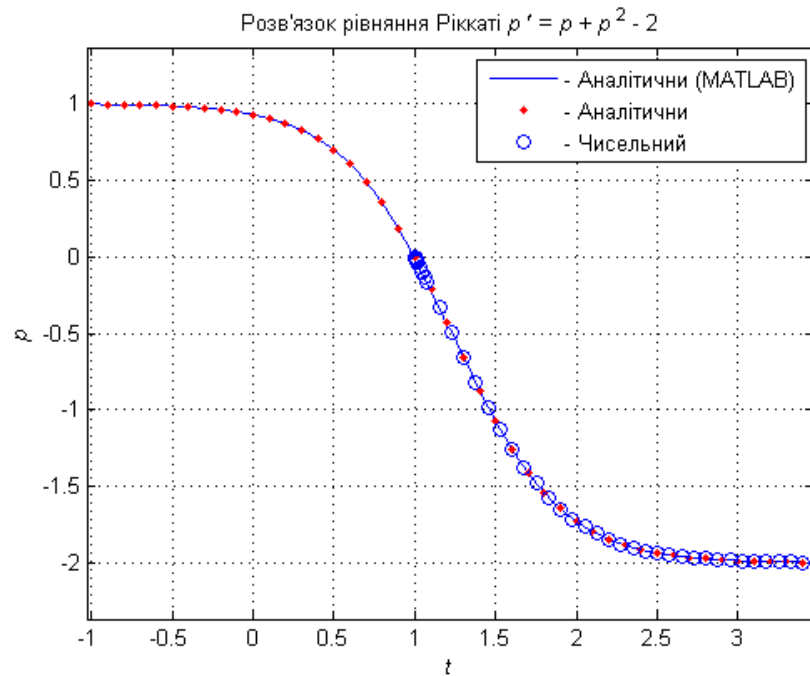
```
title('Розв'язок рівняння Ріккати {\itp} \prime
= {\itp} + {\itp} ^2 - 2')
xlabel('\itt'), ylabel('\itp'), grid on
legend('- Аналітични (MATLAB)', '- Аналітични', '-
чисельний')
hold off
```

Результат виконання програми

y =

$$\frac{\exp(3 t) - 1}{\exp(3)}$$

$$-1 - 1/2 \frac{\exp(3 t)}{\exp(3)}$$



ДОДАТОК

Використані та корисні для синтезу автоматичних систем функції Control System Toolbox.

Створення LTI-моделей

Модель системи в пакеті Control System Toolbox створюється у вигляді LTI – моделі (*Linear Time-Invariant System* – лінійні, інваріантні у часі системи); в літературі закріпилася назва *лінійних стаціонарних систем* (ЛСС).

ss - Створює модель простору стану

Синтаксис та опис

Функція **ss** (state-space) створює неперервну модель системи у просторі станів

$$\frac{dX}{dt} = AX + BU$$

$$Y = CX + DU$$

sys = ss(a,b,c,d)

Функція **ss** (state-space) створює дискретну модель системи у просторі станів з періодом квантування T_s ($T_s = -1$ або $T_s = []$ – період квантування невизначений)

$$X[n+1] = AX[n] + BU[n]$$

$$Y[n] = CX[n] + DU[n]$$

sys = ss(a,b,c,d,Ts)

Виділення статичної матриці D

sys = ss(d) або **sys = ss([], [], [], d)**

Додавання властивостей до системи (наприклад, назви входів та виходів)

sys = ss(a,b,c,d, 'Property1', value1, ..., 'PropertyN', valueN)

sys
ss(a,b,c,d,Ts,'Property1',Value1,...,'PropertyN',ValueN)

zpk - Створює модель нулі / полюси / коефіцієнти передачі;

Функція **zpk** створює неперервну модель системи ($p(i)$ - нулі, $p(j)$ - полюси, k - коефіцієнт передачі)

sys = zpk(z,p,k)

$$W(p) = k \frac{\prod_{i=0}^m (p - p(i))}{\prod_{j=0}^n (p - p(j))}, \quad n \geq m,$$

де p – змінна Лапласа.

Функція **zpk** створює дискретну модель системи ($p(i)$ - нулі, $p(j)$ - полюси, k - коефіцієнт передачі), з періодом квантування T_s

sys = zpk(z,p,k,Ts)

$$W(z) = k \frac{\prod_{i=0}^m (z - z(i))}{\prod_{j=0}^n (z - z(j))}, \quad n \geq m,$$

де z – змінна Лорана.

tf - Створює модель передатної функції

Функція **tf** створює неперервну модель системи – передатні функції

$$W(p) = \frac{\sum_{i=0}^m b_{m-i} p^{m-i}}{\sum_{j=0}^n a_{n-j} p^{n-j}}, \quad n \geq m,$$

де p – змінна Лапласа.

sys = tf(num,den)

num – вектор коефіцієнтів чисельника передатної функції;

den – вектор коефіцієнтів знаменника передатної функції.

Функція **tf** створює дискретну модель системи – передатні функції

$$W(z) = \frac{\sum_{i=0}^m b_{m-i} z^{m-i}}{\sum_{j=0}^n a_{n-j} z^{n-j}}, \quad n \geq m,$$

де z – змінна Лорана.

sys = tf(num,den,Ts)

num – вектор коефіцієнтів чисельника передатної функції;

den – вектор коефіцієнтів знаменника передатної функції.

Ts – період квантування.

dss - Визначає описувач моделі простору стану;

filt - Визначає цифровий фільтр;

set – Установка / модифікація атрибутів LTI - моделі;

ltiprops - Детальна довідка про атрибути LTI - моделей.

Процедури класичної графіки

acker - розміщення полюсів ОМ – системи.

Синтаксис

k = acker(A,b,p)

Опис

Вхідні параметри:

A, b – матриці аналізу станів;

p – вектор, елементами якого є бажані полюси системи.

Вихідні параметри:

k – матричний коефіцієнт підсилення.

place - розміщення полюсів ММ-системи.

Синтаксис

K = place(A,B,p)

[K,prec,message] = place(A,B,p)

Опис

Вхідні параметри:

A, B – матриці аналізу станів;

p – вектор, елементами якого є бажані полюси системи.

Вихідні параметри:

k – матричний коефіцієнт підсилення;

$prec$ – число точних десяткових цифр у фактичних полюсах замкненої системи;

$message$ – повідомлення, що попереджує проте, що отриманий полюс відрізняється на 10 % від заданого.

rlocus - Діаграма Еванса розміщення коренів;

rlocfind - Інтерактивне визначення ланки завданням розташування коренів;

estim - Створює оцінювач по заданому коефіцієнту передачі оцінювача;

reg - Створює регулятор по заданій матриці зворотного зв'язку і коефіцієнтам оцінювача.

Інструменти проектування LQG

lqr – проектування неперервного лінійно-квадратичного регулятора зворотного зв'язку.

Синтаксис

$$[K, S, e] = \text{lqr}(\text{sys}, Q, R)$$

$$[K, S, e] = \text{lqr}(\text{sys}, Q, R, N)$$

$$[K, S, e] = \text{lqr}(A, B, Q, R, N)$$

Опис

Вхідні параметри:

A, B – матриці аналізу станів;

Q, R, N – вагові матриці (параметри критерію якості);

sys – створена за допомогою функції ss система у просторі станів.

Вихідні параметри:

K – матричний коефіцієнт підсилення;

S – розв'язок рівнянь Рікатті;

e – власні значення замкненої системи.

dlqr – проектування дискретного лінійно-квадратичного регулятора зворотного зв'язку.

Синтаксис

$$[K, S, e] = \text{dlqr}(\text{sys}, Q, R)$$

$$[K, S, e] = \text{dlqr}(A, B, Q, R)$$

$$[K, S, e] = \text{dlqr}(A, B, Q, R, N)$$

Опис

Вхідні параметри:

A, B – матриці аналізу станів;

Q, R, N – вагові матриці (параметри критерію якості);

sys – створена за допомогою функції ss система у просторі станів.

Вихідні параметри:

K – матричний коефіцієнт підсилення;

S – розв'язок рівнянь Рікатті;

e – власні значення замкненої системи.

lqry – проектування неперервного та дискретного лінійно-квадратичного регулятора зворотного зв'язку з вихідним зважуванням (по виходу системи).

Синтаксис

$$[K, S, e] = \text{lqry}(\text{sys}, Q, R)$$

$$[K, S, e] = \text{lqry}(\text{sys}, Q, R, N)$$

Опис

Вхідні параметри:

Q, R, N – вагові матриці (параметри критерію якості);

sys – створена за допомогою функції ss система у просторі станів (якщо система неперервна, то регулятор неперервний, а якщо дискретна, то дискретний).

Вихідні параметри:

K – матричний коефіцієнт підсилення;

S – розв'язок рівнянь Рікатті;

e – власні значення замкненої системи.

lqrd – проектування дискретного лінійно-квадратичного регулятора зворотного зв'язку для неперервної системи.

Синтаксис

$[Kd, S, e] = \text{lqrd}(A, B, Q, R, Ts)$
 $[Kd, S, e] = \text{lqrd}(A, B, Q, R, N, Ts)$

Опис

Вхідні параметри:

A, B – матриці аналізу станів;

Q, R, N – вагові матриці (параметри критерію якості);

Ts – період квантування.

Вихідні параметри:

Kd – матричний коефіцієнт підсилення;

S – розв’язок рівнянь Рікатті;

e – власні значення замкненої системи.

kalman – Дискретний та неперервний фільтр Калмана;

Синтаксис

$[kest, L, P] = \text{kalman}(Sys, Qn, Rn, Nn)$

$[kest, L, P, M, Z] = \text{kalman}(Sys, Qn, Rn, Nn) \% \text{ тільки дискретний час}$

$[kest, L, P] = \text{kalman}(Sys, Qn, Rn, Nn, sensors, known)$

Опис

Вхідні параметри:

Sys – створена за допомогою функції ss система у просторі станів (якщо система неперервна, то фільтр неперервний, а якщо дискретна, то дискретний);

Qn, Rn, Nn – вагові матриці (параметри критерія якості).

Вихідні параметри:

$kest$ – спроектований фільтр Калмана;

L – матричний оптимальний коефіцієнт підсилення фільтра;

P – розв’язок рівнянь Рікатті;

M – оновлений матричний оптимальний коефіцієнт підсилення фільтра;

Z – апостеріорна оцінка.

kalmd - Дискретний фільтр Калмана для неперервної системи;

Синтаксис

$[kest, L, P, M, Z] = \text{kalmd}(sys, Qn, Rn, T)$

Опис

Вхідні параметри:

Sys – створена за допомогою функції ss система у просторі станів (якщо система неперервна, то фільтр неперервний, а якщо дискретна, то дискретний);

Qn, Rn – вагові матриці (параметри критерію якості);

T – період квантування.

Вихідні параметри:

$kest$ – спроектований фільтр Калмана;

L – матричний оптимальний коефіцієнт підсилення фільтра;

P – розв’язок рівнянь Рікатті;

M – оновлений матричний оптимальний коефіцієнт підсилення фільтра;

Z – апостеріорна оцінка.

lqgreg - Формувач ЛКГ – регулятора по ЛК – коефіцієнтах і фільтру Калмана;

Синтаксис

$r\text{lqg} = \text{lqgreg}(kest, k)$

$r\text{lqg} = \text{lqgreg}(kest, k, 'current') \% \text{ тільки дискретний час}$

$r\text{lqg} = \text{lqgreg}(kest, k, controls)$

Опис

Вхідні параметри:

$kest$ – спроектований фільтр Калмана;

k – матричний коефіцієнт підсилення;

$'current'$ – вихідний сигнал має вигляд (тільки для дискретного часу)

$$U[n] = -K\hat{X}[n|n-1].$$

$controls$ – оцінювач має додаткові входи об’єкта U_d та Y_V .

Вихідні параметри:

$r\text{lqg}$ – спроектований ЛКГ.

Розв’язання матричних рівнянь

lyap – Розв’язання неперервних рівнянь Ляпунова;

Синтаксис та опис

X = lyp(A,Q)

Розв'язує рівняння вигляду

$$AX + XA^T + Q = 0.$$

X = lyp(A,B,C)

Розв'язує загальну форму рівняння Ляпунова (також називається рівнянням Сільвестра)

$$AX + XB + C = 0.$$

X = lyp(A,Q,[],E)

Розв'язує рівняння вигляду

$$AXE^T + EXA^T + Q = 0.$$

dlyap – Розв'язання дискретних рівнянь Ляпунова;

Синтаксис та опис

X = dlyap(A,Q)

Розв'язує рівняння вигляду

$$AXA^T - X + Q = 0.$$

X = dlyap(A,B,C)

Розв'язує загальну форму рівняння Ляпунова (також називається рівнянням Сільвестра)

$$AXB - X + C = 0.$$

X = dlyap(A,Q,[],E)

Розв'язує рівняння вигляду

$$AXA^T - EXE^T + Q = 0.$$

care – Розв'язання неперервних алгебраїчних рівнянь Рікати;

Синтаксис

[X,L,G] = care(A,B,Q)

[X,L,G] = care(A,B,Q,R,S,E)

[X,L,G,report] = care(A,B,Q,...)

[X1,X2,D,L] = care(A,B,Q,...,'factor')

Обчислює рівняння вигляду

$$A^T XE + E^T XA - (E^T XB + S)R^{-1}(B^T XE + S^T) + Q = 0$$

Якщо у звертанні пропущені входні параметри R, S та E , то по замовчуванню їм присвоюються наступні значення $R = I, S = 0$ та $E = I$ (I – одинична матриця) і рівняння набуває вигляду

$$A^T X + XA - XBB^T X + Q = 0$$

Вхідні параметри:

A, B, E – матриці аналізу станів;

Q, R, S – вагові матриці;

$factor$ – параметр функції.

Опис вихідних параметрів:

X – розв'язок рівняння Рікати;

$L = eig(A - BG, E)$ – власні значення замкненої системи;

$G = R^{-1}(B^T XE + S^T)$ – матричний коефіцієнт підсилення;

$report$ – RR норма Фробеніуса матриці відносних залишків;

X_1, X_2 – дві матриці такі, що $X = D(X_2/X_1)D$;

D – діагональна матриця.

dare – Розв'язання дискретних алгебраїчних рівнянь Рікати;

Синтаксис

[X,L,G] = dare(A,B,Q,R)

[X,L,G] = dare(A,B,Q,R,S,E)

[X,L,G,report] = dare(A,B,Q,...)

[X1,X2,L,report] = dare(A,B,Q,...,'factor')

Обчислює рівняння вигляду

$$A^T XA - E^T XE - (A^T XB + S)(B^T XB + R)^{-1}(B^T XA + S^T) + Q = 0$$

або, що еквівалентно (якщо R не сингулярна)

$$E^T XE = F^T XF - F^T XB(B^T XB + R)^{-1}B^T XF + Q - SR^{-1}S^T = 0,$$

де $F = A - BR^{-1}S$.

Якщо у звертанні пропущені входні параметри R, S та E , то по замовчуванню їм присвоюються наступні значення $R = I, S = 0$ та $E = I$ (I – одинична матриця) і рівняння набуває вигляду

$$A^T XA - X - A^T XB(B^T XB + R)^{-1}B^T XA + Q = 0.$$

Вхідні параметри:

A, B, E – матриці аналізу станів;

Q, R, S – вагові матриці;

$factor$ – параметр функції.

Опис вихідних параметрів:

X – розв’язок рівняння Рікатті;

$L = eig(A - BG, E)$ – власні значення замкненої системи;

$G = (B^T X B + R)^{-1} (B^T X A + S^T)$ – матричний коефіцієнт

підсилення;

$report$ – RR норма Фробеніуса матриці відносних залишків;

X_1, X_2 – дві матриці такі, що $X = D(X_2/X_1)D$;

D – діагональна матриця.

Вилучення даних

ssdata - Вилучення матриць простору стану;

zpkdata - Вилучення даних про нулі / полюси / коефіцієнти передачі;

tfdata - Вилучення чисельника (чисельників) і знаменника (знаменників) передатної функції;

dssdata - Отримання інформації про версію описувача SSDATA;

get - Отримання інформації про значення властивостей LTI-моделі.

Отримання інформації про окремі характеристики моделі

class - Про тип моделі ('ss', 'zpk' або 'tf');

size - Про розмірність матриць входу і виходу;

isempty - Перевірка, чи є LTI-модель порожньою;

isct - Перевірка, чи є модель неперервної;

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-модель моделлю заданого типу.

Перетворення системи

ss - Перетворення в простір стану;

zpk - Перетворення в нулі / полюси / коефіцієнти передачі;

tf - Перетворення в передатні функції;

c2d - Перетворення з неперервного часу в дискретний;

d2c - Перетворення з дискретного часу в неперервний;

d2d - Перевизначення дискретної системи або додавання затримок вхідних дій.

"Арифметичні" операції

$+$ і $-$ - Додавання і віднімання LTI-систем (паралельне з'єднання);

$*$ - Множення LTI-систем (послідовне з'єднання);

\backslash - Ліве ділення - $sys1 \backslash sys2$ рівносильне $inv(sys1) * sys2$;

$/$ - Праве ділення - $sys1 / sys2$ рівнозначно $sys1 * inv(sys2)$;

$'$ - Перетранспонування;

$.'$ - Транспонування карти входу / виходу;

$[\dots]$ – Горизонтальне / вертикальне об'єднання LTI-систем;

inv - Обертання LTI-системи.

Моделі динаміки

pole, eig - Полюси системи;

tzero - Нулі системи;

pzmap - Карта нулів-полюсів;

dcgain - Коефіцієнт передачі при нульовій (низькій) частоті;

norm - Норми LTI – систем;

covar - Коваріація відгуку на білий шум;

damp - Частота власних коливань і демпфування по полюсах системи;

esort - Сортування полюсів неперервної системи по їх дійсним частинам;

dsort - Сортування полюсів дискретної системи за їх модулями;

pade - Апроксимація Паде затримок за часом;

Моделі простору стану

rss, drss - Генерування випадкових моделей простору стану;
ss2ss - Перетворення змінних стану;
canon - Канонічна форма простору стану;
ctrb, obsv - Матриці керованості і спостережуваності;
gram - Визначники Грама керованості і спостережуваності;
ssbal - Діагональне балансування матриць простору стану;
balreal - Балансування входу-виходу на основі визначника Грама;
modred - Редукція стану моделі;
minreal - Мінімальна реалізація і скорочення нулів і полюсів;
augstate - Збільшення виходу за рахунок приєднання станів.

Відгук в часі

step - Відгук на одиничний стрибок;
impulse - Відгук на одиничний імпульс;
initial - Відгук на задані початкові умови стану;
lsim - Відгук на довільні входи;
ltiview - Аналіз відгуків за допомогою графічного інтерфейсу;
gensig - Генерує періодичні сигнали для LSIM;
stepfun - Генерує одиничний стрибок.

Частотний відгук

bode - Діаграма Бодє частотного відгуку (АЧХ і ФЧХ);
sigma - Частотний графік сингулярних значень;
nyquist - Діаграма Найквіста;
nichols - Діаграма Нікольса;
ltiview - Аналіз відгуків за допомогою графічного інтерфейсу;
evalfr - Розрахунок частотного відгуку на заданій частоті;
freqresp - Частотний відгук над сіткою частот;
margin - Запаси по фазі і амплітуді.

Об'єднання систем

append - Об'єднання LTI-систем шляхом об'єднання входів і виходів;
parallel - Узагальнене паралельне з'єднання (див. також +);
series - Узагальнене послідовне з'єднання (див. також *);
feedback - Зворотне з'єднання двох систем;
star - Сполучення зіркою Редхеффера;
connect - Отримання ss-моделі з опису блок-схеми.

Розв'язання систем нелінійних рівнянь (пакет Optimization)

$$F(X) = 0,$$

де X – вектор змінних.

Синтаксис та опис

- **x = fsolve(fun,x0)** починає з точки x_0 та розв'язує описане в fun рівняння.
- **x = fsolve(fun,x0,options)** проводить оптимізацію з визначеними параметрами оптимізації.
- **x = fsolve(fun,x0,options,P1,P2,...)** передає залежні від типу задачі параметри P1, P2 і т.д. безпосередньо в функцію fun . Передає порожні матриці по замовчуванню.
- **[x,fval] = fsolve(fun,x0)** повертає значення цільової функції fun як розв'язок від x .
- **[x,fval,exitflag] = fsolve(...)** повертає значення exitflag , що містить опис вихідних умов.
- **[x,fval,exitflag,output] = fsolve(...)** повертає структурний вихід з інформацією про оптимізацію.
- **[x,fval,exitflag,output,jacobian] = fsolve(...)** повертає Якобіан від fun як розв'язок від x .

Опис вихідної умови exitflag:

- 1 – функція збігається до розв'язку від x ;
- 2 – зміна x була менша ніж вказана похибка;
- 3 – зміна в залишку була менша ніж вказана похибка;

Література

- 4 – величина напрямку пошуку була менша ніж вказана похибка;
- 0 – максимальне число оцінок функції options.FunEvals або ітерацій options.MaxIter було перевищено;
- –1 – алгоритм був закінчений функцією виводу;
- –2 – алгоритм збігається до точки, що не є розв’язком;
- –3 – довірчій радіус став дуже малим;
- –4 – лінійний пошук не може достатньо зменшитися, залишковий за поточним напрямком пошуку.

Структура, що містить інформацію про оптимізацію output:

- **iterations** – число виконаних ітерацій;
- **funcCount** – число розрахунків функцій;
- **algorithm** – алгоритм, що використовується.
- **cgiterations** – число PCG ітерацій (тільки для великомасштабного алгоритму);
- **stepsize** – остаточно прийнятий розмір кроку (тільки для середньомасштабного алгоритму);
- **firstorderopt** – вимірювання оптимальності першого порядку: норма градієнта як розв’язок від x .

1. The MathWorks, Inc. 3 Apple Hill Drive Natick, MA 01760-2098. Control System Toolbox. © COPYRIGHT 2001–2005 by The MathWorks, Inc.
2. The MathWorks, Inc. 3 Apple Hill Drive Natick, MA 01760-2098. Simulink Response Optimization User’s Guide. © COPYRIGHT 2004–2006 by The MathWorks, Inc.
3. The MathWorks, Inc. 3 Apple Hill Drive Natick, MA 01760-2098. Optimization Toolbox User’s Guide. © COPYRIGHT 1990–2006 by The MathWorks, Inc.
4. Conn A. R., N. I. M. Gould and Ph.L. Toint Trust-Region Methods, MPS/SIAM Series on Optimization, SIAM and MPS. – 2000.
5. Nocedal J. and S. J. Wright Numerical Optimization, Springer Series in Operations Research, Springer Verlag. – 1999.
6. Powell M. J. D. A Fortran Subroutine for Solving Systems of Nonlinear Algebraic Equations, Numerical Methods for Nonlinear Algebraic Equations, (P. Rabinowitz, ed.), Ch.7. – 1970.
7. Рей У. Методы управления технологическими процессами: Пер. с англ. – М.: Мир, 1983. – 368 с., ил.
8. Сейдж Э. П., Уайт Ч. С., III. Оптимальное управление системами: Пер. с англ./ Под ред. Б. Р. Левина. – М.: Радио и связь, 1982. – 392 с., ил. – (Второе изд.: США, 1977).
9. Жученко А. І. Математичні моделі цифрових систем керування. Нав. Посібник. – К.: ІЗМН 1997. 240 с.
10. Ладієва Л. Р. Оптимальне керування системами. – К.: НМЦ ВО, 2000. 187 с.
11. Жученко А. І., Ладієва Л. Р., Дубік Р. М. Нелінійне програмування з використанням MATLAB. – К.: НТУУ “КПІ”, 2007. – 140 с.
12. Лазарев Ю. Ф. Начала программирования в среде MatLAB: Учебное пособие. - К.: НТУУ "КПИ", 2003. – 424 с.
13. Ануфриев И. Е., Смирнов А. Б., Смирнова Е. Н. MATLAB 7. – СПб.: БХВ – Петербург, 2005. – 1104 с.

14. Дорф Р. Современные системы управления / Р. Дорф, Р. Бишоп; Пер. с англ. Б. И. Копылова. – М.: Лаборатория Базовых Знаний, 2004. – 832 с.: ил.
15. Куо Б. Теория и проектирования цифровых систем управления: Пер. с англ. – М.: Машиностроение, 1986. – 448 с., ил.

Зміст

Вступ	
1. Оптимальні системи керування	
1.1. Проектування оптимального лінійно квадратичного регулятора.....	
1.2. Проектування оптимального лінійно квадратичного регулятора за виходом системи.....	
1.3. Проектування оптимального лінійно квадратичного регулятора з інтегральною складовою	
1.4. Проектування оптимального дискретного лінійно квадратичного регулятора	
1.5. Проектування оптимального дискретного лінійно квадратичного регулятора по виходу системи	
1.6. Проектування оптимального дискретного лінійно квадратичного регулятора для неперервної системи	
2. Оптимальне оцінювання стану	
2.1. Неперервний фільтр Калмана.....	
2.2. Дискретний фільтр Калмана.....	
2.3. Дискретний фільтр Калмана для неперервної системи	
3. Проектування лінійно квадратичного гауссіана за заданими коефіцієнтами фільтру Калмана	
4. Розміщення полюсів за допомогою зворотного зв'язку за станом	
4.1. Задане розміщення полюсів за допомогою зворотного зв'язку за станом.....	
4.2. Застосування формули Аккермана	
5. Нелінійні системи	
6. Розв'язання систем нелінійних рівнянь	
7. Розв'язання систем нелінійних диференційних рівнянь	
Додаток	
Література	

Жученко Анатолій Іванович
Ладієва Леся Ростиславівна
Дубік Роман Миколайович

**Динамічна оптимізація з використанням MATLAB та
SIMULINK**