

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

МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИКОНАННЯ КУРСОВОЇ РОБОТИ З ДИСЦИПЛІНИ
«СУЧАСНІ ПРОБЛЕМИ АВТОМАТИЗОВАНОГО
УПРАВЛІННЯ»

Дніпро УДХТУ 2020

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

МЕТОДИЧНІ ВКАЗІВКИ

ДО ВИКОНАННЯ КУРСОВОЇ РОБОТИ З ДИСЦИПЛІНИ
«СУЧАСНІ ПРОБЛЕМИ АВТОМАТИЗОВАНОГО
УПРАВЛІННЯ»

Затверджено на засіданні кафедри
комп'ютерно-інтегрованих
технологій і метрології.
Протокол № 9 від 26.06.2019 р.

Дніпро УДХТУ 2020

Методичні вказівки до виконання курсової роботи з дисципліни «Сучасні проблеми автоматизованого управління» / Укл. : Г.І. Манко. – Дніпро : УДХТУ, 2020. – 39 с.

Укладач Г.І. Манко, канд. техн. наук

Відповідальний за випуск О.П. Мисов, канд. техн. наук

Навчальне видання

Методичні вказівки
до виконання курсової роботи з дисципліни
«Сучасні проблеми автоматизованого управління»

Укладач: МАНКО Геннадій Іванович

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

Підп. до друку _____. Формат 60x84 1/16. Папір ксерокс. Друк
різограф. Умовн. – друк. арк. _____ Облік. – вид. арк. _____ Тираж
_____ прим. Зам. № _____

Свідотство ДК №303 від 27.12.2000.

УДХТУ, 49005, Дніпро – 5, пр – т Гагаріна, 8

Видавничо – поліграфічний комплекс ІнКомЦентру

ВСТУП

Методичні вказівки призначені для використання студентами спеціальності 151 під час вивчення дисципліни «Сучасні проблеми автоматизованого управління» і мають за мету опанування ними сучасних методів і підходів до моделювання, аналізу і синтезу систем управління для різних варіантів умов, початкових даних і ситуацій. Студенти повинні знати сучасні методи рішення завдань аналізу і синтезу систем, основні способи і принципи представлення функціональних і динамічно-імітаційних моделей і прийоми їх алгоритмізації, особливості і порівняльні характеристики застосування сучасних підходів для вирішення різних практичних завдань автоматичного управління, вміти визначати оптимальні засоби і раціональні шляхи ефективного виконання завдань управління, розробляти математичні моделі об'єктів і систем різних типів і класів, проводити дослідження їх властивостей на математичних моделях.

Суть курсової роботи полягає в практичному застосуванні знань, отриманих у курсі «Сучасні проблеми автоматизованого управління», а також курсах «Теорія автоматичного управління», «Програмування та алгоритмізація» і деяких інших, для вирішення завдання побудови і дослідження системи автоматичного управління динамічним об'єктом.

У процесі виконання курсової роботи студенти повинні користуватися рекомендованою літературою, перелік якої наведений наприкінці методичних вказівок.

1 ТЕМАТИКА І ЗМІСТ КУРСОВОЇ РОБОТИ

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

Процес синтезу складається з:

- отримання моделі динамічного об'єкта у формі системи диференціальних рівнянь;
- розрахунку передатної функції об'єкта;
- переходу до лінійної моделі у формі рівнянь стану;
- вибору структури управління об'єктом;
- параметричного синтезу модального регулятора по повному вектору стану;
- проектування спостерігача стану об'єкта;
- побудови графічного інтерфейсу користувача (GUI) для автоматизації розрахунків;
- аналізу точносних і динамічних характеристик спроектованої системи.

2 ЗАВДАННЯ НА ПРОЕКТУВАННЯ

2.1 Склад курсової роботи

Для однієї з наведених нижче моделей систем автоматичного регулювання необхідно виконати наступне:

а) записати модель у формі передатної функції (всі моделі спочатку повинні бути отримані в символічній формі, а потім переведені в числову);

б) створити графічний інтерфейс користувача для інтерактивного синтезу безперервного модального регулятора по повному вектору стану, що забезпечує задану якість перехідних процесів (перерегулювання σ , час перехідного процесу t_n) з точністю $\pm 10\%$. Числові значення σ і t_n брати з таблиць вихідних даних (п. 2.2) відповідно варіанту, який задається викладачем;

в) визначити спостережуваність об'єкта і, у випадку позитивного висновку, розрахувати значення коефіцієнтів зворотних зв'язків спостерігача, сформулювати модель спостерігача и побудувати графік перехідного процесу;

г) перевірити результати моделюванням у SIMULINK.

2.2 Вибір варіанта завдання для курсової роботи

2.2.1 Система автоматичного регулювання температури

Схема системи показана на рис. 2.1.

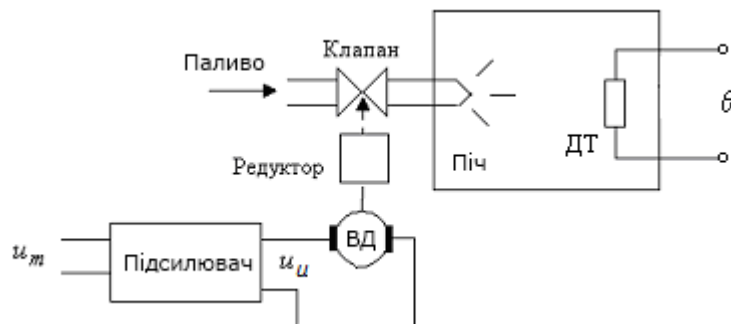


Рис. 2.1 – Схема системи автоматичного регулювання температури

Лінеаризовані рівняння елементів системи мають наступний вигляд.

Піч:

$$T_0 \cdot \frac{d}{dt} \theta + \theta = k_0 \cdot \mu$$

Підсилювач потужності:

$$T_m \frac{du_u}{dt} + u_u = k_m \cdot u_m$$

Двигун з редуктором:

$$T_{дв} \cdot \frac{d^2}{dt^2} \mu + \frac{d}{dt} \mu = k_{дв} \cdot u_u$$

У приведених рівняннях:

θ – температура печі (регульована величина);

u_m – вхідна напруга підсилювача потужності;

u_u – якірня напруга виконавчого двигуна;

μ – переміщення клапана в паливній магістралі.

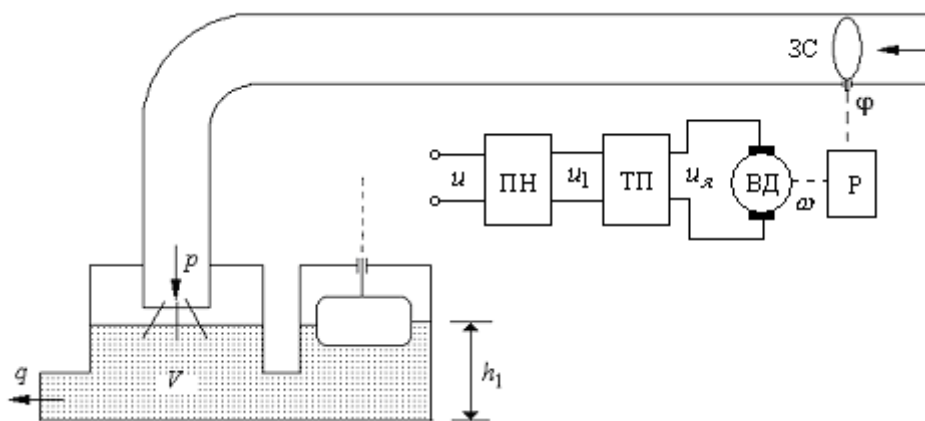
Числові дані для схеми наведені в табл. 2.1.

Таблиця 2.1

Варіант	T_0, c	$k_0, \text{град/м}$	T_m	k_m	$T_{\text{ДВ}}, c$	$k_{\text{ДВ}}, \text{см/Вс}$	$\sigma, \%$	t_n, c
1	2,0	5,0	0,01	20	0,05	0,10	20	0.1
7	1,8	4,5	0,02	15	0,04	0,15	15	0.15
13	1,7	4,0	0,02	10	0,03	0,20	10	0.2
19	1,5	3,5	0,03	15	0,02	0,25	20	0.1

2.2.2 Схема регулювання рівня рідини в відкритому баку

Схема системи показана на рис. 2.2.



ПН – підсилювач напруги; ТП – тиристорний перетворювач (підсилювач потужності); ВД – виконавчий двигун постійного струму; Р – знижуючий редуктор; ЗС – засувка

Рис. 2.2 – Схема системи автоматичного регулювання рівня рідини у відкритому баку

Лінеаризовані рівняння елементів системи мають наступний вигляд.

Підсилювачі:

$$u_1 = k_u \cdot u, \quad u_y = k_m \cdot u_1.$$

Двигун і редуктор:

$$T_{\text{ДВ}} \frac{d\omega}{dt} + \omega = k_{\text{ДВ}} \cdot u_y, \quad \frac{d\phi}{dt} = k_p \cdot \omega.$$

Трубопровід і бак:

$$T_1 \frac{dp}{dt} + p = k_1 \varphi, \quad h_1 = k_2 \cdot (p - q).$$

У приведених рівняннях

h_1 – рівень рідини у баку,

p – витрата рідини, що надходить,

q – витрата споживаної рідини,

φ – кут повороту засувки,

u – вхідна напруга підсилювача напруги,

u_1 – напруга на виході підсилювача напруги,

$u_{\text{я}}$ – якірна напруга двигуна постійного струму,

ω – кутова швидкість двигуна.

Числові дані для схеми приведені в табл. 2.2.

Таблиця 2.2

Варі- ант	k_u	k_m	$k_{\text{дв}}$, град/Вс	$T_{\text{дв}}$, с	k_p	k_1	T_1 , с	k_2	σ , %	t_n , с
2	10	1	2	0,02	0,001	10	0,3	3,5	20	0,5
8	20	20	2	0,03	0,01	100	0,2	3	15	0,5
14	4	20	1	0,02	0,002	1	0,5	4	10	1
20	4	20	1	0,02	0,002	1	0,5	4	10	1

2.2.3 Система автоматичного регулювання тиску в ресивері

Схема системи показана на рис. 2.3.

Об'єктом управління в досліджуваній системі є пневматичний пристрій – ресивер, в якому необхідне значення тиску p встановлюється за рахунок повороту засувки трубопроводу, розташованої на виході редуктора.

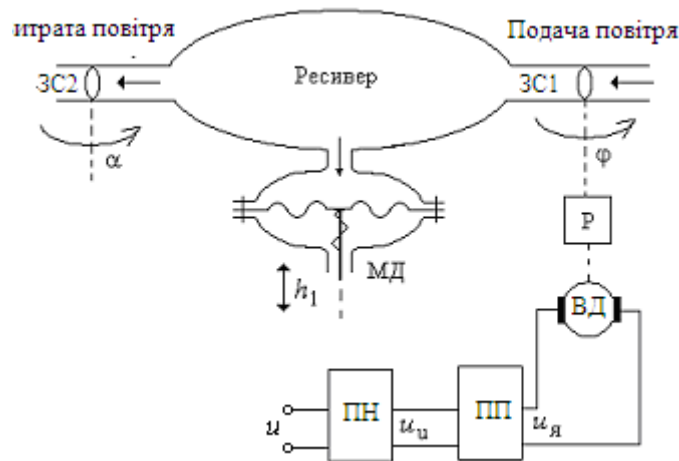
Лінеаризовані рівняння елементів системи мають вигляд.

Підсилювачі

$$u_u = k_u \cdot u; \quad T_m \frac{du_{\text{я}}}{dt} + u_{\text{я}} = k_m \cdot u_u.$$

Двигун і редуктор

$$T_{\text{дв}} \frac{d\omega}{dt} + \omega = k_{\text{дв}} \cdot u_{\text{я}}, \quad \frac{d\varphi}{dt} = k_p \cdot \omega.$$



ПН – підсилювач напруги; ПП – підсилювач потужності; ВД – виконавчий двигун постійного струму; Р – знижуючий редуктор; МД – мембранний датчик тиску; ЗС1, ЗС2 – поворотні засувки

Рис. 2.3 – Схема системи автоматичного регулювання тиску в ресивері

Ресивер і мембранний датчик

$$T_p \frac{dp}{dt} + p = k_\varphi \cdot \varphi - k_\alpha \cdot \alpha, \quad T_1 \frac{dh}{dt} + h = k_1 \cdot p.$$

У приведених рівняннях:

h – вихідний сигнал мембранного датчика;

p – тиск повітря в ресивері;

q_1, q_2 – надходження і витрата повітря;

φ, α – кути повороту вхідної і вихідної засувки;

u – напруга на вході підсилювача напруги;

u_u – напруга на виході підсилювача напруги;

u_y – якірна напруга двигуна постійного струму;

ω – кутова швидкість двигуна.

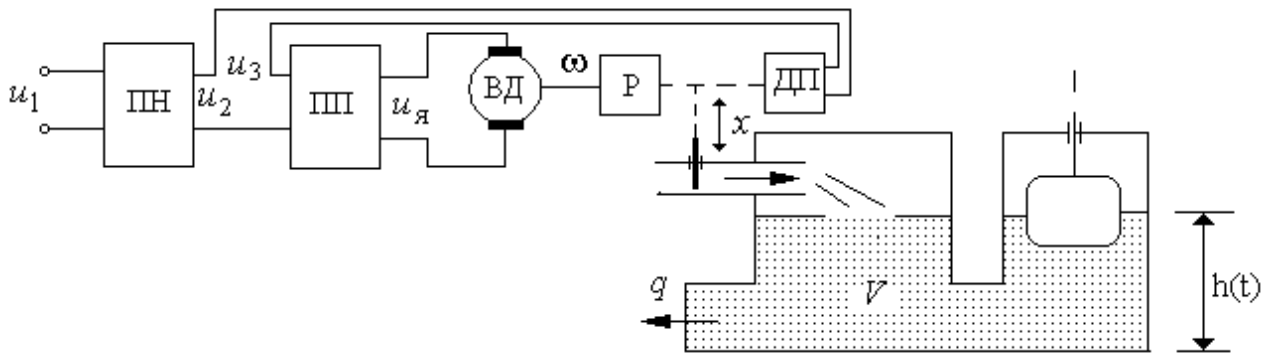
Числові дані для схеми приведені в табл. 2.3.

Таблиця 2.3

Варіант	k_u	k_m	$T_m, \text{с}$	$k_{дв}, \text{град/Вс}$	$T_{дв}, \text{с}$	k_p	$k_\varphi, \text{Пс/град}$	$k_\alpha, \text{Пс/град}$	k_1	$T_p, \text{с}$	$T_1, \text{с}$	$\sigma, \%$	$t_n, \text{с}$
3	8	5	0,03	10	0,05	0,01	50	1	0,1	2	0,1	15	0,2
9	5	4	0,02	5	0,1	0,02	20	2	0,25	3	0,2	20	0,4
15	2	10	0,03	6	0,08	0,05	10	1	0,2	2	0,15	10	0,3
21	4	8	0,02	7	0,06	0,03	30	2	0,15	3	0,12	15	0,2

2.2.4 Система регулювання рівня рідини у баку

Схема системи показана на рис. 2.4.



ПН - підсилювач напруги і випрямляч; ПП - підсилювач потужності; ВД - виконавчий двигун постійного струму; ДП - датчик положення засувки; Р - знижуючий редуктор

Рис. 2.4 – Схема системи автоматичного регулювання рівня рідини у баку

Лінеаризовані рівняння елементів системи мають наступний вигляд.

Задаючий пристрій:

$$u_1 = k \cdot h_0.$$

Підсилювачі:

$$u_2 = k_u \cdot u_1, \quad u_я = k_m \cdot (u_2 - u_3).$$

Двигун з редуктором:

$$T_{дв} \frac{d\omega}{dt} + \omega = k_{дв} \cdot u_я, \quad \frac{dx}{dt} = k_p \cdot \omega,$$

Місцевий зворотний зв'язок по положенню засувки:

$$u_3 = k_{oc} x.$$

Трубопровід і бак:

$$T_3 \frac{dp}{dt} + p = k_3 x, \quad \frac{dh}{dt} = k_2 \cdot (p - q).$$

У приведених рівняннях:

$u_1, u_2, u_я$ – напруги постійного струму;

h_0, h – задане і дійсне значення рівня рідини в баку;

p – витрата рідини, що надходить;

q – витрата споживаної рідини;

ω – кутова швидкість двигуна;

x – переміщення засувки.

Числові дані для схеми приведені в табл. 2.4.

Таблиця 2.4

Варіант	k , град/м	k_u	k_m	$k_{дв}$, рад/Вс	$T_{дв}$	k_p	k_3 , мм ²	T_3 , с	$k_{ос}$, В/м	k_2 , м/с л	σ , %	t_n , с
4	0,2	50	1	10	0,2	0,001	0.5	5	0.1	0.01	15	1
10	0,1	5	1	10	0,1	0,01	0.5	3	0,2	0.05	10	0,5
16	0,2	100	1	5	0,3	0,002	0.5	10	0.3	0.03	20	1,5
22	0,1	50	1	5	0,1	0,001	0.5	7	0.2	0.02	15	1,2

2.2.5 Система управління температурою електропечі

Електрична піч складається з послідовно сполучених тиристорного перетворювача ТП (підсилювача потужності), електродвигуна ЕД, заслінки, що регулює надходження холодного повітря в піч, і власне печі з нерегульованим нагрівальним елементом (рис. 2.5).

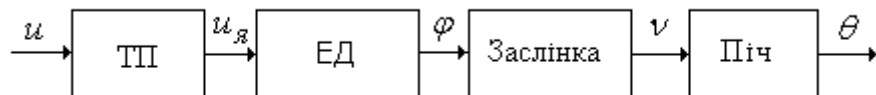


Рис. 2.5 – Структура системи автоматичного управління електропіччю

Лінеаризована модель системи управління описується наступним набором рівнянь.

Тиристорний перетворювач:

$$T_1 \frac{du_{я}}{dt} + u_{я} = k_1 u.$$

Електродвигун з редуктором:

$$T_2 \frac{d^2\varphi}{dt^2} + \frac{d\varphi}{dt} = k_2 u_{я}.$$

Піч:

$$T_3 \frac{d\theta}{dt} + \theta = k_3 v$$

Заслінка сполучена з двигуном через редуктор з коефіцієнтом передачі k_4 . Вона є лінійною статичною ланкою, що описується (разом з редуктором) співвідношенням:

$$v = k_4 \varphi.$$

На вході цієї ланки – кут повороту двигуна φ , на виході – інтенсивність охолодження печі v , яку можна виміряти об'ємом повітря, що надходить.

Тут:

θ – температура печі,

$u, u_{я}$ – задаюча і якірна напруга, відповідно,

φ – кут повороту заслінки.

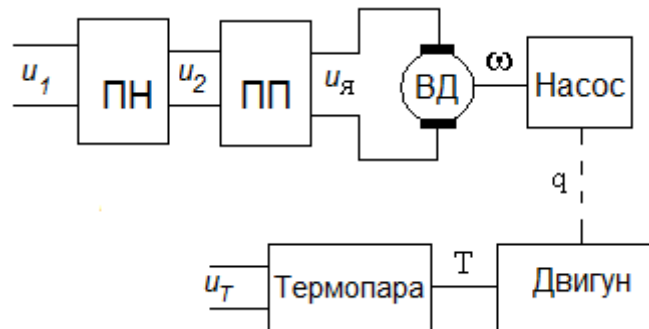
Числові дані для схеми приведені в табл. 2.5.

Таблиця 2.5

Варі- ант	T_1, c	k_1	T_2, c	$k_2, \frac{рад \cdot c}{в}$	T_3, c	$k_3, \frac{град}{л \cdot c}$	$k_4, \frac{л \cdot c}{град}$	$\sigma, \%$	t_n, c
5	0,5	1	2	$2 \cdot 10^{-5}$	50	0,4	10	10	70
11	0,4	4	1,5	$5 \cdot 10^{-4}$	40	0,2	15	15	50
17	0,5	2	2	$2 \cdot 10^{-3}$	50	0,1	20	20	50
23	0,4	3	2,5	3	50	0,3	10	10	60

2.2.6 Система стабілізації охолодження двигуна

Схема системи представлена на рис. 2.6.



ПН – підсилювач напруги; ПП – підсилювач потужності; ВД – виконавчий двигун

Рис. 2.6 – Схема системи стабілізації охолодження двигуна

Лінеаризована модель системи управління описується наступним набором рівнянь.

Підсилювачі

$$u_2 = k_u \cdot u_1; \quad u_{я} = k_m \cdot u_2.$$

Виконавчий двигун постійного струму

$$T_{дв} \frac{d\omega}{dt} + \omega = k_{дв} \cdot u_{я}.$$

Насос

$$T_h \frac{dq}{dt} + q = k_h \omega.$$

Охолоджуюча сорочка двигуна

$$T_p \frac{dT}{dt} + T = k_p q.$$

Термопара

$$T_{tp} \frac{du_T}{dt} + u_T = k_{tp} T$$

У приведених рівняннях:

u_1, u_2 – напруги постійного струму;

$u_{я}$ – якірна напруга ВД;

ω – кутова швидкість ВД;

q – витрата охолоджувальної рідини;

T – температура двигуна;

u_T – вихідна напруга термопари.

Числові дані для схеми приведені в табл. 2.6.

Таблиця 2.6

Варі-ант	k_u	k_m	$k_{дв},$ рад/В·с	$T_{дв},$ с	$k_h,$ м ³ /рад	T_h	$k_p,$ град· с/м ³	T_p	$k_{tp},$ В/град	$T_{tp},$ с	$\sigma, \%$	$t_n, \text{с}$
6	20	2	5	0,5	0,001	0,7	5	5	0.5	2	20	50
12	50	1	7	0,2	0,002	1	2	3	1	1	20	40
18	50	2	4	0,3	0,002	2	0.5	10	0.3	3	15	30
24	30	2	6	0,4	0,001	1,2	3	7	0,7	2	15	40

3 ТЕОРЕТИЧНІ ВІДОМОСТІ

3.1 Модальне управління за станом об'єкта

Характер перехідних процесів в лінійній системі визначається розташуванням коренів її характеристичного рівняння. Рішення однорідного диференціального рівняння n -го порядку має вигляд $y(t) = \sum_{i=1}^n C_i y_i(t)$, де сталі C_i визначаються початковими умовами, а складові $y_i(t)$ («моди») мають

вигляд $y_i(t) = e^{s_i t}$ – при простих або $y_i(t) = P_i(t) e^{s_i t}$ – при кратних коренях (тут $P_i(t)$ – многочлени, міри яких визначаються кратністю кореня).

Тому забезпечення бажаних перехідних процесів в системі може бути досягнуте, якщо характеристичне рівняння передатної функції замкнутої системи має задані корені.

Практично важливим є наступне питання – чи є вирішуваним поставлене завдання модального синтезу? Чи можна розмістити полюси системи у бажаних точках комплексної площини? Відомо, що коли об'єкт повністю керований і усі складові вектору стану доступні виміру (вирішується завдання з повною інформацією про стан), то при законі управління у вигляді лінійної функції від змінних стану $u(t) = -Kx(t)$ вибором K можна полюси замкнутої системи змістити у будь-які бажані точки. Якщо з n змінних стану є вимірюваними m , де $n > m$, то змістити можна m полюсів, або n , але лише в деякі обмежені області. Усе це справедливо, якщо розглядається система з одним входом. Якщо використовується r входів, то зазвичай можна довільно розмістити $m + r$ полюсів.

Рішення цієї задачі зазвичай розглядається для випадку, коли модель об'єкта записана у формі рівнянь стану. Нехай динаміка об'єкту описується рівнянням

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t). \quad (3.1)$$

Вектор стану об'єкта $\mathbf{x}(t)$ вважаємо доступним виміру. Розглянемо закон управління виду

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t). \quad (3.2)$$

Тут \mathbf{K} – матриця коефіцієнтів регулятора, що підлягає визначенню. Замкнута система об'єкт-регулятор описується рівнянням:

$$\dot{\mathbf{x}}(t) = (\mathbf{A} + \mathbf{B}\mathbf{K})\mathbf{x}(t).$$

Необхідно знайти коефіцієнти регулятора (елементи матриці \mathbf{K}), такі, щоб характеристичне рівняння мало задані корені i , відповідно, щоб характеристичний багаточлен $\det(s\mathbf{I}_n - \mathbf{A} + \mathbf{B}\mathbf{K})$ мав задані коефіцієнти.

3.2 Вибір бажаного розташування коренів характеристичного рівняння

Першим завданням, яке необхідно вирішити, приступаючи до процедури модального синтезу, є вибір розташування полюсів замкнутої системи, до якого слід прагнути. Це питання вирішується з урахуванням властивостей конкретного об'єкта.

Взагалі кажучи, вид перехідного процесу визначається не лише полюсами, але і нулями замкнутої системи, проте основний вплив на динаміку роблять полюси системи. Отже, застосовуючи модальний регулятор, ми завжди можемо забезпечити стійкість замкнутої системи, але, в загальному випадку, не можемо

відразу, без підстроювання, вибором тільки коренів характеристичного рівняння в точності гарантувати бажані показники якості регулювання.

Нулі передатної функції при усіх позитивних коефіцієнтах полінома чисельника сприяють перерегулюванню. З іншого боку, вибором коефіцієнтів полінома чисельника можна, наприклад, зменшити час перехідного процесу, або забезпечити інваріантність системи до деяких типів вхідних сигналів.

Для більшої спільності характеристики замкнених систем будуються в нормованому виді з використанням середньгеометричного кореня характеристичного рівняння.

Нехай маємо характеристичне рівняння:

$$A(s) = a_0 s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n = 0. \quad (3.3)$$

Позначимо корені характеристичного рівняння s_1, s_2, \dots, s_n і введемо поняття середньгеометричного кореня характеристичного рівняння:

$$\omega_0 = \sqrt[n]{a_n/a_0} = \sqrt[n]{|s_1 s_2 \dots s_n|}.$$

Поділивши характеристичне рівняння (3.3) на a_0 , його можна записати з використанням середньгеометричного кореня у виді

$$\begin{aligned} A(s) &= s^n + \left(\frac{a_1}{a_0}\right) s^{n-1} + \dots + \left(\frac{a_{n-1}}{a_0}\right) s + \left(\frac{a_n}{a_0}\right) = \\ &= s^n + (a_1 \omega_0^{n-1}) \omega_0 s^{n-1} + (a_2 \omega_0^{n-2}) \omega_0^2 s^{n-2} + \dots + (a_{n-1} \omega_0) \omega_0^{n-1} s + \omega_0^n = \\ &= s^n + c_1 \omega_0 s^{n-1} + c_2 \omega_0^2 s^{n-2} + \dots + c_{n-1} \omega_0^{n-1} s + \omega_0^n, \end{aligned}$$

де $c_k = a_k \omega_0^{n-k}$, $k = 1, \dots, n-1$.

Головним достоїнством нормованого запису є те, що він дозволяє окремо вибрати час перехідного процесу і характер цього процесу. Характер процесу залежить від коефіцієнтів c_k і не залежить від ω_0 . У свою чергу, зміна ω_0 , при вибраних c_k , міняє тільки його тривалість, не міняючи характеру, наприклад, перерегулювання σ . Тому середньгеометричний корінь ω_0 є мірою швидкодії системи.

Запис бажаного характеристичного рівняння в нормованому виді дозволяє використати нормовані стандартні характеристики. В цьому випадку по осі часу відкладається відносний час $\tau = \omega_0 t$.

Далі необхідно вибрати певний розподіл коренів характеристичного рівняння системи. Розглянемо декількох типових розподілів.

Найпростіший спосіб полягає в розміщенні усіх коренів характеристичного рівняння в одній точці на комплексній площині з координатами

$$(-\omega_0, j0),$$

де $\omega_0 > 0$.

Тоді характеристичний поліном обертається у біном Ньютона $(s + \omega_0)^n$, розкриваючи який, отримуємо, при різних n , наступні вирази для характеристичного полінома:

$$s + \omega_0,$$

$$s^2 + 2 \cdot \omega_0 \cdot s + \omega_0^2,$$

$$s^3 + 3 \cdot \omega_0 \cdot s^2 + 3 \cdot \omega_0^2 \cdot s + \omega_0^3,$$

$$s^4 + 4 \cdot \omega_0 \cdot s^3 + 6 \cdot \omega_0^2 \cdot s^2 + 4 \cdot \omega_0^3 \cdot s + \omega_0^4.$$

Біномний розподіл застосовується в тих випадках, коли необхідно забезпечити монотонний перехідний процес. Відповідні розташування коренів характеристичного рівняння і графіки перехідних функцій показані на рис. 3.1. Крива 1 відповідає системі першого порядку, крива 2 – другому і так далі.

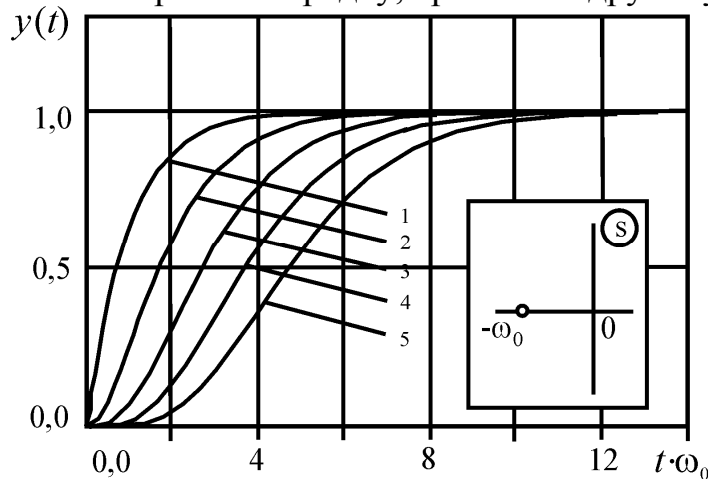


Рис. 3.1 – Розташування коренів характеристичного рівняння і графіки перехідних функцій для біномної схеми розташування коренів при різних порядках системи

Для кожного порядку системи буде свій відносний час перехідного процесу τ_n . Для систем до 4-го порядку включно воно вказане в табл. 3.1.

Таблиця 3.1 – Відносний час перехідного процесу

Порядок системи n	1	2	3	4
Час перехідного процесу τ_n у відносних одиницях (в.о.)	3	4,75	6,3	7,8

Якщо порядок системи n вибраний і є заданим бажаний час перехідного процесу t_n , то середньгеометричний корінь визначається формулою:

$$\omega_0 = \tau_n / t_n \quad (3.4)$$

Таким чином, чим більш висока швидкодія потрібна (чим менше t_n), тим більше за модулем має бути середньгеометричний корінь.

Інший спосіб, запропонований Батервортом [10], полягає в призначенні коренів в точках на півколі радіусу ω_0 у лівій частині комплексної площини, розташованих на рівній кутовій відстані один від одного. За допомогою теореми Вієта, що зв'язує корені рівняння з коефіцієнтами, можна скласти рівняння для характеристичного полінома. Нижче наводяться вирази відповідних характеристичних поліномів з першого по четвертий порядок включно:

$$s + \omega_0,$$

$$s^2 + 1,4 \cdot \omega_0 \cdot s + \omega_0^2,$$

$$s^3 + 2,0 \cdot \omega_0 \cdot s^2 + 2,0 \cdot \omega_0^2 \cdot s + \omega_0^3,$$

$$s^4 + 2,6 \cdot \omega_0 \cdot s^3 + 3,4 \cdot \omega_0^2 \cdot s^2 + 2,6 \cdot \omega_0^3 \cdot s + \omega_0^4.$$

Реакції систем Батерворта (рис. 3.2) на одиничну ступінчасту дію, в порівнянні з аналогічними реакціями біномних систем, є більш коливальними, але у багатьох випадках вони відповідають інтуїтивному уявленню про оптимальний перехідний процес.

Показники універсальних перехідних функцій для різних порядків системи, при розподілі за Батервортом представлені в табл. 3.2.

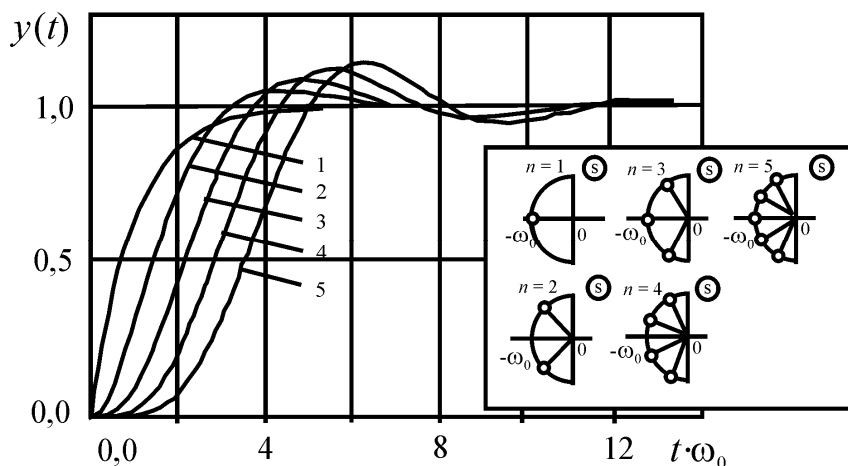


Рис. 3.2 – Розташування коренів характеристичного рівняння і графіки перехідних функцій для схеми Батерворта

Таблиця 3.2 – Показники перехідного процесу

n	1	2	3	4
τ_n , о.е.	3	3	6	7,2
σ , %	–	4,3	8	6,2

З рис. 3.2 і таблиці 3.2 видно, що при $n > 1$ розподіл Батерворта дає швидші, ніж біномний розподіл, процеси з невеликим перерегулюванням.

Середньгеометричний корінь визначається тією ж формулою (3.4).

3.3 Грубість системи

Часто початкова система має лише один або декілька полюсів, що не відповідають заданим вимогам. В цьому випадку може мати місце проблема грубості отримуваних рішень. Правильно синтезований модальний регулятор має бути працездатним не тільки при розрахункових значеннях параметрів, але і в деякому діапазоні відхилень від цих значень. Причому, чим більше цей діапазон, тим краще.

Одній з причин що приводить до не грубих рішення, являється зрушення праворуч полюсів системи, які спочатку лежать значно лівіше заданої області. На рис. 3.3 розміщення полюсів s_1, s_2, s_3, s_4 відповідає нескоректованій системі, а область Q – бажаному розміщенню полюсів. При цьому полюс s_4 відповідає швидко затухаючим стійким рухам в системі. Спроба зрушити цей полюс в область Q зазвичай призводить до недостатньо грубої системи, яка при невеликих змінах коефіцієнтів регулятора може істотно змінити свої властивості і навіть втратити стійкість. З точки зору грубості доцільно не чіпати цей полюс, обмежившись зрушенням в область Q полюсів s_1, s_2, s_3 .

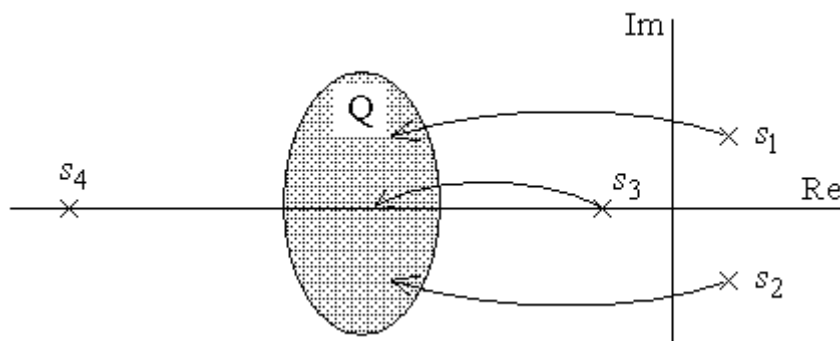


Рис. 3.3 – Зсув полюсів у задану область

Тут можна розділити початкову модель на швидку і повільну частині. Приймається, що швидка частина мало впливає на якість перехідного процесу, і в задані точки зрушуються тільки "повільні" полюси s_1, s_2, s_3 .

3.5 Методика розрахунку параметрів модального регулятора

Нехай для лінійного об'єкту, заданого у формі рівнянь стану (3.1) вимагається вибрати параметри закону управління (3.2) так, щоб замкнута система мала заданий набір полюсів $\{s_1, s_2, \dots, s_n\}$, які одночасно є власними числами матриці стану замкнутої системи і можуть бути визначені з матричного характеристичного рівняння

$$\det(\mathbf{I}s - \mathbf{A} + \mathbf{BK}) = 0.$$

Оскільки власні числа матриці однозначно визначають коефіцієнти характеристичного полінома, завдання може бути сформульоване таким чином: вибрати такі значення коефіцієнтів матриці K , при яких виконується умова

$$(s - s_1)(s - s_2) \dots (s - s_n) = \det(\mathbf{I}s - \mathbf{A} + \mathbf{BK}). \quad (3.5)$$

Прирівнюючи в (3.5) коефіцієнти при однакових степенях s , отримуємо систему алгебраїчних рівнянь відносно невідомих параметрів закону управління. Отримана система завжди має рішення, якщо об'єкт управління керований і якщо в законі управління реалізовані зворотні зв'язки по усіх змінних стану.

Описана процедура може бути представлена у вигляді наступної послідовності кроків.

Крок 1. За заданими показниками перерегулювання і часу перехідного процесу яким-небудь чином вибираємо відповідний бажаний розподіл полюсів і знаходимо бажаний характеристичний поліном замкнутої системи

$$M_{жс} = (s - s_1)(s - s_2) \dots (s - s_n) = s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n.$$

Крок 2. Знаходимо той же характеристичний поліном за допомогою формули:

$$\mathbf{M} = \det(\mathbf{I}s - \mathbf{A} + \mathbf{BK})..$$

Коефіцієнти цього полінома залежатимуть від невідомих поки параметрів – коефіцієнтів матриці K .

Крок 3. Прирівнюючи коефіцієнти поліномів, отриманих на першому і другому кроках, отримуємо систему рівнянь для визначення невідомих параметрів k_i . Вирішивши її, знаходимо шукані параметри модального регулятора.

3.6 Використання пакету Control System Toolbox™

Пакет Control System Toolbox™ середовища MATLAB містить ряд вбудованих функцій для автоматизації процесів синтезу систем управління.

Функція **tf** створює LTI-модель (від англ. Linear Time-Invariant System) у вигляді передатної функції:

$$\text{sys} = \text{tf}(\text{num}, \text{den})$$

$$\text{sys} = \text{tf}(\text{num}, \text{den}, Ts)$$

Тут:

`num` та `den` – вектори-рядки (масиви для багатовимірних моделей) коефіцієнтів поліномів чисельника і знаменника, розташованих в порядку убування степеня оператора Лапласа p или z ;

`Ts` – інтервал дискретизації (при створенні дискретних моделей).

Функція **`pole`** повертає вектор, елементами якого є полюси LTI-моделі:

```
p=pole(sys)
```

Приклад:

```
» W=tf([1 1],[1 2 10]); p=pole(W)
```

```
p= -1.0000 +3.0000i
```

```
-1.0000 -3.0000i
```

Функція **`zero`** повертає вектор, елементами якого є нулі LTI -моделі:

```
z=zero(sys)
```

```
[z, gain]= zero(sys)
```

Функція, викликана у формі `[z, gain]= zero(sys)`, окрім нулів повертає ще і узагальнений коефіцієнт посилення системи.

Функція **`pzmap`** повертає карту нулів і полюсів системи у вигляді графіку:

```
pzmap(sys)
```

```
[p, z] = pzmap(sys)
```

У останній формі запису графік не виводиться, а повертаються вектори, елементи яких є нулями і полюсами моделі системи.

Функція **`dcgain`** повертає статичний коефіцієнт передачі (матрицю таких коефіцієнтів в багатовимірному випадку) LTI-моделі:

```
k = dcgain(sys)
```

Функція **`zpk`** створює модель у вигляді передавальної функції по її заданих нулях і полюсах:

```
sys = zpk(z, p, k)
```

```
sys = zpk(z, p, k, Ts)
```

Тут `z` і `p` – відповідно, вектори-рядки (масиви для багатовимірних об'єктів) нулів і полюсів передавальної функції (матриці передавальних функцій), а `k` – узагальнений коефіцієнт посилення.

Функція `zpk(sys)` перетворить довільну LTI-модель в модель виду `zpk`.

Функція **`ss`** формує модель в просторі станів:

```
sys = ss(a, b, c, d)
```

```
sys = ss(a, b, c, d, Ts)
```

Тут `a`, `b`, `c`, і `d` - матриці моделі для змінних стану.

Функція `ss(sys)` перетворить довільну модель, задану в `tf`- або `zpk`-формі, в `ss`-модель.

Функція **`tfddata`** повертає чисельник і знаменник передатної функції:

```
[num, den] = tfdata(sys, 'v')
```

Тут аргумент `'v'` використовується для повернення результату у вигляді векторів рядків, а не у вигляді опису розмірності масиву комірок.

Функція **`zpkdata`** повертає нулі, полюси і узагальнені коефіцієнти передачі для кожного каналу LTI-моделі з ім'ям `sys`:

```
[z, p, k] = zpndata(sys, 'v')
```

Функція **ssdata** повертає матриці ss-моделі:

```
[a, b, c, d] = ssdata(sys)
```

Функція **ctrb** формує матрицю керованості для моделі в просторі станів.

Команди введення функції:

```
Co = ctrb(A, B)
```

```
Co = ctrb(sys)
```

Тут *sys* – ім'я ss-моделі, *A* і *B* - матриці цієї моделі.

Система є керованою, якщо матриця керованості має повний ранг.

Повернена величина *Co* – матриця керованості. Обчислення рангу матриці здійснюється таким чином:

```
rank(Co)
```

Функція **acker** повертає значення вектора посилення зворотного зв'язку, що забезпечує бажане розташування полюсів для одновимірних систем:

```
k = acker(A, B, p)
```

Тут аргументи *A* і *B* - матриці ss-моделі, *p* – вектор бажаних полюсів.

Повернена величина *k* – вектор коефіцієнтів посилення зворотного зв'язку.

Функція **place** за призначенням аналогічна попередній функції, але може використовуватися як для одновимірних, так і для багатовимірних систем:

```
k = place(A, B, p)
```

```
[k, prec, message] = place(A, B, p)
```

Аргументи цієї функції ті ж, що у функції *acker*. Повернені величини:

- *k* – матриця коефіцієнтів посилення;
- *prec* – число точних десяткових цифр у полюсів знайденої замкнутої системи;
- *message* – попередження, що видається, якщо деякі полюси замкнутої системи відрізняються від бажаних полюсів більш ніж на 10%.

3.7 Синтез спостерігача

Часто не всі компоненти вектора станів об'єкту управління доступні для виміру. В цьому випадку, щоб в системі управління можливо було використовувати зворотний зв'язок по стану, необхідно якимось чином оцінити вектор змінних стану системи. Пристрої, що формують вектор оцінки станів, а також що дозволяють відокремити корисний сигнал від завад, називають спостерігачами (ідентифікаторами, фільтрами).

Спостерігач може мати порядок, рівний порядку системи – спостерігач повного порядку, який оцінює вектор стану, враховуючи усі змінні стану, або менший, в порівнянні з системою – спостерігач зниженого порядку, який має розмірність на одну або кілька вимірюваних компонент менше, ніж система. При цьому формується новий вектор, в якому враховуються тільки ті компоненти вектора стану, які не доступні для виміру. Інакше кажучи, розмірність вектора стану зменшується на число компонент вимірюваного вектора.

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

$$\begin{cases} \dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{U}; \\ \mathbf{Y} = \mathbf{C}\mathbf{X}. \end{cases} \quad (3.6)$$

Припустимо, що матриці \mathbf{A} , \mathbf{B} , \mathbf{C} відомі, тоді вектор \mathbf{X} можна замінити вектором $\hat{\mathbf{X}}$ моделі, яка має той же вхід, що і об'єкт (3.6).

$$\dot{\hat{\mathbf{X}}} = \mathbf{A}\hat{\mathbf{X}} + \mathbf{B}\mathbf{U}. \quad (3.7)$$

При відновленні за допомогою моделі (3.7) не використовується вимірюваний вихід. Якість відновлення покращується, якщо ввести в модель різницю вимірюваного виходу і його оцінки $\mathbf{Y} - \mathbf{C}\hat{\mathbf{X}}$ у вигляді зворотного зв'язку:

$$\dot{\hat{\mathbf{X}}} = \mathbf{A}\hat{\mathbf{X}} + \mathbf{B}\mathbf{U} + \mathbf{L}(\mathbf{Y} - \mathbf{C}\hat{\mathbf{X}}). \quad (3.8)$$

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

Введемо помилку відновлення $\tilde{\mathbf{X}} = \mathbf{X} - \hat{\mathbf{X}}$. Віднімаючи (3.8) з диференціального рівняння (3.6), отримаємо

$$\dot{\tilde{\mathbf{X}}} = \mathbf{A}\tilde{\mathbf{X}} - \mathbf{L}(\mathbf{Y} - \mathbf{C}\hat{\mathbf{X}}) = \mathbf{A}\tilde{\mathbf{X}} - \mathbf{L}\mathbf{C}\mathbf{X} + \mathbf{L}\mathbf{C}\hat{\mathbf{X}} = \mathbf{A}\tilde{\mathbf{X}} - \mathbf{L}\mathbf{C}(\mathbf{X} - \hat{\mathbf{X}}) = (\mathbf{A} - \mathbf{L}\mathbf{C})\tilde{\mathbf{X}}. \quad (3.9)$$

Очевидно, для того, щоб помилка відновлення прагнула до нуля, необхідно вибрати матрицю \mathbf{L} так, щоб система (3.9) була асимптотично стійка.

Таким чином, вводячи зворотний зв'язок в модель відновлення, можна нівелювати помилку, навіть якщо система (3.6) нестійка.

Модель (3.8) відновлює усі складові вектору стану, тому вона називається спостерігачем стану повного порядку Калмана.

Модель спостерігача Калмана звичайно представляють у вигляді:

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

Схематично це зображається так, як показано на рис. 3.4.

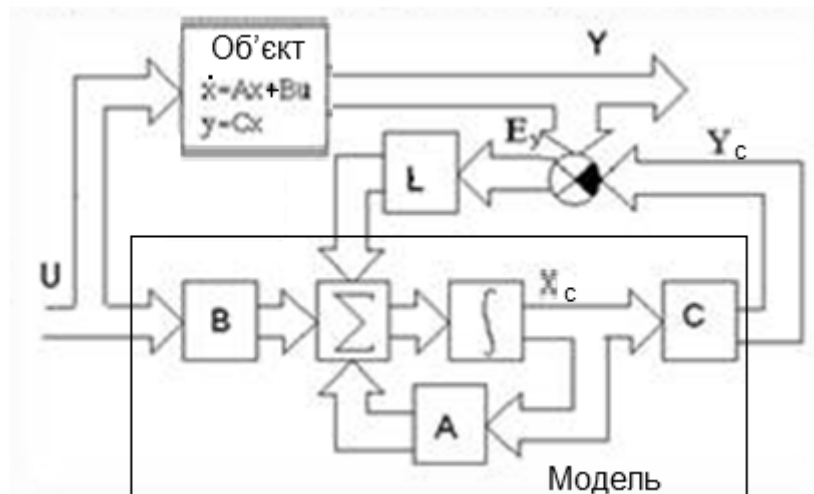


Рис. 3.4 – Структура спостерігача Калмана

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

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

Позначимо через \mathbf{w} збурення за входом, через \mathbf{v} – збурення за спостеріганням внаслідок похибок вимірів.

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

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

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

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

У загальному випадку у матричній формі:

$$\mathbf{P} = \lim M \left\{ (\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T \right\}.$$

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

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

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

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

$$[\text{kest}, \mathbf{L}, \mathbf{P}] = \text{kalman}(\text{sys}, \mathbf{Q}_n, \mathbf{R}_n, \mathbf{N}_n).$$

Ця функція повертає модель спостерігача Калмана kest в просторі станів для

- а) моделі sys спостерігача;
- б) матриці випадкових збурень \mathbf{Q}_n ;
- в) матриці завад вимірів \mathbf{R}_n ;
- г) матриці кореляційного взаємозв'язку \mathbf{N}_n .

Якщо немає статистичного зв'язку між \mathbf{Q} та \mathbf{R} , останній аргумент можна опустити.

3.8 Аналіз результатів розрахунків

Для аналізу отриманих результатів необхідно побудувати перехідні характеристики замкнутої системи, використовуючи будь-який з доступних програмних засобів. Доцільно орієнтуватися на можливості середовища Matlab.

Пакет *Simulink* є ядром інтерактивного програмного комплексу, призначеного для математичного моделювання лінійних і нелінійних динамічних систем і пристроїв, представлених своєю функціональною блок-схемою. При цьому можливі різні варіанти моделювання : в часовій області, в частотній області, з подієвим управлінням і т. д.

Simulink автоматично формує і вирішує системи алгебраїчних і диференціальних рівнянь, що описують задану функціональну схему, забезпечуючи зручний і наочний візуальний контроль за поведінкою створеного користувачем віртуального пристрою.

Аналіз синтезованої системи зручно також проводити за допомогою інструменту *LTI-Viewer*, призначеного для аналізу лінійних стаціонарних систем. За допомогою цього інструменту можна легко побудувати частотні характеристики досліджуваної системи, отримати її відгуки на одиничні ступінчасту і імпульсну дії, знайти нулі і полюси системи та ін.

Робота з вищезгаданими засобами докладно описана в методичних вказівках для практичних і лабораторних занять з дисципліни «Сучасні проблеми автоматизованого управління».

4 ПРИКЛАД СИНТЕЗУ МОДАЛЬНОГО РЕГУЛЯТОРА

4.1 Визначення передатної функції об'єкта управління

Розглянемо синтез системи управління антеною, що відстежує рух цілі в горизонтальній площині (рис. 4.1).



Рис. 4.1– Структура системи управління антеною

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

Об'єкт управління складається з антени і двигуна. Завданням є кут $\theta_r(t)$. Збуренням – момент вітрового навантаження, прикладений до антени.

Спостережувані змінні визначаються складом датчиків. Наприклад, такою змінною може бути вихідний сигнал потенціометра на валу антени, визначуваний вираженням $\eta(t) = \theta(t) + v(t)$, де $v(t)$ – шум вимірів. Кут $\theta(t)$ необхідно управляти, отже, він є керованою змінною.

Спрощену модель двигуна приймемо у вигляді

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

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

Рух антени може бути описаний диференціальним рівнянням

$$J \frac{d^2\theta(t)}{dt^2} + \beta \frac{d\theta(t)}{dt} = k_2 \mu(t) + \mu_f(t). \quad (5.9)$$

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

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

Для рівняння двигуна отримаємо:

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

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

Для антени:

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

$$W_a = \frac{\theta(s)}{\mu(s)} = \frac{k_2}{J s^2 + \beta s}.$$

Передатна функція об'єкта управління являє собою послідовне з'єднання двох ланок (рис. 4.2).

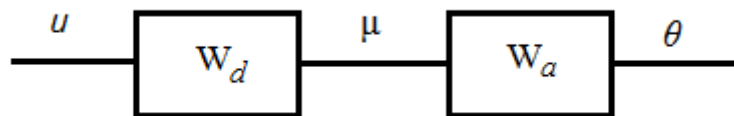


Рис. 4.2 – Структурна схема об'єкта управління

Нехай параметри передатних функцій мають значення, наведені у табл. 4.1.

Таблиця 4.1

$T_d, \text{с}$	k_1	$J, \text{кгм}^2$	$\beta, \text{Нм/с}$	k_2
0,5	0,2	40	3,8	8

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

```
k1=0.2; Td=0.5; Wd=tf(k1, [Td, 1])
k2=8; J=40; Beta=3.8; Wa=tf(k2, [J, Beta, 0])
Wo=Wd*Wa
```

Одержуємо результат:

```
Wo =
      1.6
-----
 20 s^3 + 41.9 s^2 + 3.8 s
```

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

```
P=pole(Wo) '
P =
      0      -2.0000      -0.0950
```

Як вже відзначалося, в процедурі модального синтезу системи регулювання можна виділити два основні етапи:

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

Вказувалося також, що для вирішення першого завдання можна вибрати таке розташування, виходячи з міри стійкості і коливальності замкнутої системи. Нехай треба забезпечити якість регулювання, задану параметрами $\sigma < 20\%$ і $t_n < 30$ с.

З метою автоматизації процесу синтезу модальної системи управління створимо графічний інтерфейс користувача GUI.

4.2 Побудова графічного інтерфейсу користувача

Структура графічних класів і об'єктів в Matlab представлена на рис. 4.3.

Опис графічних об'єктів наданий в табл. 4.2.

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

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

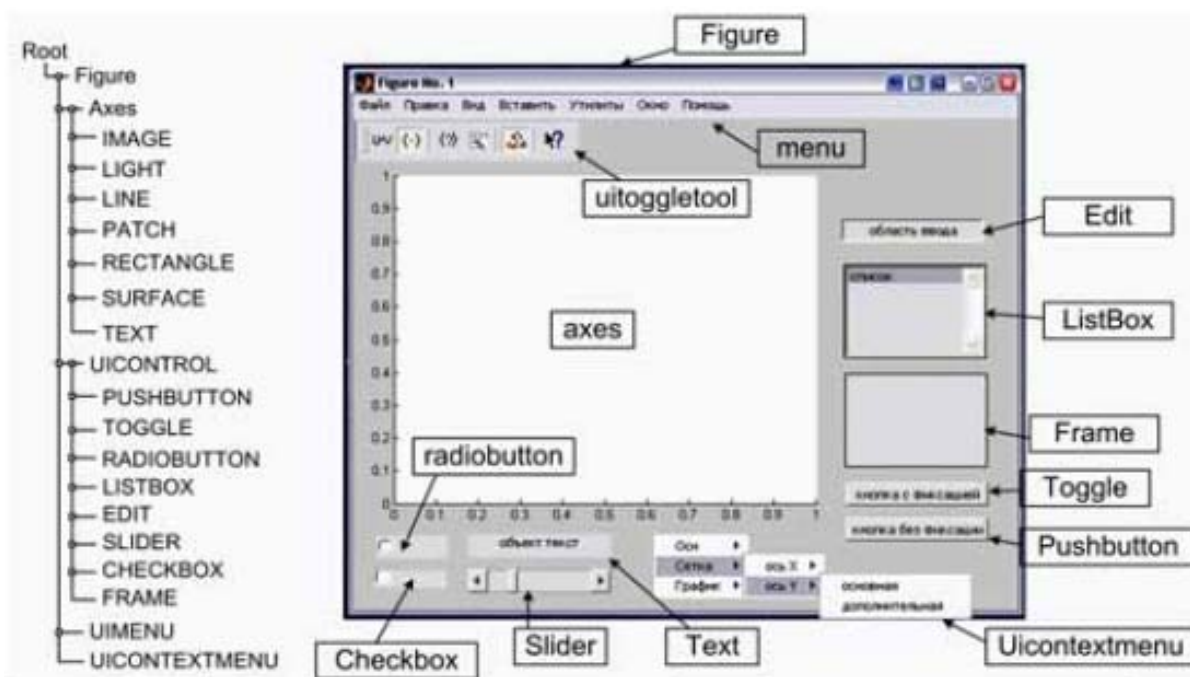


Рис. 4.2 – Структура графічних класів і об'єктів в Matlab

Таблиця 4.2 – Опис графічних об'єктів

Об'єкт	Опис
Root	Верхівка ієрархії, що представляє екран монітора
Figure	Вікно програми, що представляє графічний інтерфейс користувача
Axes	Область системи координат для побудови графіків і поверхонь
Uicontrol	Клас, що представляє набір динамічних графічних об'єктів управління (кнопки, прапори, слайдери, редактори тексту)
Uimenu	Меню вікна програми
Uicontext-menu	Контекстні меню в зоні вікна програми, що викликаються правим кліком на графічному об'єкті
Image	Двовимірні растрові картини
Light	Джерело світла, що змінює колірну карту об'єкта, на який він спрямований
Line	Лінія
Path	Полігон, що конфігурується
Rectangle	Двовимірна змінювана форма
Surface	Тривимірні матричні поверхні
Text	Статичний текст, що конфігурується, з можливістю редагування

У кожен callback-функцію передаються три параметри:

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

б) eventdata – зарезервована змінна;

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

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

```
global Wo P Wzpk
```

Для створення макету інтерфейсу користувача треба ввести команду **guide**. На екрані з'явиться вікно, в якому можна створити новий або відкрити вже існуючий інтерфейс. Якщо створюватимемо інтерфейс з нуля, то в списку шаблонів вибираємо "Blank GUI".

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

Отже, створимо макет нашої майбутньої програми. Помістимо на форму елементи і задамо їх властивості так, як показано на рис. 4.4.

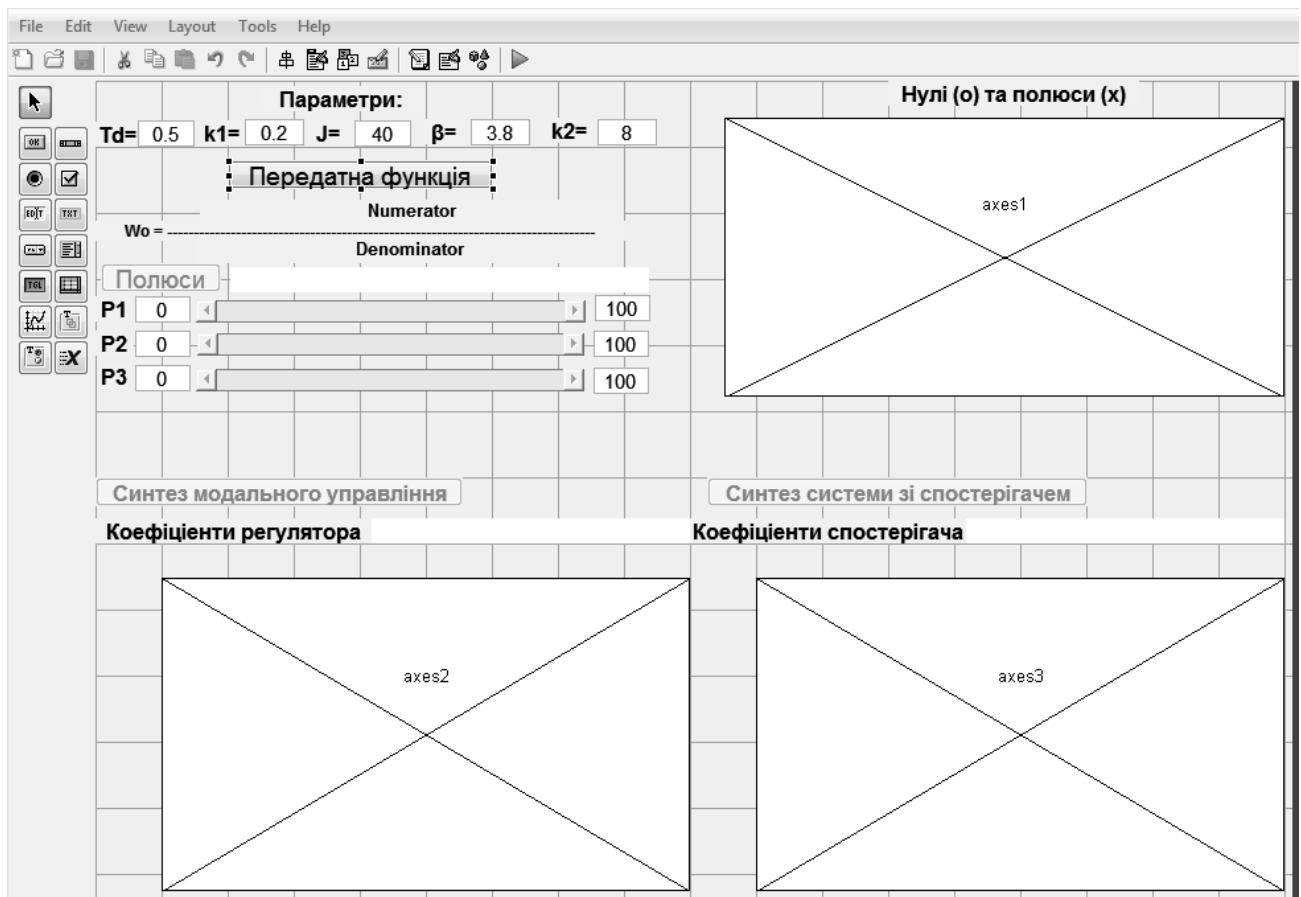





Рис. 4.4 – Макет графічного інтерфейсу користувача

На панелі контролів виберемо  (Static Text) і розміщуємо його вгорі макету. Двічі клацаємо по створеному контролю і у вікні властивостей у полі String пишемо текст «Параметри:». У відповідних полях встановлюємо бажані параметри: шрифт (FontName), кегль (FontSize), жирність (FontWeight). Тег контролю можна залишити таким, що встановлюється автоматично. Для пришвидшення роботи інші тексти GUI можна копіювати зі вже створеного описаним способом, замінюючи поля String на потрібні написи.

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


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

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

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


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

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

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

Примітка: кількість слайдерів повинна дорівнювати порядку системи.

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

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

Далі необхідно створити функції, що викликаються при задіюванні контролів.

Правою клавiшею миши клацаємо по кнопці «Передатна функція». У контекстному меню вибираємо View Callbacks та Callback. У вікні редактора програм Edit видiляється рядок:

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

Пiд цим рядком дописуємо текст функції:

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

Останнім рядком цієї функції задiюється кнопка «Полюси».

Для виведення формули передатної функції треба додати функцію TFout:

```
% Виведення формули ПФ
function TFout(hObject, eventdata, handles)
global Wo P Wzpk
[num, den]=tfdata(Wo, 'v')
m=length(num) % розмір чисельника ПФ
n=length(den) % розмір знаменника ПФ
strnum = ''; % символний рядок для чисельника
for i=1:m - 1
    if not(num(i)==0)
        strnum=[strnum, num2str(num(i)), '*s^', num2str(m-i), '+'];
    end
end
if not(num(m)==0)
    strnum = [strnum, num2str(num(m))]
end
strden = ''; % символний рядок для знаменника
for i=1:n - 1
    strden=[strden, num2str(den(i)), '·s^', num2str(n-i), ' + ']
end
```

```

if den(n)~=0
    strden = [strden,num2str(den(n))]
else
    strden = strden(1:length(strden) - 3)
end
set(handles.txtNum, 'String', strnum) ;
set(handles.txtDen, 'String', strden) ;

```

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

```

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

```

додаємо текст:

```

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

```

Тут викликається функція побудови графіка нулів і полюсів:

```

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

```

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

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

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

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

```
% - - - Executes on button press in btnSint.
function btnSint_Callback(hObject, eventdata, handles)
% Функція синтезу системи модального управління
global Wo P
% Модель у просторі станів
Wss=ss(Wo);
% Вилучення матриць коефіцієнтів
A=Wss.a; B=Wss.b; C=Wss.c; D=Wss.d;
% Будуємо матрицю керованості
Q = ctrb(A, B);
% Визначаємо ранг матриці Q:
if rank(Q)~=length(B)
    error('система не є керованою, модальне управління
неможливе', 'Проблема!');
    set(handles.btnSint, 'Enable', 'off');
    return;
end
% значення коефіцієнтів зворотних зв'язків
K=acker(A,B,P);
set(handles.txtK, 'String', num2str(K));
```

```

% Матриця А коефіцієнтів замкненої системи:
As=A - B*K;
% Модель замкненої системи:
sys=ss(As,B,C,D);
% ПФ замкненої системи:
Wcls=tf(sys);
% Розрахунок перехідного процесу
[out,t]=step(Wcls);
% обчислення коефіцієнта масштабування
k0=1/out(length(out));
% Будуємо графік
axes(handles.axes2); % вибір вікна
cla; % очистка графіку
plot(t, k0*out)
xlabel('Час')
ylabel('Вихід')
grid on % нанесення сітки
set(handles.btnEst, 'Enable', 'on'); % задіюється кнопка
«Синтез системи зі спостерігачем»

```

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

```

% - - - Executes on button press in btnEst.
function btnEst_Callback(hObject, eventdata, handles)
% Функція синтезу системи зі спостерігачем
global Wo P
% Модель у просторі станів
Wss=ss(Wo);
% Вилучення матриць коефіцієнтів
A=Wss.a; B=Wss.b; C=Wss.c; D=Wss.d;
% Будуємо матрицю спостережуванності
N = obsv(A, C);
% Перевірка спостережуванності
if det(N)==0
    errordlg('Система не є спостережуваною','Проблема!');
    set(handles.btnEst, 'Enable', 'off');
    return;
end
% Бажане розміщення полюсів спостерігача:
Pe=1.2*P; % лівіше полюсів системи
% Матриця коефіцієнтів спостерігача L:
Lt=acker(A', C', Pe); L=Lt'
set(handles.txtL, 'String', num2str(Lt)) ;

```




```

% Матриця А коефіцієнтів замкненої системи:
K=acker(A,B,P);
As=A - B*K;
% Модель замкненої системи:
sys=ss(As,B,C,D);
% Формуємо спостерігаючий пристрій
rsys=reg(sys,K,L)
% Передатна функція спостерігача
We=tf(rsys);
% перехідний процес спостерігача
axes(handles.axes3); % вибір вікна
cla; % очистка графіку
step(We)
xlabel('Час')
ylabel('Вихід')
grid on % нанесення сітки
% Експорт результатів у робочий простір Matlab
checkLabels = {'Зберегти матрицю А під іменем:' ...
               'Зберегти матрицю В під іменем:' ...
               'Зберегти матрицю С під іменем:' ...
               'Зберегти матрицю D під іменем:' ...
               'Зберегти матрицю К під іменем:' ...
               'Зберегти матрицю L під іменем:'};
varNames = {'A','B','C','D','K','L'};
items = {A,B,C,D,K,L};
export2wsdlg(checkLabels,varNames,items,...
             'Зберегти?');

```

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

Побудований GUI треба зберегти під будь-яким іменем, наприклад, Anten.fig. Опробування проводиться натисненням кнопки , комбінацією клавіш Ctrl+T або через меню Tools→Run.

Відлагоджений інтерфейс можна запускати на виконання, набравши у командному вікні його ім'я без розширення:

```
>>Anten
```

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

система керована, формується матриця коефіцієнтів регулятора і будується графік перехідного процесу.

Тиснемо кнопку «Синтез системи зі спостерігачем». Якщо система не є спостережуваною, видається повідомлення про це, і робота припиняється. В іншому випадку формується матриця коефіцієнтів спостерігача і будується графік перехідного процесу спостерігача. Під кінець виводиться вікно експорту результатів (рис. 4.5). Ці результати необхідні для використання в Simulink-моделі синтезованої системи (див. п. 4.3).

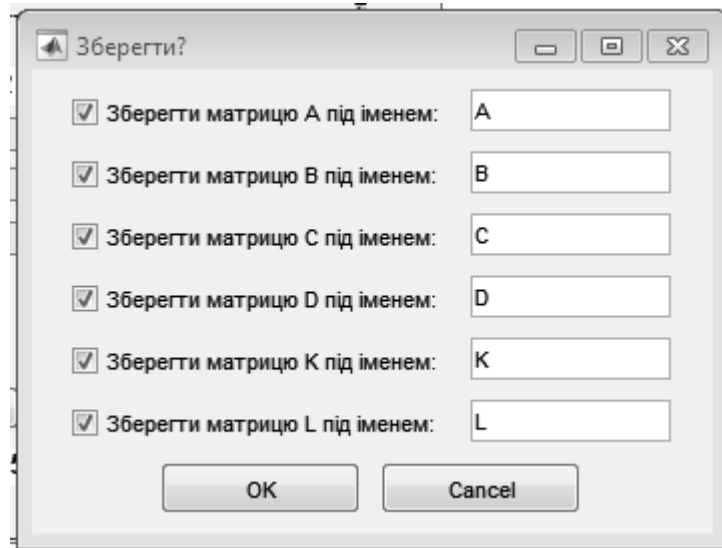


Рис. 4.5 – Вигляд вікна експорту результатів

Кінцевий вигляд вікна GUI показаний на рис. 4.6. Як можна побачити, час встановлення виходу спостерігача на порядок менший часу встановлення системи модального управління.

4.3 Аналіз результатів

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

Модель спостерігача будується у відповідності з рівнянням:

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

Для моделі об'єкта управління використовується блок State-Space з бібліотеки Continuous. Вигляд Simulink – моделі показаний на рис. 4.7.

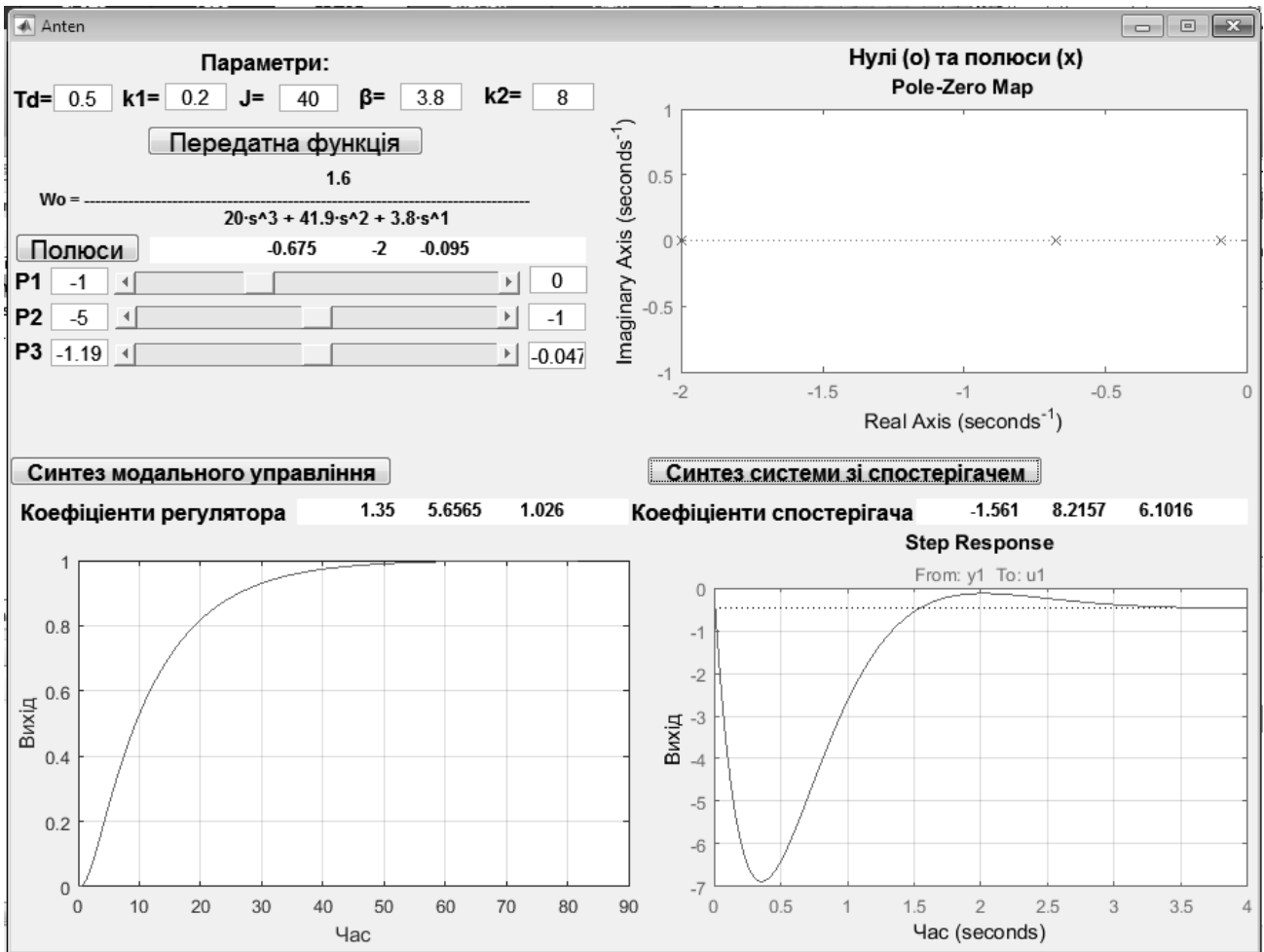


Рис. 4.6 – Результати розрахунків за допомогою GUI

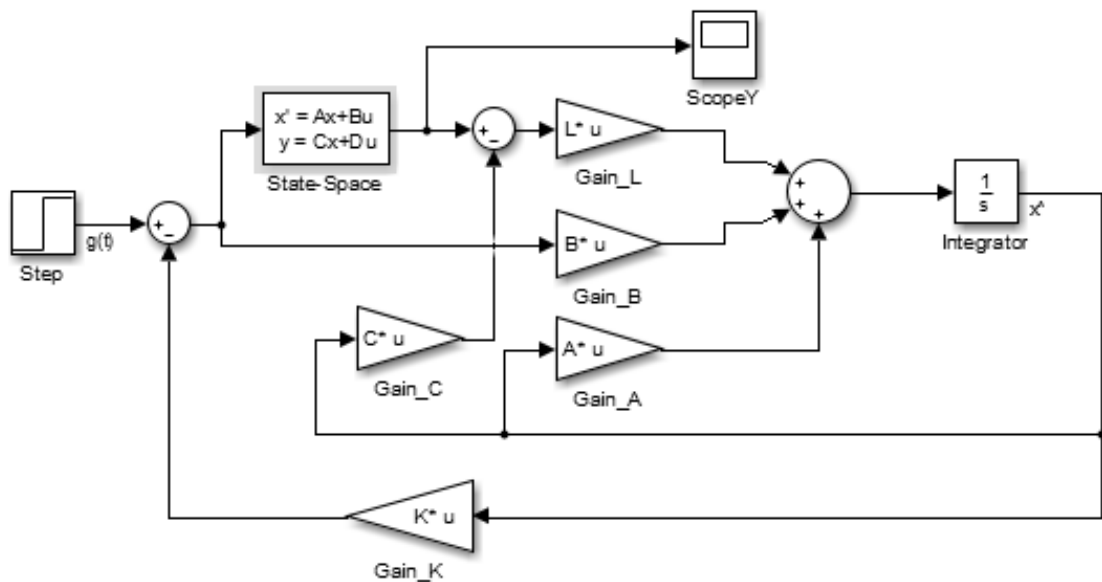


Рис. 4.7 – Simulink – модель синтезованої системи зі спостерігачем

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

Параметри блоку State-Space установлюємо відповідно рис. 4.8.

Рис. 4.8 – Налаштування блоку State-Space

Примітка: кількість елементів вектора початкових умов (Initial conditions) повинна дорівнювати порядку системи.

Для блоку Step установлюємо кінцеве значення 0.2.

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

Перед початком моделювання необхідно виконати синтез системи і експорт матриць у робочий простір MATLAB згідно п. 4.2. Результат моделювання показаний на рис. 4.9.

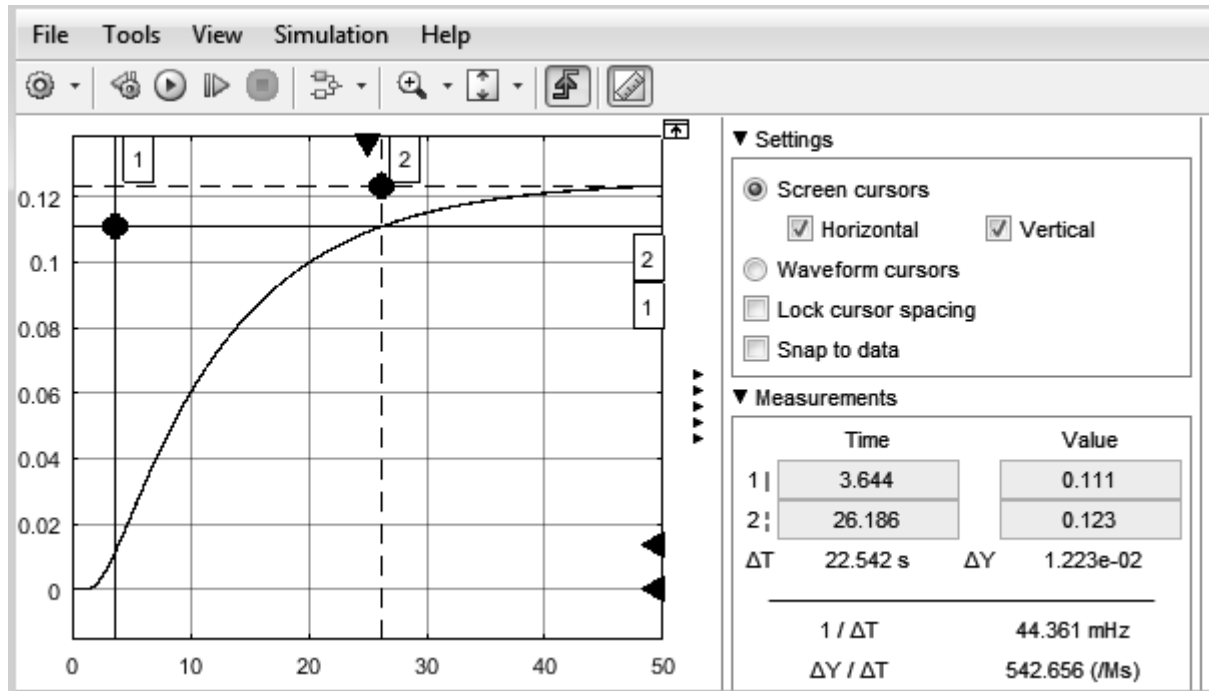



Рис. 4.9 – Результат моделювання з відкритим вікном Cursor Measurements

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

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

Щоб оцінити запас стійкості, треба побудувати діаграму Бодє. У бібліотеці Simulink Control Design / Linear Analysis Plots вибираємо блок Bode Plot і переносимо його у модель. Клацнувши правою клавiшею миші по лініях зв'язку для вхідного і вихідного сигналів, вибираємо у контекстному меню

Linear Analysis Points опції Input Perturbation і Output Measurement відповідно. Двічі клацаємо по доданому блоку і у вікні властивостей блоку тиснемо кнопки Get the linearization I/Os (показана стрілкою на рис. 4.11) і Show Plot. Виводиться вікно Bode Plot (рис. 4.10). На графіку тиснемо кружечок і читаємо значення запасу стійкості Gain Margin 33.7 dB.

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

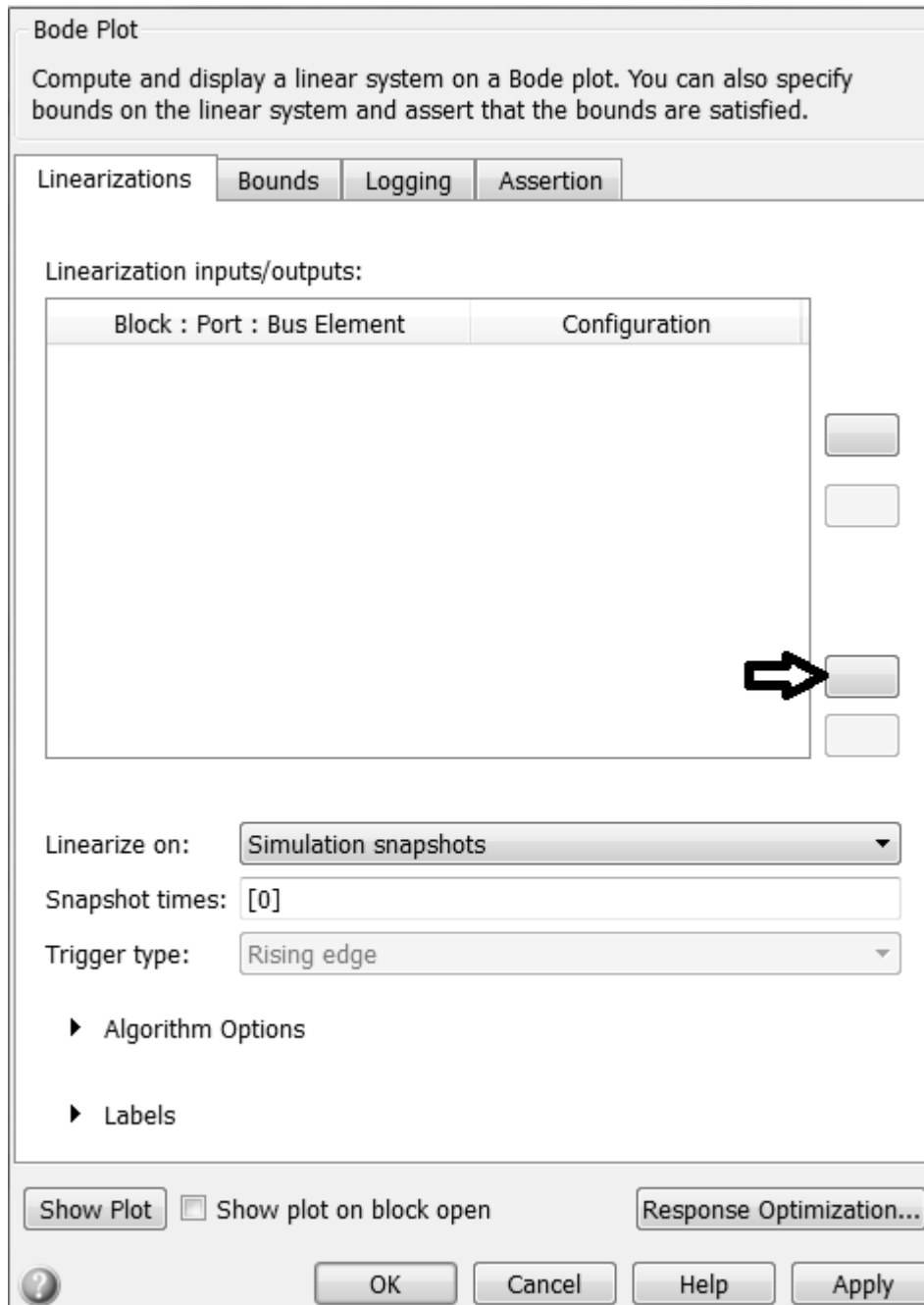


Рис. 4.10 – Вікно властивостей блоку Bode Plot

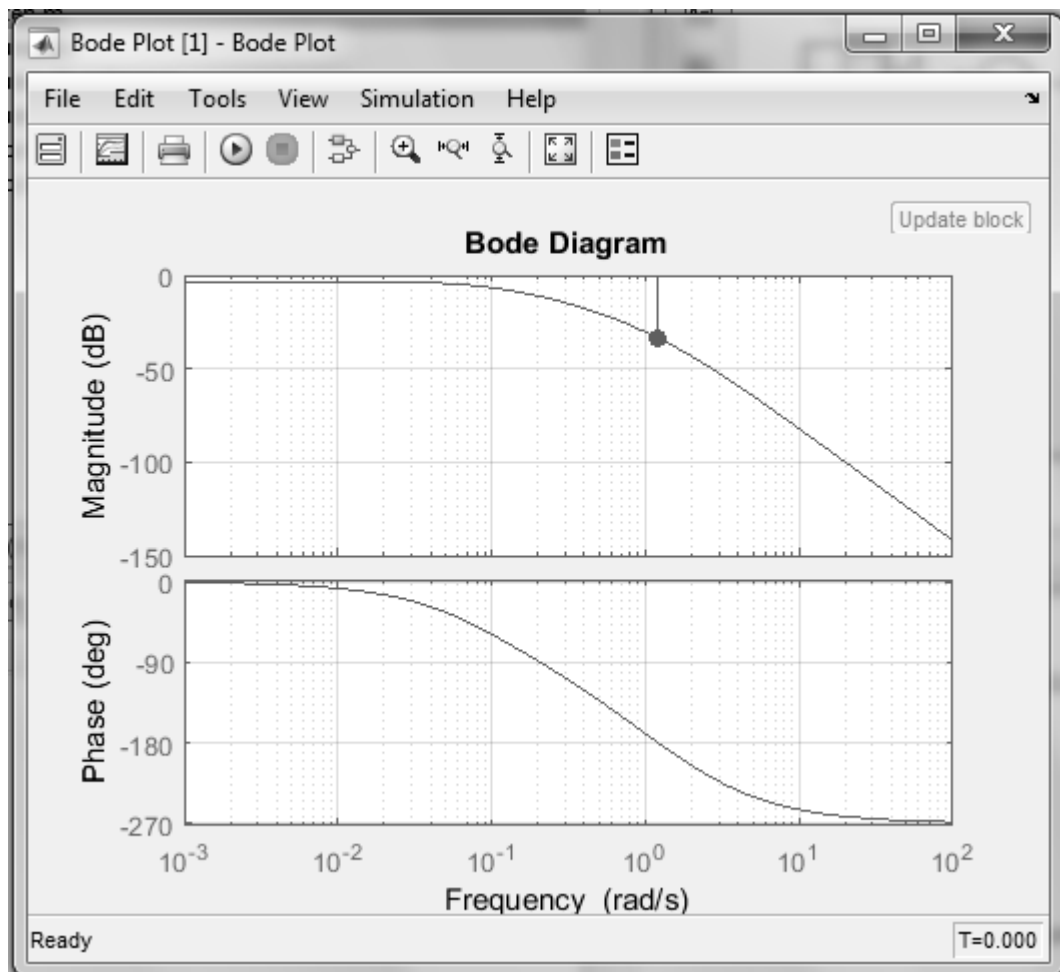


Рис. 4.11 – Диаграмма Боде

6 ВИКОРИСТАНА ЛІТЕРАТУРА

1. Александров, А.Г. Синтез регуляторов многомерных систем. – М.: Машиностроение, 1986. – 272 с.
2. Анализ и синтез систем управления / Д.Х. Имаев и др. – СПб.: Инф. изд. центр СГУ, 1998. – 169с.
3. Андриевский, Б.Р. Избранные главы теории автоматического управления/ Б.Р. Андриевский, А.Л. Фрадков. – СПб.: Наука, 1999.
4. Боднер, В.А. Теория автоматического управления полетом. – М.: Наука, 1964. – 698 с.
5. Дьяконов, В.П. MATLAB 6/6.1/6.5 + SIMULINK 4/5 в математике и моделировании.– М.: СОЛОН – Пресс, 2003 – 565 с.
6. Заде, Л. Теория линейных систем. Метод пространства состояний/ Л. Заде, Ч. Дезоер. – М.: Наука, 1970.
7. Ильинский, Н.Ф. Общий курс электропривода/ Н.Ф. Ильинский, В.Ф.

Козаченко. – М.: Энергоатомиздат, 1992.

8. Кузовков, Н.Т. Модальное управление и наблюдающие устройства. – М. : Машиностроение, 1976. – 184 с.

9. Мезенцев, А.А. Техническое и программное обеспечение лабораторного комплекса «Организация пультов управления современных АСУ ТП» : учебное пособие / А.А. Мезенцев, В.М. Павлов, К.И. Байструков – Томск : Изд-во Томского политехнического университета, 2011. – 120 с.

10. Озеров, Л.А. Автоматизированное проектирование систем : учебное пособие. – Челябинск: Издательский центр ЮУрГУ, 2009. – Ч.1.– 66 с.

11. Параев, Ю.И. Алгебраические методы в теории линейных систем управления. – Томск : Томск. гос. ун – т, 1980. – 139 с.

12. Теория автоматического управления: рабочая программа, метод. указания и контр. задания для студентов спец. 220301 «Автоматизация технологических процессов и производств (в нефтегазовой отрасли)» ИДО сост. А. В. Воронин. – Томск: Изд – во ТПУ, 2013. – 28 с.

7 РЕКОМЕНДОВАНА ЛИТЕРАТУРА

1. Дорф, Р. Современные системы управления/ Р. Дорф, Р. Бишоп. Пер. Б. И. Копылова. — М. : Лаборатория Базовых Знаний, 2002. — 832 с

2. Теория автоматического управления: В 2 ч. / Под ред. А. А. Воронова. 2 – е изд. – М. : Высш. шк., 1986. – 364 с.

3. Кузовков, Н. Т. Модальное управление и наблюдающие устройства / Н. Т. Кузовков. – М. : «Машиностроение», 1976, 184 с

4. Коровин, С. К. Наблюдатели состояния для линейных систем с неопределенностью / С. К. Коровин, В. В. Фомичев. – М. : ФИЗМАТЛИТ, 2007 — 224 с.

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

6. Поляков, К.Ю. Исследование систем автоматического управления в Matlab [Электронный ресурс] / К.Ю. Поляков. – Режим доступа: <http://kpolyakov.spb.ru/uni/labs.htm>

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

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

9. Дьяконов, В. SIMULINK 4 : Спец. справочник.– СПб. : Питер, 2002. – 528 с.

10. Советов, Б. Я. Моделирование систем / Б. Л. Советов, С.А. Яковлев. – М. : Высш. шк., 2001. – 343 с. – Режим доступа : <http://simulation.su/uploads/files/default/2001-uchebnik-sovetov-yakovlev-.pdf>