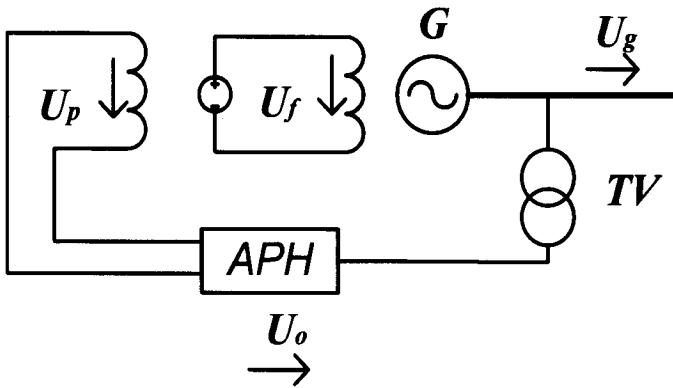


В. М. Кутін, В. О. Лесько, Ю. Ю. Півнюк

## ТЕОРІЯ АВТОМАТИЧНОГО КЕРУВАННЯ



Міністерство освіти і науки України  
Вінницький національний технічний університет

**В. М. Кутін, В. О. Лесько, Ю. Ю. Півнюк**

# **ТЕОРІЯ АВТОМАТИЧНОГО КЕРУВАННЯ**

**Лабораторний практикум**

Вінниця  
ВНТУ  
2019

**УДК 681.5.017**

**K95**

Рекомендовано до друку Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 11 від 25.04.2019 р.)

Рецензенти:

**М. Й. Бурбело**, доктор технічних наук, професор

**В. В. Кухарчук**, доктор технічних наук, професор

**О. М. Сінчук**, доктор технічних наук, професор

**Кутін, В. М.**

**K95** Теорія автоматичного керування : лабораторний практикум / В. М. Кутін, В. О. Лесько, Ю. Ю. Півнюк. – Вінниця : ВНТУ, 2019. – 106 с.

Лабораторний практикум призначений для виконання лабораторних робіт з дисципліни «Теорія автоматичного керування» і може бути корисним для студентів спеціальності 141 – «Електроенергетика, електротехніка та електромеханіка» денної та заочної форм навчання.

**УДК 681.5.017**

## ЗМІСТ

ВСТУП .....	4
ЛАБОРАТОРНА РОБОТА № 1 Дослідження розімкненої лінійної системи .....	5
ЛАБОРАТОРНА РОБОТА № 2 Проектування регулятора для лінійної системи .....	20
ЛАБОРАТОРНА РОБОТА № 3 Моделювання систем керування в пакеті Simulink .....	37
ЛАБОРАТОРНА РОБОТА № 4 Моделювання нелінійних систем керування.....	52
ЛАБОРАТОРНА РОБОТА № 5 Програмування в середовищі Matlab .....	63
ЛАБОРАТОРНА РОБОТА № 6 Оптимізація нелінійних систем .....	73
ВИСНОВКИ.....	85
СПИСОК ЛІТЕРАТУРИ.....	86
ДОДАТОК А.....	87
ДОДАТОК Б .....	88

## ВСТУП

Теорія автоматичного керування є базовою наукою про керування. Основні питання та методи теорії автоматичного керування розглянуті в класичній теорії, що продовжує інтенсивно розвиватися в бік створення теорії інтелектуальних систем керування.

Автоматичні системи регулювання та керування є невід'ємною частиною сучасних електронних, електричних, механічних пристроїв та систем. При незадовільній роботі систем керування придатність до експлуатації сучасних технічних пристроїв втрачається. Кожен кваліфікований спеціаліст має мати знання та навички контролю і корекції систем керування, розуміти процеси, які в них проходять, закони керування, принципи побудови й функціонування систем керування, методи їх дослідження та влаштування.

Лабораторний практикум призначений для виконання лабораторних робіт з дисципліни «Теорія автоматичного керування» студентами всіх форм навчання спеціальності 141 – «Електроенергетика, електротехніка та електромеханіка».

Структура та послідовність виконання лабораторних робіт запозичені з [1] і адаптовані під спеціальність 141 – «Електроенергетика, електротехніка та електромеханіка».

### Вимоги до звіту з лабораторної роботи

1. Звіт з лабораторної роботи виконується у вигляді зв'язного (читаного) тексту у файлі формату *Microsoft Word* (шрифт основного тексту *Times New Roman*, 14 пунктів, через 1,5 інтервалу, вирівнювання «по ширині»). На титульному листі вказується: номер та назва роботи, назва дисципліни, шифр групи, прізвище та ініціали студента, прізвище та ініціали викладача.

2. Звіт має містити: номер варіанта, короткий опис досліджуваної системи, результати обчислень, графіки, висновки.

При складанні звіту рекомендується копіювати необхідну інформацію через буфер обміну з робочого вікна середовища *MATLAB*. Для цих даних використовуйте шрифт *Courier New*, в якому ширина всіх символів однакова.

# ЛАБОРАТОРНА РОБОТА № 1

## ДОСЛІДЖЕННЯ РОЗІМКНЕНОЇ ЛІНІЙНОЇ СИСТЕМИ

**Мета роботи:** освоєння методів аналізу одновимірної лінійної безперервної системи за допомогою середовища MATLAB [1].

### Завдання роботи:

- ввести модель системи у вигляді передавальної функції;
- побудувати еквівалентні моделі в просторі станів і у формі «нулі-полюси»;
- визначити коефіцієнт посилення в усталеному режимі і смугу пропускання системи;
- навчитися будувати імпульсну і перехідну характеристики, карту розташування нулів і полюсів, частотну характеристику;
- навчитися використовувати вікно **LTViewer** для побудови різних характеристик;
- навчитися будувати процеси на виході лінійної системи при довільному вхідному сигналі.

## 1.1 Основні теоретичні відомості

### 1.1.1 Моделі лінійних систем

Для опису лінійних систем можуть застосовуватися декілька способів:

- диференціальні рівняння,
- моделі в просторі станів,
- передавальні функції,
- моделі вигляду «нулі-полюси».

Перші два способи називаються *тимчасовими*, оскільки описують поведінку системи в тимчасовій області і відображають внутрішні зв'язки між сигналами. Передавальні функції і моделі вигляду «нулі-полюси» відносять до частотних способів опису, оскільки вони безпосередньо пов'язані з частотними характеристиками системи і відображають тільки вхід-вихідні властивості (тобто, не повністю описують динаміку).

Частотні методи дозволяють застосовувати для аналізу і синтезу методи алгебри, що часто спрощує розрахунки. З іншого боку, для автоматичних обчислень придатні методи, засновані на моделях в просторі станів, оскільки вони використовують обчислювально стійкі алгоритми лінійної алгебри.

Початкові рівняння динаміки об'єктів, які будуються на основі законів фізики, мають вигляд нелінійних диференціальних рівнянь. Для наближеного аналізу й синтезу зазвичай проводять їх лінеаризацію в околі усталеного режиму та отримують лінійні диференціальні рівняння.

Лінійне рівняння  $\ddot{y} + 2\dot{y} + 3y = 4\dot{u} + 5u$  можна записати в операторній формі

$$(p^2 + 2p + 3)y = (4p + 5)u \quad (1.1)$$

або

$$D(p)y = N(p)u,$$

де  $u(t)$  – вхідний сигнал,  $y(t)$  – сигнал виходу,  $p = \frac{d}{dt}$  – оператор диференціювання,  $D(p) = p^2 + 2p + 3$  і  $N(p) = 4p + 5$  – операторні поліноми [2].

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

$$W(s) = \frac{Y(s)}{U(s)}, \quad Y(s) = \int_0^{\infty} y(t)e^{-st} dt, \quad U(s) = \int_0^{\infty} u(t)e^{-st} dt. \quad (1.2)$$

Передавальна функція ланки, яка описується наведеним вище рівнянням, визначається як  $W(s) = \frac{4s+5}{s^2+2s+3}$ , тобто, збігається з відношенням операторних поліномів  $N(p)/D(p)$  при заміні змінної  $p$  на  $s$ .

Передавальна функція в середовищі MATLAB вводиться у вигляді відношення двох многочленів (поліномів) від комплексної змінної  $s$ . Поліноми зберігаються як масиви коефіцієнтів, записаних за зменшенням степенів. Наприклад, передавальна функція

$$F(s) = \frac{2s+4}{s^3+1.5s^2+1.5s+1}$$

вводиться таким чином (на екрані монітора чорним кольором позначається введення користувача, синім – відповідь середовища MATLAB)

```
>> n = [2 4]
```

```
n =
```

```
2 4
```

```
>> d = [1 1.5 1.5 1]
```

```
d =
```

```
1.0000 1.5000 1.5000 1.0000
```

```
>> f = tf ( n, d )
```

```
Transfer function:
```

```
2 s + 4
```

```
-----  
s^3 + 1.5 s^2 + 1.5 s + 1
```

або відразу, без попередньої побудови чисельника і знаменника,

```
>> f = tf ( [2 4], [1 1.5 1.5 1] );
```

У пам'яті створюється об'єкт класу **tf**, що описує передавальну функцію. Крапка з комою в кінці команди пригнічує виведення на екран.

За передавальною функцією можна легко побудувати модель у формі «нулі-полюси»

```
>> f_zpk = zpk(f)
Zero/pole/gain:
      2 (s+2)
-----
(s+1) (s^2 + 0.5s + 1)
```

*Нулями* називається корені чисельника, *полюсами* – корені знаменника. Ця функція має один нуль в точці  $s=-2$  і три полюси в точках  $s=-1$  і  $s=-0,25 \pm 0.9682i$ . Парі комплексних полюсів відповідає квадратний тричлен.

*Модель в просторі станів* пов'язана з записом диференціальних рівнянь в стандартній формі Коші (у вигляді системи рівнянь першого порядку)

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (1.3)$$

Тут  $x$  – вектор змінних стану розміру  $n \times 1$ ,  $u$  – вектор вхідних сигналів (вектор керування) розміру  $m \times 1$  і  $y$  – вектор вихідних сигналів розміру  $p \times 1$ . Крім того,  $A$ ,  $B$ ,  $C$  і  $D$  – постійні матриці. Згідно з правилами матричних обчислень, матриця  $A$  має бути квадратною розміру  $n \times n$ , матриця  $B$  має розмір  $n \times m$ , матриця  $C$  –  $p \times n$  і матриця  $D$  –  $p \times m$ . Для систем з одним входом і одним виходом матриця  $D$  – скалярна величина (у зарубіжній літературі для одновимірних систем використовується скорочення SISO = *Single Input Single Output*) [3].

Для перетворення передавальної функції в *модель в просторі станів* використовується команда

```
>> f_ss = ss ( f )
a =
      x1      x2      x3
x1  -1.5  -0.1875 -0.03125
x2   8      0      0
x3   0      4      0
b =
      u1
x1  0.5
x2  0
x3  0
```



$$c = \begin{matrix} & x_1 & x_2 & x_3 \\ y_1 & 0 & 0.5 & 0.25 \end{matrix}$$

$$d = \begin{matrix} u_1 \\ y_1 & 0 \end{matrix}$$

Це означає, що матриці моделі мають вигляд

$$A = \begin{bmatrix} -1.5 & -0.1875 & -0.03125 \\ 8 & 0 & 0 \\ 0 & 4 & 0 \end{bmatrix}, B = \begin{bmatrix} 0.5 \\ 0 \\ 0 \end{bmatrix}, C = [0 \quad 0.5 \quad 0.25], D = 0.$$

Модель в просторі станів можна побудувати не для всіх передавальних функцій, а тільки для правильних, у яких степінь чисельника не вищий, ніж степінь знаменника. Наприклад, передавальна функція

$W(s) = \frac{2s^2 + 3s + 1}{s + 5}$  неправильна, вона не може бути перетворена в модель в просторі станів.

Використовують також поняття *строго правильної функції*, у якої степінь чисельника менший, ніж степінь знаменника. Якщо побудувати модель в просторі станів для такої функції, матриця  $D$  дорівнюватиме нулю, тобто, пряма передача з входу на вихід відсутня (при стрибкоподібній зміні входу сигнал на виході буде безперервним)

### 1.1.2 Коефіцієнт посилення в усталеному режимі

Одна з найважливіших характеристик лінійної системи – коефіцієнт підсилення в усталеному режимі або *статичний коефіцієнт підсилення (static gain, Dc-gain)*. Його можна визначити як стале значення сигналу виходу при постійному входньому сигналі, що дорівнює одиниці. Розмірність цієї величини дорівнює відношенню розмірностей сигналів входу і виходу.

Розглянемо диференціальне рівняння  $\ddot{y} + 2\dot{y} + 3y = 4\dot{u} + 5u$ .

Вважаючи, що всі похідні (в усталеному режимі) дорівнюють нулю, отримуємо  $3y = 5u \Rightarrow y = \frac{5}{3}u$ . Статичний коефіцієнт підсилення дорівнює  $k_s = 5/3$ .

Якщо задана передавальна функція, для обчислення  $k_s$  потрібно підставити в неї  $s=0$ , оскільки змінна  $s$  відповідає операторові диференціювання. Розглянутому вище рівнянню можна зіставити передавальну функцію  $W(s) = \frac{4s+5}{s^2+2s+3}$ . Тоді  $k_s = \lim_{s \rightarrow 0} W(s) = \frac{5}{3}$ .

Якщо система містить інтегровальну ланку (передавальна функція має полюс в точці  $s=0$ ), то границя інтегрування дорівнює нескінченності,

тобто, при постійному сигналі вихід нескінченно збільшується або зменшується, не досягаючи усталеного режиму.

Той же результат можна отримати за допомогою еквівалентної моделі в просторі станів. За допомогою середовища MATLAB знаходимо  $A = \begin{bmatrix} -2 & -1.5 \\ 2 & 0 \end{bmatrix}$ ,  $B = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$ ,  $C = [2 \quad 1.25]$ ,  $D = 0$ .

Вважаючи, що  $\dot{x} = 0$ , отримуємо модель, яка визначає усталений режим

$$\begin{aligned} 0 &= Ax + Bu &\Rightarrow x &= -A^{-1}Bu \\ y &= Cx + Du &\Rightarrow y &= (-CA^{-1}B + D)u, \end{aligned}$$

звідки випливає, що  $k_s = -CA^{-1}B + D$ .

Для нашої системи, як і раніше, отримуємо  $k_s = \frac{5}{3}$ . Відмітьте, що для того, щоб статичний коефіцієнт підсилення був кінцевий, потрібна оборотність матриці  $A$ , тобто, відсутність інтегровальних ланок (полюси передавальної функції є власними числами матриці  $A$ . Таким чином, якщо у передавальної функції є полюс в точці  $s = 0$ , матриця  $A$  буде виродженою).

Щоб знайти статичний коефіцієнт підсилення моделі  $f$  в MATLAB, використовується команда `>> k = dsgain ( f )`.

### 1.1.3 Імпульсна характеристика

*Імпульсною характеристикою* (ваговою функцією)  $w(t)$  називається реакція системи на одиничний нескінченний імпульс (дельта-функцію або функцію Дірака) за нульових початкових умов (рис.1.1). Дельта-функція визначається рівністю

$$\delta(t) = \begin{cases} 0, & t \neq 0 \\ \infty, & t = 0 \end{cases}, \quad \int_{-\infty}^{\infty} \delta(t) dt = 1. \quad (1.4)$$

Це *узгаальнена функція* – математичний об'єкт, що є ідеальним сигналом, ніякий реальний пристрій не здатний його відтворити. Дельта-функцію можна розглядати як границю прямокутного імпульсу одиничної площі з центром в точці  $t = 0$  при прагненні ширини імпульсу до нуля.

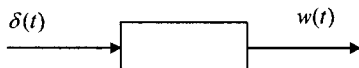


Рисунок 1.1 – Імпульсна характеристика (вагова функція)  $w(t)$

Друга назва – *вагова функція* – пов'язана з тим, що для довільного вхідного сигналу  $u(t)$  виход системи  $y(t)$  обчислюється як згортка

$$y(t) = \int_{-\infty}^t u(\tau) w(t-\tau) d\tau = \int_0^{\infty} u(t-\tau) w(\tau) d\tau. \quad (1.5)$$

Тут функція  $w(t)$  ніби «зважує» вхідний сигнал в підінтегральному виразі. [1]

Імпульсна характеристика відображає лише вхід-вихідні співвідношення за нульових початкових умов, тобто, не може повністю описувати динаміку системи.

Поняття імпульсної характеристики використовується, головним чином, для систем, передавальні функції яких строго правильні. Якщо передавальна функція правильна, але не строго правильна, коефіцієнт прямої передачі з входу на вихід (матриця  $D$  моделі в просторі станів) не дорівнює нулю, тому нескінченний імпульс на вході в момент  $t=0$  передається на вихід. Таку нескінченну (за величиною) імпульсну характеристику неможливо побудувати. Система MATLAB в цьому випадку буде імпульсну характеристику для строго правильної частини, приймаючи  $D=0$ . Це один з тих випадків, коли комп'ютер видає якісно неправильний або недостовірний результат.

Якщо система не містить інтеграторів, імпульсна характеристика прагне до нуля. Це витікає з теореми про граничне значення

$$\lim_{t \rightarrow \infty} w(t) = \lim_{s \rightarrow 0} s W(s), \quad (1.6)$$

де  $W(s)$  – передавальна функція системи, яка є перетворенням Лапласа для  $w(t)$ .

Імпульсна характеристика системи з одним інтегратором прагне до постійної величини, що дорівнює статичному коефіцієнту передачі системи без інтегратора. Для системи з двома інтеграторами імпульсна характеристика асимптотично прагне до прямої, з трьома інтеграторами – до параболи і так далі [2].

#### 1.1.4 Перехідна характеристика

Перехідною характеристикою (перехідною функцією)  $h(t)$  називається реакція системи (за нульових початкових умов) на одиничний ступінчастий сигнал (одиничний стрибок) (рис. 1.2)

$$I(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}. \quad (1.7)$$

Імпульсна і перехідна функції пов'язані виразами

$$w(t) = \frac{dh(t)}{dt}, \quad h(t) = \int_0^t w(\tau) d\tau. \quad (1.8)$$

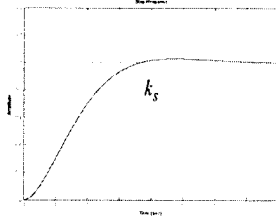


Рисунок 1.2 – Перехідна характеристика (перехідна функція)  $h(t)$

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

За визначенням, граничне значення перехідної функції  $h(t)$  при  $t \rightarrow \infty$  є статичним коефіцієнтом підсилення

$$k_s = \lim_{t \rightarrow \infty} h(t). \quad (1.9)$$

Ця величина має сенс тільки для стійких систем, оскільки при нестійкості перехідний процес не сходиться до кінцевого значення.

Якщо передавальна функція правильна, але не строго правильна (матриця  $D$  моделі в просторі станів не дорівнює нулю), стрибкоподібна зміна вхідного сигналу миттєво приводить до стрибкоподібної зміни виходу. Величина цього стрибка дорівнює відношенню коефіцієнтів при старших степенях чисельника і знаменника передавальної функції (або матриці  $D$  моделі в просторі станів).

За перехідною характеристикою можна знайти найважливіші *показники якості системи – перерегулювання (overshoot) і час перехідного процесу (settling time)* (рис.1.3).

*Перерегулювання* визначається як

$$\sigma = \frac{h_{\max} - h_{\infty}}{h_{\infty}} \times 100\%, \quad (1.10)$$

де  $h_{\max}$  – максимальне значення функції  $h(t)$ , а  $h_{\infty} = \lim_{t \rightarrow \infty} h(t)$  – усталене значення виходу.

*Час перехідного процесу* – це час, після якого сигнал виходу відрізняється від усталеного значення не більш, ніж на задану малу величину (у середовищі МАТЛАВ за замовчуванням використовується точність 2%).

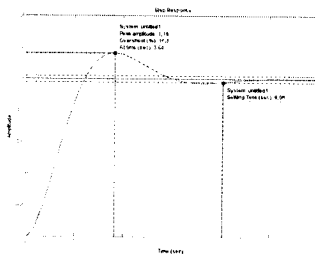


Рисунок 1.3 – Перерегулювання (overshoot) і час перехідного процесу (settling time)

### 1.1.5 Частотна характеристика

При подачі на вхід лінійної системи гармонічного (синусоїдального) сигналу  $u(t) = \sin \omega t$  з частотою  $\omega$  (вона вимірюється в радіанах в секунду), на виході буде також гармонічний сигнал тієї ж частоти, але іншої амплітуди і фази  $y(t) = A \sin(\omega t + \varphi)$ , де  $A$  – амплітуда і  $\varphi$  – зсув фази (це не стосується нелінійних систем).

Частотна характеристика визначається як реакція системи на комплексний експоненціальний сигнал  $e^{j\omega t} = \cos \omega t + j \sin \omega t$ . Для її побудови потрібно використовувати підстановку  $s = j\omega$  у передавальній функції  $W(s)$ . Вигляд  $W(j\omega)$  називається частотною передавальною функцією або амплітудно-фазовою частотною характеристикою системи (АФЧХ).

Залежність модуля величини  $W(j\omega)$  від частоти називають амплітудно-частотною характеристикою (АЧХ) (рис. 1.4), а залежність аргументу комплексного числа (фази)  $W(j\omega)$  від частоти – фазочастотною характеристикою (ФЧХ):

$$A(\omega) = |W(j\omega)|, \quad \varphi(\omega) = \arg W(j\omega) = \arctg \frac{\text{Im } W(j\omega)}{\text{Re } W(j\omega)}. \quad (1.11)$$

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

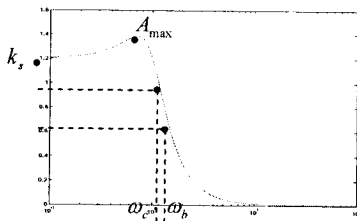


Рисунок 1.4 – Амплітудно-частотна характеристика

Реальні об'єкти мають строго правильну передавальну функцію, тому їх АЧХ убиває зі зростанням частоти і асимптотично прагне до нуля. Говорять, що такий об'єкт має *властивість фільтра* – фільтрує (не пропускає) високочастотні сигнали (завади, шуми вимірювань). Ця властивість слугує основою для використання методу гармонічного балансу.

Частота, після якої значення АЧХ зменшується нижче 0 дБ (коефіцієнт підсилення менший 1, сигнал ослаблюється), називається *частотою зрізу* системи  $\omega_c$ . Частота, після якої значення АЧХ падає нижче -3 дБ (коефіцієнт підсилення менший 0.708), називається *смугою пропускання* системи  $\omega_s$ . Для її обчислення використовують команду `>> b = bandwidth ( f )`.

Максимум АЧХ відповідає частоті, на якій підсилення найбільше. *Значення АЧХ при  $\omega = 0$  дорівнює підсиленню при постійному сигналі, тобто, статичному коефіцієнту підсилення  $k_s$* . Це впливає і з рівності

$$\lim_{\omega \rightarrow 0} A(\omega) = \lim_{\omega \rightarrow 0} |W(j\omega)| = \lim_{s \rightarrow 0} W(s) = k_s. \quad (1.12)$$

Для систем з інтегровальними ланками частотна характеристика прагне до нескінченності при  $\omega \rightarrow 0$ . Це означає, що їх вихід нескінченно збільшується або зменшується при постійному вхідному сигналі.

Щоб побудувати частотні характеристики в MATLAB, потрібно спочатку створити масив частот в потрібному діапазоні. Для цього можна використовувати функції `linspace` (рівномірний розподіл точок за лінійною шкалою) і `logspace` (рівномірний розподіл точок за логарифмічною шкалою). Команда `>> w = linspace ( 0, 10, 100 )`; будувє масив зі 100 крапок з рівномірним кроком в інтервалі від 0 до 10, а команда `>> w = logspace (-1, 2, 100)`; – масив з 100 точок з рівномірним кроком за логарифмічною шкалою в інтервалі від  $10^{-1}$  до  $10^2$ .

Частотна характеристика на сітці `w` для лінійної моделі `f` (заданою як передавальна функція, модель в просторі станів або у формі «нулі-полюси») обчислюється за допомогою функції `freqresp`

`>> r = freqresp(f, w);`

Функція `freqresp` повертає тривимірний масив. Це пов'язано з тим, що вона застосовна і до багатовимірних моделей (з декількома входами і виходами), передавальна функція яких є матрицею. Перші два індекси позначають рядок і стовпець в цій матриці, а третій – номер точки частотної характеристики. Для системи з одним входом і одним виходом зручно перетворити тривимірний масив в одновимірний командою `>> r = r(:);`

Для виведення графіка АЧХ на екран можна використовувати команди MATLAB

```
>> plot ( w, abs(r) );
>> semilogx ( w, abs(r) );
>> loglog ( w, abs(r) );
```

У першому випадку масштаб обох осей координат лінійний, в другому – використовують логарифмічний масштаб по осі абсцис (частот), в останньому – логарифмічний масштаб по обох осях. Для обчислення фази (у градусах) використовують команду  $\gg \text{phi} = \text{angle}(r) * 180 / \text{pi}$ ; після чого можна будувати ФЧХ, наприклад,  $\gg \text{semilogx} ( w, \text{phi} );$

### 1.1.6 Полюси і нулі

Багато динамічних властивостей системи (наприклад, швидкодія, перерегулювання) визначається полюсами передавальної функції (або, що те ж саме, власними числами матриці моделі в просторі станів).

Передавальну функцію можна записати як добуток передавальних функцій елементарних ланок першого і другого порядків. Таким чином, безліч полюсів передавальної функції стійкої системи складають полюси передавальних функцій двох типів простих ланок: аперіодичних і коливальних [3].

**Аперіодична ланка** з передавальною функцією вигляду  $F(s) = \frac{1}{Ts+1}$  має єдину характеристику – постійну часу  $T$ . Починаючи приблизно з частоти  $\omega_0 = 1/T$ , АЧХ такої ланки починає убувати, наближаючись до нуля (значення  $\omega_0$  повертається функцією **damp** як власна частота для дійсного полюсу).

**Коливальна ланка** має передавальну функцію

$$F(s) = \frac{1}{T^2 s^2 + 2\zeta Ts + 1}, \quad (1.13)$$

де  $T$  – постійна часу і  $0 < \zeta < 1$ .

Частота  $\omega_0 = 1/T$  називається *власною частотою (natural frequency)*, а параметр  $\zeta$  – *параметром загасання* або *безрозмірним коефіцієнтом демпфування (damping factor)*. При зменшенні  $\zeta$  імпульсна і перехідна функції набувають яскраво вираженого коливального характеру, а на АЧХ з'являється «горб» в районі частоти  $\omega_0$ . У граничному випадку при  $\zeta = 0$  колювання стають незагасимими, а ланка називається *консервативною*. З іншого боку при  $\zeta = 1$  корені знаменника стають дійсними, і ланка перетворюється на аперіодичну ланку другого порядку.

Для знаходження полюсів передавальної функції **f** можна використовувати функцію  $\gg \text{p} = \text{pole} ( \text{f} )$ .

Виклик функції  $\gg [\text{w0}, \text{zeta}, \text{p}] = \text{damp} ( \text{f} )$  дозволяє знайти не тільки полюси **p**, але також відповідні їм власні частоти **w0** і коефіцієнти демпфування **zeta** у вигляді масивів.

Нулі передавальної функції **f** обчислюються як  $\gg \text{z} = \text{zero} ( \text{f} );$

Стійкість системи не залежить від розташування нулів, але вони істотно впливають на перехідні процеси. Команда  $\gg \text{pzmap} ( \text{f} );$  будує карту

розташування нулів (вони позначаються кружками) і полюсів (хрестики) системи на комплексній площині.










## 1.2 Послідовність виконання роботи




Основна частина команд вводиться в командному вікні середовища MATLAB. Команди, які потрібно застосовувати в інших вікнах, позначені іконками відповідних програм. [1]

Таблиця 1.1 – Послідовність виконання роботи

Етап виконання завдання	Команди MATLAB
1. Очистіть робочий простір MATLAB (пам'ять).	clear all
2. Очистіть вікно MATLAB.	Clc
3. Подивіться коротку довідку щодо команди tf.	help tf
4. Визначте адресу файлу, який виконує цю команду.	which('tf')
5. Введіть передавальну функцію $F(s) = \frac{n_2 s^2 + n_1 s + n_0}{s^3 + d_2 s^2 + d_1 s + d_0}$ як об'єкт tf. (Всі коефіцієнти потрібно взяти з таблиці в кінці файлу).	n = [n2 n1 n0] d = [1 d2 d1 d0] f = tf ( n, d )
6. Перевірте, як «витагувати» з цього об'єкта чисельник і знаменник передавальної функції.	[n1,d1] = tfdata ( f, 'v' )
7. Знайдіть нулі і полюси передавальної функції.	z = zero ( f ) p = pole ( f )
8. Знайдіть коефіцієнт підсилення ланки в усталеному режимі.	k = dcgain ( f )
9. Визначите смугу пропускання системи (найменшу частоту, на якій АЧХ стає менше, ніж -3 дБ).	b = bandwidth ( f )
10. Побудуйте модель системи в просторі стану.	f_ss = ss ( f )
11. Зробіть так, щоб коефіцієнт прямої передачі ланки дорівнював 1.	f_ss.d = 1
12. Знайдіть новий коефіцієнт підсилення ланки в усталеному режимі.	k1 = dcgain ( f_ss )
13. Як пов'язані коефіцієнти $k$ і $k_1$ ? Чому?	



14. Побудуйте модель початкової системи у формі «нулі-полюси».	$f_{zp} = zp_k (f)$
15. Перевірте, які змінні є в робочому просторі.	who чи whos (в чому різниця?)
16. Побудуйте на графіку розташування нулів і полюсів системи.	pzmap (f)
17. Визначите коефіцієнти демпфування і власні частоти для всіх елементарних ланок (першого і другого порядку).	[wc,ksi,p] = damp (f)
18. Запустіть модуль <b>LTIViewer</b> .	Ltview
19. Загрузіть модель f.	 LTI Viewer File – Import
20. Побудуйте імпульсну характеристику (вагову функцію) цієї системи.	 ПКМ – Plot Types - Impulse
21. Загрузіть модель f_ss.	 LTI Viewer File – Import
22. Перевірте, чи побудована імпульсна характеристика другої системи?	 LTI Viewer ПКМ – Systems
23. Відключіть систему f. Чому однаково побудовані імпульсні характеристики різних систем?	 LTI Viewer ПКМ – Systems
24. Підключіть обидві системи.	 LTI Viewer ПКМ – Systems
25. Побудуйте перехідні характеристики систем.	 LTI Viewer ПКМ – Plot Types – Step
26. Зробіть, щоб на графіку для кожної функції були відмічені: <ul style="list-style-type: none"> <li>• максимум</li> <li>• час перехідного процесу</li> </ul> (за замовчуванням в MATLAB час перехідного процесу визначається для двовідсоткового відхилення від сталого значення) <ul style="list-style-type: none"> <li>• час наростання (від 10% до 90% усталеного значення)</li> <li>• усталене значення</li> </ul>	 ПКМ – Characteristics: <ul style="list-style-type: none"> <li>• Peak Response</li> <li>• Settling Time</li> <li>• Rise Time</li> <li>• Steady State</li> </ul>
27. Натискаючи мишею по мітках-кружках, виведіть на екран рамки з чисельними значеннями цих параметрів і розташуйте їх так, щоб всі числа були видні.	
28. Експортуйте побудований графік в окреме вікно.	 LTI Viewer File – Print to Figure
29. Скопіюйте графік в буфер обміну в форматі векторного метафайлу.	(ввести у командному вікні MATLAB) print –dmeta

30. Вставте графік з буферу обміну в звіт ( <i>Microsoft Word</i> ).	 ПКМ – Вставить
31. Закрийте вікно <b>LTIVIEWER</b> .	
32. Створіть масив частот для побудови частотної характеристики (100 точок в інтервалі від $10^{-1}$ до $10^2$ з рівномірним розподіленням на логарифмічній шкалі) (крапка з комою в кінці команди пригнічує виведення на екран результату виконання. Це зручно при роботі з великими масивами).	<code>w = logspace(-1, 2, 100);</code>
33. Порахуйте частотну характеристику вихідної системи... (Частотна характеристика повертається у вигляді тривимірного масиву, в якому кожен елемент має 3 індекси: рядок, стовпець (для багатовимірних моделей) і номер точки частотної характеристики. Для системи з одним входом і одним виходом команда <code>r = r(:)</code> ; перетворить ці дані у звичайний одно-вимірний масив).	<code>r = freqresp ( f, w );</code> <code>r = r(:);</code>
34. ... і побудуйте її на осях з логарифмічним масштабом по осі абсцис.	<code>semilogx ( w, abs(r) )</code>
35. Скопіюйте графік в буфер обміну в форматі векторного метафайлу.	<code>print -dmeta</code>
36. Вставте графік з буферу обміну в звіт ( <i>Microsoft Word</i> ). Поясніть, де на графіку можна знайти коефіцієнт підсилення в статичному режимі і як визначити смугу пропускання системи.	 ПКМ – Вставить
37. Закрийте всі зайві вікна, крім командного вікна <b>MATLAB</b> .	
38. Побудуйте сигнал, імітуючий прямокутні імпульси одиничної амплітуди з періодом 4 секунди (всього 5 імпульсів).	<code>[u,t] = gensig('square',4);</code>
39. Виконаєте моделювання і побудуйте на графіку сигнал виходу системи <code>f</code> при даному вході.	<code>lsim ( f, u, t )</code>
40. Скопіюйте графік в буфер обміну у форматі векторного метафайла.	<code>print -dmeta</code>
41. Вставте графік із буфера обміну в звіт ( <i>Microsoft Word</i> ).	 ПКМ – Вставить

Таблиця 1.2 –Таблиця коефіцієнтів

Варіант	$n_2$	$n_1$	$n_0$	$d_2$	$d_1$	$d_0$
1	2.0	0.60	-0.360	1.2000	0.7406	0.2734
2	2.1	1.68	0.315	1.3250	0.8281	0.2734
3	2.2	-2.42	0.616	1.3059	0.7696	0.2076
4	2.3	-0.46	-0.552	1.4235	0.8401	0.2076
5	2.4	0.24	-0.480	1.3889	0.7531	0.1543
6	2.5	2.25	0.500	1.5000	0.8086	0.1543
7	2.6	0.26	-0.780	1.2421	0.6139	0.1108
8	2.7	-0.27	-0.810	1.1368	0.5717	0.1108
9	2.8	0.28	-0.840	0.8000	0.3700	0.0500
10	2.9	3.19	0.870	0.7000	0.3500	0.0500

### 1.3 Контрольні питання до захисту лабораторної роботи

#### 1. Що таке

- передавальна функція,
- нулі і полюси передавальної функції,
- імпульсна характеристика (вагова функція),
- перехідна функція,
- частотна характеристика,
- модель в просторі станів,
- модель вигляду «нулі-полюси»,
- коефіцієнт підсилення в статичному режимі,
- смуга пропускання системи,
- час перехідного процесу,
- частота зрізу системи,
- власна частота коливальної ланки,
- коефіцієнт демпфування коливальної ланки?

#### 2. У яких одиницях вимірюються:

- коефіцієнт підсилення в статичному режимі,
- смуга пропускання системи,
- час перехідного процесу,
- частота зрізу системи,
- власна частота коливальної ланки,
- коефіцієнт демпфування коливальної ланки?

#### 3. Як пов'язана власна частота з постійною часу коливальної ланки?

#### 4. Чи може четвірка матриць

$$A = \begin{bmatrix} 1 & 2 & 1 \\ -1 & -3 & -1 \\ 2 & 2 & 2 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, C = [1 \quad 2], D = 0$$

бути моделлю системи в просторі станів? Чому? Які співвідношення між матрицями мають виконуватися в загальному випадку?

5. Як отримати коротку довідку про будь-яку команду MATLAB?
6. У чому різниця між командами MATLAB *who* і *whos* *clear all* і *clc*?
7. Як ввести передавальну функцію  $F(s) = \frac{2s+3}{s^2+4s+5}$ ?
8. Як впливає зміна коефіцієнта прямої передачі (матриці  $D$  в моделі в просторі станів) на статичний коефіцієнт підсилення?
9. Які можливості надає модуль **LTIVIEWER**?
10. Що можна сказати про імпульсну характеристику системи `f_ss`? Чому вона не була побудована правильно?
11. Як знайти:
  - коефіцієнт підсилення в сталому режимі за АЧХ,
  - смугу пропускання системи за АЧХ?
12. Як скопіювати графік з вікна MATLAB в іншу програму?
13. Як побудувати масив з 200 значень в інтервалі від  $10^{-3}$  до  $10^3$  з рівномірним розподілом на логарифмічній шкалі?
14. Які величини відкладаються по осях на графіку АЧХ?

## ЛАБОРАТОРНА РОБОТА № 2 ПРОЕКТУВАННЯ РЕГУЛЯТОРА ДЛЯ ЛІНІЙНОЇ СИСТЕМИ

**Мета роботи:** освоєння методів проектування регулятора для одновимірної лінійної безперервної системи за допомогою середовища MATLAB [1].

### Завдання роботи:

- навчитися будувати моделі з'єднань лінійних ланок;
- навчитися використовувати модуль **Sisotool** для проектування простих регуляторів.

## 2.1 Основні теоретичні відомості

### 2.1.1 Моделі з'єднань систем

Для побудови моделей з'єднань систем в MATLAB використовуються знаки арифметичних дій. Ці операції перевантажені, тобто, перевизначені спеціальним чином для об'єктів класів **tf**, **ss** і **zpk**.

Введемо початкові моделі, з якими виконуватимемо всі операції:

```
>> f = tf(1, [1 1]);
```

```
>> g = tf(1, [2 1]);
```

- паралельне з'єднання (рис. 2.1)

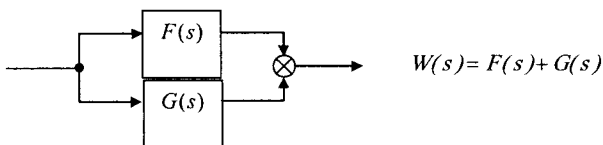


Рисунок 2.1 – Паралельне з'єднання

```
>> w = f + g
```

Transfer function:

```
 3 s + 2  
-----  
2 s^2 + 3 s + 1
```

- послідовне з'єднання (рис. 2.2)

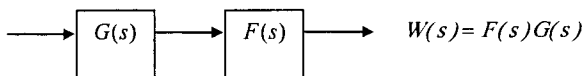


Рисунок 2.2 – Послідовне з'єднання

```
>> w = f * g
Transfer function:
      1
```

---


$$2s^2 + 3s + 1$$

- контур з від'ємним зворотним зв'язком (рис. 2.3)

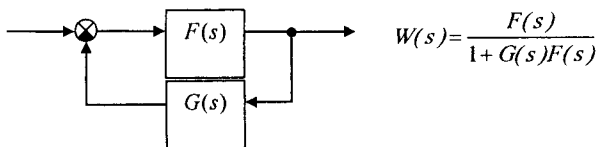


Рисунок 2.3 – Контур з від'ємним зворотним зв'язком

```
>> w = feedback(f, g)
Transfer function:
      2s + 1
```

---


$$2s^2 + 3s + 2$$

Можна обчислити цю передавальну функцію і так

```
>> w = f / (1 + g*f)
```

```
Transfer function:
      2s^2 + 3s + 1
```

---


$$2s^3 + 5s^2 + 5s + 2$$

Цей результат може показатися несподіваним. Річ у тому, що обидві передавальні функції мають перший порядок, тобто, описуються диференціальним рівнянням (ДР) першого порядку. Тому вся система має описуватися другим порядком, а ми отримали третій. Щоб розібратися в цьому, перетворимо модель до форми «нулі-полюси»

```
>> w_zpk = zpk(w)
```

```
Zero/pole/gain:
(s+1) (s+0.5)
-----
(s+1) (s^2 + 1.5s + 1)
```

Видно, що чисельник і знаменник передавальної функції містять спільний множник  $s+1$ , який можна скоротити, і залишається система другого порядку. Для цього потрібно побудувати *мінімальну реалізацію*, скоротивши спільні множники

```
>> w = minreal ( w )
Transfer function:
s + 0.5
-----
s^2 + 1.5 s + 1
```

Ця передавальна функція збігається з тією, що видає функція **feedback**.

- контур з додатним зворотним зв'язком (рис. 2.4)

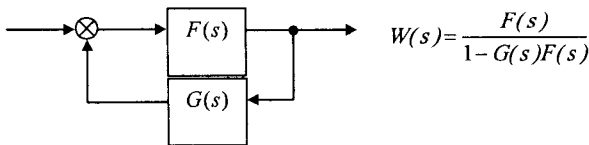


Рисунок 2.4 – Контур з додатним зворотним зв'язком

```
>> w = feedback(f, -g)
або
>> w = feedback(f, g, 1)
або
>> w = minreal ( f/(1 - g*f) )
Transfer function:
2 s + 1
-----
2 s^2 + 3 s
```

### 2.1.2 Кореневий годограф

Багато важливих властивостей системи (наприклад, швидкодія, перерегулювання) визначаються розташуванням коренів характеристичного рівняння на комплексній площині.

Простий спосіб корекції системи – застосувати П-регулятор (підсилювач з коефіцієнтом  $k$ ), який змінює коефіцієнт підсилення розімкненої системи і розташування цих коренів. При зміні  $k$  від 0 до  $\infty$  корені описують криві, які називаються *кореневим годографом* (цим терміном також називається сукупність траєкторій, які описують корені характеристичного рівняння при зміні *будь-якого* числового коефіцієнта в системі) (рис. 2.5).

За допомогою модуля **Sisotool** (скорочення *Siso*=single Input Single Output позначає систему з одним входом і одним виходом) можна вибрати потрібне розташування коренів (і відповідний коефіцієнт підсилення), «перетягуючи» їх мишкою. Відмітимо, що при переміщенні одного кореня зміщуються і всі інші, оскільки система має один ступінь свободи – коефіцієнт підсилення контуру, що змінюється [3].

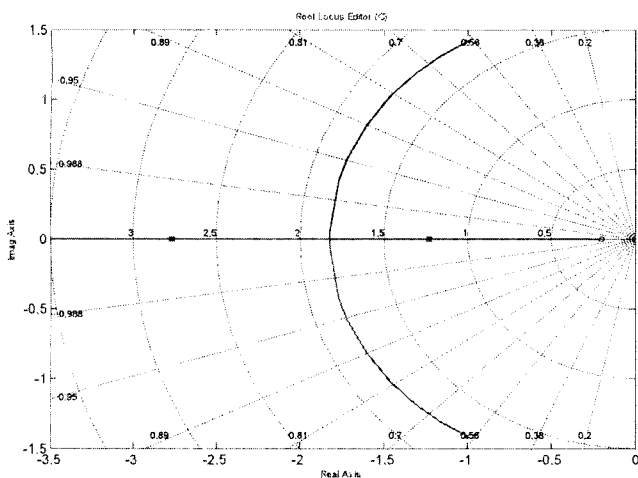


Рисунок 2.5 – Кореневий годограф

Корені при вибраному коефіцієнті підсилення  $k$  зображуються фіолетовими квадратиками. Кінці годографа для кожного кореня помічені хрестиком ( $k = 0$ ) і кружком ( $k = \infty$ ). Сітка (для її виведення потрібно натиснути ПКМ на графіці і вибрати пункт **Grid**) показує лінії однакових показників коливальності (коефіцієнта демпфування, *damping factor*) – прямі, що виходять з початку координат, і лінії рівних власних частот (*natural frequency*) – кола з центром у початку координат.

У контекстному меню (ПКМ) можна встановити обмеження на розташування полюсів так, щоб перерегулювання і час перехідного процесу не перевищували заданих. Для цього потрібно вибрати пункт **Design Constraints - New** і вибрати у низхідному списку **Percent Overshoot** (пере-



регулювання у відсотках) або **Settling Time** (час перехідного процесу з 2% точністю). Обмеження показуються у вигляді меж заборонених зон (рис. 2.6).

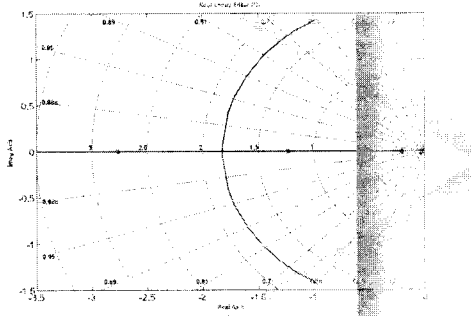


Рисунок 2.6 – Межі заборонених зон кореневого годографа

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

$$T_m \approx -\frac{\ln \Delta}{\eta} \approx \frac{3,912}{\eta}, \quad (2.1)$$

де  $\Delta$  – величина допустимої помилки (у МАТЛАВ вона приймається такою, що дорівнює 2% або 0,02). Таким чином, при обмеженні тільки на  $T_m$  область допустимого розташування коренів  $p_i$  є півплощина  $Re p_i < -\frac{3,912}{T}$ .

Вимоги до коефіцієнта демпфування додають обмеження у вигляді сектора

$$\beta = \max_i \left| \frac{Im p_i}{Re p_i} \right| < \frac{\sqrt{1-\zeta^2}}{\zeta^2}. \quad (2.2)$$

Число  $\beta$  називають *коливальністю* або *ступенем коливальності* замкнутої системи. Кожному заданому  $\zeta$  відповідає деяке значення  $\beta$ .

Перерегулювання (у відсотках) оцінюється за формулою

$$\sigma \approx e^{-\frac{\pi \zeta}{\sqrt{1-\zeta^2}}} \times 100\%. \quad (2.3)$$

Кожному перерегулюванню відповідає своє значення  $\zeta$  і свій сектор, що обмежує розташування коренів.

Таким чином, при використанні двох обмежень (перше – на  $T_n$ , друге – на  $\sigma$  чи  $\zeta$ ) областю допустимого розташування коренів є зрізаний сектор в лівій частині рисунка 2.6. Якщо перетяганням коренів (тобто, зміною підсилення контуру) не вдається мати в своєму розпорядженні полюси в цій області, потрібно ускладнювати регулятор, додаючи його нулі і полюси (ПКМ – Add Pole/zero або ПКМ – Edit Compensator).

### 2.1.3 Синтез за допомогою ЛАФЧХ

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

Нехай розімкнена система має передавальну функцію  $W(s)$ . ЛАФЧХ містить дві криві – амплітудно-частотну характеристику (ЛАЧХ)  $L_m(\omega) = 20 \lg |W(j\omega)|$  і фазочастотну (ЛФЧХ)  $\varphi(\omega) = \arg W(j\omega)$ .

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

Розклавши чисельник і знаменник передавальної функції  $W(s)$  на співмножники першого і другого порядків, можна подати ЛАЧХ системи як суму ЛАЧХ елементарних ланок (аперіодичних, коливальних, інтегровальних, диференціальних та ін.) Для  $W(j\omega) = \frac{N_1(j\omega) \dots N_n(j\omega)}{D_1(j\omega) \dots D_q(j\omega)}$  отримуємо, використовуючи властивості логарифма,  $20 \lg |W(j\omega)| = 20 \lg |N_1(j\omega)| + \dots + 20 \lg |N_n(j\omega)| - 20 \lg |D_1(j\omega)| - \dots - 20 \lg |D_q(j\omega)|$ .

Раніше будували асимптотичні ЛАЧХ вручну, підсумовуючи ЛАЧХ окремих ланок. У середовищі MATLAB існують засоби, що дозволяють автоматизувати побудову точних (не асимптотичних) ЛАФЧХ. При цьому можна використовувати накопичений за багато років класичний досвід проектування.

Низькочастотна частина ЛАЧХ визначає точність системи, середньочастотна (поблизу частоти зрізу  $\omega_c$ ) – стійкість і якість перехідного процесу, високочастотна – чутливість до завад. Якщо система містить інтегратор, низькочастотна частина має ненульовий нахил (20 дБ на декаду для одного інтегратора), постійний сигнал відстежується без усталеної помилки. Для системи з двома інтеграторами ЛАФЧХ має в області низьких частот нахил

40 дБ на декаду, без усталеної помилки відстежується не тільки постійний, а й сигнал, що лінійно зростає. Складніші вимоги до точності приводять до того, що ЛАЧХ не має заходити в деякі заборонені області.

*Запас стійкості за амплітудою*  $g_m$  (в дБ) – це відстань від ЛАЧХ до горизонтальної прямої  $L_m = 0$  дБ на частоті, на якій фазова характеристика перетинає пряму  $\varphi = -180^\circ$ . На цій частоті коефіцієнт підсилення системи має бути менше 1 (або  $L_m(\omega) < 0$ ) (рис. 2.7).

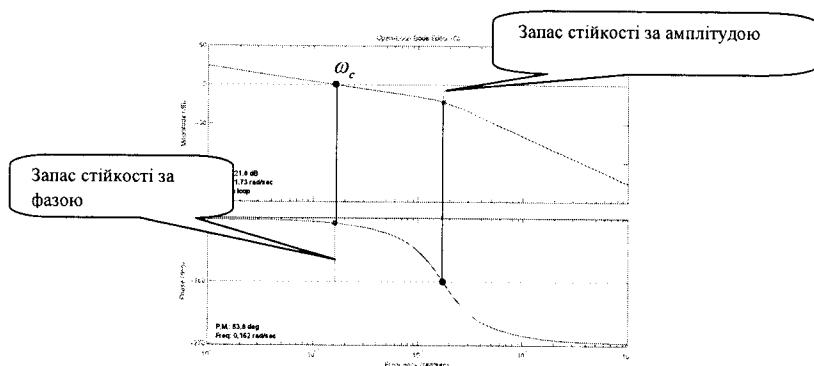


Рисунок 2.7 – Запаси стійкості логарифмічної амплітудної частотної характеристики

*Запас стійкості за фазою*  $\varphi_m$  (в градусах) – це відстань від частотної характеристики до горизонтальної прямої  $\varphi = -180^\circ$  на частоті зрізу  $\omega_c$ . На цій частоті фазова характеристика має мати значення *більше*  $-180^\circ$ .

Допустимим вважається запас за амплітудою не менше 6 дБ і запас за фазою не менше 30 градусів.

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

Зазвичай потрібно, щоб система мала високу точність (великий коефіцієнт підсилення за контуром) для низьких частот і пригнічувала високо-

частотні завади (мала низьке підсилення в області високих частот). Частота зрізу вибирається, виходячи з вимог до швидкодії. Таким чином, типова ЛАЧХ має вигляд, показаний на рис. 2.8. Сірим кольором показані заборонені області, які визначаються вимогами до точності і придушення завад.

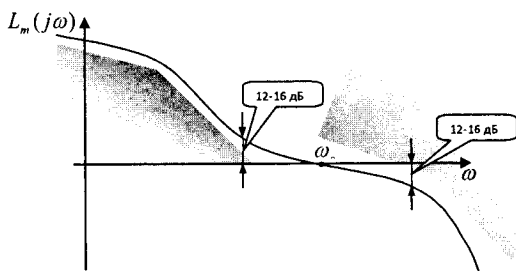


Рисунок 2.8 – Типова логарифмічна амплітудна частотна характеристика із забороненими областями

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

У загальному випадку будується бажана ЛАЧХ  $L_A(j\omega)$ , що задовольняє вимоги до системи, потім ЛАЧХ послідовного коректувального пристрою визначається як різниця між  $L_A(j\omega)$  і ЛАЧХ існуючої розімкненої системи [2].

#### 2.1.4 Точність в усталеному режимі

Нехай передавальну функцію розімкненої системи можна подати у вигляді

$$W(s) = KW_1(s), \quad (2.4)$$

де  $\lim_{s \rightarrow 0} W_1(s) = 1$ .

Тоді передавальна функція замкнутої системи помилково дорівнює

$$\Phi_{\varepsilon}(s) = \frac{1}{1 + KW_1(s)}. \quad (2.5)$$

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

$$\varepsilon_{\infty} = \lim_{t \rightarrow \infty} \varepsilon(t) = \lim_{s \rightarrow 0} s \Phi_{\varepsilon}(s) X(s) = \frac{x_0}{1 + K}. \quad (2.6)$$

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

Тепер нехай

$$W(s) = \frac{K}{s^{\nu}} W_1(s), \quad (2.7)$$

де  $\nu$  – ціле число і  $\lim_{s \rightarrow 0} W_1(s) = 1$ .

Тоді для всіх вхідних сигналів вигляду  $x(t) = x_0 + x_1 t + x_2 t^2 + \dots + x_{\nu-1} t^{\nu-1}$  система забезпечуватиме нульову сталу помилку при будь-яких значеннях коефіцієнтів  $x_0, \dots, x_{\nu-1}$ . Таким чином, при  $\nu > 0$  система відстежує постійний сигнал без сталої помилки. Такі системи називають *астатичними*.

Число  $\nu$  називається *порядком астатизму*. Для сигналу

$$x(t) = x_{\nu} t^{\nu}, \quad X(s) = \frac{\nu! x_{\nu}}{s^{\nu+1}} \quad (2.8)$$

усталена помилка дорівнює

$$\varepsilon_{\infty} = \frac{\nu! x_{\nu}}{K}. \quad (2.9)$$

Вище розглянуто випадок астатизму стосовно задавального впливу. Аналогічно може йти мова про астатизм стосовно збурювального впливу.

### 2.1.5 Прості типи регуляторів

**П-регулятор.** Простий пропорційний регулятор (П-регулятор) – це звичайний підсилювач з передавальною функцією

$$C(s) = K_c. \quad (2.10)$$

**ПД-регулятор.** Для поліпшення якості регулювання та підвищення швидкодії в закон управління вводять похідну від сигналу помилки, так що отримана передавальна функція пропорційно-диференціального регулятора (ПД-регулятора) може бути подана у вигляді

$$C(s) = K_c(1 + T_D s), \quad (2.11)$$

де  $T_D$  – постійна часу диференціувальної ланки. На практиці реалізувати ідеальне диференціювання неможливо, оскільки частотна характеристика ланки нескінченно збільшується на високих частотах. Тому використовують диференціувальну ланку з додатковим фільтром

$$C(s) = K_c \left( 1 + \frac{T_D s}{T_I s + 1} \right). \quad (2.12)$$

Тут постійна часу фільтра  $T_I$  зазвичай у 3–10 разів менша, ніж  $T_D$ . Надмірне збільшення  $T_D$  може призвести до нестійкості системи, зменшення цієї величини затягує перехідний процес.

**ПІД-регулятор.** На відміну від ПД-регулятора, він містить інтегратор і система стає астатичною як за задавальною дією, так і за збуренням (тобто, постійне збурення повністю компенсується). Його передавальна функція має вигляд

$$C(s) = K_c \left( 1 + \frac{T_D s}{T_I s + 1} + \frac{1}{T_I s} \right). \quad (2.13)$$

де  $T_I$  – постійна часу інтегрувальної ланки.

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

За допомогою правильно налаштованого ПІД-регулятора в більшості випадків вдається забезпечити виконання всіх вимог до системи. Завдяки своїй простоті вони стали найпоширенішими. За статистикою більше 90% всіх промислових регуляторів є саме ПІД-регуляторами [3].

## 2.2 Опис системи

У роботі розглядається система регулювання напруги генератора. Її спрощена а) та структурна б) схеми АСК показано на рисунку 2.9.

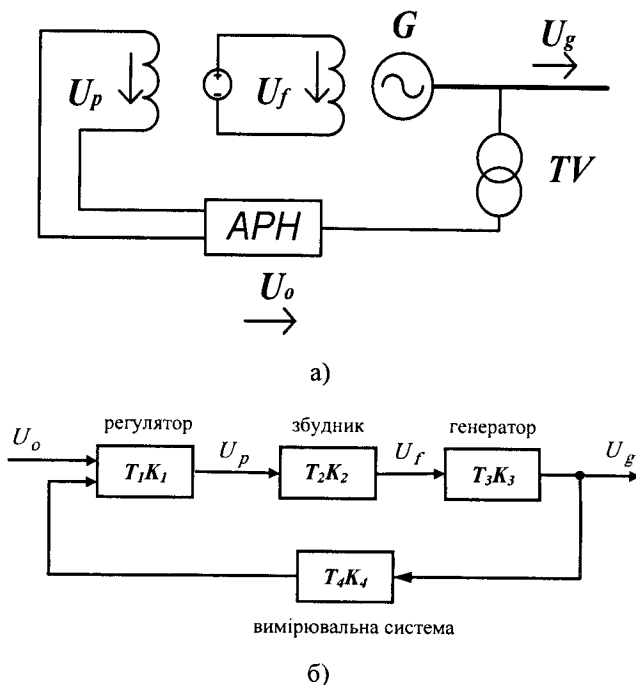


Рисунок 2.9 – Спрощена а) та структурна б) схеми системи регулювання напруги генератора

Система автоматичного регулювання складається з трьох інерційних ланок першого порядку (рис. 2.9, б)), які описуються такими диференціальними рівняннями в операторній формі:

– ланка регулятора

$$(T_1 p + 1)U_p = k_1(U_0 - U_g). \quad (2.14)$$

– ланка збудника

$$(T_2 p + 1)U_f = k_2 U_p. \quad (2.15)$$

– ланка генератора

$$(T_3 p + 1)U_g = k_3 U_f, \quad (2.16)$$

де  $U_g, U_f, U_p, U_0$  – відповідно, напруга на виходах ланок генератора, збудника, регулятора і задавального пристрою;  $T_3, T_2, T_1$  – постійні часу, відповідно, ланок генератора, збудника і регулятора.

Генератор потужністю  $P_g=25$  МВт, напруга  $U_g=6,3$  кВ,  $T_g=9,9$  с.

Збудник  $P_{z\sigma}=130$  кВт, напруга  $U_f=250$  В, швидкість зміни напруги 330 В/с.

$T_3=T_2=250/330=0,76$  с;  $U_{\sigma}-U_g^{ycm}=22$  В.

Коефіцієнт трансформації  $n_{TV} = 6300/220$ ; напруга на виході  $U_p=220$  В; коефіцієнт підсилення регулятора  $\kappa_p=220/22=10$ ; постійна часу  $T_p=T_I=0,5$  с;  $\kappa_I = \kappa_p / n_{TV} = (220/6300) \cdot 10 = 0,35$ ;  $\kappa_2 = (250/220) = 1,13$ ;  $\kappa_3 = 6300/250 = 25$ .

Передавальна функція від напруги збудника до напруги генератора запишеться у вигляді

$$P(s) = \frac{K_3}{T_s s + 1}. \quad (2.17)$$

Збудник приблизно моделюється як інтегрувальна ланка

$$R_0(s) = \frac{K_2}{T_R s + 1}. \quad (2.18)$$

Для вимірювання напруги на затискачах генератора використовується вольтметр, математична модель якого записується у вигляді аперіодичної ланки першого порядку з передавальною функцією

$$H(s) = \frac{1}{T_{oc} s + 1}. \quad (2.19)$$




Чисельні значення  $T_s, T_R$  і  $T_{oc}$  потрібно взяти з таблиці в кінці файлу.

### 2.3 Послідовність виконання роботи







Основна частина команд вводиться в командному вікні середовища МАТЛАВ. Команди, які треба застосовувати в інших вікнах, позначені іконами відповідних програм [1].



Таблиця 2.1 – Послідовність виконання роботи

Етап виконання завдання	Команди MATLAB
1. Введіть передавальну функцію моделі генератора $P(s) = \frac{K_3}{T_s s + 1}$ як об'єкт <b>tf</b> .	$P = \text{tf} ( K_3, [T_s \ 1] )$
2. Введіть передавальну функцію моделі інтегрувальної ланки $R_0(s) = \frac{1}{T_R s}$ .	$R_0 = \text{tf} ( 1, [T_R \ 0] )$
3. Побудуйте передавальну функцію плавного перемикача, замкнувши інтегратор одиничним від'ємним зворотним зв'язком	$R = \text{feedback} ( R_0, 1 )$
4. Побудуйте передавальну функцію послідовного з'єднання генератора зі збудником.	$G = P * R$
5. Побудуйте перехідну характеристику для отриманої моделі та скопіюйте її в звіт через буфер обміну. Поясніть, чому функція нескінченно зростає і прагне до прямої. Який коефіцієнт нахилу цієї прямої? Закрийте вікно з графіком.	$\text{step} ( G )$
6. Побудуйте передавальну функцію вимірювального пристрою $H(s) = \frac{1}{T_{oc} s + 1}$ .	$H = \text{tf} ( 1, [T_{oc} \ 1] )$
7. Побудуйте передавальну функцію розімкненого контуру.	$L = G * H$
8. Побудуйте ЛАФЧХ розімкненої системи (у зарубіжній літературі ЛАФЧХ називають діаграмою Боде)	$\text{bode} ( L )$
9. Відзначте точки, що визначають перетин ЛАЧХ з прямою 0 дБ і перетин ЛФЧХ з прямою $-180^\circ$ .	 <b>Figure No. 1</b> ПКМ – Characteristics – Stability (Minimum Margin)
10. Визначте, чи є замкнута система стійкою? Які запаси стійкості по амплітуді ( <i>Gain margin</i> ) і фазі ( <i>Phase margin</i> )? Який регулятор неявно використовується в цьому випадку? Скопіюйте графік ЛАФЧХ у звіт.	 <b>Figure No. 1</b> ЛКМ на мітках-кружках
11. Знайдіть максимальний коефіцієнт підсилення розімкненої системи. Поясніть цей результат.	 <b>Figure No. 1</b> ПКМ – Characteristics – Peak Response

Продовження таблиці 2.1

<p>12. Закрийте вікно з ЛАФЧХ і запустіть модуль <b>SISOTool</b> (Single Input Single Output, система з одним входом і виходом)</p>	<p>Sisotool</p>
<p>13. Імпортуйте передавальну функцію <math>G</math> як модель об'єкта (<i>Plant</i>) і <math>H</math> як модель датчика (<i>Sensor</i>). Блоки <math>F</math> (передфільтр) і <math>C</math> (регулятор) залиште без змін (дорівнюють 1).</p>	<p> File - Import</p>
<p>14. Відключіть зображення кореневого годографа так, щоб у вікні залишилась тільки ЛАФЧХ.</p>	<p> View – Root Locus (отключить)</p>
<p>15. Для того, щоб відразу бачити зміни перехідних процесів, запустіть <b>LTIVIEWER</b> (Linear Time-Invariant, лінійна стаціонарна система) з верхнього меню вікна <b>SISOTool</b>. Розташуйте два вікна поряд, щоб вони не перекривали один одного.</p>	<p> Analysis – Response to Step Command</p>
<p>16. Залиште тільки графік перехідного процесу на виході, відключивши вивід сигналу керування.</p>	<p> ПКМ – Systems – Closed loop r to u</p>
<p>17. Визначте перерегулювання <math>\sigma</math> і час перехідного процесу <math>T_p</math> (за замовчуванням в Matlab час перехідного процесу визначається для двовідсоткового відхилення від сталого значення) Скопіюйте графік у звіт.</p>	<p> ПКМ – Characteristics –</p> <ul style="list-style-type: none"> <li>• Peak Response</li> <li>• Settling Time</li> </ul>
<p>18. Перейдіть у вікно <b>SISOTool</b>. Визначте коефіцієнт підсилення, при якому перерегулювання приблизно дорівнює 10%. Як змінився час перехідного процесу? Які запаси стійкості в цьому випадку? Скопіюйте графік в звіт.</p>	<p> перетягання мишею ЛАЧХ, редагування в полі <i>Current Compensator</i></p>

Продовження таблиці 2.1

<p>19. Перейдіть у вікно середовища MATLAB і введіть передавальну функцію пропорційно-диференціального (ПД) регулятора</p> $C_{pd}(s) = 1 + \frac{T_s s}{T_v s + 1}, \text{ де } T_v = 1 \text{ сек,}$ <p>а <math>T_s</math> – постійна часу генератора.</p>	$Cpd = 1 + tf([Ts\ 0], [Tv\ 1])$
<p>20. Перейдіть у вікно <b>SISOTool</b>. Імпортуйте регулятор <math>Cpd</math> як базову модель для блока <math>C</math>.</p>	<p><b>SISO Design Tool</b> File – Import, Cpd -&gt; C</p>
<p>21. Визначите додатковий коефіцієнт посилення, при якому перерегулювання приблизно рівне 10%. Знайдіть час перехідного процесу і запаси стійкості. Порівняйте пропорційний і ПД-регулятори. Скопіюйте у звіт графік перехідного процесу.</p>	<p><b>SISO Design Tool</b> перетягання мишею ЛАЧХ, редагування в полі <i>Current Compensator</i></p>
<p>22. Визначите додатковий коефіцієнт посилення, при якому час перехідного процесу мінімальний. Скопіюйте у звіт графік перехідного процесу.</p>	<p><b>SISO Design Tool</b> перетягання мишею ЛАЧХ, редагування в полі <i>Current Compensator</i></p>
<p>23. Експортуйте отриманий регулятор у робочу область MATLAB.</p>	<p><b>SISO Design Tool</b> File – Export в стовпці <i>Export as</i> змінити ім'я Cpd на C кнопка <i>Export to workspace</i></p>
<p>24. Побудуйте передавальну функцію отриманої замкнутої системи. Подумайте, чому вийшов такий громіздкий вираз. Який має бути порядок передавальної функції?</p>	$W = C * G / (1 + C * G * H)$
<p>25. Побудуйте мінімальну реалізацію передавальної функції <math>W</math>.</p>	$W = \text{minreal}(W)$
<p>26. Визначте полюси передавальної функції замкнутої системи. Що означає близькість деяких полюсів до уявної осі? Чи правильно, що в цьому випадку буде малий запас стійкості?</p>	$\text{pole}(W)$

Продовження таблиці 2.1

<p>27. Знайдіть коефіцієнт підсилення системи в усталеному режимі. Поясніть результат. Чи є у такої системи статична помилка при відслідковуванні постійного сигналу? Чому? А для сигналу, що лінійно зростає?</p>	<p>dcgain ( W )</p>
<p>28. Побудуйте мінімальну реалізацію передавальної функції замкнутої системи від входу до сигналу управління (виходу регулятора).</p>	<p><math>W_u = \text{minreal}(C/(1 + C*G*H))</math></p>
<p>29. Побудуйте зміну сигналу управління при одиничному ступінчастому вхідному сигналі і скопіюйте графік в звіт. Поясніть, чому сигнал управління прагне до нуля.</p>	<p>step ( Wu )</p>

Таблиця 2.2 – Таблиця коефіцієнтів

Варіант	$T_s$ , сек	$T_R$ , сек	$T_{oc}$ , сек
1.	9.8	0.2	5
2.	9.9	0.3	6
3.	10.2	0.4	1
4.	12.6	0.5	2
5.	13.8	0.6	3
6.	14.4	0.7	4
7.	15.6	0.8	5
8.	17.8	0.9	6
9.	18	1	1
10.	18.2	2	6

## 2.4 Контрольні питання до захисту лабораторної роботи

1. Що означають скорочення SISO, LTI?
2. Як отримати передавальну функцію за лінійними диференціальними рівняннями системи?
3. Як ввести передавальну функцію у вікні MATLAB?
4. За допомогою яких операцій (функцій) будуються в MATLAB моделі паралельного і послідовного з'єднань, системи зі зворотним зв'язком?
5. Як побудувати ЛАФЧХ розімкненої системи?
6. Як визначаються запаси стійкості за амплітудою та фазою? Що означають ці величини? У яких одиницях вони вимірюються?
7. Які можливості надає модуль **SISOTool**?
8. Що таке
  - кореневий годограф,
  - пере регулювання,
  - час перехідного процесу?
9. Як впливає збільшення коефіцієнта посилення контуру на ЛАФЧХ?
10. Чому в диференціовальній частині ПД-регулятора використовується додатковий фільтр у вигляді аперіодичної ланки з постійною часу  $T_v$ ?
11. Які переваги дає використання ПД-регулятора порівняно з П-регулятором?
12. Як впливає збільшення коефіцієнта підсилення контуру на перерегулювання і час перехідного процесу?
13. Що таке астатична система? Що таке порядок астатизму?

# ЛАБОРАТОРНА РОБОТА № 3

## МОДЕЛЮВАННЯ СИСТЕМ КЕРУВАННЯ В ПАКЕТИ SIMULINK

**Мета роботи:** освоєння методів моделювання лінійних систем в пакеті Simulink [1].

### **Завдання роботи:**

- навчитися будувати і редагувати моделі систем управління в пакеті SIMULINK;
- навчитися змінювати параметри блоків;
- навчитися будувати перехідні процеси;
- навчитися оформляти результати моделювання;
- вивчити метод компенсації постійних збурень за допомогою ПІД-регулятора.

### **3.1 Основні теоретичні відомості**

#### **3.1.1 Створення моделей в SIMULINK**

Пакет SIMULINK призначений для моделювання систем. Вся модель будується з блоків, що мають входи і виходи. Існує бібліотека стандартних блоків, крім того, можна створювати свої власні блоки будь-якої складності. Існує дві групи спеціальних пристроїв – джерела сигналів (**Sources**) і пристрою виведення (**Sinks**).

Блоки мають назви. Для того, щоб змінити назву, потрібно клацнути по ньому ЛКМ і відредагувати текст (у деяких версіях МАТЛАВ існують проблеми з російською буквою «Я», тому рекомендується використовувати англійські назви блоків).

Кожен блок має свої властивості, що налаштовуються. Для їх зміни потрібно двічі клацнути на блоці і змінити потрібні значення в діалоговому вікні.

Для того, щоб повернути блок на 90 градусів, потрібно виділити його і натиснути клавіші **Ctrl+R**. Комбінація **Ctrl+I** дозволяє виконати дзеркальне відображення входів і виходів.


Верхнє меню **Format** призначене для зміни оформлення виділеного блока. Також для цієї мети можна використовувати контекстне меню **ПКМ - Format**. Для виділеного блока можна змінити колір тексту і лінії (**Foreground color**), колір фону (**Background color**), вивести тінь (**Show drop shadow**), перемістити назву на іншу сторону (**Flip name**).

Для виділення одного блока або сполучної лінії потрібно клацнути ЛКМ на потрібному елементі. Для того, щоб виділити декілька блоків, потрібно «обвести» їх при натиснутій ЛКМ. Клавіша **Delete** видаляє виділену частину. Щоб скопіювати блок (або виділену частину), потрібно «перетягнути» його при натиснутій *правій* кнопці миші (ПКМ) [4].

Блоки з'єднуються лініями зв'язку, по яких розповсюджуються сигнали. Для того, щоб з'єднати блоки, потрібно клацнути ЛКМ по джерелу сигналу і потім, при натиснутій клавіші **Ctrl**, по блоку-приймачу. Можна також «протягнути» мишкою лінію зв'язку між потрібним виходом і входом.

Щоб подати один сигнал на два блоки (зробити «розвилку»), потрібно спочатку створити одну лінію звичайним способом. Щоб провести другу лінію, потрібно натиснути *праву* кнопку миші на лінії в точці розвилки і протягнути лінію до другого блока.

Модель можна скопіювати в буфер обміну у вигляді растрового рисунка. Для цього у вікні моделі потрібно вибрати у верхньому меню пункт **Edit - Copy model to clipboard**. Заздалегідь краще зменшити розміри вікна до мінімальних, щоб не було білих полів.

Для того, щоб запустити моделювання, потрібно клацнути ЛКМ по кнопці  на панелі інструментів. Ця ж кнопка дозволяє зупинити моделювання за необхідності.

Параметри моделювання (метод інтеграції, обробка помилок) встановлюються за допомогою вікна **Simulation - Parameters**. Найважливіші параметри – це час моделювання (**Stop time**) і метод чисельної інтеграції рівнянь (**Solver options**).

### 3.1.2 Основні джерела сигналів (Sources)



**Constant** – сигнал постійної величини.



**Step** – ступінчастий сигнал, міняється час стрибка (**Step Time**), початкове (**Initial Value**) і кінцеве (**Final Value**) значення.



**Ramp** – сигнал, який лінійно зростає з заданим нахилом (**Slope**). Можна задати також час початку зміни сигналу (**Start Time**) і початкове значення (**Initial Value**).



**Pulse Generator** – генератор прямокутних імпульсів, задаються амплітуда (**Amplitude**), період (**Period**), ширина (**Pulse Width**, у відсотках від періоду), фаза (**Phase Delay**).



**Repeating Sequence** – послідовність імпульсів, їх форма задається у вигляді пар чисел (час; величина сигналу)



**Sine Wave** – синусоїдний сигнал, задається амплітуда (**Amplitude**), частота (**Frequency**), фаза (**Phase**) і середнє значення (**Bias**).



**Signal Builder** – будівник сигналів, що дозволяє задавати форму сигналу, «перетягуючи» мишею опорні точки.



**Random Number** – випадкові числа з нормальним (гаусовим) розподілом. Можна задати середнє значення (**Mean Value**), дисперсію (**Variance**), період зміни сигналу (**Sample Time**).



**Uniform Random Number** – випадкові числа з рівномірним розподілом в заданому інтервалі від **Minimum** до **Maximum**.



**Band Limited White Noise** – випадковий сигнал, обмежений, за смугою, білий шум (що має рівномірний спектр до деякої частоти). Блок використовується як джерело білого шуму для моделей непе-

рвних систем. Задається інтенсивність (**Noise Power**) і інтервал дискретизації (**Sample Time**), протягом якого утримується постійне значення сигналу. Чим менший інтервал, тим точніше моделювання, проте більші обчислювальні витрати [4].

### 3.1.3 Основні пристрої виведення (Sinks)



**Display** – цифровий дисплей, показує зміну вхідного сигналу в цифровому вигляді.



**Scope** – осцилограф, показує зміну сигналу у вигляді графіка, дозволяє передавати дані в робочу область MATLAB для подальшої обробки і оформлення.

### 3.1.4 Лінійні системи (Continuous)



**Transfer Fcn** – передавальна функція, в параметрах задаються чисельник (**Numerator**) і знаменник (**Denominator**) у вигляді поліномів.



**State Space** – модель в просторі станів, в параметрах задається четвірка матриць, що визначають модель і початкові умови для вектора стану (**Initial conditions**).



**Zero-Pole** – модель у формі «нулі-полюси», в параметрах задаються масиви нулів (**Zeros**), полюсів (**Poles**), а також коефіцієнт підсилення (**Gain**).



**Integrator** – інтегратор з можливістю встановлення початкових умов (**Initial condition**), а також меж насичення (**Lower saturation limit** і **Upper saturation limit**).

### 3.1.5 Інші часто використовувані блоки

#### Math Operations



**Gain** – підсилювач, задається коефіцієнт посилення (**Gain**).



**Sum** – суматор, використовується для складання і віднімання входів. Параметр **List of signs** задає кількість входів, їх знаки («+» для складання і «-» для віднімання). Проміжки між входами (позначаються знаком **)**).



**Trigonometric Function** – тригонометрична функція.

#### Signal Routing



**Manual Switch** – ручний перемикач, дозволяє подвійним клацанням перемикати вихід на один з двох вхідних сигналів.



**Mux** – мультиплексор, об'єднує декілька сигналів в один «джгут» (векторний сигнал), в параметрах задається число входів (**Number of Inputs**).



**Demux** – демультиплексор, дозволяє «розбити» векторний сигнал на декілька скалярних, в параметрах задається число виходів (**Number of Outputs**).



### 3.1.6 Блок Scope

У вікні блока **Scope** зображається графік зміни вхідного сигналу (рис. 3.1). Якщо вхід з'єднаний з виходом мультиплексора, відразу будується декілька графіків (за розмірністю вхідного «джгута»).

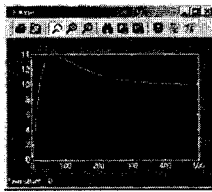





Рисунок 3.1 – Вікно блока Scope

За замовчуванням на осі ординат використовується діапазон від -5 до 5. Якщо цей варіант не підходить, вибрати масштаб автоматично (так, щоб весь графік був видний) можна за допомогою кнопки . Сусідня кнопка  зберігає ці налаштування для наступних запусків.

Кнопка  відкриває вікно налаштувань, причому найбільш важливі дані містяться на вкладці **Data history**. Якщо не скинути прапорець **Limit data points**, в пам'яті зберігатиметься тільки задане число точок графіка, тобто, при великому часі моделювання початок графіка буде втрачений.

Відзначивши на цій же сторінці прапорець **Save data to workspace** можна відразу передати результати моделювання в робочу область MATLAB для того, щоб їх можна було далі обробляти, виводити на графіки і зберігати у файлі. Поле **Variable name** задає ім'я змінної в робочій області, в якій зберігаються дані. У простому випадку вибирається формат **Array** (у списку **Format**). Це означає, що дані зберігатимуться в масиві з декількох стовпців (перший стовпець – час, другий – перший сигнал, третій – другий сигнал і так далі, за порядком входів мультиплексора) [4].

### 3.1.7 Оформлення графіків

Для створення нового вікна для рисунка в MATLAB використовується команда `>> figure(1);`

Замість одиниці можна ставити будь-який номер рисунка. Якщо рисунок з таким номером вже є, він стає активним і виводиться на перший план. Якщо такого рисунка немає, він створюється і стає активним.

У MATLAB є можливість будувати декілька графіків на одному рисунку. Інакше кажучи, рисунок можна розбити на «клітки», в кожній з яких буду-

ється окремий графік. Для цього потрібно зробити активним необхідний рисунок і застосувати команду `>> subplot(2, 1, 1);`

Перше число в команді `subplot` показує кількість «рядків» в такій матриці, друге – кількість стовпців, третє – який, за рахунком, графік зробити активним (рахуючи стовпці справа наліво і рядки зверху вниз). Всі подальші команди (`plot`, `title`, `xlabel`, `ylabel`, `legend` і ін.) відносять до цього «підграфіка».

У командах можна передавати як аргументи не цілі масиви, а їх частини. Наприклад, за командою `>> plot(x(1:20), y(11:30));`будується графік, на якому по осі абсцис відкладаються значення елементів масиву `x` з номерами від 1 до 20, а по осі ординат – відповідні їм значення з масиву `y` з номерами від 11 до 30.

Двокрапка означає «всі рядки» або «всі стовпці». Наприклад, за командою `>> plot(x(:,1), x(:,2));` будується залежність між першим і другим стовпцями масиву `x` (тут двокрапка замість першого індексу позначає «всі рядки»).

За допомогою команди `plot` (й інших подібних – `semilogx`, `semilogy`, `loglog`) можна будувати декілька ліній на одному графіку. Для цього серед аргументів перераховуються пари масивів `>> plot(x, y, v, z);`

Перша лінія показуватиме залежність `y` від `x`, а друга – залежність `z` від `v`. Масиви в кожній парі мають бути однакової довжини. За бажання можна вказати кольори для кожної лінії, наприклад, `>> plot(x, y, 'b', v, z, 'g');`

Перша лінія (залежність `y` від `x`) буде синьою, друга (залежність `z` від `v`) – зеленою. Можна використовувати такі кольори:

- `b` синій (`blue`)
- `g` зелений (`green`)
- `r` червоний (`red`)
- `c` голубий (`cyan`)
- `m` фіолетовий (`magenta`)
- `y` жовтий (`yellow`)
- `k` чорний (`black`)

За замовчуванням перша лінія – синя, друга – зелена і так далі, в порядку перерахування кольорів в списку. Додатково можна вказати тип лінії

- `-` – суцільна;
- `:` – точкова;
- `-.` – штрих-пунктирна;
- `--` – штрихова.

Наприклад, `>> plot(x, y, 'b:', v, z, 'g--');`

Перша лінія – точкова синього кольору, друга – штрихова зеленого кольору. За замовчуванням всі лінії суцільні.

Для оформлення графіка також використовуються команди:

**title** – заголовок графіка;

**xlabel** – назва осі абсцис;

**ylabel** – назва осі ординат.

У всіх цих команд обов'язковий один аргумент – текст в апострофах.

Команда **legend** служить для виведення легенди графіка. Легенда потрібна, якщо на графіці є декілька ліній і потрібно показати, що позначає кожна з них. Параметрами команди **legend** є символічні рядки, їх має бути стільки, скільки побудовано ліній.

У написах можна використовувати деякі команди системи TeX (TeX – краща в світі система підготовки текстів з математичними формулами, розроблена математиком і програмістом Дональдом Кнутом (автором багатотомного видання «Мистецтво програмування для ЕОМ»). Найчастіше використовуються так звані макропакети (надбудови), що розширюють можливості ядра TeX, наприклад LaTeX чи AMSTeX). Наприклад, грецькі букви записуються у вигляді « $\backslash\alpha$ », « $\backslash\beta$ » і т. д. Верхній індекс (ступінь) позначається знаком « $\wedge$ », Наприклад,  $a^2$  запишеться як « $a^2$ ». Для позначення індексу використовують нижнє підкреслення, наприклад  $a_{22}$  кодується як « $a_{22}$ » [5].

### 3.1.8 Компенсація постійних збурень

На будь-який генератор в реальних умовах діють *збурювальні сили*, викликані аварійними режимами роботи енергосистеми. Деякі з них (наприклад, короткі замикання) містять постійну складову, тобто, їх середнє значення не дорівнює нулю. Проте система управління має підтримувати значення номінальної напруги навіть в таких умовах. Збурювальні сили і моменти прикладені безпосередньо до входу об'єкта управління (рис. 3.2).

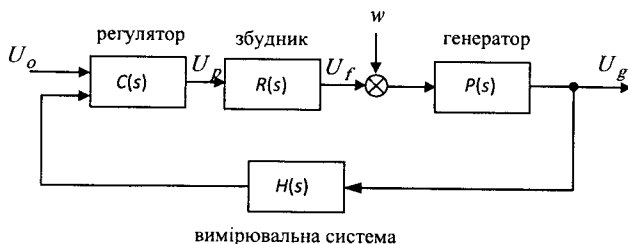


Рисунок 3.2 – Структурна схема системи управління

Придушення збурень (позначених на схемі через  $w$ ) визначається передавальною функцією системи зі збурення, тобто, передавальною функцією від входу  $w$  до виходу  $\varphi$

$$W_w(s) = \frac{P(s)}{1 + R(s)C(s)H(s)P(s)}. \quad (3.1)$$

Якщо вона містить нуль в точці  $s = 0$ , відповідна АЧХ дорівнює нулю на нульовій частоті, тобто, постійні збурення в усталеному режимі компенсуються повністю. Для цього потрібно, щоб інтегратор входив в модель приводу, зворотного зв'язку або регулятора. Таким чином, якщо регулятор містить інтегральний канал (І-канал), в системі немає статичної помилки при постійному збуренні  $w$ .

### 3.2 Опис системи

У роботі розглядається система регулювання напруги генератора. Її спрощена а) та структурна б) схеми АСК показані на рисунку 3.3.

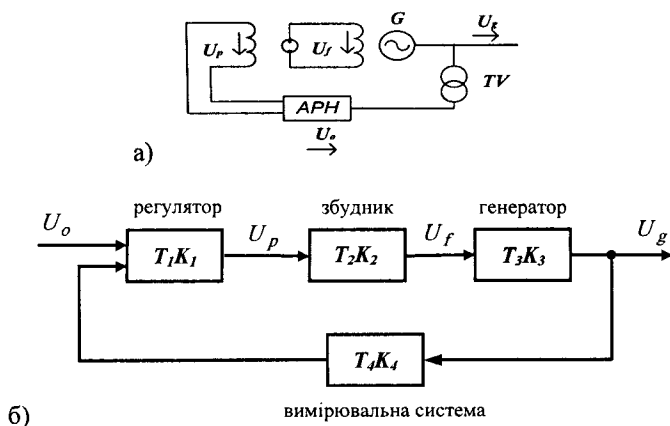


Рис. 3.3 – Спрощена а) та структурна б) схеми системи регулювання напруги генератора

Система автоматичного регулювання складається з трьох інерційних ланок першого порядку (рис. 3.3, б), які описуються нижченаведеними диференціальними рівняннями в операторній формі:

– ланка регулятора  $(T_1 p + 1)U_p = k_1(U_0 - U_g); \quad (3.2)$

$$- \text{ ланка збудника } (T_2 p + 1) U_f = k_2 U_p; \quad (3.3)$$

$$- \text{ ланка генератора } (T_3 p + 1) U_g = k_3 U_f, \quad (3.4)$$

де  $U_g, U_f, U_p, U_0$  – відповідно, напруга на виходах ланок генератора, збудника, регулятора та задавального пристрою;  $T_3, T_2, T_1$  – постійні часу, відповідно, ланок генератора, збудника та регулятора.

Генератор потужністю  $P_g=25$  МВт, напруга  $U_g=6,3$  кВ,  $T_g=9,9$  с.

Збудник  $P_{зб}=130$  кВт, напруга  $U_f=250$  В, швидкість зміни напруги  $330$  В/с.

$$T_3=T_2=250/330=0,76 \text{ с}; U_0-U_g^{ycm}=22 \text{ В.}$$

Коефіцієнт трансформації  $n_{TV} = 6300/220$ ; напруга на виході  $U_p=220$  В; коефіцієнт підсилення регулятора  $\kappa_p=220/22=10$ ; постійна часу  $T_p=T_l=0,5$  с;  $\kappa_l = \kappa_p / n_{TV} = (220/6300) \cdot 10 = 0,35$ ;  $\kappa_2 = (250/220) = 1,13$ ;  $\kappa_3 = 6300/250 = 25$ .

Передавальна функція від напруги збудника до напруги генератора записується у вигляді

$$P(s) = \frac{K_3}{T_s s + 1}. \quad (3.5)$$

Збудник приблизно моделюється як інтегрувальна ланка

$$R_0(s) = \frac{K_2}{T_R s + 1}. \quad (3.6)$$

Для вимірювання напруги на затискачах генератора використовується вольтметр, математична модель якого записується у вигляді аперіодичної ланки першого порядку з передавальною функцією

$$H(s) = \frac{1}{T_{oc} s + 1}. \quad (3.7)$$

Досліджуються перехідні процеси в системі при використанні ПД-регулятора



$$C(s) = K_c \left( 1 + \frac{T_s s}{T_v s + 1} \right), \quad (3.8)$$

і ПІД-регулятора

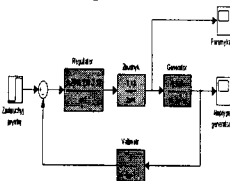
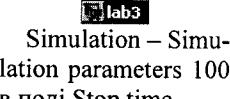
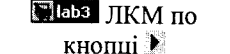
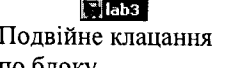
$$C(s) = K_c \left( 1 + \frac{T_s s}{T_v s + 1} \right) + \frac{1}{T_I s}. \quad (3.9)$$

### 3.3 Послідовність виконання роботи









Таблиця 3.1 – Послідовність виконання роботи [1]

Етап виконання завдання	Команди MATLAB
1. Для запуску пакета SIMULINK натисніть кнопку  у командному вікні MATLAB чи введіть команду <b>simulink</b> у командному рядку.	simulink
2. Створіть нову модель за допомогою верхнього меню висхідного вікна <i>Simulink Library Browser</i> .	 <b>Simulink Library Browser</b> File – New – Model
3. Перетягніть блок <b>Transfer Fcn</b> (передавальна функція) з вікна <i>Simulink Library Browser</i> (група <i>Continuous</i> ) у вікно моделі і введіть чисельник і знаменник передавальної функції моделі генератора.	Подвійне клацання на блоці a. Numerator [K] miniator [Ts 1 0]
4. Дайте блоку назву <b>Generator</b> .	ЛКМ на імені блока
5. Аналогічно додайте ще три блоки типу <b>Transfer Fcn</b> , назвіть їх <b>Zbydnuk</b> , <b>Regulator</b> і <b>Voltmetr</b> , введіть потрібні параметри.	
6. Збережіть модель у своїй папці під іменем <b>lab3.mdl</b> (Всі файли моделей в пакеті SIMULINK мають розширення <b>.mdl</b> )	File – Save
7. Виділіть блок <b>Voltmetr</b> і поверніть його в іншу сторону.	Натиснути Ctrl+I чи двічі натиснути Ctrl+R.
8. Зробіть, щоб назва блоків <b>Generator</b> , <b>Zbydnuk</b> і <b>Regulator</b> були над блоками.	ПКМ на блоці, Format - Flip name
9. Виберіть колір блоків на свій смак.	ПКМ на блоці, Bacground color
10. Перетягніть у вікно моделі блок <b>Sum</b> з групи <b>Math Operations</b> і встановіть його зліва від регулятора.	ЛКМ
11. Зробіть так, щоб другий вхід враховувався у сумі зі знаком мінус (від'ємний зворотний зв'язок).	Подвійне клацання на блоці, ввести  +- в поле List of signs
12. Перетягніть у вікно моделі блок <b>Step</b> з групи <b>Sources</b> і установіть його зліва від суматора. Дайте йому ім'я <b>Zadane znachenia</b> .	

Продовження таблиці 3.1




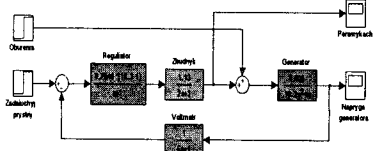

<p>13. Установіть час подачі сигналу 0 і величину сигналу 6,3 (досліджуємо встановлення номінальної напруги 6,3 кВ).</p>	<p>Подвійне клацання на блоці, 0 в полі Step time; 6,3 в полі Final value</p>
<p>14. З'єднайте всі блоки потрібним способом.</p>	<p>ЛКМ на джерелі, утримувати Ctrl і ЛКМ на приймачі, чи перетягнути ЛКМ від виходу одного блока до входу іншого</p>
<p>15. Перетягніть у вікно моделі два блоки <b>Scope</b> (осцилограф) з групи <b>Sinks</b> і установіть їх у правій частині. Назвіть їх <b>Peremykach</b> і <b>Napruga</b>.</p>	
<p>16. Зробіть, щоб на перший блок <b>Scope</b> надходив сигнал керування (значення зміни напруги, після блока <b>Zbydnuak</b>), а на другий – сигнал виходу (напруга генератора). Збережіть модель.</p>	<p>Натиснути ПКМ на лінії в точці відбору сигналу, потім, не відпускаючи ПКМ, тягнути лінію до входу блока.</p>
<p>17. Зменшіть вікно до мінімального розміру, при якому видні всі елементи, і скопіюйте модель в буфер обміну. Потім вставте її з буфера обміну у звіт.</p>	<p></p>
<p>18. Установіть час моделювання 100 с.</p>	<p></p>
<p>19. Виконайте моделювання.</p>	<p></p>
<p>20. Прогляньте результати моделювання, відкривши вікна для блоків <b>Peremykach</b> і <b>Napruga</b>.</p>	<p></p>

Продовження таблиці 3.1


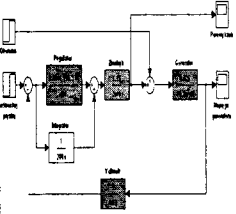

<p>21. Налаштуйте масштаб по осях у вікнах обох блоків.</p>	<p> Курс ЛКМ по кнопці  – установити оптимальний масштаб</p>
<p>22. Збережіть налаштування.</p>	<p> Курс ЛКМ по кнопці </p>
<p>23. Зробіть так, щоб результати моделювання передавались з обох блоків <b>Scope</b> у робочу область MATLAB у вигляді матриць, у яких перший стовпець – час, а другий – сигнал (напряга генератора чи значення зміни напруги).</p>	<p> Курс ЛКМ по кнопці  вкладка Data history <input type="checkbox"/> Limit data points <input checked="" type="checkbox"/> Save data to workspace Variable name: phi (напряга) чи delta (перемикач) Format: Array</p>
<p>24. Виконайте моделювання ще раз.</p>	<p> lab3 ЛКМ по кнопці </p>
<p>25. Перейдіть в командне вікно MATLAB і створіть нове вікно для графіка. В одному вікні будуть побудовані дві криві на різних осях.</p>	<p>figure(1);</p>
<p>26. Розбийте вікно на 2 частини по вертикалі і зробіть активним перший графік. Перше число в команді <b>subplot</b> означає кількість комірок з графіками по вертикалі, друге – по горизонталі, третє – номер комірки, яку потрібно зробити активною. (При введенні цієї і наступних команд вікно з графіком не з'являється на екрані. Щоб побачити зміни, треба вручну зробити його активним, клацнувши мишею на відповідній кнопці в панелі задач.)</p>	<p>subplot(2, 1, 1);</p>
<p>27. Побудуйте графік зміни курсу. В команді <b>plot</b> спочатку вказують масив абсцис, потім – масив ординат. Двокрапка означає, що використовуються всі рядки.</p>	<p>plot(phi(:,1),phi(:,2)) ;</p>
<p>28. Введіть заголовок графіка.</p>	<p>title('Напряга генератора');</p>



Продовження таблиці 3.1

<p>29. Введіть назви осей координат. Всередині апострофів для введення грецьких букв дозволяється використовувати команди LaTeX, Наприклад, «\phi» означає грецьку букву <math>\phi</math>, а «\delta» – букву <math>\delta</math>.</p>	<p>xlabel('Час, сек'); ylabel('\phi, напруга');</p>
<p>30. Аналогічно побудуйте у другій комірці графік зміни кута повороту керма, використовуючи дані з масиву <b>delta</b>, отриманого в результаті моделювання.</p>	<pre>subplot(2, 1, 2); plot(delta(:, 1), delta(:, 2)); title('Положення перемикача'); xlabel('Час, сек'); ylabel('\delta, відпайки');</pre>
<p>31. Скопіюйте побудований графік у звіт.</p>	<p>print -dmeta</p>
<p>32. Видаліть у вікні моделі зв'язок між приводом і об'єктом.</p>	<p> ЛКМ по лінії, натиснути Delete.</p>
<p>33. Додайте ще один блок <b>Sum</b> з групи <b>Math Operations</b> і встановіть його на звільнене місце. Налаштуйте розташування входів і виходу так, щоб перший вхід був у верхній частині кола.</p>	<p> Подвійне клацання по блоку ++  в полі List of signs</p>
<p>34. Досліджуємо реакцію системи на постійний сигнал, прикладений безпосередньо до входу об'єкта. Він може моделювати якусь постійну збудувальну дію, наприклад, коротке замикання.</p>	
<p>35. Скопіюйте блок <b>Zadaiuchyj prystrij</b>, перетягнувши його правою кнопкою миші, і встановіть для нього величину стрибка 2 кА. Дайте йому назву <b>Oburenia</b>. Підключіть його вихід до нового суматора. Добудуйте потрібні з'єднувальні лінії.</p>	<p> Перетягування ПКМ. Подвійне клацання по блоку 2 в полі Final Value Подвійне клацання по імені</p>
<p>36. Скопіюйте отриману модель у звіт.</p> 	<p> Edit – Copy model to clipboard</p>

Продовження таблиці 3.1

<p>37. Збільшіть час моделювання до 500 і виконайте моделювання. Перевірте, чи збереглося значення номінальної напруги .</p>	<p> <b>lab3</b> Simulation – Simulation parameters – Stop time ЛКМ по кнопці ▶</p>
<p>38. Перейдіть в командне вікно MATLAB і запомніть результати моделювання в нових масивах. Вони знадобляться для того, щоб порівняти вихідний і скоректований варіанти системи.</p>	<p>phi0 = phi; delta = delta;</p>
<p>39. Щоб регулятор компенсував постійну складову збурення, потрібно додати до нього інтегральний канал. Отже, виходить ПІД-регулятор. Підключіть паралельно регулятору інтегровальну ланку з передавальною функцією <math>\frac{1}{T_i s}</math>, <math>T_i = 200</math> сек. Збережіть модель і скопіюйте її у звіт.</p>	
<p>40. Виконайте моделювання. Перевірте, чи збереглося значення номінальної напруги.</p>	<p> <b>lab3</b> ЛКМ по кнопці ▶</p>
<p>41. Побудуйте у верхній частині графіка 2 кривих – перехідні процеси за напругою генератора для ПІД- і ПІД-регуляторів. В команді <code>plot</code> можна перераховувати декілька пар масивів – перша пара відповідає першому графіку, друга – другому і т. д. Три крапки в кінці рядка означають перенесення команди на наступний рядок. Команда <code>legend</code> служить для виведення легенди – символічних рядків, що описують кожний з побудованих графіків.</p>	<pre>subplot(2, 1, 1); plot(phi0(:,1),       phi0(:,2),...       phi(:,1), phi(:,2)); title ('Напруга на шинах генератора'); xlabel('Час, сек'); ylabel('\phi, %'); legend('ПІД- регулятор', ... 'ПІД-регулятор');</pre>
<p>42. Аналогічно побудуйте в нижній частині графіка 2 кривих – зміни кута перекладання керма для ПІД- і ПІД-регуляторів, використовуючи дані з масивів <code>delta0</code> і <code>delta</code>.</p>	
<p>43. Скопіюйте побудований графік у звіт через буфер обміну. Зробіть висновки про вплив інтегрального каналу на перехідні процеси у системі.</p>	

Таблиця 3.2 –Таблиця коефіцієнтів

Варіант	$T_s, c$	$T_R, c$	$T_{oc}, c$
1.	9.8	0.2	5
2.	9.9	0.3	6
3.	10.2	0.4	1
4.	12.6	0.5	2
5.	13.8	0.6	3
6.	14.4	0.7	4
7.	15.6	0.8	5
8.	17.8	0.9	6
9.	18	1	1
10.	18.2	2	6

### 3.4 Контрольні питання до захисту лабораторної роботи

1. Як знайти передавальну функцію інтегратора, охопленого зворотним зв'язком?
2. Як запустити пакет SIMULINK?
3. Що таке *Library Browser*?
4. Яке розширення мають файли-моделі SIMULINK?
5. Як створити нову модель?
6. Як з'єднати два блоки, що мають відповідно вільний вихід і вільний вхід?
7. Як зробити, щоб один і той же сигнал надходив на декілька блоків?
8. Як передати результати моделювання у робочу область MATLAB? В якому вигляді вони передаються?
9. Як видалити блок чи зв'язок між блоками?
10. Як визначити потрібні масштаби для осей координат у вікнах **Scope** і запам'ятати їх?
11. Як скопіювати блок у вікні моделі?
12. Як змінити знаки арифметичних операцій у суматорі?
13. Як скопіювати зображення моделі в документ *Microsoft Word*?
14. Як змінити час моделювання?
15. Як змінити назву блока?
16. Як зробити, щоб назва блока була з іншої сторони?
17. Як змінити колір фона блока? колір надпису?
18. Як ввести параметри блока **Transfer Fcn** (передавальна функція)?
19. Як знайти передавальну функцію системи за збуренням?
20. Чому при використанні ПД-регулятора система не компенсує постійне збурення?

21. Як, знаючи статичний коефіцієнт підсилення за збуренням, визначити установлене відхилення від заданого курсу?
22. Які властивості має мати передавальна функція зі збурення для того, щоб постійне збурення повністю компенсувалось?
23. Які властивості має мати регулятор для того, щоб постійне збурення повністю компенсувалось?
24. Які переваги дає використання інтегрального каналу в ПІД-регуляторі?
25. Чому порядок передаточної функції замкнутої системи за збуренням з ПІД-регулятором на 1 більше, ніж для системи з ПІД-регулятором?
26. Які параметри приймає команда `subplot`?
27. Що означає двокрапка в записі `phi(:,1)`?
28. Як вивести на графік заголовок і назву осей?
29. Як побудувати в одному вікні два різних графіка?
30. Як на одному графіку побудувати декілька кривих?
31. Що таке легенда? Як вивести легенду на графік?
32. Як виводити на графіку букви грецького алфавіта?

## ЛАБОРАТОРНА РОБОТА № 4 МОДЕЛЮВАННЯ НЕЛІНІЙНИХ СИСТЕМ КЕРУВАННЯ

**Мета роботи:** освоєння методів моделювання нелінійних систем в пакеті Simulink [1].

### Завдання роботи:

- навчитися будувати і редагувати моделі систем з підсистемами;
- навчитися використовувати нелінійні ланки типу «насичення»;
- навчитися будувати декілька графіків одночасно на одному осцилографі;
- навчитися створювати, редагувати і налагоджувати скрипти;
- навчитися змінювати властивості елементів графіка (шрифт, товщину лінії);

### 4.1 Основні теоретичні відомості

#### 4.1.1 Моделі нелінійних ланок (*Discontinuities*)



**Saturation** – насичення, в параметрах задаються верхня і нижня межі (**Upper limit** и **Lower limit**).



**Dead zone** – нечутливість, «мертва зона». У параметрах задаються межі нечутливості (**Start of dead zone** и **End of dead zone**).



**Rate Limiter** – обмежувач швидкості зміни сигналу, в параметрах задаються межі на швидкість збільшення (**Rising slew rate**) і на швидкість зменшення (**Falling slew rate**).



**Relay** – реле, в параметрах задаються точки перемикання (**Switch on point** и **Switch off point**), також величини сигналів в режимах «ввімкнено» (**Output when on**) і «вимкнено» (**Output when off**).



**Backlash** – люфт, «мертвий хід». У параметрах задаються величина мертвого ходу (**Deadband width**) і початкове значення виходу (**Initial output**).



**Coulomb and Viscous Friction** – кулонівське і в'язке тертя.

#### 4.1.2 Підсистеми

Якщо на схемі багато блоків, вона стає громіздкою і працювати з такою схемою незручно. Щоб не перенавантажувати схеми, можна об'єднувати блоки в *підсистеми*. Найпростіше виділити потрібні блоки мишкою («обвести» при натиснутій ЛКМ) і натиснути клавіші **Ctrl+G** (або

вибрати пункт меню **Edit – Create subsystem**). На основній схемі підсистема зображається як блок типу **Subsystem** (з групи **Ports and Subsystems**). Цей блок можна додати і вручну, перетягнувши з вікна *Library Browser*.

За допомогою подвійного клацання ЛКМ можна «відкрити» підсистему. Входи позначаються блоками **In**, а виходи – блоками **Out** (також із групи **Ports and Subsystems**). Можна додавати нові входи і виходи, видаляти непотрібні, міняти назви, працюючи з ними так само, як з рештою блоків.

Усередині підсистем можна створювати вкладені підсистеми. У підсистемі не може бути блоків з однаковими назвами, однак в різних підсистемах можуть бути такі блоки.

Зберегти модель можна з вікна будь-якої підсистеми, але закриття вікна підсистеми не приводить до закриття моделі [4].

### 4.1.3 Блок *Scope* (декілька сигналів)

До входу осцилографа (блок **Scope**) можна підключати декілька сигналів одночасно. Для цього їх потрібно об'єднати в один векторний сигнал («жгут») за допомогою блока **Mux** (мультиплексор) з групи **Signal Routing** (рис. 4.1).



Рисунок 4.1 – Блок *Scope* (декілька сигналів)

Якщо використовується два вхідних сигнали, перший зображується жовтою лінією, другий – фіолетовою. При передачі даних в робочу область **MATLAB** у форматі **Array** масив міститиме три стовпці – час, перший сигнал і другий сигнал. Якщо сигналів більше, відповідно збільшується кількість стовпців масиву.

### 4.1.4 Скрипти

При роботі в **MATLAB** часто для отримання потрібного результату потрібно ввести послідовно декілька команд. Якщо з'ясується, що в якійсь команді була зроблена помилка або потрібно змінити початкові дані, всі команди доведеться вводити знову. Щоб не набирати їх вручну, можна записати всю послідовність команд на диск у вигляді текстового файлу (**M-**

файлу з розширенням **.m**), а потім виконувати його, викликаючи за іменем. Такий файл називається *скриптом*.

*Скрипт* – це програма, яка є списком команд мовою системи **МАТЛАВ**. Скрипти можна створювати та редагувати в будь-якому простому текстовому редакторі (наприклад, в *Блокноті*), проте найзручніше використовувати вбудований редактор **МАТЛАВ**, в якому є підсвічування синтаксису (команди, символічні рядки, коментарі й інші елементи програми виділяються різними кольорами).

У **M**-файлі перераховуються послідовно всі необхідні команди. Крапка з комою в кінці команди пригнічує виведення результату на екран. Можна розташовувати в одному рядку декілька команд, розділяючи їх комою (якщо потрібне виведення результату на екран) або крапкою з комою. Якщо потрібно перенести довгу команду на наступний рядок, в кінці рядка ставиться трикрапка.

Коментарем вважається все, що розташоване праворуч від знака **%** до кінця рядка. Його можна ставити в будь-якому місці рядка, наприклад, праворуч від команди **МАТЛАВ**.

Для виклику скрипту потрібно набрати його ім'я в командному вікні **МАТЛАВ** і натиснути клавішу **Enter**. Скрипт має знаходитися в робочій папці (ім'я показане у вікні **Current directory** у верхній частині командного вікна) або в одній з папок, що входять в *шлях для пошуку*. У шлях для пошуку внесені папки, в яких знаходяться **M**-файли для стандартних функцій системи **МАТЛАВ** і додаткових пакетів (*toolbox*). Якщо потрібно, щоб скрипт запускався з будь-якої папки, цю папку вносять в шлях для пошуку (команда **File - Set Path** верхнього меню).

Запустити скрипт можна безпосередньо з вікна редактора **МАТЛАВ**, натиснувши на клавішу **F5**. Можна виконати не весь скрипт, а тільки деякі рядки – їх потрібно виділити і натиснути клавішу **F9**. Можна розташувати два вікна (редактор і командне вікно **МАТЛАВ**) поряд так, щоб вони не перекривали один одного. Тоді при виконанні скрипту (або окремих команд) відразу буде видний результат.

Якщо в командах скрипту є помилки (або вони виникли при виконанні), відповідні повідомлення виводяться в командне вікно **Matlab**.

#### 4.1.5 Форматування графіка

За замовчуванням команда **plot** рисує на активному графіку нову криву, стираючи старі лінії. Для того, щоб існуючі криві збереглися, перед викликом **plot** потрібно виконати команду **>> hold on**

Обернена їй команда **>> hold off** відновлює стандартний режим.

Кожний об'єкт на графіку (осі координат, написи, лінії і тому подібне) являють собою *об'єкти*, які мають властивості. Для того, щоб отримати значення властивості, використовують команду **get**, а для зміни властивості – команду **set**. Скорочення **gca** позначає поточні (активні) осі координат (*get current axes*). Команда **>> get(gca)** виводить на екран всі властивості

осей і їх значення. Для управління розміром шрифту (він вимірюється в пунктах) використовується властивість **Font Size**:

```
>> get(gca, 'FontSize') % визначити розмір шрифту
ans =
    10
```

```
>> set(gca, 'FontSize', 16) % змінити розмір шрифту
```

Для того, щоб налаштувати властивості окремої лінії, потрібно спочатку отримати її *хендл* (*handle* – ручка, рукоятка, покажчик). Так називається унікальний числовий код об'єкта, через який до цього об'єкта можна звертатися. Команда `gca` насправді повертає хендл поточних координатних осей. Команда `>> h = get(gca, 'Children')` запише в змінну `h` масив хендлів – покажчиків на об'єкти лінії. Для кожної лінії можна встановити товщину (у пунктах, за замовчуванням – 0,5 пункту).

```
>> set(h(1), 'LineWidth', 1.5)
```

```
>> set(h(2), 'LineWidth', 1.5)
```

Аналогічно можна управляти й іншими властивостями [5].

## 4.2 Опис системи

У роботі розглядається система регулювання напруги генератора. Її спрощена а) та структурна б) схеми АСК показані на рисунку 4.2.

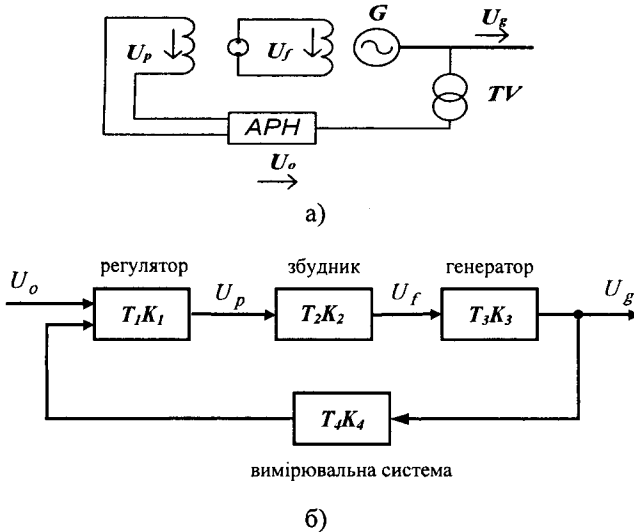


Рисунок 4.2 – Спрощена а) та структурна б) схеми системи регулювання напруги генератора



Система автоматичного регулювання складається з трьох інерційних ланок першого порядку (рис. 4.2, б), які описуються такими диференціальними рівняннями в операторній формі:

– ланка регулятора

$$(T_1 p + 1)U_p = k_1(U_0 - U_g); \quad (4.1)$$

– ланка збудника

$$(T_2 p + 1)U_f = k_2 U_p; \quad (4.2)$$

– ланка генератора

$$(T_3 p + 1)U_g = k_3 U_f, \quad (4.3)$$

де  $U_g, U_f, U_p, U_0$  – відповідно, напруга на виходах ланок генератора, збудника, регулятора та задавального пристрою;  $T_3, T_2, T_1$  – постійні часу відповідно ланок генератора, збудника і регулятора.

Генератор потужністю  $P_g = 25$  МВт, напруга  $U_g = 6,3$  кВ,  $T_g = 9,9$  с.

Збудник  $P_{z6} = 130$  кВт, напруга  $U_f = 250$  В, швидкість зміни напруги 330 В/с.

$T_3 = T_2 = 250/330 = 0,76$  с;  $U_0 - U_g^{ycm} = 22$  В.

Коефіцієнт трансформації  $n_{TV} = 6300/220$ ; напруга на виході  $U_p = 220$  В; коефіцієнт підсилення регулятора  $k_p = 220/22 = 10$ ; постійна часу  $T_p = T_1 = 0,5$  с;  $k_1 = k_p / n_{TV} = (220/6300) \cdot 10 = 0,35$ ;  $k_2 = (250/220) = 1,13$ ;  $k_3 = 6300/250 = 25$ .

Передавальна функція від напруги збудника до напруги генератора запишеться у вигляді

$$P(s) = \frac{K_3}{T_3 s + 1}. \quad (4.4)$$

Збудник приблизно моделюється як інтегрувальна ланка

$$R_0(s) = \frac{K_2}{T_R s + 1}. \quad (4.5)$$

На значення напруги та швидкість її зміни на шинах генератора накладаються нелінійні обмеження  $|\dot{\delta}(t)| < 10$  В/сек,  $|\delta(t)| < 30$  В.

Для вимірювання напруги на затискачах генератора використовується вольтметр, математична модель якого записується у вигляді аперіодичної ланки першого порядку з передавальною функцією

$$H(s) = \frac{1}{T_{oc} s + 1}. \quad (4.6)$$

Як пристрій керування використовується ПІД-регулятор з передавальною функцією




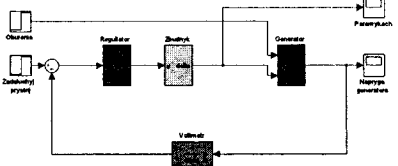

$$C(s) = K_c \left( 1 + \frac{T_d s + 1}{T_v + 1} \right) + \frac{1}{T_i s}, \quad (4.7)$$

де  $T_v = 1$  с і  $T_i = 200$  с.


Числові значення  $T_s$ ,  $T_k$  і  $T_{oc}$  потрібно взяти з таблиці в кінці файлу. Вони мають збігатися з даними, які використовувались Вами в лабораторних роботах № 2 і 3.

### 4.3 Послідовність виконання роботи

Таблиця 4.1 – Послідовність виконання роботи [1]

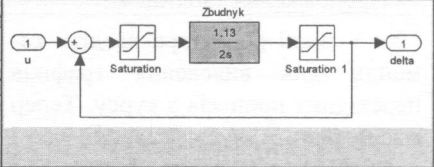
Етап виконання завдання	Команди MATLAB
1. Зробіть свою папку робочою папкою MATLAB.	ЛКМ по кнопці  справа від поля Current Directory
2. Відкрийте модель, побудовану в лабораторній роботі № 3.	Перетягніть файл lab3.mdl в робочу область MATLAB
3. Збережіть модель у своїй папці під іменем lab4.mdl.	 File – Save as ...
4. Виділіть мишею регулятор разом з інтегратором і перетворіть його в підсистему.	 Edit – Create subsystem
5. Дайте підсистемі ім'я <b>Regulator</b> , розташуйте назву зверху і виберіть фоновий колір на свій смак.	Подвійне клацання на імені ПКМ – Format – Flip name ПКМ – Background color
6. Визначте для входу і виходу цього блока імена <b>e</b> і <b>u</b> відповідно.	Подвійне клацання на блоці ЛКМ на імені входу чи виходу
7. Аналогічно побудуйте підсистему <b>Zbudnyk</b> з входом <b>u</b> і виходом <b>delta</b> і підсистему <b>Generator</b> з входами <b>f</b> і <b>delta</b> і виходом <b>phi</b> . Збережіть модель і скопіюйте її через буфер обміну в звіт.	
8. Обведіть мишею (при натиснутій ЛКМ) всі блоки, крім джерел сигналів і осцилографів. Створіть підсистему <b>Лінійна система</b> з входами <b>r</b> (задавальний пристрій) і <b>f</b> (збурення) і виходами <b>phi</b> і <b>delta</b> . Приведіть модель до відповідного вигляду, розтягуючи лінії.	 Edit – Create subsystem

Продовження таблиці 4.1

<p>9. Скопіюйте блок <b>Лінійна система</b> і змініть його ім'я на <b>Нелінійна система</b>. Підключіть до входу нового блока ті ж сигнали (задавальний пристрій і збурення), що і для першого блока. Встановіть для блока лінійної системи жовтий фоновий колір, а для нелінійної – фіолетовий.</p>	<p>Перетягніть з допомогою ПКМ ПКМ – Background color</p>
<p>10. Відкрийте підсистему <b>Zbudnyk</b> в нелінійній системі. Ми побудуємо нелінійну модель приводу, враховуючи обмеження на кут перекладання керма і швидкість його зміни.</p>	<p>Подвійне клацання на блоці</p>
<p>11. Видаліть з'єднувальні лінії.</p>	<p>ЛКМ на елементі, натиснути Delete.</p>
<p>12. Змініть передавальну функцію на <math>R_o(s) = \frac{1}{T_R s}</math>.</p>	<p>Подвійне клацання на блоці • Denominator</p>
<p>13. Відкрийте вікно <i>Library Browser</i>.</p>	<p> View – Library Browser</p>
<p>14. Перетягніть у вікно моделі блок <b>Sum</b> з групи <b>Math Operations</b>. Змініть його так, щоб організувати від'ємний зворотний зв'язок.</p>	<p>Подвійне клацання на блоці  +- в полі List of signs</p>
<p>15. Перетягніть у вікно моделі два блоки <b>Saturation</b> (насичення) з групи <b>Discontinuities</b>. Розташуйте один блок перед інтегратором (обмеження на швидкість зміни напруги), другий – після (обмеження на значення напруги).</p>	
<p>16. Введіть потрібні границі допустимих значень так, щоб швидкість зміни напруги була не більше 10 вольт в секунду, а значення зміни напруги – не більше 30 вольт. У звіті вкажіть всі встановлені значення.</p>	<p>Подвійне клацання на блоці • Upper limit • Lower limit</p>

Продовження таблиці 4.1

17. З'єднайте блоки потрібним способом. Збережіть модель. Скопіюйте схему нелінійної підсистеми **Zbudnyk** у звіт.



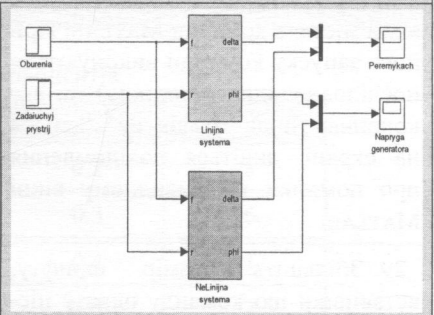
18. Закрийте зайві вікна і перейдіть в головне вікно моделі. Звільніть обидва осцилографів від зв'язків, перетягнувши їх вправо при натиснутій клавіші **Shift**.

19. Перетягніть у вікно моделі два блоки **Mux** (мультиплексор) з групи **Signal Routing**. Ці блоки слугують для об'єднання сигналів в «джгут» (многожильний кабель).

20. З'єднайте входи першого блока з сигналами керування (**delta**) лінійної і нелінійної систем, а вихід – з входом осцилографа **Peremukach**

21. Аналогічно з'єднайте входи другого мультиплексора з сигналами виходу (**phi**) лінійної і нелінійної систем, а вихід – з входом осцилографа **Napruga**.

22. Збережіть модель і скопіюйте її через буфер обміну у звіт.



23. Установіть задане значення напруги 6,3 кВ і збурення 0. Виконайте моделювання і подивіться результати. Жовтий графік показує зміни першого входу осцилографів – (лінійна система), фіолетовий – другого (нелінійна система).

24. Поясніть розбіжність між результатами моделювання лінійної та нелінійної систем. Яка нелінійна ланка суттєво впливає на результат?

25. Створіть новий М-файл (робоча область Matlab).

File – New – M-file

Продовження таблиці 4.1

<p>26. У вікні редактора введіть команди для виведення графіків перехідних процесів з курсу. Тепер масив <b>phi</b> містить 3 стовпці: час і сигнали з двох входів осцилографа (виходи лінійної і нелінійної системи). Весь текст справа від знака % вважається коментарем. Третій параметр команди <b>plot</b> означає колір: 'b' – синій, 'g' – зелений, 'r' – червоний і т. д. (див. довідку щодо цієї команди). Команда <b>hold on</b> означає, що не потрібно стирати старий графік, <b>hold off</b> – потрібно.</p>	<pre>figure(1); % відкрити рис. # subplot(2,1,1); plot(phi(:,1),phi(:,2),'b'); hold on; plot(phi(:,1),phi(:,3),'g'); hold off; legend('Лінійна система', ... 'Нелінійна система')</pre>
<p>27. Збережіть файл під іменем <b>lab4graph.m</b>.</p>	<p>File – Save</p>
<p>28. Запустіть файл (скрипт (файл, який містить команди MATLAB. При його запуску команди виконуються послідовно одна за одною) на виконання. Якщо графік не з'явився на екрані, дивіться повідомлення про помилки в командному вікні MATLAB.</p>	<p>клавіша F5</p>
<p>29. Збільшіть розмір шрифту, вставивши цю команду одразу після виклику <b>subplot</b>. Тут <b>gca</b> означає поточні осі координат (<i>get current axis</i>). Ще раз запустіть скрипт.</p>	<pre>set(gca,'FontSize',16);</pre>
<p>30. Додайте в скрипт назву графіка і осей координат так само, як і в роботі № 3.</p>	<pre>Title ('Встановлення номінальної напруги') xlabel('Час, сек'); ylabel('\phi, напруга');</pre>

Продовження таблиці 4.1

<p>31. Збільшіть товщину ліній. За допомогою функції <code>get</code> ми спочатку отримуємо масив покажчиків (хендлів, <i>handle</i>) на всі об'єкти-лінії. Потім, за допомогою функції <code>set</code>, встановлюємо для кожної лінії властивість <b>LineWidth</b> (товщина лінії), рівне 1,5 пункту. Збережіть файл і запустіть його на виконання.</p>	<pre>h = get(gca, 'Children') set(h(1), 'LineWidth', 1.5) set(h(2), 'LineWidth', 1.5)</pre>
<p>32. Додайте в скрипт команди, за допомогою яких в нижній половині вікна на одному графіку будуються криві зміни сигналу керування в лінійній і нелінійній системах. Не додавайте заголовок для цього графіка (він буде заважати верхньому графіку).</p>	
<p>33. Зробіть так, щоб скрипт працював правильно. Скопіюйте текст скрипту у звіт.</p>	
<p>34. Запустіть скрипт на виконання. Скопіюйте отриманий графік у звіт.</p>	

Таблиця 4.2 – Таблиця коефіцієнтів

Варіант	$T_s$ , сек	$T_R$ , сек	$T_{oc}$ , сек
1	9.8	0.2	5
2	9.9	0.3	6
3	10.2	0.4	1
4	12.6	0.5	2
5	13.8	0.6	3
6	14.4	0.7	4
7	15.6	0.8	5
8	17.8	0.9	6
9	18	1	1
10	18.2	2	6

#### 4.4 Контрольні питання до захисту лабораторної роботи

1. Як побудувати підсистему з декількох існуючих блоків моделі?
2. Скільки входів і виходів може мати підсистема?
3. Як редагувати підсистему?
4. Як змінити назву входів і виходів підсистеми?
5. Як скопіювати існуючий блок чи підсистему?
6. Як видалити блок чи з'єднувальну лінію?
7. Поясніть структуру нелінійної моделі збудника.
8. Чому в нелінійній моделі не можна використовувати спільну передавальну функцію лінійної моделі приводу  $R(s) = \frac{1}{T_R s + 1}$ ?
9. Як звільнити блок від зв'язків з іншими блоками?
10. Поясніть призначення блока **Mux**.
11. Як розібратися, який сигнал надходить на перший вхід осцилографа (через мультиплексор), який – на другий?
12. Що таке скрипт в середовищі MATLAB?
13. Що означає знак % всередині скрипту?
14. Як вводити декілька команд в одному рядку?
15. Як правильно перенести довгу команду на наступний рядок?
16. Що означає третій параметр при виклику функції **plot**?
17. Що означають команди **hold on** і **hold off**?
18. Як запустити скрипт на виконання?
19. Як виконати тільки деякі команди з скрипту?
20. Що означає **gca**?
21. Для чого служать функції **set** і **get**?
22. Як узнати і змінити розмір шрифту на графіку?
23. Навіщо потрібен хендл (*handle*) графічного об'єкта?
24. Як змінити товщину лінії на графіку?
25. Де виводяться повідомлення про помилки в скрипті?

## ЛАБОРАТОРНА РОБОТА № 5 ПРОГРАМУВАННЯ В СЕРЕДОВИЩІ MATLAB

**Мета роботи:** освоєння методів програмування в середовищі Matlab [1].

### Завдання роботи:

- навчитися передавати дані з робочої області MATLAB в модель SIMULINK;
- навчитися складати додаткові функції MATLAB;
- освоїти деякі способи автоматизації обрахунків;

## 5.1 Основні теоретичні відомості

### 5.1.1 Передача даних в модель Simulink

Якщо параметри блоків моделі часто змінюються, зручніше зробити так, щоб їх можна було міняти прямо в командному вікні MATLAB або навіть зі скрипту. Для цього при заданні параметрів блоків потрібно використовувати не числові значення, а імена змінних. При запуску моделювання SIMULINK шукатиме змінні з такими іменами в робочій області MATLAB і підставляти їх значення. Наприклад, в параметрах блока **Transfer Fcn** можна задати

*Numerator:* **n**

*Denominator:* **d**

Тоді для того, щоб ланка відповідала передавальній функції  $\frac{s+2}{2s^2+3s+1}$

у командному вікні MATLAB потрібно задати значення для цих масивів

```
>> n = [1 2]
n =
     1     2
>> d = [2 3 1]
d =
     2     3     1
```

При будь-якій зміні цих масивів в робочій області змінюються і властивості відповідного блока в моделі. Можна використовувати і складніші вирази, наприклад, в полі *Numerator* можна ввести  $K_c \cdot [Ts \ 1 \ 0]$ . Це означає, що чисельник має вигляд  $K_c(T_c s^2 + s)$ , при старті моделювання з робочої області MATLAB будуть завантажені значення двох змінних з іменами  $K_c$  і  $T_c$ .

### 5.1.2 Функції MATLAB

Всі М-файли, з якими працює система MATLAB, діляться на дві категорії: *скрипти* і *функції*. Скрипт – це просто послідовність команд, в якій ви-



користуються змінні з основного робочого простору MATLAB. Функція – це підпрограма, яка приймає аргументи (параметри) і повертає результати. На відміну від функцій в більшості мов програмування, функція MATLAB може повертати декілька результатів (а не один). Функція відрізняється від скрипту тим, що має заголовок, який починається словом **function**. Наприклад, заголовок **function [a,b,c,d] = qq ( x, y, z )** визначає функцію з ім'ям **qq**, яка приймає три параметри (x, y і z) і повертає 4 результати (a, b, z і d). На відміну від більшості сучасних мов, типи змінних (ціла, вагова, символна, масив і так далі) не визначаються заздалегідь, кожна з них містить будь-які допустимі в MATLAB дані.

Функція записується в M-файл (з розширенням **.m**), ім'я якого має збігатися з ім'ям функції. Наприклад, функція **qq** має бути записана у файл **qq.m**. Сучасні версії MATLAB взагалі не звертають увагу на ім'я функції (у заголовку), важливе тільки ім'я файлу.

Функція має свій простір змінних і не може безпосередньо звертатися до змінних основного робочого простору. Усередині функції доступні аргументи, крім того, можна вводити і використовувати нові змінні. Для того, щоб повернути потрібні значення, потрібно записати їх змінні **a, b, z** і **d**. Оператор **return** використовується для дострокового повернення з функції (до цього у всі вихідні змінні мають бути записані потрібні значення).

Функції можуть викликатися за іменем з командного рядка MATLAB, зі скрипту або з іншої функції. Наприклад, для виклику розглянутої вище функції **qq** потрібно набрати команду вигляду **[w,e,r,t] = qq ( x1, 4\*x2, 3 )**.

В даному випадку при роботі функції замість **x** використовується значення змінної **x1**, замість **y** – значення виразу **4\*x2**, а замість **z** – число 3. Результати функції записуються в змінні **w, e, r** і **t**.

При виклику функції кількість вхідних і вихідних змінних може бути менша (але не більша!), ніж в заголовку функції. Усередині функції число вхідних параметрів зберігається в спеціальній змінній **nargin**, а число вихідних величин – в змінній **nargout**.

### 5.1.3 Деякі стандартні функції MATLAB

Система MATLAB містить великий набір стандартних функцій. Більшість цих функцій також оформлена у вигляді M-файлів, їх адресу можна дізнатися за допомогою команди **which** (див. роботу № 1). У цій роботі використовуються три функції:

- **abs** – обчислення модуля числа чи модуля кожного елемента масиву;
- **max** – обчислення максимального значення в масиві (є також функція **min** для обчислення мінімального значення);
- **find** – пошук елементів масиву, що відповідають заданій умові.

Робота з функціями **abs** і **max** достатньо очевидна, тому розглянемо тільки функцію **find**. Вона повертає індекси (номери) елементів масиву, які задовольняють задану умову. Наприклад, якщо **A** – масив, а **b** – число, то команда **>> ind = find ( A > b )** у змінну **ind** записує масив номерів елемен-

тів масиву A, які більші b. В умові можна використовувати знаки  $<$ ,  $>$ ,  $<=$ ,  $>=$  (дорівнює),  $\neq$  (не дорівнює). Складні умови будуються за допомогою операцій  $\sim$  (НЕ),  $\&$  (І) і  $|$  (АБО) так само, як і в інших мовах програмування [5].

## 5.2 Опис системи

У роботі розглядається система регулювання напруги генератора. Її спрощена (а) та структурна (б) схеми АСК показані на рисунку 5.1.

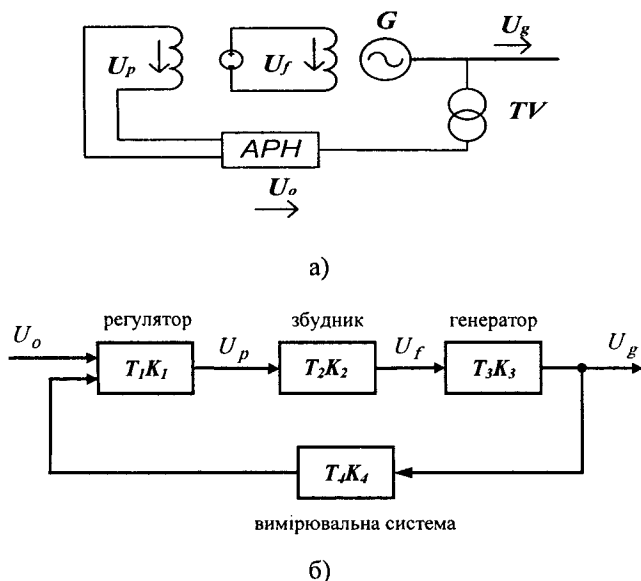


Рисунок 5.1 – Спрощена а) та структурна б) схеми системи регулювання напруги генератора

Система автоматичного регулювання складається з трьох інерційних ланок першого порядку (рис. 5.1, б), які описуються такими диференціальними рівняннями в операторній формі:

– ланка регулятора

$$(T_1 p + 1) U_p = k_1 (U_0 - U_g); \quad (5.1)$$

– ланка збудника

$$(T_2 p + 1) U_f = k_2 U_p; \quad (5.2)$$

– ланка генератора

$$(T_3 p + 1) U_g = k_3 U_f, \quad (5.3)$$

де  $U_g, U_f, U_p, U_0$  – відповідно, напруга на виходах ланок генератора, збудника, регулятора і задавального пристрою;  $T_3, T_2, T_1$  – постійні часу відповідно ланок генератора, збудника і регулятора.

Генератор потужністю  $P_g=25$  МВт, напруга  $U_g=6,3$  кВ,  $T_g=9,9$  с.

Збудник  $P_{зб}=130$  кВт, напруга  $U_f=250$  В, швидкість зміни напруги 330 В/с.

$T_3=T_2=250/330=0,76$  с;  $U_0-U_g^{ycm}=22$  В.

Коефіцієнт трансформації  $n_{TV} = 6300/220$ ; напруга на виході  $U_p=220$  В; коефіцієнт підсилення регулятора  $\kappa_p=220/22=10$ ; постійна часу  $T_p=T_f=0,5$  с;  $\kappa_I = \kappa_p / n_{TV} = (220/6300) \cdot 10 = 0,35$ ;  $\kappa_2 = (250/220) = 1,13$ ;  $\kappa_3 = 6300/250 = 25$ .

Передавальна функція від напруги збудника до напруги генератора записується у вигляді

$$P(s) = \frac{K_3}{T_s s + 1}. \quad (5.4)$$

Збудник приблизно моделюється як інтегрувальна ланка

$$R_0(s) = \frac{K_2}{T_R s + 1}. \quad (5.5)$$

На значення напруги і швидкість її зміни на шини генератора накладаються нелінійні обмеження  $|\dot{\delta}(t)| < 10$  В/сек,  $|\delta(t)| < 30$  В.

Для вимірювання напруги на затискачах генератора використовується вольтметр, математична модель якого записується у вигляді аперіодичної ланки першого порядку з передавальною функцією

$$H(s) = \frac{1}{T_{oc} s + 1}. \quad (5.6)$$

Як пристрій керування використовується ПІД-регулятор з передавальною функцією



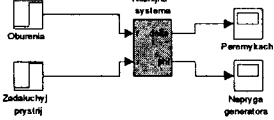

$$C(s) = K_c \left( 1 + \frac{T_s s + 1}{T_v + 1} \right) + \frac{1}{T_I s}, \quad (5.7)$$

де  $T_v = 1$  с і  $T_I = 200$  с.


Числові значення  $T_s, T_R$  і  $T_{oc}$  потрібно взяти з таблиці в кінці файлу.

### 5.3 Послідовність виконання роботи

Таблиця 5.1 – Послідовність виконання роботи [1]

Етап виконання завдання	Команди MATLAB
1. Зробіть свою папку робочою папкою MATLAB.	ЛКМ на кнопці  справа від поля Current Directory
2. Відкрийте модель, побудовану в лабораторній роботі № 4.	Перетягніть файл lab4.mdl в робочу область Matlab
3. Збережіть модель в своїй папці під іменем lab5.mdl.	 File – Save as ...
4. Видаліть модель лінійної системи, обидва мультиплектори та зайві з'єднувальні лінії. Добудуйте схему, як показано на рисунку.	
5. Замініть числові значення постійного збурення і задавального пристрою на символіні: fConst і phiZad відповідно.	 Подвійне клацання на блоці Final value
6. Відкрийте блок Nelinejna systema і замініть всі числові значення на імена змінних у всіх підсистемах. Через deltaMax позначимо максимальну зміну напруги (6,5 кВ), а через ddMax – максимальну швидкість зміни напруги (340 В / сек ).	Генератор, Numerator: K Denominator: [Ts 1 0] ПД-регулятор: Numerator: Kc*[Ts+1 1] І-канал, Denominator: [TI 0] Вольтметр, Denominator: [Toc 1] Збудник, Denominator: [TR 0] Обмеження швидкості зміни напруги: $\pm TR * ddMax$ Обмеження значення напруги: $\pm deltaMax$
7. Вставте у звіт схеми всіх підсистем нелінійної моделі. Після цього закрийте всі вікна підсистем, крім головного вікна моделі.	
8. Перейдіть в команді вікно MATLAB і створіть новий М-файл.	File – New – M-file
9. Введіть числові значення всіх параметрів моделі для свого варіанта. Установіть значення напруги 6,3 кВ. Збережіть файл під іменем sysdata.m.	clear all; clc; K = 0.0694; Ts = 18.2; TR = 2; Toc = 6; ddMax = 340; deltaMax = 6,5; phiZad = 6,3; fConst = 0; TI = 200; Kc = 0.7045;

Продовження таблиці 5.1

<p>10. Виконайте скрипт, після цього запустіть модель. Зробіть так, щоб моделювання було успішно виконано. Якщо цього не відбувається, повідомлення про помилки потрібно шукати в командному вікні MATLAB.</p>	<p>клавиша F5   ЛКМ по кнопці ▶</p>
<p>11. Відкрийте в редакторі скрипт <b>lab4graph.m</b> і збережіть його під іменем <b>lab5graph.m</b>.</p>	<p>Подвійне клацання на lab4graph.m File – Save as...</p>
<p>12. Додайте на початок скрипту рядок, показаний справа. При цьому скрипт стає <i>функцією</i>, приймаючи два масиви – <b>phi</b> і <b>delta</b>. Ця функція нічого не повертає, а тільки буде графік в окремому вікні. Вона має власний простір змінних і не може (без спеціальних прийомів) використовувати змінні робочої області MATLAB.</p>	<p>function lab5graph ( phi, delta )</p>
<p>13. Видаліть з функції зайві рядки, враховуючи, що тепер масиви <b>phi</b> і <b>delta</b> мають тільки 2 стовпці (будуються тільки процеси в нелінійній системі).</p>	
<p>14. Змініть заголовок на «Перехідні процеси при зміні напруги». Збережіть файл і скопіюйте текст функції у звіт.</p>	<p>title ( ... )</p>
<p>15. Створіть новий М-файл, введіть в першому рядку виклик скрипту <b>sysdata</b> (завантаження параметрів моделі). Додайте рядки для запуску моделювання (модель <b>lab5.mdl</b>) і виведення результатів на екран. Збережіть скрипт під іменем <b>lab5go.m</b> і скопіюйте його у звіт.</p>	<p>sysdata;  sim ('lab5')  lab5graph ( phi, delta )</p>

## Продовження таблиці 5.1

16. Тепер побудуємо функцію, яка розраховує перерегулювання і час перехідного процесу. Створіть новий М-файл в редакторі та введіть в нього текст функції (номери рядків вводити не потрібно, вони відображаються автоматично на полях у вікні редактора):

```
1 function [sigma, Tpp] = overshoot ( t, y );  
2 yInf = y(end);  
3 diff = (y - yInf) / abs (yInf);  
4 sigma = max(diff) * 100;  
5 i = find(abs(diff) > 0.02);  
6 Tpp = t(max(i)+1).
```

Коментар:

1 – оголошення функції `overshoot`, яка приймає два параметри-масиви (час  $t$  і перехідний процес  $y$ ) і повертає два значення (перерегулювання у відсотках  $\sigma$  і час перехідного процесу  $T_{pp}$ );

2 – розрахунок *останього* значення масиву  $y$ , яке приймається за установлене;

3 – розрахунок відносно відхилення в кожній точці графіка;

4 – розрахунок перерегулювання у відсотках;

5 – в масив  $i$  записуються *номери* всіх елементів масива  $diff$ , які за модулем більші 0.02 (для визначення часу перехідного процесу використовується відхилення 2%);

6 – розраховується час перехідного процесу як перший елемент масиву  $t$ , після якого всі елементи масиву  $y$  відхиляються від усталеного значення не більше, ніж на 2%.

17. Збережіть файл з іменем `overshoot.m` і скопіюйте його у звіт.

18. Перейдіть у вікно скрипту `lab5go`. Видаліть останні два рядки, залишивши тільки завантаження вихідних даних. Збережіть файл з ім'ям `lab5go1.m`. В останій частині роботи ми досліджуємо вплив параметрів моделі на показники якості перехідних процесів. Спочатку подивимось, як впливає постійна часу генератора  $T_i$  (властивість системи зберігати стійкість і вагові показники якості при малій зміні характеристик об'єкта керування порівняно з розрахунковими значеннями називають *грубістю* чи *робастністю* (*robustness*). Негрубі системи непридатні для використання на практиці).

### Продовження таблиці 5.1

19. Додайте в кінець скрипту текст програми (без номерів рядків)

```
1 Ts0 = Ts;
2 aTs = linspace(0.8, 1.2, 100) * Ts0;
3 aSi = []; aTpp = [];
4 for Ts=aTs
5     sim ('lab5' )
6     [si,Tpp] = overshoot ( phi(:,1), phi(:,2) );
7     aSi = [aSi si];
8     aTpp = [aTpp Tpp];
9 end;
```

Коментар:

- 1 – зберігаємо номінальне значення постійної часу у змінній Ts0;
- 2 – створюється масив зі 100 постійних часу, які змінюються в діапазоні від 80 до 120% від номінального (розрахункового) значення;
- 3 – створюються порожні масиви aSi (для зберігання значень перерегулювання) і aTpp (для зберігання значень часу перехідного процесу);
- 4 – початок циклу, змінна Ts приймає послідовно всі значення з масиву aTs;
- 5 – моделювання при новому значенні Ts;
- 6 – розрахунок перерегулювання і часу перехідного процесу;
- 7 – в кінець масиву aSi додається нове значення;
- 8 – в кінець масиву aTpp додається нове значення;
- 9 – кінець циклу.

20. Щоб регулятор не змінювався при зміні Ts, потрібно в його моделі змінити в чисельнику Ts на Ts0.

ПД-регулятор:  
 Numerator: Kc\*[Ts0+1 I]

21. Запустіть скрипт на виконання. Якщо відкрити вікно осцилографа **Napruga generatora**, можна спостерігати, як змінюється перехідний процес при зміні постійної часу.


клавіша F5

22. Додайте в кінець скрипту рядки для побудови графіка. У верхній частині потрібно побудувати графік зміни перерегулювання, в нижній – зміни часу перехідного процесу. Тут зручно використовувати елементи з файлу **lab5graph.m**, виправивши їх відповідним чином.

23. Збережіть скрипт і запустіть на виконання тільки нові рядки. Щоб не проводити моделювання заново, можна виділити їх в редакторі і натиснути F9. Скопіюйте відлагоджений скрипт у звіт.

виділити рядки, клавіша F9

### Продовження таблиці 5.1

24. Якщо графік зміни часу перехідного процесу має ступінчастий чи стрибкоподібний характер, зменшіть максимально допустимий крок інтегрування і повторіть моделювання (врахуйте, що час виконання скрипту збільшиться).	 Simulation – Simulation parameters – Max step size = 0.2
25. Скопіюйте отриманий графік у звіт.	
26. Збережіть скрипт з іменем <b>lab5go2.m</b> .	

Таблиця 5.2 –Таблиця коефіцієнтів

Варіант	$T_r$ , с	$T_R$ , с	$T_{oc}$ , с
1.	9.8	0.2	5
2.	9.9	0.3	6
3.	10.2	0.4	1
4.	12.6	0.5	2
5.	13.8	0.6	3
6.	14.4	0.7	4
7.	15.6	0.8	5
8.	17.8	0.9	6
9.	18	1	1
10.	18.2	2	6

### 5.4 Контрольні питання до захисту лабораторної роботи

1. В параметрах блока моделі вказано ім'я змінної. Як задати її значення?
2. Як ввести обмеження на швидкість зміни напруги, якщо відома постійна часу збудника?
3. Що таке М-файл?
4. Як створити новий М-файл?
5. Де виводяться повідомлення про помилки при виконанні скрипту чи функції?
6. Як будується заголовок функції в М-файлі?
7. Може функція повертати декілька величин?
8. Чи можна звертатися до змінних робочої області MATLAB всередині функції?



9. Як викликається функція, записана в М-файл?
10. Як виділити останній елемент масиву?
11. Чому дорівнює результат операції  $A \cdot x$ , де  $A$  – масив, а  $x$  – число?
12. Як працює функція `find`?
13. Як змінити функцію `overshoot`, щоб вона визначала час перехідного процесу з точністю 5%?
14. Що таке грубість (робастність) системи?
15. Що означає запис
  - `x = []`;
  - `x = [x y]`;
  - `phi(:,1)`
  - `phi(1,:)`?
16. Як виконати тільки декілька рядків із скрипту?

## ЛАБОРАТОРНА РОБОТА № 6 ОПТИМІЗАЦІЯ НЕЛІНІЙНИХ СИСТЕМ

**Мета роботи:** освоєння методів оптимізації нелінійних систем в середовищі Matlab [1].

### Завдання роботи:

- навчитися копіювати підсистеми з однієї моделі в іншу;
- вивчити прийоми, що дозволяють частково компенсувати нелінійність типу «насичення» в системі з ПІД-регулятором;
- навчитися використовувати пакет **Simulink Response Optimization**.

## 6.1 Основні теоретичні відомості

### 6.1.1 Ефект насичення

У реальних системах завжди є обмеження на максимальну величину керувальної дії. У генераторних системах управління це, наприклад, гранична швидкість зміни напруги, граничне значення відхилення напруги на шинах генератора. При малих відхиленнях напруги такими нелінійними обмеженнями можна знехтувати, проте при великих величинах сигналів вони істотно змінюють властивості системи.

Нелінійності такого типу називаються «насиченням»:

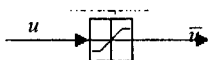


Рисунок 6.1 – Насичення

Вони описуються рівнянням

$$\bar{u} = \begin{cases} u_{\min}, & \text{при } u < u_{\min} \\ u, & \text{при } u_{\min} \leq u \leq u_{\max}, \\ u_{\max}, & \text{при } u > u_{\max} \end{cases}, \quad (6.1)$$

де  $u$  – сигнал на вході ланки,  $\bar{u}$  – сигнал на виході (з урахуванням насичення),  $u_{\min}$  і  $u_{\max}$  – допустимі границі.

Для компенсації постійних збурень в регулятор часто вводиться інтегровальна ланка. При цьому в системах з насиченням спостерігається ефект «залипання» інтегратора. Він полягає в тому, що управлінський сигнал вже досяг граничного значення, а інтегратор продовжує інтегрувати («намотувати», *windup*) помилку, хоча збільшувати управління вже не можна. Коли

помилка змінить знак, потрібно буде переключити кермо в інший бік, але цього не відбудеться, оскільки вихід інтегратора дуже великий. В результаті збільшується перерегулювання і час перехідного процесу. На практиці така поведінка системи може виявитися неприпустимою.

### 6.1.2 Компенсація ефекту насичення (*anti-windup*)

Для того, щоб запобігти «намотуванню» інтегратора, використовуються спеціальні прийоми нелінійної корекції. Вони зводяться до одного з двох варіантів:

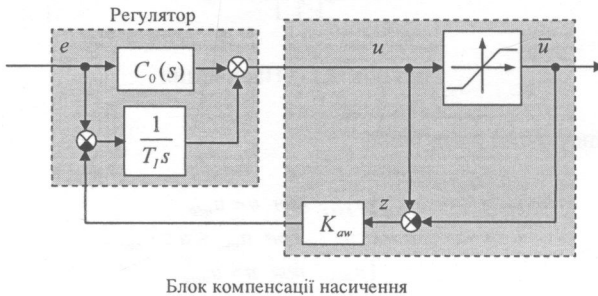
1. *Умовна інтеграція*: якщо сигнал керування досягає граничного значення, інтегратор відключається, інтеграція зупиняється.

2. *Техніка anti-windup*: з входу інтегратора віднімається сигнал, який надходить з блока компенсації насичення. Саме цей варіант ми досліджуватимемо.

Хай інтегратор під'єднується паралельно решті частини регулятора, тобто, має вигляд

$$C(s) = C_0(s) + \frac{1}{T_I s}, \quad (6.2)$$

де  $C_0(s)$  – частина регулятора, яка не містить інтегровувальних ланок, а  $T_I$  – постійна часу інтегровувальної ланки (рис. 6.2).



Блок компенсації насичення

Рисунок 6.2 – Блок-схема системи з паралельно ввімкненим інтегратором

Блок компенсації насичення генерує сигнал

$$z = u - \bar{u}, \quad (6.3)$$

де  $u$  – сигнал на виході регулятора, а  $\bar{u}$  – сигнал з урахуванням насичення (6.1), причому границі  $u_{\min}$  і  $u_{\max}$  вибираються такі ж, як і допустимі границі для вихідного сигналу збудника. Якщо сигнал  $u$  знаходиться в допустимих межах (в інтервалі  $[u_{\min}, u_{\max}]$ ), то  $z = 0$  і система працює в лінійному режимі, без насичення. Якщо ж сигнал  $u$  виходить з діапазону  $[u_{\min}, u_{\max}]$ , сигнал  $z$ , який не дорівнює нулю, подається через підсилювач на вхід інтегратора зі зворотним знаком, протидіючи «намотуванню» інтегратора і збільшенню його вихідного сигналу.

Зауважимо, що замість підсилювача з коефіцієнтом  $K_{ав}$  можна встановлювати динамічну ланку, однак простішого варіанта часто буває достатньо.

Аналогічна ідея може бути використана і в тому випадку, коли інтегратор під'єднується послідовно,

$$C(s) = \frac{1}{s} C_0(s), \quad (6.4)$$

де  $C_0(s)$  – частина регулятора, яка не містить інтегровальних ланок. Відповідна блок-схема системи показана на рис. 6.3.

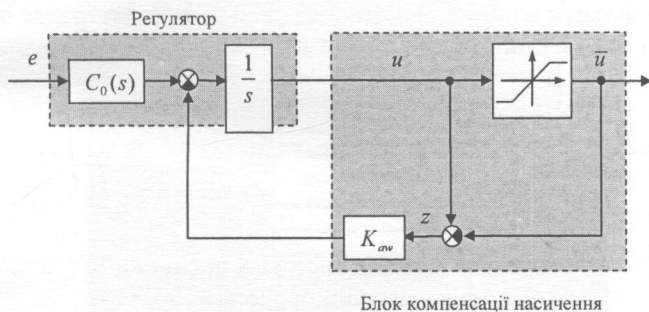


Рисунок 6.3 – Блок-схема системи з послідовно під'єднаним інтегратором

Для того, щоб вибрати коефіцієнт  $K_{ав}$ , можна використовувати різні методи теорії нелінійних систем або математичне моделювання. З інженерної точки зору простіше другий спосіб, який дозволяє в інтерактивному режимі методом проб і помилок вибрати відповідне значення  $K_{ав}$ . У середовищі MATLAB пошук можна автоматизувати за допомогою пакета *Simulink Response Optimization*. [4].

### 6.1.3 Пакет Simulink Response Optimization

Пакет *Simulink Response Optimization* призначений для налаштування параметрів нелінійної моделі методом чисельної оптимізації за перехідним процесом.

За допомогою ламаних ліній задається ділянка, з якої не має виходити перехідний процес. Інтервал, на якому виконується моделювання, розбивається на невеликі ділянки шириною  $h$ . Для цих точок будується система нерівностей, яким має задовольняти функція, що описує перехідний процес. На рисунку 6.4 відрізками червоного кольору показано, де ці нерівності порушені. Потрібно вибрати параметри моделі так, щоб порушень було якомога менше і величини відхилень були мінімальні. У ідеалі весь перехідний процес вписується в допустиму область, порушень взагалі немає.

Для вирішення цього завдання в пакеті *Simulink Response Optimization* використовуються процедури нелінійної оптимізації з обмеженнями з пакета *Optimization Toolbox*.

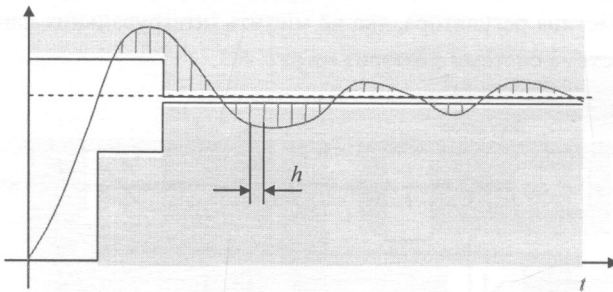


Рисунок 6.4 – Метод чисельної оптимізації за перехідним процесом

Спочатку треба перетягнути в модель SIMULINK блок **Signal Constraint** з групи *Simulink Response Optimization* і подати на його вхід сигнал, який треба «вписати» в задану область. За замовчуванням межі області встановлюються так, щоб стало значення сигналу дорівнювало одиниці. Якщо це не так, на вході блока **Signal Constraint** можна поставити додатковий підсилювач (блок **Gain**), який змінить масштаб. Наприклад, якщо стало значення дорівнює 10, коефіцієнт підсилення має дорівнювати 0.1, щоб стало значення на вході блока **Signal Constraint** дорівнювало 1.

Подвійне клацання по блоку **Signal Constraint** відкриває робоче вікно для підбору параметра (рис. 6.5).

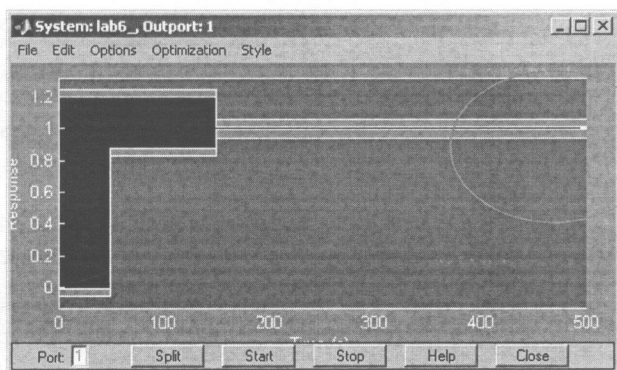


Рисунок 6.5 – Робоче вікно для підбору параметра

Перетягуючи червоні смужки вгору і вниз, можна змінювати межі допустимої області (вона залита чорним кольором). Можна також перетягувати вліво і управо вертикальні межі. Кладання ПКМ по червоній смужці дозволяє задати параметри обмеження точніше в діалоговому вікні.

Для того, щоб розбити смужку на дві (зробити точнішу межу) потрібно виділити її клацанням миші і клацнути по кнопці **Split**.

Щоб задати параметри пошуку, потрібно вибрати в цьому вікні пункт верхнього меню **Optimization – Optimization Parameters** (рис. 6.6)

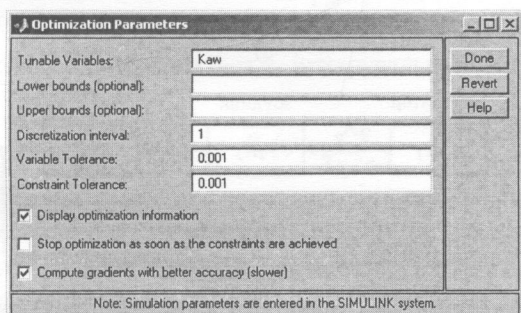


Рисунок 6.6 – Задання параметрів пошуку

У полі **Tunable variables** вводяться через пропуск імена змінних, значення яких потрібно підібрати. Поля **Lower bounds** (нижні межі значень змінних) і **Upper bounds** (верхні межі) необов'язкові для заповнення.

У полі **Discretization interval** потрібно ввести величину кроку  $h$  (див. рисунок вище). Від кроку залежить кількість інтервалів і кількість обмежень. Чим менший крок, тим більше задасться обмежень і повільніше пра-

цює процедура пошуку. З іншого боку, при дуже великому кроці знижується точність. Рекомендується вибрати цей параметр таким, що дорівнює 1–2% від загального часу моделювання.

Перед запуском процедури оптимізації потрібно ввести перше наближення для невідомих параметрів в командному вікні MATLAB:

**Kaw = 0.2;**

Після цього слід клацнути по кнопці **Start** у вікні блока **Signal Constraint**. Інформацію про хід процесу і повідомлення про помилки можна спостерігати в командному вікні Matlab. Зазвичай для того, щоб домогтися якісних перехідних процесів, доводиться кілька разів запускати процедуру оптимізації, змінюючи обмеження і послідовно покращуючи результат [5].

## 6.2 Опис системи

У роботі розглядається система регулювання напруги генератора. Її спрощена а) та структурна б) схеми АСК показані на рисунку 6.7.

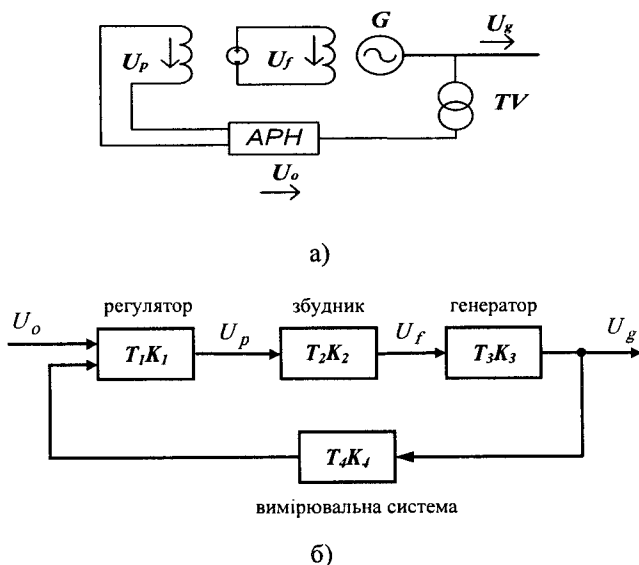


Рисунок 6.7 – Спрощена а) та структурна б) схеми системи регулювання напруги генератора

Система автоматичного регулювання складається з трьох інерційних ланок першого порядку (див. рис. 6.7, б), які описуються нижченаведеними диференціальними рівняннями в операторній формі:

– ланка регулятора  $(T_1 p + 1)U_p = k_1(U_0 - U_g);$  (6.5)

– ланка збудника  $(T_2 p + 1)U_f = k_2 U_p;$  (6.6)

– ланка генератора  $(T_3 p + 1)U_g = k_3 U_f,$  (6.7)

де  $U_g, U_f, U_p, U_0$  – відповідно, напруга на виходах ланок генератора, збудника, регулятора і задавального пристрою;  $T_3, T_2, T_1$  – постійні часу, відповідно, ланок генератора, збудника і регулятора.

Генератор потужністю  $P_g = 25$  МВт, напруга  $U_g = 6,3$  кВ,  $T_g = 9,9$  с.

Збудник  $P_{зб} = 130$  кВт, напруга  $U_f = 250$  В, швидкість зміни напруги 330 В/с.

$$T_3 = T_2 = 250/330 = 0,76 \text{ с}; U_0 - U_g^{ном} = 22 \text{ В}.$$

Коефіцієнт трансформації  $n_{TV} = 6300/220$ ; напруга на виході  $U_p = 220$  В; коефіцієнт підсилення регулятора  $\kappa_p = 220/22=10$ ; постійна часу  $T_p = T_1 = 0,5$  с;  $\kappa_1 = \kappa_p / n_{TV} = (220/6300) \cdot 10 = 0,35$ ;  $\kappa_2 = (250/220)=1,13$ ;  $\kappa_3 = 6300/250 = 25$ .

Передавальна функція від напруги збудника до напруги генератора запишеться у вигляді

$$P(s) = \frac{K_3}{T_3 s + 1}. \quad (6.8)$$

Збудник приблизно моделюється як інтегральна ланка

$$R_0(s) = \frac{K_2}{T_R s + 1}. \quad (6.9)$$

На значення напруги і на швидкість її зміни на шинах генератора накладаються нелінійні обмеження  $|\dot{\delta}(t)| < 10 \text{ В/сек}$ ,  $|\delta(t)| < 30 \text{ В}$ .

Для вимірювання напруги на затискачах генератора використовується вольтметр, математична модель якого записується у вигляді аперіодичної ланки першого порядку з передавальною функцією

$$H(s) = \frac{1}{T_{oc} s + 1}. \quad (6.10)$$

Як пристрій керування використовується ПІД-регулятор з передавальною функцією



$$C(s) = K_c \left( 1 + \frac{T_s s + 1}{T_v + 1} \right) + \frac{1}{T_i s}, \quad (6.11)$$

де  $T_v = 1$  с і  $T_i = 200$  с.

Числові значення  $T_s$ ,  $T_R$  і  $T_{oc}$  потрібно взяти з таблиці в кінці файлу.

Для компенсації ефекту насичення, викликаного обмеженою кількістю відпайок перемикача, використовується схема з внутрішнім нелінійним зворотним зв'язком, що охоплює інтегратор у складі регулятора (рис. 6.8).

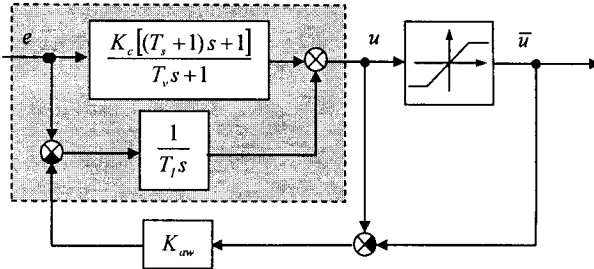


Рисунок 6.8 – Схема з внутрішнім нелінійним зворотним зв'язком, що охоплює інтегратор у складі регулятора



Базовий регулятор, побудований за лінійною моделлю, виділений сірим фоном. Сигнал  $u$  на його виході є бажана відпайка перемикача. Для блока нелінійної корекції типу «насичення» вибираються межі, що дорівнюють обмеженням на відпайки перемикача.

Якщо насичення немає, різниця сигналів  $u - \bar{u}$  дорівнює нулю, і зворотний зв'язок не працює; використовується закон управління, синтезований для лінійної системи. Якщо сигнал  $u$  перевищує допустимі межі, різниця  $u - \bar{u}$  подається (зі знаком «мінус») на вхід інтегратора через підсилювач. Таким чином, при насиченні сигнал на вході інтегратора ослабляється тим сильніше, чим більше різниця між бажаним і допустимим кутами переключення. Такий метод корекції отримав в літературі назву *anti-windup* (протидія «намотуванню»).

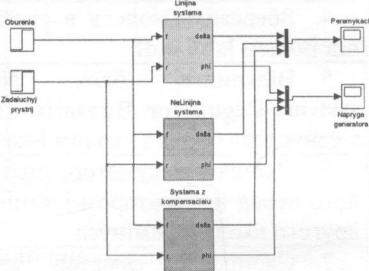
В ході виконання роботи для вибору оптимального значення коефіцієнта  $K_{ав}$  застосовуються процедури чисельної оптимізації пакета **Simulink Response Optimization**.

### 6.3 Послідовність виконання роботи

Таблиця 6.1 – Послідовність виконання роботи [1]

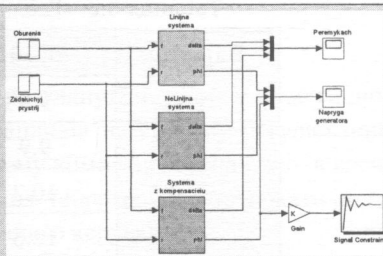
Етап виконання завдання	Команди MATLAB
1. Зробіть свою папку робочою папкою MATLAB.	ЛКМ по кнопці  справа від поля Current Directory
2. Створіть новий М-файл, в першому рядку наберіть ім'я скрипту <b>sysdata</b> для завантаження вихідних даних. Збережіть файл з ім'ям <b>lab6go.m</b> і виконайте його.	File – New – M-file File – Save клавіша F5
3. Відкрийте модель, побудовану в лабораторній роботі № 5.	Перетягніть файл lab5.mdl в робочу область MATLAB
4. Збережіть модель в своїй папці під іменем <b>lab6.mdl</b> .	 File – Save as ...
5. Відкрийте блок <b>Nelinijna systema-Regulator</b> . Видаліть лінію, що з'єднує вхід блока з входом І-канала.	виділити ЛКМ, натиснути Delete
6. Скопіюйте суматор, розташуйте його перед інтегратором і змініть знак другого входу на «мінус».	перетягнути ПКМ  +- в полі List of signs
7. Скопіюйте суматор ще раз, розверніть його на 180 градусів і змініть розташування входів.	перетягніть ПКМ, натисніть Ctrl+I+  в полі List of signs
8. Відкрийте блок <b>Nelinijna systema-Zbudnyk</b> і перетягніть у вікно <b>Regulator</b> другий нелінійний блок (обмеження на кількість відпайок перемикача). Розверніть його на 180 градусів.	перетягніть ЛКМ натисніть Ctrl+I
9. Додайте блок <b>Gain</b> із групи <b>Math Operations</b> і встановіть коефіцієнт підсилення, що дорівнює <b>Kaw</b> .	Kaw в полі Gain
10. Зберіть схему, показану на рисунку. Скопіюйте її у звіт.	
11. Додайте в скрипт <b>lab6go</b> команду, яка встановлює початкове значення для <b>Kaw</b> . Збережіть скрипт і виконайте його.	Kaw = 1;

Продовження таблиці 6.1

<p>12. Далі ми порівнюємо перехідні процеси в цій системі з результатами моделювання лінійної і вихідної нелінійної систем. Закрийте вікна всіх підсистем, крім головного вікна моделі. Дайте блоку назву <b>Systema z kompensacieu</b> і поміняйте колір фону на світло-голубий (цим кольором будуть відображатись на осцилографах відповідні графіки).</p>	
<p>13. Скопіюйте в нову модель підсистему <b>Linijna systema</b> з моделі <b>lab4.mdl</b> і підсистему <b>Nelinijna systema</b> з моделі <b>lab5.mdl</b>.</p>	<p>відкрити потрібну модель, перетягування ЛКМ</p>
<p>14. Додайте в модель мультиплексор (блок <b>Mux</b> з групи <b>Signal Routing</b>) і встановіть для нього кількість входів 3.</p>	<p>подвійне клацання ЛКМ 3 в полі Number of inputs</p>
<p>15. Створіть копію мультиплексора і зберіть схему за рисунком так, щоб на кожному з осцилографів відображались по 3 сигнали з різних моделей. Входи <b>f</b> всіх систем потрібно з'єднати з виходом блока <b>Oburenia</b>, а входи <b>r</b> – з виходом блока <b>Zadaiuchyj prystrij</b>.</p>	
<p>16. Перейдіть у вікно скрипту <b>lab6go</b> в редакторі. Додайте рядки для запуску процедури моделювання і для виведення графіка, враховуючи, що на кожній половині рисунка потрібно виводити по 3 графіки, масиви <b>phi</b> і <b>delta</b> містять по 4 стовпці. Зручно скопіювати і трішки змінити команди для оформлення графіка з файлу <b>lab4graph.m</b>.</p>	<p>sim ('lab6' ) figure(1)</p>
<p>17. Запустіть скрипт. Після відкладки збережіть скрипт на диску і скопіюйте його у звіт. Скопіюйте у звіт побудований графік. Опишіть отриманий ефект.</p>	
<p>18. Додайте в основне вікно моделі підсилювач (блок <b>Gain</b> з групи <b>Math Operations</b>) і встановіть коефіцієнт підсилення, що дорівнює <b>1/phiZad</b>. Цей (нормувальний) блок буде слугувати для того, щоб на виході усталене значення дорівнювало 1 (нормування).</p>	<p>View – Library Browser перетягування ЛКМ подвійне клацання по блоку 1/phiZad в полі Gain</p>

Продовження таблиці 6.1

19. Додайте в модель блок **Signal Constraint** з пакета **Simulink Response Optimization** і з'єднайте його через нормувальний підсилювач (див. вище) з виходом  $\phi_i$  системи з компенсацією насичення. Збережіть схему в звіті.

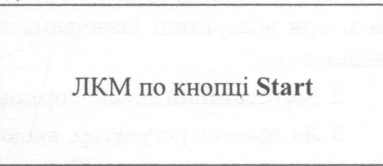


20. Відкрийте блок **Signal Constraint**. Мета оптимізації – вибрати деякі коефіцієнти системи так, щоб перехідний процес залишався всередині чорної зони графіка. Горизонтальні і вертикальні границі обмежень (червоні смуги) можна зсовувати мишкою. Смуга виділяється натисканням ЛКМ, при цьому вона стає білою. Кнопка **Split** дозволяє розбити виділену смугу на дві незалежні частини. Натиснувши ПКМ на червоній смугі, можна редагувати обмеження в числовому вигляді.

21. Зсуньте верхнє обмеження так, щоб перерегулювання складало приблизно 10%. Праву межу першого блока зсуньте приблизно на час 60–65 с.

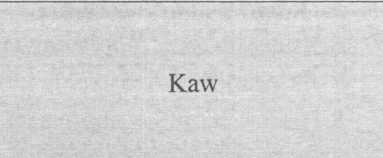
22. Виберіть пункт верхнього меню **Optimization – Parameters** і введіть **Kaw** в поле **Tunable variables** (налаштовувальні параметри). В полі **Lower bound** (нижня межа) ведіть 0 (нуль).

23. Запустіть процедуру оптимізації. Вона може займати декілька хвилин. Ознакою закінчення буде поява тексту *Optimization Converged Successfully* в командному вікні MATLAB.

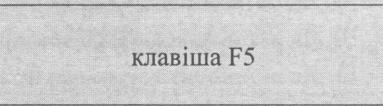


24. Спробуйте покращити перехідний процес, зсовуючи обмеження і знову запускаючи процедуру оптимізації.

25. Визначте знайдене оптимальне значення **Kaw**. Для цього потрібно ввести **Kaw** в командному вікні MATLAB. Запишіть його в звіт і перенесіть в скрипт **lab6go**.



26. Запустіть скрипт **lab6go** на виконання і скопіюйте отриманий графік у звіт.



27. Визначте перерегулювання і час перехідного процесу, використовуючи функцію **overshoot**.

```
[si, Tpp] = overshoot(phi(:,1),...
                    phi(:,4));
```

Таблиця 6.2 –Таблиця коефіцієнтів

Варіант	$T_s$ , с	$T_R$ , с	$T_{oc}$ , с
1.	9.8	0.2	5
2.	9.9	0.3	6
3.	10.2	0.4	1
4.	12.6	0.5	2
5.	13.8	0.6	3
6.	14.4	0.7	4
7.	15.6	0.8	5
8.	17.8	0.9	6
9.	18	1	1
10.	18.2	2	6

#### 6.4 Контрольні питання до захисту лабораторної роботи

1. Нехай лінеаризована система стійка. Чи може система стати нестійкою при врахуванні насичення збудника? На яких режимах це особливо небезпечно?

2. Застосований спосіб корекції є лінійним чи нелінійним? Чому?

3. Як працює регулятор, якщо сигнали не виходять за допустимі межі? Чому (доведіть за схемою)?

4. Що таке *anti-windup*?

5. Поясніть дію блока **Gain**.

6. Як скопіювати блок з однієї моделі в іншу?

7. Як змінити число входів мультиплексора?

8. Навіщо потрібен нормувальний підсилювач?

9. Як потрібно під'єднувати блок **Signal Constraint**?

10. Як встановлювати обмеження на перехідну функцію?

11. Як встановити межі для пошуку рішення в блоці **Signal Constraint**?

## ВИСНОВКИ

В даному лабораторному практикумі розглянуті питання математичного моделювання основних керувальних впливів на систему регулювання напруги на шинах генератора в програмному комплексі Matlab, що, в свою чергу, забезпечує більш глибоке засвоєння теоретичного та практичного матеріалу з дисципліни «Теорія автоматичного керування».

Ознайомлення з можливостями і проведення математичного моделювання в програмному комплексі Matlab дає студентам практичні навички для застосування цього програмного комплексу для подальших науково-дослідних робіт.

## СПИСОК ЛИТЕРАТУРИ

1. Поляков К. Ю. Исследование систем автоматического управления в Matlab [Электронный ресурс] Режим доступа : <http://kpolyakov.spb.ru/uni/labs.htm> – Назва з екрана.
2. Гудвин Г. К. Проектирование систем управления / Гудвин Г. К., Гребне С. Ф., Сальгадо М. Э. – М. : Бином, Лаборатория базовых знаний, 2004. – 342 с.
3. Дорф Р. Современные системы управления / Дорф Р., Бишоп Р. – М. : Бином, Лаборатория базовых знаний, 2004. – 412 с.
4. Дьяконов В. П. MATLAB. Обработка сигналов и изображений. Специальный справочник / В. П. Дьяконов, И. В. Абраменкова. – СПб. : «Питер», 2002. – 608 с.
5. Дьяконов В. П. Справочник по применению системы РС MATLAB. / Дьяконов В. П. – М. : «Физматлит», 1993. – 112 с.

## ДОДАТОК А

### Форма титульного листа звіту до лабораторної роботи

Міністерство освіти і науки України  
Вінницький національний технічний університет  
Факультет електроенергетики і електромеханіки  
Кафедра електричних станцій та систем

Звіт до лабораторної роботи № ...  
«назва»  
з дисципліни «Теорія автоматичного керування»

Виконали: ст. гр. ІЕ-196  
Олійнічук К. Б.  
Богачук І. Ю.

Перевірив: к.т.н., ст. викл.  
Макітрак О. О.

Вінниця 2019 р.



**ДОДАТОК Б**  
**Приклади оформлення звіту**

**ЛАБОРАТОРНА РОБОТА № 1**  
**ДОСЛІДЖЕННЯ РОЗІМКНЕНОЇ ЛІНІЙНОЇ СИСТЕМИ**

Варіант № 10

**1. Опис системи**

Досліджується система, що описується математичною моделлю у вигляді передавальної функції  $F(s) = \frac{2.9s^2 + 3.19s + 0.87}{s^3 + 0.7s^2 + 0.35s + 0.05}$ .

**2. Результати дослідження**

- адреса файлу tf.m:  
E:\MAT\LAB\toolbox\control\control\@tf\tf.m
- нулі передавальної функції  
-0.6000  
-0.5000
- полюси передавальної функції  
-0.2500 + 0.4330i  
-0.2500 - 0.4330i  
-0.2000
- коефіцієнт підсилення ланки в усталому режимі  
k = 17.4000
- смуга пропускання системи  
b = 0.4808 рад/сек
- модель системи в просторі станів

$$a = \begin{bmatrix} -0.7000 & -0.1750 & -0.0500 \\ 2.0000 & 0 & 0 \\ 0 & 0.5000 & 0 \end{bmatrix}$$

$$b = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$$
$$c = \begin{bmatrix} 1.4500 & 0.7975 & 0.4350 \end{bmatrix}$$
$$d = 0$$

- статичний коефіцієнт підсилення після зміни матриці  $D$   
k1 = 18.4000
- модель в формі «нулі-полюси»  
2.9 (s+0.6) (s+0.5)

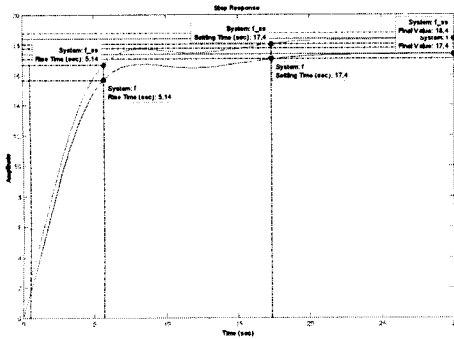
-----

$$(s+0.2) (s^2 + 0.5s + 0.25)$$

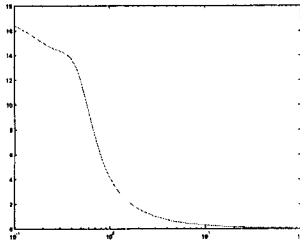
- коефіцієнти демпфування і частоти зрізу

Полос передавальної функції	Власна частота, рад/с	Постійна часу, с	Коефіцієнт демпфування
-0.2000	0.2000	5	1.0000
-0.2500 + 0.4330i	0.5000	2	0.5000
-0.2500 - 0.4330i	0.5000	2	0.5000

- Перехідні процеси початкової і модифікованої систем



- амплітудно-частотна характеристика



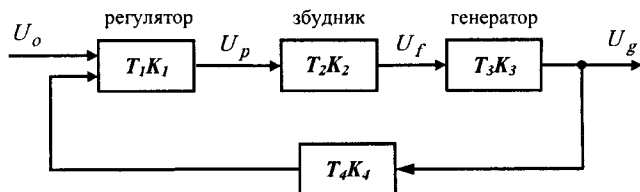
- реакція на сигнал, що складається з прямокутних імпульсів

# ЛАБОРАТОРНА РОБОТА № 2 ПРОЕКТУВАННЯ РЕГУЛЯТОРА ДЛЯ ЛІНІЙНОЇ СИСТЕМИ

Варіант № 10

## 1. Опис системи

Досліджується система регулювання напруги генератора. Її структурна схема АСК показана на рисунку



вимірювальна система

Значення напруги генератора описується лінійною математичною моделлю у вигляді передавальної функції

$$P(s) = \frac{K_3}{(T_s s + 1)}, \quad \text{де } K_3 = 25 \text{ рад/с}, \quad T_s = 18.2 \text{ с.}$$

Збудник моделюється як інтегрувальна ланка

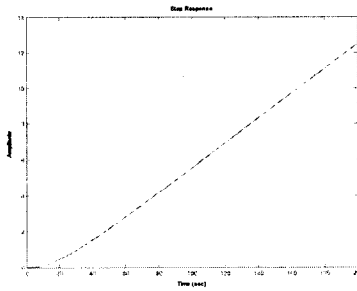
$$R_0(s) = \frac{K_2}{T_R s + 1}, \quad T_R = 2 \text{ с.}$$

Моделлю вимірювального пристрою є аперіодична ланка з передавальною функцією

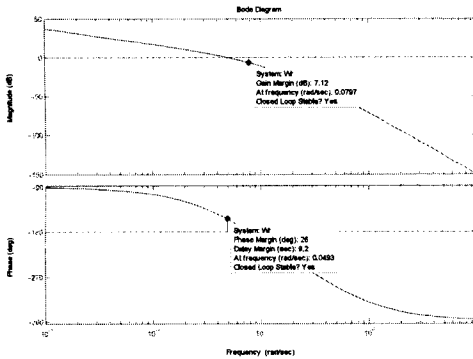
$$H(s) = \frac{1}{T_{oc} s + 1}, \quad T_{oc} = 6 \text{ с.}$$

## 2. Дослідження різмікненої системи

- Передавальна функція плавного перемикача  $R(s) = \frac{1,13}{2s+1}$ .
- Передавальна функція послідовного з'єднання генератора з збудником  $G(s) = \frac{25}{36,4s^3 + 20,2s^2 + s}$ .
- Перехідна характеристика цієї моделі:

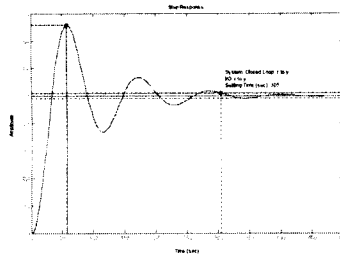


- ЛАФЧХ різьмкненої системи

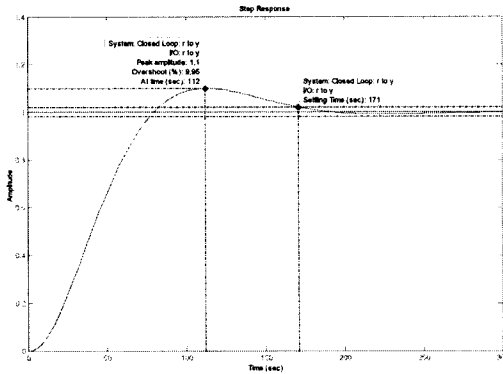


- Система з регулятором  $C(s)=...$  стійка, запаси стійкості: за амплітудою – 7,12 дБ, за фазою – 26 градусів.

### 3. Дослідження системи з пропорційним (П-) регулятором



- Час перехідного процесу  $T_{nn}=305$  с, перерегулювання  $\sigma = 51\%$ .
- Для забезпечення перерегулювання не більше 10% потрібно зменшити коефіцієнт підсилення регулятора до значення  $C(s)=0.348$
- Перехідна функція скоректованої замкнутої системи при  $C(s)=0.348$



- Час перехідного процесу  $T_{mn} = 171$  с,
- Запаси стійкості: за амплітудою 16,3 дБ, за фазою 57.9 градусів.

#### 4. Дослідження системи з пропорційно-диференціальним (ПД-) регулятором

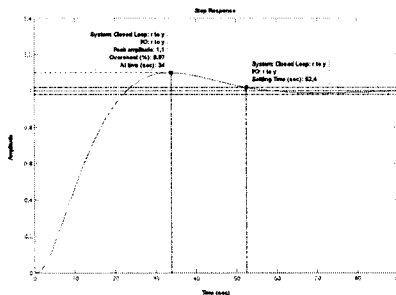
- Загальний вигляд передавальної функції регулятора

$$C(s) = K_c \left( 1 + \frac{T_d s}{T_v + 1} \right), \quad \text{де } T_d = 18.2 \text{ с}, T_v = 1 \text{ с},$$

а коефіцієнт  $K_c$  має бути вибраний в процесі проектування відповідно до вимог системи.

##### 4.1 Регулятор, що забезпечує перерегулювання 10%

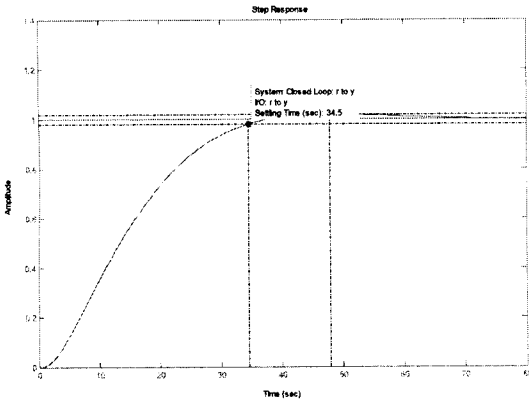
- Для забезпечення перерегулювання 10% потрібно вибрати  $K_c = 0.942$ .
- Перехідна функція скоректованої замкнутої системи



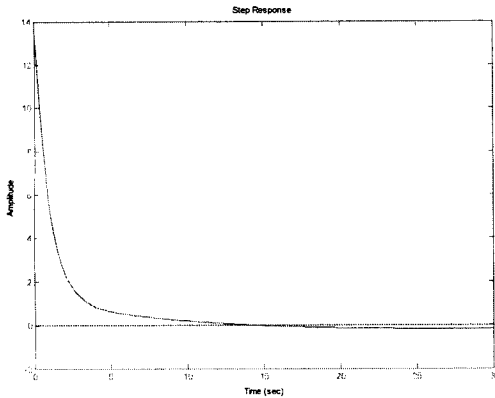
- Час перехідного процесу  $T_{mn} = 52$  с.
- Запаси стійкості: за амплітудою – 15.9 дБ, за фазою – 60.2 градуса.
- Порівняно з П-регулятором, використання ПД-регулятора дозволяє...

#### 4.2 Регулятор, який забезпечує найкоротший перехідний процес

- Для забезпечення мінімального часу перехідного процесу потрібно вибрати  $K_c = 0.704$ .
- Перехідна функція скоректованої замкнутої системи



- Час перехідного процесу  $T_{nn} = 34$  с.
- Зміна сигналу управління при одиничному ступінчастому вхідному сигналі

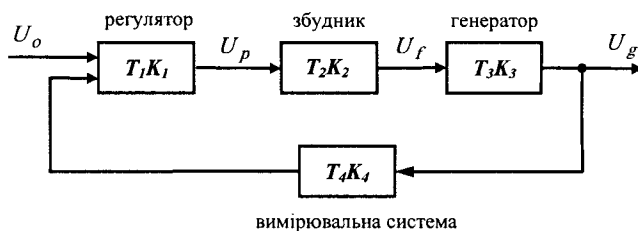


# ЛАБОРАТОРНА РОБОТА № 3 МОДЕЛЮВАННЯ СИСТЕМ КЕРУВАННЯ В ПАКЕТІ SIMULINK

Варіант № 10

## 1. Опис системи

Досліджується система регулювання напруги генератора. Її структурна схема АСК показана на рисунку



Значення напруги генератора описується лінійною математичною моделлю у вигляді передавальної функції

$$P(s) = \frac{K_3}{(T_s s + 1)},$$

де  $K_3 = 25$  рад/с,  $T_s = 18.2$  с.

Збудник моделюється як інтегральна ланка

$$R_0(s) = \frac{K_2}{T_R s + 1}, \quad T_R = 2 \text{ с.}$$

Моделлю вимірювального пристрою є аперіодична ланка з передавальною функцією

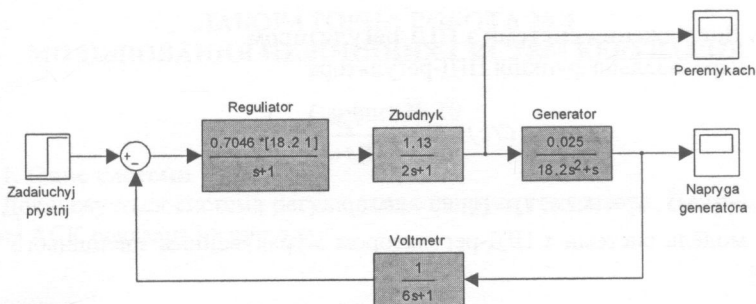
$$H(s) = \frac{1}{T_{oc} s + 1}, \quad T_{oc} = 6 \text{ с.}$$

## 2. Дослідження системи з ПД-регулятором

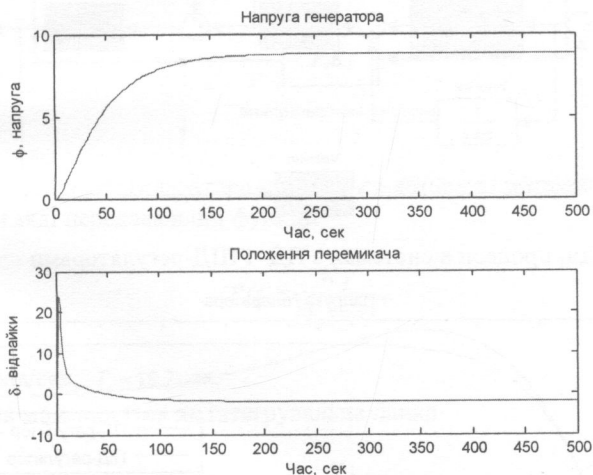
• передавальна функція ПД-регулятора, що забезпечує перехідний процес мінімальної тривалості

$$C(s) = K_c \left( 1 + \frac{T_s s}{T_v s + 1} \right), \quad \text{де } K_c = 0.7045, T_s = 18.2 \text{ с}, T_v = 1 \text{ с},$$

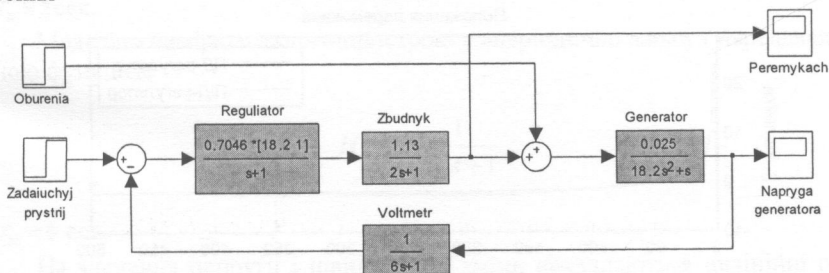
• модель системи з ПД-регулятором



- перехідні процеси в системі з ПД-регулятором при встановленні напруги генератора на 6,3 кВ



- модель системи з ПД-регулятором з урахуванням зовнішнього обурення





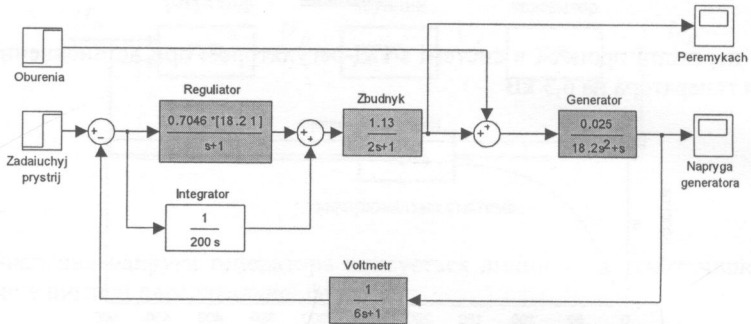
### 3. Дослідження системи з ПІД-регулятором

- передавальна функція ПІД-регулятора

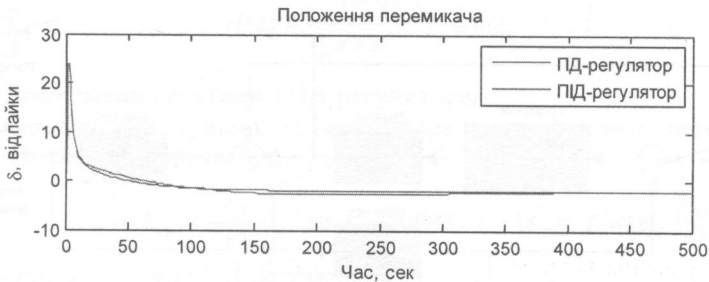
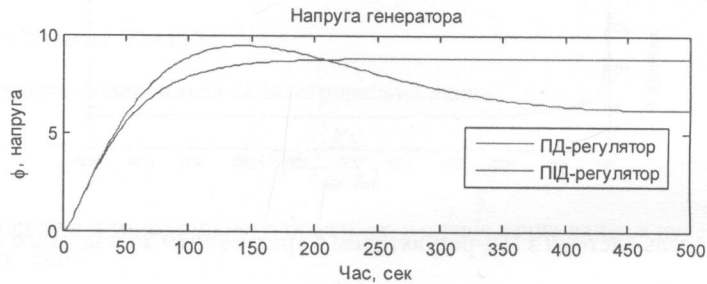
$$C(s) = K_c \left( 1 + \frac{T_s s}{T_i s + 1} \right) + \frac{1}{T_i s},$$

де  $K_c = 0.7045$ ,  $T_s = 18.2$  с,  $T_v = 1$  с,  $T_i = 200$  с,

- модель системи з ПІД-регулятором з урахуванням зовнішнього збурення



- перехідні процеси в системах з ПД- і ПІД-регуляторами

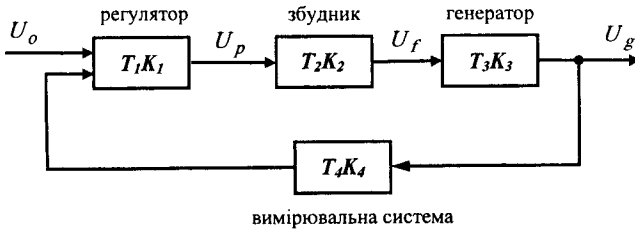


# ЛАБОРАТОРНА РОБОТА № 4 МОДЕЛЮВАННЯ НЕЛІНІЙНИХ СИСТЕМ КЕРУВАННЯ

Варіант № 10

## 1. Опис системи

Досліджується система регулювання напруги генератора. Її структурна схема АСК показана на рисунку



Значення напруги генератора описується лінійною математичною моделлю у вигляді передавальної функції

$$P(s) = \frac{K_3}{(T_s s + 1)},$$

де  $K_3 = 25$  рад/сек,  $T_s = 18.2$  сек,

Збудник моделюється як інтегрувальна ланка

$$R_0(s) = \frac{K_2}{T_R s + 1},$$

$T_R = 2$  сек.

Моделлю вимірювального пристрою є аперіодична ланка з передавальною функцією

$$H(s) = \frac{1}{T_{oc} s + 1},$$

$T_{oc} = 6$  сек,

На значення напруги і швидкість її зміни накладаються нелінійні обмеження

$$|\dot{\delta}(t)| < 10 \text{ В/сек}, \quad |\delta(t)| < 30 \text{ В}.$$

Вимірювальний пристрій (вольтметр) моделюється як аперіодична ланка з передавальною функцією

$$H(s) = \frac{1}{T_{oc}s + 1},$$

$T_{oc} = 6$  сек,

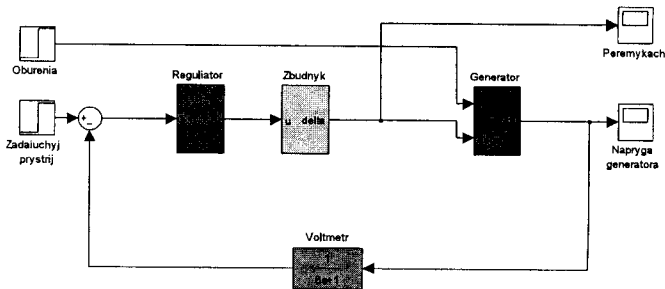
Як керувальний пристрій використовується ПІД-регулятор з передавальною функцією

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

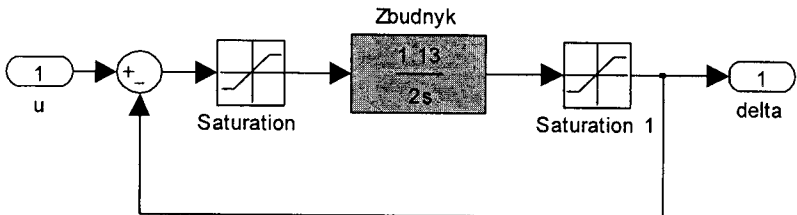
де  $K_c = 0.7045$ ,  $T_s = 18.2$  сек,  $T_v = 1$  сек,  $T_i = 200$  сек,

## 2. Побудова нелінійної моделі

- модель системи керування з виділеними підсистемами



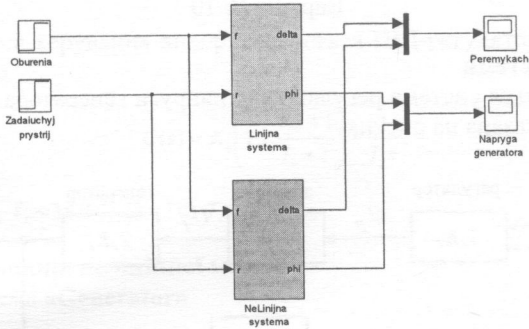
- нелінійна модель збудника



- нижня і верхня межі насичення  
 блок **Saturation**: від  $-10$  до  $10$   
 блок **Saturation1**: від  $-30$  до  $30$

### 3. Порівняння лінійної і нелінійної моделей

- структурна схема системи для порівняння лінійної і нелінійної моделей



- скрипт для побудови і оформлення графіків

```
close(1);
```

```
figure(1);
```

```
subplot(2,1,1);
```

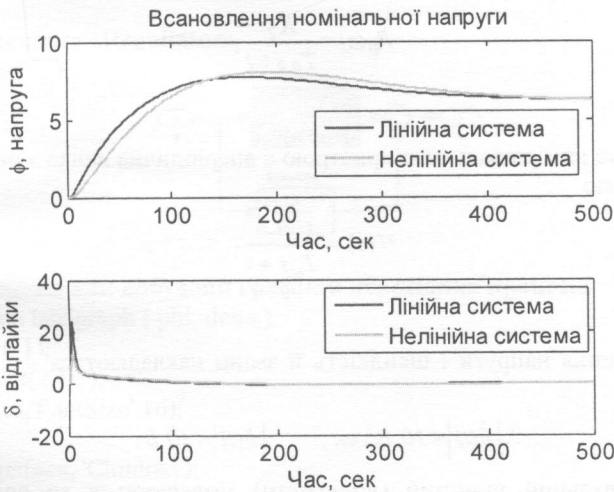
```
...
```

```
set(h(1),'LineWidth',1.5)
```

```
set(h(2),'LineWidth',1.5)
```

- перехідні процеси при встановленні номінального значення напруги

6,3 кВ

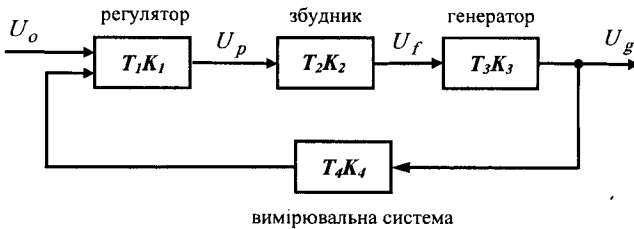


# ЛАБОРАТОРНА РОБОТА № 5 ПРОГРАМУВАННЯ В СЕРЕДОВИЩІ МАТЛАВ

Варіант № 10

## 1. Опис системи

Досліджується система регулювання напруги генератора. Її структурна схема АСК показана на рисунку



Значення напруги генератора описується лінійною математичною моделлю у вигляді передавальної функції

$$P(s) = \frac{K_3}{(T_s s + 1)},$$

де  $K_3 = 25$  рад/с,  $T_s = 18.2$  с.

Збудник моделюється як інтегрувальна ланка

$$R_0(s) = \frac{K_2}{T_R s + 1},$$

$T_R = 2$  с.

Моделлю вимірювального пристрою є аперіодична ланка з передавальною функцією

$$H(s) = \frac{1}{T_{oc} s + 1},$$

$T_{oc} = 6$  с,

На значення напруги і швидкість її зміни накладаються нелінійні обмеження

$$|\dot{\delta}(t)| < 10 \text{ В/сек}, \quad |\delta(t)| < 30 \text{ В}.$$

Вимірювальний пристрій (вольтметр) моделюється як аперіодична ланка з передавальною функцією

$$H(s) = \frac{1}{T_{oc}s + 1},$$

$T_{oc} = 6$  с,

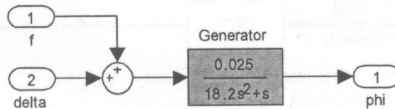
Як пристрій керування використовується ПД-регулятор з передавальною функцією

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

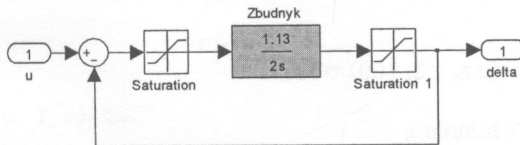
де  $K_c = 0.7045$ ,  $T_s = 18.2$  с,  $T_v = 1$  с,  $T_I = 200$  с.

## 2. Модифікація нелінійної моделі

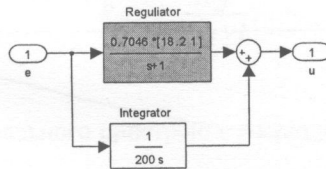
- підсистема «Generator»



- підсистема «Zbudnyk»



- підсистема «Regulator»



## 3. Функція для побудови графіків перехідних процесів

function lab5graph ( phi, delta )

```
figure(1);
subplot(2,1,1);
set(gca,'FontSize',16);
```

...

```
h = get(gca, 'Children');
set(h(1), 'LineWidth', 1.5)
```

## 4. Скрипт sysdata.m для завантаження вихідних даних

```
clear all;
```

```

clc;
K = 0.0694;
Ts = 18.2;
...
phiZad = 5;
fConst = 0;

```

### 5. Скрипт lab5go.m для запуску моделі

```

sysdata;
sim ('lab5')
lab5graph ( phi, delta )

```

### 6. Функція overshoot

```

function [sigma, Tpp] = overshoot ( t, y )
yInf = y(end);
diff = (y - yInf) / abs (yInf);
sigma = max(diff) * 100;
i = find(abs(diff) > 0.02);
Tpp = t(max(i)+1);

```

### 7. Вплив постійної часу судна на показники якості

- скрипт lab5go1.m для проведення розрахунків

```

sysdata;
Ts0 = Ts;
aTs = linspace(0.8, 1.2, 100) * Ts0;
...

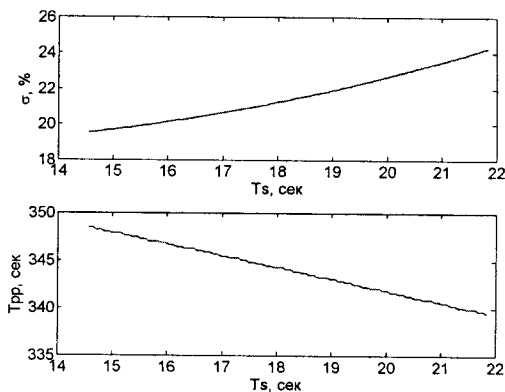
```

```

h = get(gca, 'Children');
set(h(1), 'LineWidth', 1.5)

```

- графіки зміни перерегулювання і часу перехідного процесу

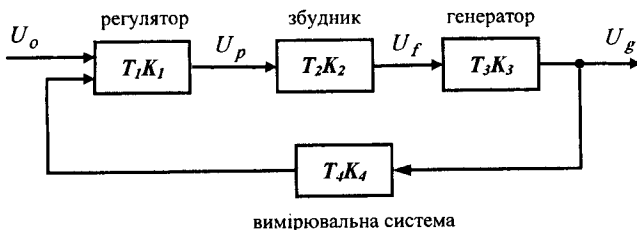


# ЛАБОРАТОРНА РОБОТА № 6 ОПТИМІЗАЦІЯ НЕЛІНІЙНИХ СИСТЕМ В СЕРЕДОВИЩІ МАТЛАВ

Варіант № 10

## 1. Опис системи

Досліджується система регулювання напруги генератора. Її структурна схема АСК показана на рисунку



Значення напруги генератора описується лінійною математичною моделлю у вигляді передавальної функції

$$P(s) = \frac{K_3}{(T_s s + 1)},$$

де  $K_3 = 25$  рад/с,  $T_s = 18.2$  с.

Збудник моделюється як інтегрувальна ланка

$$R_0(s) = \frac{K_2}{T_R s + 1},$$

$T_R = 2$  с.

Моделлю вимірювального пристрою є аперіодична ланка з передавальною функцією

$$H(s) = \frac{1}{T_{oc} s + 1},$$

$T_{oc} = 6$  с.

На значення напруги і швидкість її зміни накладаються нелінійні обмеження

$$|\dot{\delta}(t)| < 10 \text{ В/с}, \quad |\delta(t)| < 30 \text{ В}.$$

Вимірювальний пристрій (вольтметр) моделюється як аперіодична ланка з передавальною функцією



$$H(s) = \frac{1}{T_{oc}s + 1},$$

$T_{oc} = 6$  с.

Як пристрій керування використовується ПІД-регулятор з передавальною функцією

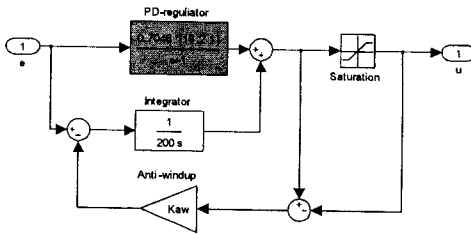
$$C(s) = K_c \left( 1 + \frac{T_I s + 1}{T_I} \right) + \frac{1}{T_I s},$$

де  $K_c = 0.7045$ ,  $T_I = 18.2$  с,  $T_v = 1$  с,  $T_f = 200$  с.

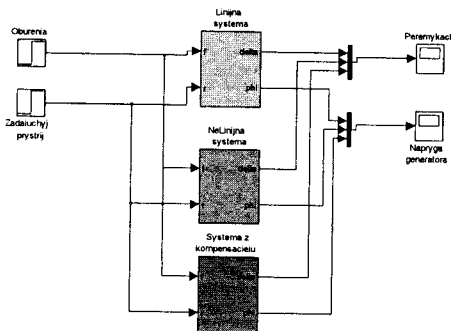
Для компенсації ефекту насичення, викликаного обмеженою кількістю відпайок перемикача, використовується схема з внутрішнім нелінійним зворотним зв'язком, що містить інтегратор у складі регулятора. В ході роботи потрібно вибрати оптимальне значення коефіцієнта підсилення у зворотньому зв'язку за допомогою пакета **Simulink Response Optimization**.

## 2. Блок компенсації насичення

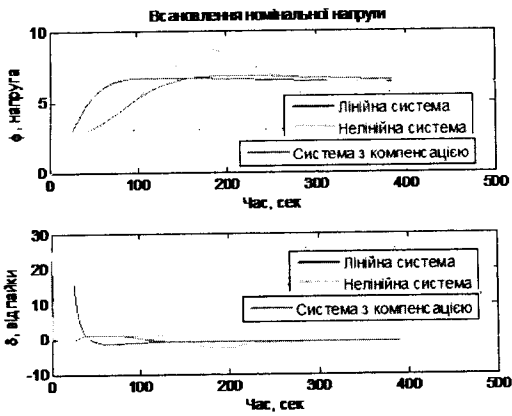
- підсистема «Regulator»



- межі для блока **Saturation**  $\pm 30$  В
- номінальне значення  $K_{aw} = 1$
- модель для порівняння трьох типів систем

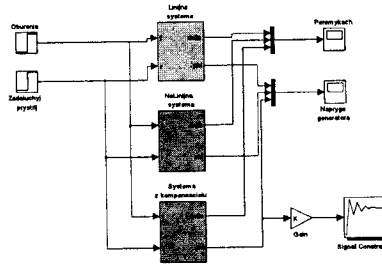


- перехідні процеси при  $K_{aw} = 1$

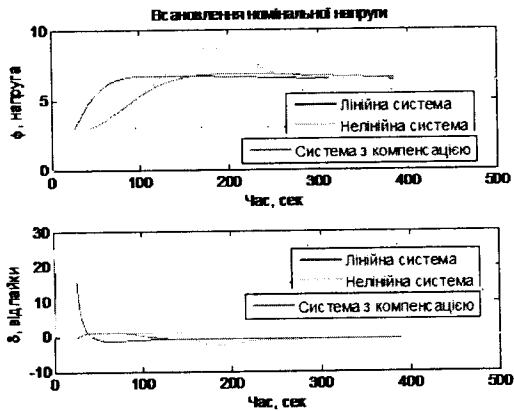


### 3. Оптимальний вибір $K_{сн}$

- модель для оптимізації



- перехідні процеси при  $K_{сн} = 3.084$



*Навчальне видання*

**Кутін Василь Михайлович  
Лесько Владислав Олександрович  
Півнюк Юрій Юрійович**

## **ТЕОРІЯ АВТОМАТИЧНОГО КЕРУВАННЯ**

Лабораторний практикум

Рукопис оформлено *Ю. Півнюком*

Редактор *В. Дружиніна*

Оригінал-макет виготовлено *О. Ткачуком*

Підписано до друку 04.11.2019.  
Формат 29,7×42<sup>1</sup>/<sub>4</sub>. Папір офсетний.  
Гарнітура Times New Roman.  
Друк різнографічний. Ум. друк. арк. 6,36.  
Наклад 50 (1-й запуск 1–21) пр. Зам. 2019-143.

Видавець та виготовлювач  
інформаційний редакційно-видавничий центр.

ВНТУ, ГНК, к. 114.  
Хмельницьке шосе, 95,  
м. Вінниця, 21021.  
Тел. (0432) 65-18-06.

**press.vntu.edu.ua;**

*E-mail:* kivc.vntu@gmail.com.

Свідоцтво суб'єкта видавничої справи  
серія ДК № 3516 від 01.07.2009 р.



**Кутін Василь Михайлович**

Працює у Вінницькому національному технічному університеті на посаді професора кафедри «Електричні станції та системи»

Доктор технічних наук, професор.

Галузь наукових інтересів – технічна діагностика, релейний захист і автоматика електрообладнання.



**Лесько Владислав Олександрович**

Працює у Вінницькому національному технічному університеті на посаді доцента кафедри «Електричні станції та системи».

Кандидат технічних наук, доцент.

Галузь наукових інтересів – автоматизація оптимального керування нормальними режимами електроенергетичних систем.



**Півнюк Юрій Юрійович**

Працює у Вінницькому національному технічному університеті на посаді асистента кафедри «Електричні станції та системи».

Галузь наукових інтересів – оптимізація режимів роботи електричних систем та розподільних електричних мереж.