

Затверджено науково-методичною  
радою ЖДТУ  
протокол від «\_\_»\_\_\_\_\_ 20\_\_ р.  
№\_\_

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ**  
для проведення лабораторних робіт

з навчальної дисципліни  
**«ІНТЕЛЕКТУАЛЬНІ МЕХАТРОННІ СИСТЕМИ»**

для студентів освітнього рівня «магістр»  
для денної та заочної форми навчання

спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»  
(шифр та назва спеціальності)  
освітньо-професійна програма «магістр»  
(назва)

факультет комп'ютерно-інтегрованих технологій, мехатроніки і  
робототехніки  
кафедра автоматизації та комп'ютерно-інтегрованих технологій

Розглянуто і рекомендовано  
на засіданні кафедри

\_\_\_\_\_  
(назва кафедри)  
протокол від «\_\_» \_\_\_\_\_  
201\_\_ р. № \_\_\_\_\_

Розробник: старший викладач каф А та КІТ Богдановський М.В.

Житомир  
2018 – 2019 н.р.

## Лабораторна робота №1

### Дослідження класичних законів управління електромеханічними системами

**Мета роботи:** Здобути практичні навички проектування та оптимізації ПІ та ПІД регуляторів електромеханічними об'єктами.

Здобути практичні навички проектування гібридних фази-ПІ та фази-ПІД регуляторів електромеханічних систем.

#### Короткі теоретичні відомості

Дослідження проводяться з використанням типової структури електромеханічної САР швидкості об'єкта керування, схема якої зображена на рис.1. У цій схемі вжиті наступні позначення:  $\omega$  - швидкість вала двигуна або робочого механізму, яка є регульованою величиною;  $\omega^*$  - завдання швидкості;  $e$  - помилка регулювання;  $u$  - керуюча величина;  $f$  - зовнішнє збурення;  $W_r(p)$  - передавальна функція регулятора швидкості;  $W_o(p)$  - передавальна функція об'єкта керування;  $K_\omega$  - коефіцієнт передачі давача швидкості;  $p = d/dt$ .

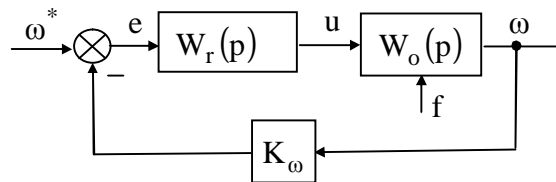


Рис.1.1

В межах даної структури САР у роботі будуть вивчатися регулятори швидкості для керування двома варіантами електромеханічних систем (ЕМС): на основі двигуна постійного струму (ЕМС1) та на основі асинхронного двигуна з перетворювачем частоти (ЕМС2).

#### 1.1. Синтез ПІ регулятора швидкості ЕМС1

Структурна схема ЕМО1, що являє собою електропривод деякого робочого механізму,

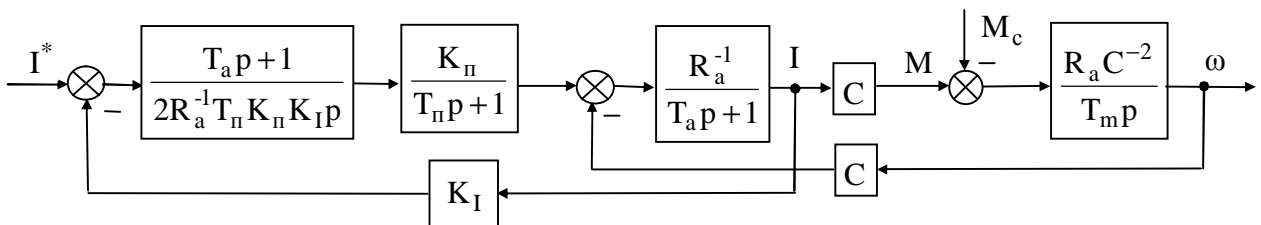


Рис.1.2.

побудований на основі двигуна постійного струму з незалежним збудженням, де контур регулювання струму якоря вже є замкненим, зображена на рис.1.2.

Вжиті у цій схемі позначення означають наступне:  $I$  - струм якоря;  $I^*$  - завдання струму якоря;  $M$  - електромагнітний момент двигуна;  $M_c$  - момент сил опору на валу;  $R_a$  - опір якорного ланцюга;  $T_a$  - стала часу якорного кола;  $T_m$  - електромеханічна стала часу;  $C$  -

постійний коефіцієнт;  $K_{\Pi}$ ,  $T_{\Pi}$  - відповідно коефіцієнт передачі та стала часу перетворювача напруги;  $K_I$  - коефіцієнт передачі давача струму. Регулятор струму в ЕМС1 є аналоговим з налаштуванням на модульний оптимум. Двигун характеризується номінальними значеннями вихідної потужності, кількості обертів та момента на валу такими, як:  $P = 10 \text{ кВт}$ ;  $n_n = 500 \text{ об/хв}$ ;  $M_n = 191 \text{ Н} \cdot \text{м}$ . Значення параметрів моделі ЕМС1 подані в табл.1.1 у відповідних одиницях системи СІ.

Табл.1.1

Параметр	$R_a$	$T_a$	$T_m$	$C$	$K_{\Pi}$	$K_I$	$K_w$
Значення	0,61	0,0754	0,126	3,67	43	0,022	0,1

Синтезуємо дискретний ПІ регулятор швидкості ЕМС1 за методом аналогового прототипу. ПІ закон керування має вигляд

$$u(t) = K_r \left[ e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau \right]. \quad (1.1)$$

При налаштуванні ПІ регулятора швидкості ЕМС1 на симетричний оптимум його коефіцієнт передачі  $K_r$  та стала інтегрування  $T_i$  дорівнюватимуть

$$K_r = T_m C K_I / 4 T_{\Pi} R_a K_{\omega}; \quad T_i = 8 T_{\Pi}, \quad (1.2)$$

де  $T_{\Pi}$  - мала некомпенсована стала часу у контурі регулювання струму двигуна. Диференціюючи (1.1) отримаємо

$$\dot{u}(t) = K_r \left[ \dot{e}(t) + T_i^{-1} e(t) \right]. \quad (1.3)$$

Перейшовши до дискретного часу в (1.3) отримаємо

$$\frac{u(k) - u(k-1)}{T_0} = K_r \left[ \frac{e(k) - e(k-1)}{T_0} + \frac{1}{T_i} e(k) \right], \quad (4)$$

де  $T_0$  - період квантування;  $k = 0, 1, 2, \dots$  - номер періоду квантування;  $t = k T_0$ . З (1.4) отримаємо закон функціонування дискретного ПІ регулятора у вигляді

$$u(k) = K_r \frac{T_0}{T_i} \left[ \frac{T_i}{T_0} \Delta e(k) + e(k) \right] + u(k-1), \quad (1.5)$$

де  $\Delta e(k) = e(k) - e(k-1)$ . За (5) побудована структура ПІ регулятора, що подана на рис.3.

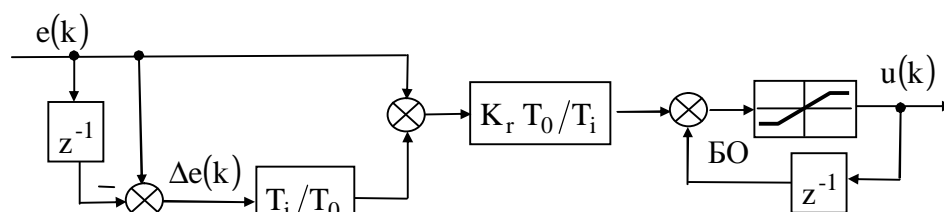


Рис.1.3.

У цій схемі  $z$  - змінна  $Z$ -перетворення, а БО - блок обмеження сигналу керування. При цьому БО застосовано таким чином, що унеможливується насичування ПІ регулятора

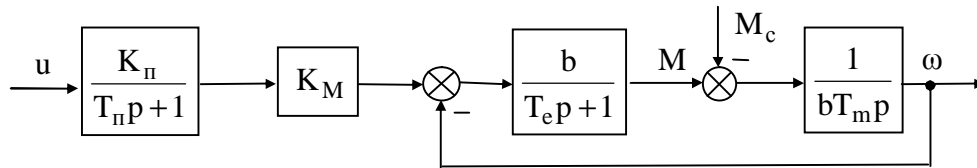


Рис.1.4.

(обмеження виходу регулятора з антинасиченням інтегратора).

## 1.2. Синтез ПІД регулятора швидкості ЕМС2

Структурна схема ЕМС2, що являє собою частотно-керований асинхронний електропривод деякого робочого механізму, зображена на рис.4.

Двигун характеризується номінальними значеннями вихідної потужності, кількості обертів та моменту на валу такими, як:  $P = 2,2 \text{ кВт}$ ;  $n_n = 2841 \text{ об/хв}$ ;  $M_n = 7,4 \text{ Н} \cdot \text{м}$ .

Значення параметрів моделі ЕМС2 подані в табл.2 у відповідних одиницях системи СІ.

Табл.1.2

Параметр	$b$	$T_e$	$T_m$	$K_\pi$	$K_M$	$K_w$
Значення	0,448	0,0091	0,0323	5,0	6,28	0,0318

Синтезуємо дискретний ПІД регулятор швидкості ЕМС2 за методом аналогового прототипу. ПІД закон керування має вигляд

$$u(t) = K_r \left[ e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \dot{e}(t) \right], \quad (1.6)$$

При налаштуванні ПІД регулятора швидкості ЕМС2 на модульний оптимум його коефіцієнт передачі  $K_r$ , стала інтегрування  $T_i$  та стала диференціювання  $T_d$  дорівнюватимуть

$$K_r = T_m / 2T_\pi K_\pi K_M K_w; \quad T_i = T_m; \quad T_d = T_e, \quad (1.7)$$

де  $T_\pi$  - мала некомпенсована стала часу.

Побудуємо ПІД регулятор у вигляді паралельно з'єднаних ПІ та ПД регуляторів. Рівняння (1.6) розкладемо на два складники

$$u(t) = u_i(t) + u_d(t), \quad (1.8)$$

де

$$u_i(t) = K_r \left[ \frac{1}{2} e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau \right]; \quad (1.9)$$

$$u_d(t) = K_r \left[ \frac{1}{2} e(t) + T_d \dot{e}(t) \right]. \quad (1.10)$$

Продиференціювавши (9) та перейшовши до дискретного часу, як це було зроблено вище для ПІ регулятора, отримаємо вираз

$$u_i(k) = K_r \frac{T_0}{T_i} \left[ \frac{T_i}{2T_0} \Delta e(k) + e(k) \right] + u_i(k-1), \quad (1.11)$$

де  $\Delta e(k) = e(k) - e(k-1)$ . Далі, дискретизувавши (10) за часом, отримаємо

$$u_d(k) = \frac{K_r}{2} \left[ \frac{2T_d}{T_0} \Delta e(k) + e(k) \right]. \quad (1.12)$$

У підсумку закон функціонування дискретного ПІД регулятора матиме вигляд

$$u(k) = u_i(k) + u_d(k), \quad (1.13)$$

де  $u_i(k)$  та  $u_d(k)$  визначаються за (1.11) та (1.12) відповідно. За (1.13) побудована структура ПІД регулятора, схема якої подана на рис.5. У цій схемі БО1 обмежує значення ПІ складника сигналу керування з реалізацією антинасічування інтегратора, а БО2 обмежує значення виходу ПІД регулятора на бажаному рівні.

### Програма досліджень

1.2.1. Побудувати модель САР з ПІ регулятором швидкості EMC1, взявши значення сталої часу  $T_n$  з табл.1.3 для вказаного викладачем варіанта.

Табл.1.3

Варіант завдання	1	2	3	4	5	6
$T_n, c$	0,007	0,009	0,006	0,008	0,01	0,005
$m$	63	81	54	72	90	45
$m_i$	5,9	5,9	5,9	5,9	5,9	5,9
$m_d$	6,7	6,7	6,7	6,7	6,7	6,7

1.2.2. Промоделювати систему з ПІ регулятором швидкості EMC1 та побудувати графіки перехідних процесів при стрибкоподібній зміні завдання та навантаження. Визначити показники якості керування даної САР.

1.2.3. Побудувати модель САР з ПІД регулятором швидкості EMC2, взявши значення сталої часу  $T_n$  з табл.1.3 для заданого варіанта.

1.2.4. Промоделювати систему з ПІД регулятором швидкості EMC2 та побудувати графіки перехідних процесів при стрибкоподібній зміні завдання та навантаження. Визначити показники якості керування даної САР.

### Порядок виконання роботи

1.3.1. У системі SIMULINK побудувати модель EMC1 за схемою на рис.1.1. Значення параметрів моделі взяти з табл.1.1. При моделюванні тут і нижче у бібліотечних елементах SIMULINKа доцільно задавати параметри у символному вигляді, присвоївши кожній величині своє ім'я – ідентифікатор. Чисельні значення параметрів при цьому вводяться у робочий простір MATLABа безпосередньо через його командне вікно (**MATLAB Command Window**) або шляхом створення відповідного файлу-сценарію та наступним його запуском на виконання.

1.3.2. Побудувати модель дискретного ПІ регулятора за структурною схемою на рис.1.3. Для налаштування регулятора на симетричний оптимум його параметри слід визначити за виразами (2). Період квантування прийняти  $T_0 = 0,001$  с. Межові значення в БО задати рівними  $\pm 10$ .

1.3.3. Побудувати модель САР за схемою на рис.1.1. з ПІ регулятором швидкості EMC1.

1.3.4. Промоделювати систему з ПІ регулятором швидкості ЕМО1 та побудувати графіки перехідних процесів при відпрацюванні змін завдання та збурення. Зміну завдання швидкості  $w^*$  слід здійснити стрибкоподібно з 0 до 10 рад/с при  $M_c = 0$ . Зміну моменту навантаження двигуна  $M_c$  слід здійснити стрибкоподібно з 0 до  $0,3M_n$  при усталеному русі двигуна на швидкості  $w = 10$  рад/с.

1.3.5. Визначити показники якості керування САР та занести їх в табл.1.4. Зберегти

Табл.1.4.

Регулятор	Об'єкт керування	Показники якості керування				
		$t_p, c$	$s, \%$	$m, \%$	$\Delta w_{max}, \text{рад/с}$	$t_k, c$
ПІ	EMC1					
ФПІ	EMC1					
ПІД	EMC2					
ФПІД	EMC2					

модель САР та результати моделювання на дискеті для майбутнього оформлення звіту.

Показниками якості відпрацювання системою завдання є: 1) час регулювання  $t_p$ , що визначається інтервалом часу від моменту стрибкоподібної зміни завдання до моменту, коли керована величина  $y$  остаточно ввійде у зону  $\pm 5\%$  від нового усталеного значення  $y_{уст}$ ; 2) перерегулювання  $s$ , що визначається відношенням максимального відхилення керованої величини  $\Delta y_{max} = y_{max} - y_{уст}$  від усталеного значення як  $s = (\Delta y_{max} / y_{уст}) \times 100\%$ ; 3) коливальність  $m$ , що оцінюється відношенням сусідніх максимумів відхилення  $\Delta y_{max1}$  та  $\Delta y_{max2}$  кривої перехідного процесу від усталеного значення як  $m = (\Delta y_{max2} / \Delta y_{max1}) \times 100\%$ . Показниками якості компенсування системою змін навантаження є: 1) час компенсування  $t_k$ , що визначається інтервалом часу від моменту стрибкоподібного накиду навантаження до моменту, коли відхилення  $w$  від усталеного значення  $\Delta w = w - w_{уст}$  стане  $|\Delta \omega| \leq 0,2$  рад/с; 2) максимальне відхилення швидкості, що визначається як  $\Delta w_{max} = \max \{ |w - w_{уст}| \}$ .

1.3.7. Побудувати модель EMC2 за схемою на рис.1.4. Значення параметрів моделі взяти з табл.1.2.

1.3.8. Зробити висновки по результатах перехідних процесів щодо якості отриманих систем.

### Зміст звіту

1.4.1. Мета роботи, структура електромеханічної САР швидкості, що подана на рис.1.1.

1.4.2. Складені у SIMULINKу моделі 2-х варіантів САР: 1) з ПІ регулятором швидкості EMC1; 2) з ПІД регулятором швидкості EMC2; 4).

1.4.3. Графіки перехідних процесів при відпрацюванні зміни завдання та компенсуванні накиду навантаження для 2-х варіантів САР швидкості.

1.3.6. Результати порівняльного дослідження якості керування для класичних регуляторів у вигляді заповненої отриманими в роботі даними табл. 1.4.

1.3.7. Висновки по роботі.

### **Контрольні питання**

1. Які закони керування відносяться до класичних?
2. Які переваги та недоліки має ПД регулятор перед ПІ?
3. Яку операцію відтворює оператор  $z^{-1}$  (призначення)?
4. Який показник якості характеризує час тривалості перехідного процесу?
5. Як впливає період квантування на якість перехідного процесу?

## Лабораторна робота №2

### Дослідження комбінованих регуляторів електромеханічними системами на базі нечіткої логіки

**Мета роботи:** Здобути практичні навички проектування комбінованих фазі-ПІ та фазі-ПІД регуляторів електромеханічних систем.

#### Короткі теоретичні відомості

##### 2.1. Побудова фазі-регуляторів швидкості

Синтез фазі-ПІ (ФПІ) регулятора виконаємо на основі побудованого вище дискретного ПІ регулятора. В даному випадку на вхід блока фазі-логіки БФЛ, який є статичним, подаються вхідні змінні, що є пропорційними помилці (вхідному сигналу ПІ регулятора)

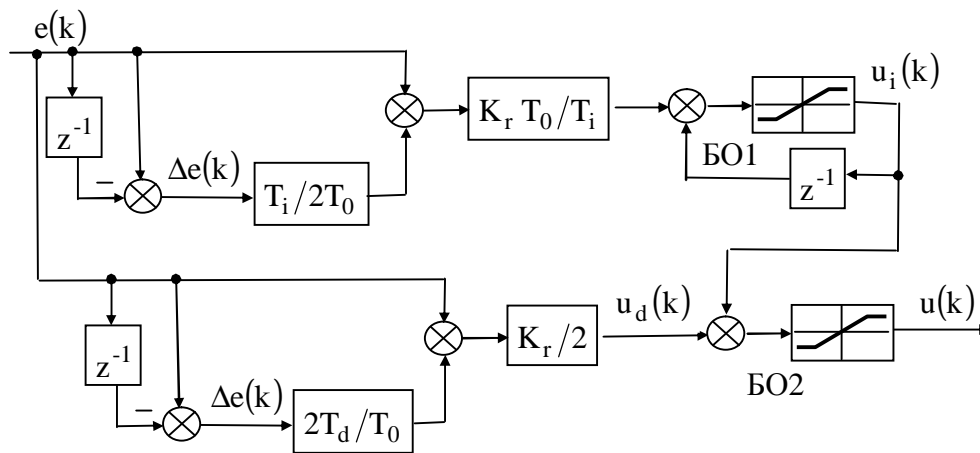


Рис.2.1.

та похідній від помилки. Саме помилка та її перша різниця підсумовуються у схемі на рис.2.1 Отже, замінивши у цій схемі суматор на БФЛ можна перейти до схеми ФПІ. Отримана таким шляхом структурна схема ФПІ зображена на рис.2.2. де з метою спрощення проектування забезпечується нормування вхідних та вихідної змінної БФЛ в

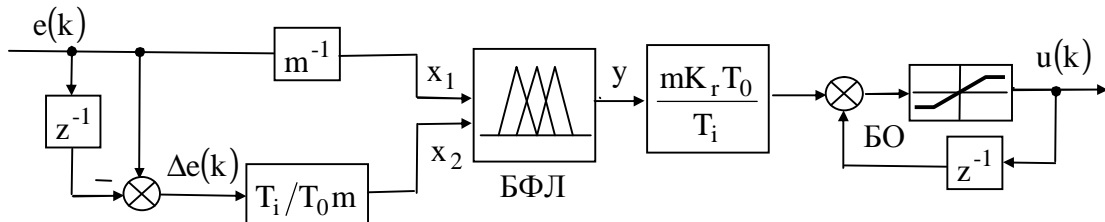


Рис.2.2.

інтервалі  $[-1, 1]$  за рахунок вибору відповідного значення коефіцієнта нормування  $m$ . Аналогічним чином із структури на рис.2.2. можна отримати структурна схему нормалізованого фазі-ПІД (ФПІД) регулятора, що зображена на рис.2.3. Схема є



поєднанням ФПІ регулятора з фазі-блоком БФЛ1 та коефіцієнта нормування змінних  $m_i$ , а також ФПД регулятора з фазі-блоком БФЛ2 та коефіцієнта нормування змінних  $m_d$ .

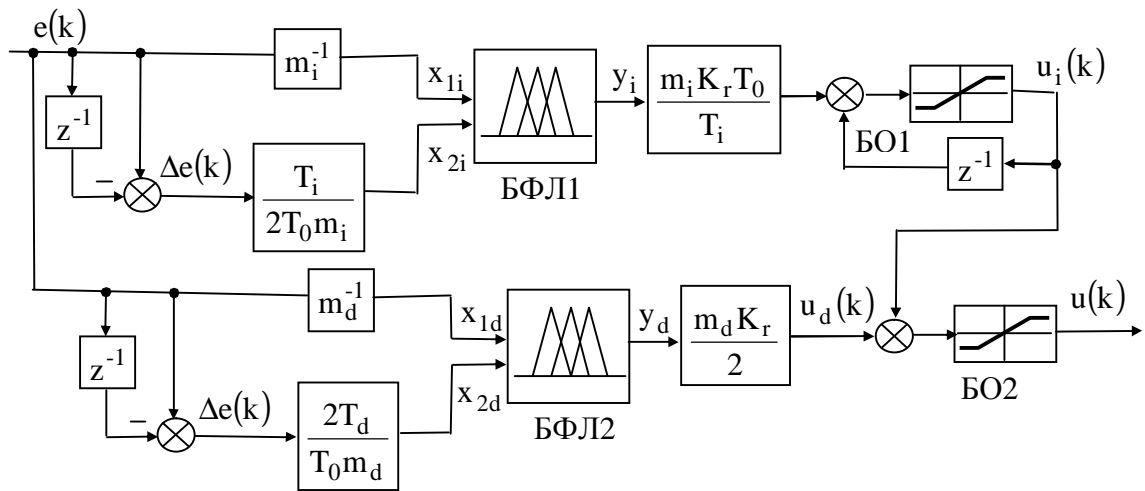


Рис.2.3.

На рис.8 представлена симетрична функція належності нечіткої змінної, що складається з семи лінгвістичних термів. Імена, форми та параметри цих термів подані в табл.2.1. Дані функція належності, а також база правил, яка відображена в табл.2.2, будуть використовуватися при побудові БФЛ для ФПІ та ФПД, про що докладніше йтиметься нижче у методичних вказівках.

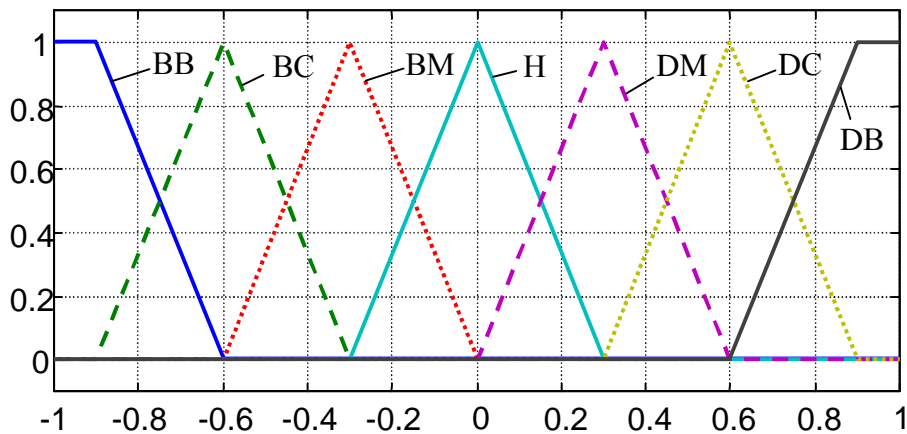


Рис.2.4.

Табл. 2.1

Терм	Форма терму	Характерні точки
<b>ВВ</b> (Від'ємне велике)	трапеція	[-10, -10, -0.9, -0.6]
<b>ВС</b> (Від'ємне середнє)	трикутник	[-0.9, -0.6, -0.3]
<b>ВМ</b> (Від'ємне мале)	трикутник	[-0.6, -0.3, 0]
<b>Н</b> (Нуль)	трикутник	[-0.3, 0, 0.3]
<b>ДМ</b> (Додатне мале)	трикутник	[0, 0.3, 0.6]

<b>DC</b> (Додатне середнє)	трикутник	[0.3, 0.6, 0.9]
<b>DB</b> (Додатне велике)	трапеція	[0.6, 0.9, 10, 10]

Табл.2.2

		Значення $x_1(k)$						
		BB	BC	BM	H	DM	DC	DB
Значення $x_2(k)$	DB	H	DM	DC	DB	DB	DB	DB
	DC	BM	H	DM	DC	DB	DB	DB
	DM	BC	BM	H	DM	DC	DB	DB
	H	BB	BC	BM	H	DM	DC	DB
	BM	BB	BB	BC	BM	H	DM	DC
	BC	BB	BB	BB	BC	BM	H	DM
	BB	BB	BB	BB	BB	BC	BM	H

### Програма досліджень

2.2.1. Синтезувати ФПІ регулятор на основі раніше отриманого дискретного ПІ регулятора, взявши значення коефіцієнта нормування  $m$  з табл.1.3 для заданого варіанта.

2.2.2. Побудувати модель САР з ФПІ регулятором швидкості ЕМС1, промоделювати цю систему та побудувати графіки перехідних процесів при стрибкоподібній зміні завдання та навантаження. Визначити показники якості керування даної САР.

2.2.3. Синтезувати ФПІД регулятор на основі раніше отриманого дискретного ПІД регулятора, взявши значення коефіцієнтів нормування  $m_f$  та  $m_d$  з табл.1.3 для заданого варіанта.

2.2.4. Побудувати модель САР з ФПІД регулятором швидкості ЕМС2, промоделювати цю систему та побудувати графіки перехідних процесів при стрибкоподібній зміні завдання та навантаження. Визначити показники якості керування даної САР.

2.2.5. Провести порівняльний аналіз показників якості керування систем зі звичайними та нечіткими регуляторами, зробити відповідні висновки.

### Порядок виконання роботи

2.3.1. Синтезувати ФПІ регулятор на основі раніше отриманого дискретного ПІ регулятора швидкості ЕМО1. У поданій на рис.2.1. структурі для моделювання БФЛ слід використати блок **Fuzzy-Logic-Controller** з бібліотеки **fuzblock**.

2.3.2. Для того, щоб викликати **FIS Editor** з **Fuzzy Logic Toolbox**, який є інструментарієм для створення та редагування фази-блоків в межах середовища **MATLAB**, необхідно ввести з командного вікна ім'я *fuzzy*. Після цього відкривається вікно редактора фази-блоків, вигляд якого представлений на рис.2.5. Функції належності для обох вхідних змінних та вихідної змінної БФЛ слід задати такими, як це зображено на рис.2.4. Форму та параметри лінгвістичних термів слід взяти із табл.2.1. Після цього треба сформулювати базу фази-правил за табл.2.2., для чого використовується редактор правил. Щоб його відкрити, треба вибрати в меню **View**→**Edit rules**. Складання фази-правил проводиться автоматично на основі опису вхідних та вихідних лінгвістичних змінних, який був зроблений за допомогою редактора фази-блоків. Для того, щоб утворити нове правило, треба натиснути на кнопку **Add rule**, після чого обрати значення кожної вхідної змінної, логічний зв'язок між вхідними змінними та значення вихідної змінної, що відповідає створеній *ЯКЩО-частині* фази-правила. У випадку, якщо правило не містить якоїсь вхідної величини, треба

обрати як вхідну величину **none**. Якщо змінна має бути інвертованою, треба поставити прапорець напроти **not**. Для проєктованого БФЛ пропонується використати такі установки інференції:

- оператор імплікації – мінімум-оператор; оператор агрегації – максимум-оператор.
- дефазіфікацію здійснювати за гравітаційним методом (центр ваги) - *centroid*.

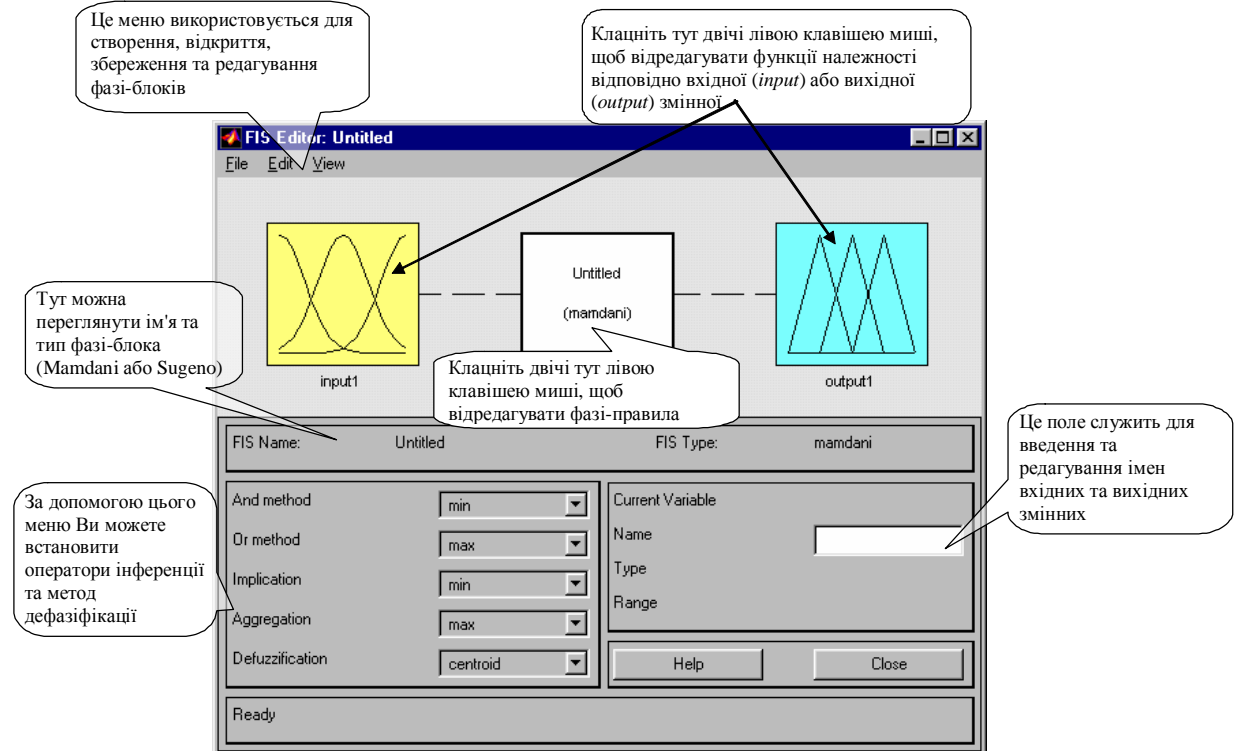


Рис. 2.5.

Діагностика розробленого фазі-блока здійснюється за допомогою переглядача тривимірної функції (**Surface Viewer**) та переглядача правил (**Rule Viewer**). Розроблений фазі-блок у вигляді FIS-матриці треба зберегти в робочому просторі під іменем, яке використовується як параметр у блоці **Fuzzy-Logic-Controller**. Для того, щоб зберегти FIS-матрицю у робочому просторі, треба обрати в меню редактора фазі-блоків **File**→**Save to workspace** або **File**→**Save to workspace as**. У випадку, якщо фазі-блок був раніше спроектований та збережений на твердому диску, то FIS-матрицю можна зчитати з диска та занести у робочий простір **MATLABa**.

2.3.3. Промоделювати систему з ФПІ регулятором швидкості EMC1 та побудувати графіки перехідних процесів при відпрацюванні змін завдання та збурення за п.1.3.4.

2.3.4. Визначити показники якості керування САР за п.1.3.5 та занести їх в табл.1.4. Зберегти модель САР та результати моделювання на дискеті для майбутнього оформлення звіту.

2.3.5. Синтезувати ФПД регулятор на основі раніше отриманого дискретного ПД регулятора швидкості EMC2. У поданій на рис.2.3. структурі для моделювання БФЛ1 та БФЛ2 слід використати блок **Fuzzy-Logic-Controller** з FIS-матрицею, що була отримана вище при синтезі ФПІ регулятора.

2.3.6. Промоделювати систему з ФПД регулятором швидкості EMC2 та побудувати графіки перехідних процесів при відпрацюванні змін завдання та збурення за п.1.3.4.

2.3.7. Визначити показники якості керування САР за п.1.3.5 та занести їх в табл.1.4. Зберегти модель САР та результати моделювання на дискеті для майбутнього оформлення звіту.

### Зміст звіту

- 2.4.1. Мета роботи, структура електромеханічної САР швидкості, що подана на рис.1.1.
- 2.4.2. Складені у SIMULINKу моделі 2-х варіантів САР: 1) з ФПІ регулятором швидкості ЕМС1; 2) з ФПД регулятором швидкості ЕМС2.
- 2.4.3. Діаграму синтезованого в роботі БФЛ, де позначаються вхідні та вихідні змінні, вигляд їх функцій належності, кількість фазі-правил та тип системи інференції (використати функцію *plotfis (fismat)*, де *fismat* – ім'я FIS-матриці, що використовувалась в БФЛ );
- 2.4.4. Тривимірний графік передавальної функції синтезованого в роботі БФЛ (використати функцію *gensurf (fismat)*);
- 2.4.5. Графіки перехідних процесів при відпрацюванні зміни завдання та компенсуванні накиду навантаження для 2-х варіантів САР швидкості.
- 2.4.6. Результати порівняльного дослідження якості керування для звичайних та нечітких регуляторів у вигляді заповненої отриманими в роботі даними табл.1.4.
- 2.4.7. Висновки по роботі.

### Контрольні запитання

6. Дайте визначення нечіткої множини.
7. Що таке функція належності в теорії нечітких множин?
8. Назвіть основні типи функцій належності.
9. Дайте визначення таких операцій на фазі-множинах як перетин, поєднання та заперечення.
10. Що таке система фазі-логіки типу Мамдані?
11. Що таке система фазі-логіки типу Сугено?
12. Які етапи нечіткого логічного виведення ви знаєте?
13. Назвіть основні методи дефазифікації.

## Лабораторна робота №3

## Інтелектуальні системи управління на базі штучних нейронних мереж

**Мета роботи:** Отримання практичних навичок з розділу теорії штучних нейронних мереж (ШНМ) та їх застосування в електромеханічних системах керування. Вивчення ШНМ прямого поширення сигналу щодо придатності до апроксимації складних функцій, здатності до навчання та узагальнення, можливості відтворення властивостей динамічних об'єктів. Ознайомлення з явищем перенавчання ШНМ та способами його запобігання.

## Теоретичні відомості

Зазвичай під штучною НМ мається на увазі сукупність елементарних нейроноподібних перетворювачів інформації (штучних нейронів), що з'єднані між собою певним чином для обміну інформацією та для спільної роботи.

Серед відомих на сьогодні типів НМ найбільше застосування в задачах автоматичного керування знаходять мережі прямого поширення (передавання) сигналу (the feedforward networks), які називають також багатошаровими персептронами. Процедура синтезу чи проектування ШНМ прямого поширення в загальному випадку полягає у виборі кількості захованих шарів нейронів, тобто шарів, що розташовані між входами мережі та вихідним шаром нейронів; у визначенні кількості нейронів в захованих шарах; у виборі функції активації нейронів, а також у навчанні (тренуванні) мережі, завдяки якому визначаються чисельні значення параметрів її внутрішніх зв'язків.

На рис.1 представлена архітектура двошарової НМ прямого поширення, де кружечками умовно позначені нейрони, а стрілками – зв'язки між нейронами. На рис.3.1.

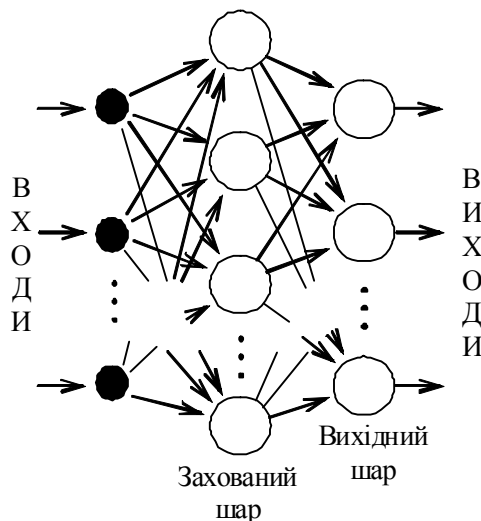


Рис.3.1. Архітектура НМ з прямим поширенням сигналів

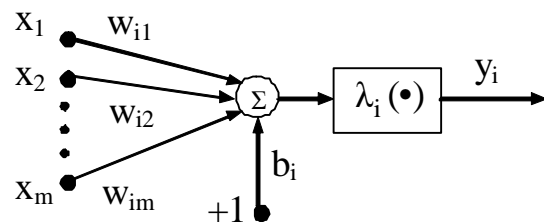


Рис.3.2 Схема нейрона

схематично зображено і-й нейрон, вихід якого  $y_i$  визначається як

$$y_i = I_i \left( \sum_{j=1}^m x_j w_{ij} + b_i \right), \quad (3.1)$$

де  $x_1, x_2, \dots, x_m$  – входи нейрона;  $w_{i1}, w_{i2}, \dots, w_{im}$  – вагові коефіцієнти зв'язків або синаптичні ваги;  $b_i$  – зміщення або зсув нейрона (the bias);  $I_i(\cdot)$  – активаційна функція нейрона. У нейронах застосовуються функції активації лінійного та нелінійного типу. Серед останніх досить часто вживається функція гіперболічного тангенсу

$$I(x) = th(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}. \quad (3.2)$$

Мережі прямого поширення навчаються за допомогою методів супервізорного типу (навчання з учителем), серед яких широке застосування знайшов метод зворотного поширення помилки (backpropagation), зокрема його модифікація з використанням алгоритму оптимізації Левенберга-Марквардта. Для навчання НМ треба сформулювати навчальну множину даних, куди ввійдуть масиви векторів  $\bar{X}(k)$ ,  $k = \overline{1, N}$  та  $\bar{Y}(k)$ ,  $k = \overline{1, N}$ , де  $N$  – обсяг навчальної множини;  $\bar{X} = [X_1, X_2, \dots, X_n]^T$  – вектор входу мережі;  $n$  – кількість входів НМ;  $\bar{Y} = [Y_1, Y_2, \dots, Y_q]^T$  – вектор виходу мережі;  $q$  – кількість виходів НМ, що за визначенням дорівнює кількості нейронів у її вихідному шарі. В процесі навчання при подачі на вхід НМ вектора  $\bar{X}(k)$  на виході мережі формується вектор оцінки  $\hat{Y}(k)$  та визначається помилка мережі  $e_{net}(k) = \sum_{i=1}^q (Y_i(k) - \hat{Y}_i(k))$ . Після досягнення  $k = N$ , закінчується епоха (цикл) навчання та обчислюється певний функціонал якості навчання, наприклад середній квадрат помилки мережі  $E_{net} = N^{-1} \sum_{k=1}^N e_{net}^2(k)$ . Залежно від значення  $E_{net}$  програма навчання змінює коефіцієнти зв'язків між нейронами та зміщення нейронів. Процес навчання триває доти, поки  $E_{net}$  не стане меншим від заданого значення або поки не буде досягнуто встановленої кількості епох навчання мережі.

Досить часто у літературних джерелах застосовують умовне позначення архітектури НМ прямого поширення у вигляді  $S0 - S1 - S2 - \dots - Sq$ , де  $S0$  та  $Si, i = \overline{1, q}$  – означають відповідно кількість входів НМ та кількість нейронів  $i$ -го шару мережі. Наприклад, НМ з двома входами, одним захованим шаром з трьома нейронами та з одним нейроном у вихідному шарі матиме умовне позначення у вигляді 2-3-1.

Оцінка якості навчання мережі може бути отримана шляхом обчислення лінійної регресії між навчальними та відповідними їм реальними значеннями виходу мережі. Лінія регресії, що будується в координатах  $X_{rg} = Y$ ,  $Y_{rg} = \hat{Y}$ , описується як  $Y_{rg} = mX_{rg} + b$ , де  $m$  – коефіцієнт нахилу лінії регресії,  $b$  – точка перетину цієї лінії з віссю ординат. Очевидно, що для добре навченої НМ лінія регресії буде досить наближеною до ідеальної  $Y_{rg} = X_{rg}$ . При обчисленні лінійної регресії визначається також коефіцієнт регресії  $R$ , що являє собою міру кореляції між навчальними та відповідними їм реальними значеннями виходу мережі. Для бездоганно навченої мережі матимемо  $R = 1$ .

### 3.1. Апроксимація ШНМ складних функцій

До класичних задач, що існують в прикладних галузях науки, відноситься задача апроксимації експериментальних даних. Для високоякісного розв'язання цієї задачі НМ, які часто називають універсальними апроксиматорами, мають великі перспективи. Математиками доведено, що при достатньо великій кількості нейронів у захованому шарі, мережа прямого поширення здатна апроксимувати із заданою точністю любую нелінійну



функцію багатьох аргументів із обмеженою кількістю точок розриву. Мінімально необхідна кількість нейронів у захваному шарі при цьому визначається евристично, виходячи із допустимої помилки апроксимації.

Здатність ШНМ до високоякісної апроксимації складних залежностей досягається завдяки сегментуванню вхідного простору на підпростори дією нелінійних функцій активації нейронів та наступного інтерполювання нейронами даних на кожному сегменті. Аналіз роботи ШНМ засвідчує, що при одних вхідних сигналах частина нейронів працює а інші є пасивними, проте при інших вхідних сигналах працюючими вже є інші нейрони.

У даній роботі здатність НМ до апроксимування вивчатимемо на прикладі нелінійної векторної багатоекстремальної функції

$$F(x_1, x_2) = A_1(1 - x_1)^2 e^{-x_1^2 - (x_2 + 1)^2} + A_2(x_1/3 - x_1^3 - x_2^2) e^{-x_1^2 - x_2^2}, \quad (3.3)$$

де  $x_1, x_2 \in [-3, 3]$ ;  $A_1$  та  $A_2$  - сталі величини.

Явище перенавчання ШНМ. При навчанні ШНМ може існувати явище, яке називають явищем перенавчання. Це явище полягає у тому, що помилки навчання на даних навчальної множини виявляються дуже малими, але коли на мережу подаються нові дані, то помилки істотно зростають. Це значить, що ШНМ завчила приклади, проте не навчилася розв'язувати задачу. Іншими словами, мережа не здатна розпізнавати нові дані, а значить не набула властивості узагальнення.

Одним із способів виправити ситуацію полягає у тому, щоб використати мережу з меншою кількістю нейронів, тобто мережу меншої потужності, яка би не змогла точно відтворювати на виході задані значення. Проте, оскільки наперед важко передбачити наскільки потужною має бути мережа для конкретної задачі, цей спосіб вимагає емпіричного вирішення.

Відомі також два систематичних підходи до уникнення явища перенавчання ШНМ. Перший з них полягає у застосуванні метода регуляризації, а другий – у формуванні репрезентативної навчальної множини.

Синтез нейронного ПД регулятора Структурна схема електромеханічної САР швидкості об'єкта керування зображена на рис.3.3. У цій схемі вжиті наступні позначення:

$w$  - швидкість вала двигуна або робочого механізму, яка є регульованою величиною;  $w^*$  - завдання швидкості;  $e$  - помилка регулювання;  $u$  - керуюча величина;  $f$  - зовнішнє збурення;  $W_r(p)$  - передавальна функція регулятора швидкості;  $W_o(p)$  - передавальна функція об'єкта керування;  $K_\omega$  - коефіцієнт передачі давача швидкості;  $p = d/dt$ . Для даної САР у роботі ставиться задача синтезу нейронного

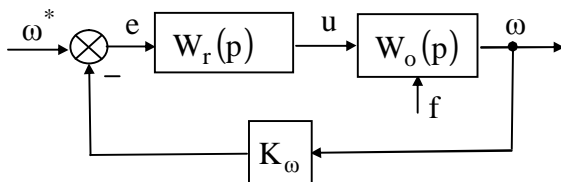


Рис.3.3.

ПД (НПД) регулятора, що з високою точністю відтворюватиме властивості звичайного ПД регулятора швидкості електромеханічного об'єкта (ЕМС).

Структурна схема ЕМО, що являє собою частотно-керований асинхронний електропривод деякого робочого механізму, зображена на рис.3.4.

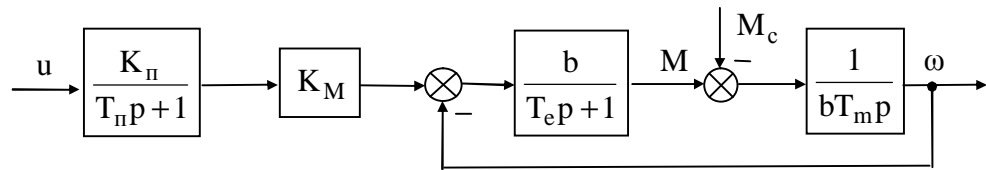


Рис.3.4.

Двигун характеризується номінальними значеннями вихідної потужності, кількості обертів та момента на валу такими, як:  $P = 2,2 \text{ кВт}$ ;  $n_n = 2841 \text{ об/хв}$ ;  $M_n = 7,4 \text{ Н} \cdot \text{м}$ . Значення параметрів моделі ЕМО подані в табл.1 у відповідних одиницях системи СІ.

Табл.3.1.

Параметр	$b$	$T_e$	$T_m$	$K_\pi$	$K_M$	$K_w$
Значення	0,448	0,0091	0,0323	5,0	6,28	0,0318

Синтезуємо дискретний ПІД регулятор швидкості ЕМС за методом аналогового прототипу. ПІД закон керування має вигляд

$$u(t) = K_r \left[ e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \dot{e}(t) \right], \quad (3.4)$$

При налаштуванні аналогового ПІД регулятора швидкості ЕМС на модульний оптимум його коефіцієнт передачі  $K_r$ , стала інтегрування  $T_i$  та стала диференціювання  $T_d$  дорівнюватимуть

$$K_r = T_m / 2T_\pi K_\pi K_M K_w; \quad T_i = T_m; \quad T_d = T_e, \quad (3.5)$$

де  $T_\pi$  - мала некомпенсована стала часу.

Перейшовши до дискретного часу в (3.4) шляхом апроксимації інтегралу за формулою прямокутників, а похідної лівою різницею, отримаємо закон функціонування дискретного ПІД регулятора у вигляді різницевого рівняння

$$u(k) = q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) + u(k-1), \quad (3.6)$$

де коефіцієнти визначаються за формулами

$$q_0 = K_r (1 + T_d / T_0); \quad q_1 = -K_r (1 + 2T_d / T_0 - T_0 / T_i); \quad q_2 = K_r T_d / T_0, \quad (3.7)$$

а  $T_0$  - період квантування;  $k = 0, 1, 2, \dots$  - номер періоду квантування;  $t = kT_0$ . Дискретну передавальну функцію деякого динамічного об'єкта в загальному вигляді можна записати як

$$W(z) = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2} + \dots + q_m z^{-m}}{p_0 + p_1 z^{-1} + p_2 z^{-2} + \dots + p_n z^{-n}}, \quad (3.8)$$



де  $z^{-1}$ - оператор затримки в часі на  $T_0$ . За формою (8) запишемо дискретну передавальну функцію ПД регулятора на основі (3.5) у вигляді

$$W_r(z) = \frac{u(z)}{e(z)} = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-1}}. \quad (3.9)$$

У даній роботі завдання полягає у побудові нейромережного регулятора із властивостями дискретного ПД регулятора швидкості ЕМС з передавальною функцією (3.9). Відомо, що для отримання динамічних властивостей у статичній НМ, якою є зображена на рис.1 мережа прямого поширення, вона має бути доповнена елементами затримки сигналів у часі. Це дозволяє подавати на входи мережі крім поточних значень вхідного  $e(t)$  та вихідного  $u(t)$  сигналів також і їх попередні значення.

Для синтезу ШНМ нижче використовуватимемо програмні засоби системи MATLAB, зокрема його підсистему програм Neural Network Toolbox. Процес синтезу може бути організовано наступним чином.

Треба згенерувати сигнал навчання на вході регулятора  $e^*(k)$  у вигляді стрибкоподібного сигналу з випадковим значенням амплітуди, а також сигнал навчання на виході регулятора  $u^*(k)$ , що є реакцією динамічної системи з ПФ (3.9) на  $e^*(k)$ . Вектори входу та виходу мережі задамо відповідно як  $\bar{X}(k) = [e^*(k), e^*(k-1), u^*(k-1)]^T$ ,  $\bar{Y}(k) = [u^*(k)]^T$ . Для навчання НМ слід сформуванати навчальну множину даних, куди ввійдуть масиви векторів  $\bar{X}(1) \dots \bar{X}(N)$  та  $\bar{Y}(1) \dots \bar{Y}(N)$ .

Побудову та навчання ШНМ у MATLABі можна здійснити наступним програмним блоком:

- 1) `merzha = newff(minmax(Xnav),[1],{'purelin'},'trainlm');`
- 2) `merzha.trainParam.epochs = 100;`
- 3) `merzha.trainParam.goal = 0.001^2;`
- 4) `merzha = train(merzha,Xnav,Ynav); grid on`

Тут першим оператором шляхом використання функції `newff()` створюється ШНМ з іменем `merzha`, яка складається з одного нейрона з лінійною функцією активації, а для її навчання використовується варіант Левенберга-Марквардта методу зворотного поширення помилки `trainlm()`. Другим оператором встановлюється максимальна кількість епох навчання, рівна 100, а третій оператор задає максимальну помилку мережі  $E_{\max} = 1 \cdot 10^{-6}$ . Четвертим оператором, шляхом використання функції `train()` здійснюється навчання НМ на навчальній множині даних, яку складають масиви вектора входу  $X_{nav}$  та вектора виходу  $Y_{nav}$ . Процес навчання супроводжується візуалізацією помилки НМ у залежності від реалізованих епох тренування.

Для побудови моделі синтезованої НМ у системі SIMULINK використовується функція *gensim(mezzha, T0)*. Дану НМ слід доповнити 3-ма елементами затримки сигналів на період квантування  $T_0$ . У підсумку отримаємо модель нейронного ПІД регулятора, схема якого представлена на рис.5.

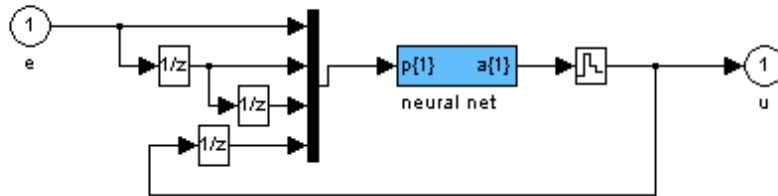


Рис.3.5.

### Програма досліджень

3.1.1. Дослідити апроксимувальні властивості НМ прямого поширення на прикладі функції (3), де коефіцієнти  $A1$  та  $A2$  взяти з табл.2 для вказаного викладачем варіанта.

Табл.3.2.

Варіант завдання	1	2	3	4	5	6
A1	0,1	-0,1	1	-2	10	-7
A2	-100	50	-10	7	-10	7
S1	25	30	20	27	23	33
$T_{II}$ , с	0,006	0,005	0,007	0,009	0,008	0,01

3.1.2. Вивчити явище перенавчання ШНМ на прикладі апроксимації гармонічної функції двошаровою мережею прямого поширення, де кількість нейронів у захваному шарі S1 взяти з табл.3.2 для заданого варіанта.

3.1.3. Побудувати модель САР з дискретним ПІД регулятором швидкості ЕМС, взявши значення сталої часу  $T_{II}$  з табл.3.2 для заданого варіанта.

3.1.4. Синтезувати нейромережний регулятор, що відтворюватиме характеристики ПІД регулятора швидкості ЕМС.

3.1.5. Промодельювати для САР швидкості ЕМС з ПІД та НПІД регуляторами та побудувати графіки перехідних процесів при стрибкоподібній зміні завдання та навантаження.

3.1.6. Порівняти показники якості керування системи зі звичайним та нейронним регуляторами.

3.1.7. Зробити висновки за проведеними у роботі дослідженнями.

### Порядок виконання роботи

3.2.1. Використовуючи лекційні матеріали, рекомендовану літературу та дані методичні вказівки вивчити принципи побудови та функціонування ШНМ.

3.2.2. Для вивчення апроксимувальної здатності ШНМ скласти відповідну програму у MATLABi, яка повинна будувати поверхню функції (3.3), створювати та навчати двошарову НМ прямого поширення для апроксимації даної функції, обчислювати лінійну регресію НМ та будувати відповідний графік. За допомогою створеної програми

дослідити залежність якості апроксимації від кількості нейронів у захованому шарі ШНМ та заповнити отриманими даними табл.3.

Табл.3.3

Показник якості апроксимації	Кількість нейронів у захованому шарі				
	10	15	20	25	30
Помилка мережі $E_{net}$					
Коефіцієнт регресії R					

3.2.3. При складанні зазначеної у попередньому пункті програми рекомендується ознайомитись з лістингом програми А, що поданий у додатку.

3.2.4. Для вивчення явища перенавчання і НМ скласти відповідну програму у MATLABі, яка повинна апроксимувати гармонічну функцію при кількості нейронів у захованому шарі рівній S1, значення якого взято з табл.3.2, а також при S1=5.

3.2.5. При складанні зазначеної у попередньому пункті програми рекомендується ознайомитись з лістингом програми В, що поданий у додатку.

3.2.6. У системі SIMULINK побудувати модель EMC за схемою на рис.3.4. Значення параметрів моделі взяти з табл.3.1. При моделюванні тут і нижче у бібліотечних елементах SIMULINKа доцільно задавати параметри у символічному вигляді, присвоївши кожній величині своє ім'я – ідентифікатор. Чисельні значення параметрів при цьому вводяться у робочий простір MATLABа безпосередньо через його командне вікно (MATLAB Command Window) або шляхом створення відповідного файла-сценарію та наступним його запуском на виконання.

3.2.7. Побудувати модель дискретного ПІД регулятора з передавальною функцією (3.9). Для налаштування регулятора на модульний оптимум його параметри слід визначити за виразами (3.5) та (3.7). Період квантування прийняти  $T_0 = 0,001$  с.

3.2.8. Побудувати модель САР за схемою на рис.3 з ПІД регулятором швидкості EMC.

3.2.9. Промодельювати систему з ПІД регулятором швидкості EMC та побудувати графіки перехідних процесів при відпрацюванні змін завдання та збурення. Зміну завдання швидкості  $w^*$  слід здійснити стрибкоподібно з 0 до 10 рад/с при  $M_c = 0$ . Зміну моменту навантаження двигуна  $M_c$  слід здійснити стрибкоподібно з 0 до  $0,3M_n$  при усталеному русі двигуна на швидкості  $w = 10$  рад/с. Зберегти модель САР та результати моделювання на дискеті для майбутнього оформлення звіту.

3.2.10. Скласти програму у MATLABі, яка повинна створити ШНМ прямого поширення, що складається з одного нейрона з лінійною функцією активації, та навчити цю НМ за характеристиками раніше отриманого дискретного ПІД регулятора швидкості EMC.

3.2.11. При складанні зазначеної у попередньому пункті програми рекомендується ознайомитись з лістингом програми С, що поданий у додатку.

3.2.12. Створити модель синтезованої ШНМ у SIMULINKу, для чого необхідно викликати з командного вікна функцію *gensim(merezha,T0)*.

3.2.13. На основі моделі синтезованої ШНМ у SIMULINK створити модель НПІД регулятора згідно з рис. 3.5.

3.2.14. Промодельювати систему з НПІД регулятором швидкості EMC та побудувати графіки перехідних процесів при відпрацюванні змін завдання та збурення за п.3.2.9. Зберегти модель САР та результати моделювання на дискеті для майбутнього оформлення звіту.

### Зміст звіту

3.3.1. Мета роботи та програма досліджень.

3.3.2. Зображення поверхонь вихідної функції (3.3) та її апроксимації мережею з кількістю нейронів у захованому шарі  $S1=10$  та  $S1=30$ . Графіки лінійної регресії для ШНМ, що апроксимує функцію (3.3), при  $S1=10$  та  $S1=30$ . Результати дослідження якості апроксимації в залежності від кількості нейронів у захованому шарі ШНМ у вигляді заповненої отриманими даними табл.3 та побудованих за цими даними графіків  $E_{net}(S1)$  та  $R(S1)$ .

3.3.3. Графік апроксимації гармонічної функції перенавченою ШНМ, де  $S1$  взято з табл.2, та нормально навченою ШНМ, де  $S1=5$ , а також графік з помилками апроксимації для обох мереж.

3.3.4. Складені у SIMULINKу моделі 2-х варіантів САР: 1) з ПІД регулятором швидкості ЕМО; 2) з НПІД регулятором швидкості ЕМС.

3.3.5. Графіки перехідних процесів при відпрацюванні зміни завдання та компенсуванні накиду навантаження для обох варіантів САР швидкості.

3.3.6. Висновки з роботи.

### Контрольні питання

1. Нарисуйте структуру НМ типу 2-3-1.
2. Нарисуйте схему штучного нейрона та поясніть його роботу.
3. Назвіть основні типи функцій активації нейронів.
4. Які типи НМ вам відомі?
5. Які принципи та правила навчання НМ ви знаєте?
6. Яким методом навчають НМ прямого поширення?
7. Поясніть у чому полягає здатність НМ до узагальнення?
8. Як можна перевірити якість навчання мережі?
9. Що таке перенавчання мережі та як його можна уникнути?
10. Як отримати динамічну систему на основі НМ прямого поширення?

## Лабораторна робота №2

**Апроксимація складних функцій з використанням штучний нейронних мереж**

Мета роботи: Навчитися використовувати нейронні мережі для апроксимації функції.

Завдання: У середовищі Matlab необхідно побудувати і навчити нейронну мережу для апроксимації таблично заданої функції,  $i = 1, 20$ . Розробити програму, яка реалізує нейромережевий алгоритм апроксимації і виводить результати апроксимації у вигляді графіків.

## Теоретичні відомості

Поняття штучного нейрона і штучних нейронних мереж. Під штучними нейронними мережами ШНМ маються на увазі обчислювальні структури, які моделюють прості біологічні процеси, зазвичай асоціюються з процесами людського мозку. Вони являють собою розподілені і паралельні системи, здатні до адаптивного навчання шляхом аналізу позитивних і негативних впливів. Елементарним перетворювачем в даних мережах є штучний нейрон або просто нейрон, названий так за аналогією з біологічним прототипом. До теперішнього часу запропоновано і вивчено велику кількість моделей нейроноподібних елементів і нейронних мереж.

Нейрон є складовою частиною нейронної мережі. Загальна структура нейрона наведена на рис. 4.1.

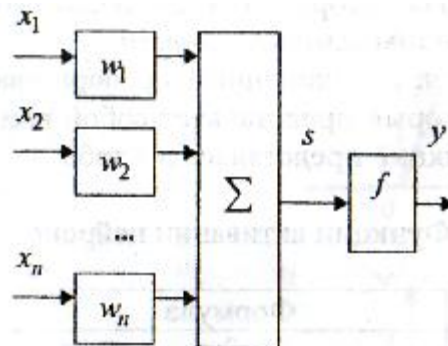


Рис.4.1.

Він складається з елементів трьох типів: помножувачів (синапсів), суматора і нелінійного перетворювача. Синапси здійснюють зв'язок між нейронами, множать вхідний сигнал на число, що характеризує силу зв'язку (вага синапсу). Суматор виконує додавання сигналів, що надходять по синаптичних зв'язків від інших нейронів, і зовнішніх вхідних сигналів. Нелінійний перетворювач реалізує нелінійну функцію одного аргументу - виходу суматора. Ця функція називається функцією активації або передавальною функцією нейрона. Нейрон в цілому реалізує скалярну функцію векторного аргументу. Математична модель нейрона наведена на рисунку 4.2.

$$S = \sum_{i=1}^n w_i x_i + b,$$

$$y = f(S),$$

Рис 4.2.

де  $S$  - результат підсумовування (sum);  $w_i$  - вага (weight) синапсу,  $i = 1, n$ ;  $x$  - компонент вхідного вектора (вхідний сигнал),  $i = 1, n$ ;  $b$  - значення зміщення (bias);  $n$  -

число входів нейрона;  $y$  - вихідний сигнал нейрона;  $f$  - нелінійне перетворення (функція активації).

У загальному випадку вхідний сигнал, вагові коефіцієнти і зміщення можуть приймати дійсні значення, а в багатьох практичних завданнях - лише деякі фіксовані значення. Вихід у визначається видом функції активації і може бути як дійсним, так і цілим.

Синаптичні зв'язки з позитивною вагою називають збудливими, з негативними - гальмують. Описаний обчислювальний елемент можна вважати спрощеною математичною моделлю біологічних нейронів. Щоб підкреслити відмінність нейронів біологічних і штучних, другі іноді називають нейроподібними елементами, або формальними нейронами.

На вхідний сигнал  $S$  нелінійний перетворювач відповідає вихідним сигналом  $f(S)$ , який представляє собою вихід у нейрона. Приклади активаційних функцій наведено на рисунку 4.3.

Название	Формула	Область значений
1	2	3
Линейная	$f(s) = ks$	$(-\infty, \infty)$
Полулинейная	$f(s) = \begin{cases} ks, & s > 0, \\ 0, & s \leq 0 \end{cases}$	$(0, \infty)$
Логистическая (сигмоидальная)	$f(s) = \frac{1}{1 + e^{-as}}$	$(0, \infty)$
Гиперболический тангенс (сигмоидальная)	$th(s) = \frac{e^{as} - e^{-as}}{e^{as} + e^{-as}}$	$(-1, 1)$
Экспоненциальная	$f(s) = e^{-as}$	$(0, \infty)$
Синусоидальная	$f(s) = \sin(s)$	$(-1, 1)$

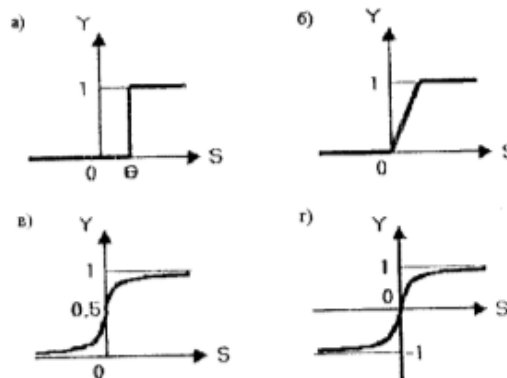


Рис.4.3.

а) функція одиничного стрибка; б) лінійний поріг (гістерезис); в) сигмоїд (логістична функція); г) сигмоїд (гіперболічний тангенс).

Однією з найбільш поширених є нелінійна функція активації, так звана логістична функція, або сигмоїд (функція  $S$  - образного виду):

$$f(s) = \frac{1}{1 + e^{-as}}.$$

(4.1.)

При зменшенні  $a$  сигмоїд стає більш пологим, в межі при  $a = 0$  вироджуючись в горизонтальну лінію на рівні 0,5, при збільшенні  $a$  сигмоїд наближається до виду одиничного стрибка з порогом  $\theta$ . Очевидно, що вихідне значення нейрона лежить в діапазоні  $(0,1)$ . Одна з цінних властивостей сигмоїдальної функції - простий вираз її похідної:



$$f'(s) = a \cdot f(s)[1 - f(s)]. \quad (4.2.)$$

### Порядок виконання роботи

4.1. У середовищі Matlab побудувати і навчити нейронну мережу для апроксимації таблично заданої функції  $y_i = f(x_i) = [2.09 \ 2.05 \ 2.19 \ 2.18 \ 2.17 \ 2.27 \ 2.58 \ 2.73 \ 2.82 \ 3.04 \ 3.03 \ 3.45 \ 3.62 \ 3.85 \ 4.19 \ 4.45 \ 4.89 \ 5.06 \ 5.63 \ 5.91]$ ,  $i = 1, 20$ .

Для цього ц математичної середовищі Matlab створюємо новий M-File, в якому записуємо код програми створення і навчання нейронної мережі з використанням вбудованих функцій пакету мереж Neural Networks Toolbox.

Для вирішення скористаємося функцією newff (.) - створення «класичної» багатошарової НС з навчанням за методом зворотного поширення помилки.

Код програми:

```
P = zeros (1,20);
for i = 1:20% створення масиву
P (i) = i * 0.1; % Вхідні дані (аргумент)
end
T = [2.09 2.05 2.19 2.18 2.17 2.27 2.58 2.73 2.82 3.04 3.03 3.45 3.62 3.85 4.19 4.45 4.89
5.06 5.63 5.91]; % Вхідні дані (значення функції)
net = newff ([0.1 2], [5 1], { 'tansig' 'purelin'}); % Створення нейронної мережі
net.trainParam.epochs = 100; % Завдання числа епох навчання
net = train (net, P, T); % Навчання мережі
y = sim (net, P); % Опитування навченої мережі
figure (1);
hold on;
xlabel ( 'P');
ylabel ( 'T');
plot (P, T, P, y, 'o'), grid; % Промальовування графіка вихідних даних і функції,
сформованою нейронною мережею
```

Результат роботи нейронної мережі привести на графіках: графік процесу навчання і результат апроксимації заданої послідовності.

4.2. Змінити кількість нейронів 1 шару в більшу і меншу сторону на одиницю. Зняти графіки процесу навчання і результату апроксимації заданої послідовності.

4.3. Збільшити кількість шарів базової компонування мережі. Зняти графіки процесу навчання і результату апроксимації заданої послідовності.

4.4. Поміняти місцями (використовувати інші) функції активації (аксонів) в шарах базової компонування мережі. Зняти графіки процесу навчання і результату апроксимації заданої послідовності.

4.5. Побудувати графіки певних функцій активації.

4.6. Зробити висновки про тривалості процесу навчання і як апроксимації нейронної мережі.

### Контрольні питання

1. Зобразить структуру простого нейрона (персептрона)?
2. Що таке функція активації нейрона та за якими принципами можна її обрати?
3. Яка роль вагових коефіцієнтів при кожному нейроні?
4. В чому полягає метод зворотного поширення?

5. Що таке епоха навчання і як вона визначає процес навчання?
6. Які існують умови завершення процесу навчання?



## Додаток 1.

У середовищі MatLab використовуються наступні функції для створення нейронної мережі

Функції створення нової мережі.

\* Network - створення нейронної мережі користувача.

запис:

```
net = network
```

```
net == nework (numInputs, numLayers, biasConnect, inputConnect, layerConnect, outputConnect, targetConnect)
```

Опис. Функція повертає створену нейронну мережу з ім'ям net і з наступними характеристиками (в дужках дані значення за замовчуванням):

numInputs - кількість входів (0)

numLayers - кількість шарів (0),

biasConnect - булевський вектор з числом елементів, що дорівнює кількості шарів (нулі),

inputConnect - булевська матриця з числом рядків, що дорівнює кількості шарів і числом рядків, що дорівнює кількості входів (нулі),

layerConnect - булевська матриця з числом рядків і стовпців, що дорівнює кількості шарів (нулі),

outputConnect - булевський вектор - рядок з числом елементів, що дорівнює кількості шарів (нулі),

targetConnect - вектор - рядок, така ж як попередня.

\* Newc - створення конкурентного шару

```
net = newc (PR, S, KLR, CLR) - функція створення шару Кохонена.
```

Аргументи функції:

PR - R x 2 матриця мінімальних і максимальних значень для R вхідних елементів,

S - число нейронів,

KLR - коефіцієнт навчання Кохонена (за замовчуванням 0.01)

CLR - коефіцієнт «справедливості» (за замовчуванням 0.001).

\* Newcf - створення каскадних спрямованої мережі

```
net = newcf (PR, [S1 S2 ... SNI], {TF1 TF2 ... TFNI}, BTF, BLF, PF) - функція створення різновиду багатошарової нейронної мережі зі зворотним поширенням помилки - так званої каскадної нейронної мережі. Така мережа містить прихованих NI шарів, використовує вхідні функції типу dotprod і netsum, ініціалізація мережі здійснюється функцією initnw.
```

Аргументи функції:

PR - R x 2 матриця мінімальних і максимальних значень для R вхідних елементів,

Si - розмір i - го прихованого шару, для NI шарів,

TFi - функція активації нейронів i - го шару, за замовчуванням 'tansig',

BTF - функція навчання мережі, по умовчанням 'trained',

BLF - функція настройки ваг і зсуву, за замовчуванням 'learnngdm',

PF - функція помилки, за замовчуванням 'mse'.

\* Newelm - створення мережі зворотного поширення Елмана (Elman)

```
net = newelm (PR, [S1 S2 ... SNI], {TF1 TF2 ... TFN1}, BTF, BLF, PF) - функція створення мережі Елмана. Аргументи такі ж, як і у попередньої функції.
```

\* Newff - створення односпрямованої мережі  
net = newff (PR, [SI S2 ... SNI], {TFI TF2 ... TFNI}, BTF, BLF, PF) - функція створення «класичної» багатошарової нейронної мережі з навчанням за методом зворотного поширення помилки.

\* Newfftd - створення односпрямованої мережі з вхідними затримками  
net = newfftd (PR, ID, [SI S2 ... SNI], {TFI TF2 ... TFNI}, BTF, BLF, PF) - те саме, що і попередня функція, але з наявністю затримок по входах. Додатковий аргумент ID - вектор вхідних затримок.