

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування
імені адмірала Макарова

В. С. БЛІНЦОВ, Г. С. ГРУДІНІНА

**ІНТЕЛЕКТУАЛЬНЕ КЕРУВАННЯ
ЗАСОБАМИ МОРСЬКОЇ РОБОТОТЕХНІКИ**

Методичні вказівки
до виконання лабораторних робіт

Рекомендовано Методичною радою НУК



ВИДАВНИЦТВО
НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ
КОРАБЛЕБУДУВАННЯ
ІМ. АДМІРАЛА МАКАРОВА

2021

УДК 004.8:621.313(076)

Б69

Автори: В. С. Блінцов, доктор технічних наук, професор, проректор НУК з наукової роботи, віцепрезидент Академії наук суднобудування України, академік Міжнародної академії морських наук, технологій і інновацій, заслужений діяч науки й техніки України, лауреат Державної премії України в галузі науки й техніки;

Г. С. Грудініна, викладач кафедри електричної інженерії судових та роботизованих комплексів

Рецензент В. Л. Тимченко, доктор технічних наук, професор кафедри Морського приладобудування

Рекомендовано Методичною радою НУК

Блінцов В. С.

Б69 Інтелектуальне керування засобами морської робототехніки : методичні вказівки до виконання лабораторних робіт / В. С. Блінцов, Г. С. Грудініна. – Миколаїв : НУК, 2021. – 44 с.

Вміщено матеріал необхідний для вивчення студентами теорії автоматичного керування. Подано приклад розв'язання поставлених задач за розробленою методикою з використанням інструменту для синтезу й аналізу нейронних мереж – Neural Network Toolbox та інструменту для побудови й аналізу нечітких множин – Fuzzy Logic Toolbox. Отримано графіки перехідних процесів за якими можна робити висновки про якість регулювання.

Призначено для студентів за напрямом підготовки 14 «Електрична інженерія».

004.8:621.313(076)

© Блінцов В. С., Грудініна Г. С., 2021

© Національний університет кораблебудування імені адмірала Макарова, 2021

ЗМІСТ

1. Загальні вимоги та вказівки до виконання лабораторних робіт	4
2. Завдання для лабораторних занять	6
Лабораторна робота № 1	6
Лабораторна робота № 2	10
Лабораторна робота № 3	17
Лабораторна робота № 4	21
Лабораторна робота № 5	26
Лабораторна робота № 6	34
Лабораторна робота № 7	37
Рекомендована література	42

1. ЗАГАЛЬНІ ВИМОГИ ТА ВКАЗІВКИ ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ

До виконання лабораторних робіт кожен студент повинен пройти інструктаж з техніки безпеки, про що робиться відповідний запис у журналі, та вивчити інструкції користувача комп'ютерних комплексів. *Виконання правил техніки безпеки є обов'язковим.*

Лабораторні роботи виконуються студентами індивідуально або по бригадам згідно з поставленим завданням.

Виконання лабораторних робіт передбачає домашню підготовку, яка включає в себе вивчення теоретичних відомостей необхідних для виконання лабораторної роботи, виконання рекомендацій індивідуально вказаних для кожної роботи, підготовку робочих шаблонів у разі необхідності.

Детально зміст попередньої підготовки наводиться в описанні кожної лабораторної роботи. На початку заняття викладач перевіряє наявність попередньої теоретичної та практичної підготовки.

Непідготовлені студенти до роботи не допускаються.

Під час лабораторної роботи студенти проводять дослідження динамічних характеристик об'єктів першого та другого порядку, а також розробляють інтелектуальні системи керування електродвигуном постійного струму незалежного збудження та автономним підводним апаратом. У роботах використовується інструмент для синтезу й аналізу нейронних мереж – Neural Network Toolbox та інструмент для побудови й аналізу нечітких множин – Fuzzy Logic Toolbox.

Результати досліджень перевіряються, підписуються викладачем та додаються до звіту. За результатами роботи кожним студентом оформлюється звіт, який має містити наступні розділи:

- назву роботи;
- мету;
- короткі теоретичні відомості, які опановано й використано під час проведення роботи;
- конкретне завдання згідно з варіантом та порядок виконання роботи;
- результати моделювання у вигляді графіків відповідних залежностей або результати виконання поставленої задачі з відповідними поясненнями й аналізом;
- висновки.

Звіт оформлюється на аркушах формату А4 з рамками. Штмп на кожному аркуші необов'язковий. Титульний аркуш містить штмп висотою 55 мм, де вказується: назва роботи, номер спеціальності, номер групи та номер лабораторної роботи, прізвище студента та викладача, дата виконання та здавання.

Висновки мають містити аналіз результатів отриманих у процесі моделювання та виконання поставленої задачі.

Звіт з роботи подається викладачу на перевірку перед наступним заняттям. Якщо робота виконана з помилками, чи оформлення звіту не відповідає вимогам, він повертається на доробку. Захист звітів лабораторних робіт може відбуватися як у процесі занять, так і на додаткових консультаціях або на спеціально відведеному занятті. Умовою для зарахування роботи є наявність правильного звіту і спроможність студента відповісти на питання викладача за тематикою даної роботи. Якість відповіді під час захисту звіту може оцінюватися викладачем і враховуватися під час екзамену з дисципліни.

2. ЗАВДАННЯ ДЛЯ ЛАБОРАТОРНИХ ЗАНЯТЬ

Лабораторна робота № 1

Тема. Розробка пропорційно-інтегрального регулятора для об'єкта керування першого порядку (апериодична ланка).

Мета роботи: ознайомитися з пакетом математичних програм Matlab. Вивчити основні поняття теорії управління.

Короткі теоретичні відомості

Показники якості керування. Якість керування, що забезпечується системами регулювання, визначається поведінкою системи як в усталеному режимі, так і під час перехідного процесу (рис. 1.1).

Основні показники перехідного процесу: перерегулювання

$$y = \frac{y_{\max} - y_0}{y_0} \cdot 100 \% .$$

Тривалість перехідного процесу T_0 – час з моменту виникнення стрибка задавального впливу, після якого відхилення регульованої величини не перевищує заданого припустимого значення Δe (рис. 1.1).

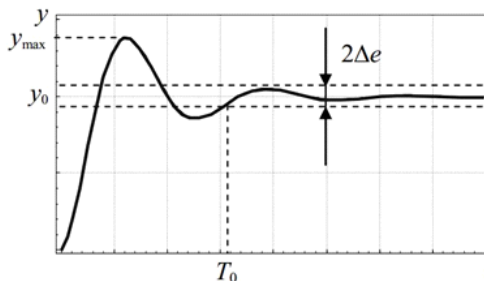


Рис. 1.1. Характеристики перехідного процесу

Якість керування в усталеному режимі визначається інтегральними оцінками, які дають змогу характеризувати перехідний процес у системі одним числом. На практиці найбільшого поширення набули такі інтегральні оцінки:

квадратична інтегральна оцінка

$$I = \int_0^{+\infty} e(t)^2 dt,$$

де $e(t) = x(t) - y(t)$ – похибка регулювання;

поліпшена квадратична інтегральна оцінка

$$I = \int_0^{+\infty} \left(e(t)^2 + T^2 \left[\frac{d}{dt} e(t) \right]^2 \right) dt.$$

Перша із вказаних квадратичних оцінок тим менша, чим перехідний процес у розглядуваній системі ближчий до задавального впливу, а друга – чим він ближчий до перехідного процесу на виході інерційної ланки зі сталою часу T .

Хід роботи

Створимо модель одноконтурної системи автоматичного регулювання на основі ПІ-регулятора (рис. 1.2).

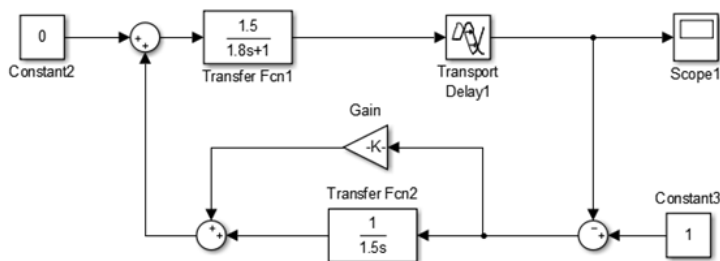


Рис. 1.2. Структурна схема моделі одноконтурної системи автоматичного регулювання на основі ПІ-регулятора

Запускаємо Matlab, у командному рядку обираємо вкладку Simulink library. Створюємо нову модель – «New Model», зберігаємо її в необхідну папку.

У Simulink, у правій колонці подано набір інструментів необхідних для отримання математичної моделі заданої системи керування. У вкладці «math operations» оберемо необхідні блоки та перетягнемо їх у вікно із нашою моделлю.

У вкладці «sources» знаходяться блоки необхідні для задання початкових умов та інших впливів (джерела). Також перетягніть необхідний блок до своєї моделі.

Подібним чином зберіть задану на рис. 1.2 модель.

Виконаємо налаштування заданої моделі. Як об'єкт управління обираємо аперіодичну ланку з передатною функцією (для першого варіанта, для всіх інших за табл. 1.1):

$$H(p) = \frac{k}{Tp + 1} = \frac{1,5}{1,8p + 1}.$$

Таблиця 1.1. Параметри об'єкта керування за варіантом

Варіант		1	2	3	4	5	6	7	8
k		1,5	1,8	1,3	1,9	2,0	2,2	1,4	2,5
T		1,8	1,8	1,5	1,5	1,3	1,5	1,5	1,6
9	10	11	12	13	14	15	16	17	18
2,3	2,4	3,5	3,8	3,3	3,9	3,0	3,2	3,4	4,0
1,4	1,7	1,8	1,5	1,5	1,5	1,8	1,2	1,4	1,9

Налаштуємо регулятор таким чином, щоб перехідний процес відповідав заданим критеріям регулювання. Налаштування ПІ-регулятора визначається за формульним методом (табл. 1.2), де K_1 , K_0 – налаштування пропорційної та інтегральної частин ПІ-регулятора відповідно.

Таблиця 1.2. Параметри налаштування ПІ-регулятора

Регулятор	Аперіодичний процес	Процес з перерегулюванням 20 %	Процес з мінімальним часом регулювання
ПІ	$K_1 = \frac{0,6 \cdot T}{k \cdot \tau}$ $K_0 = \frac{1}{k \cdot \tau}$	$K_1 = \frac{0,7 \cdot T}{k \cdot \tau}$ $K_0 = \frac{1}{k \cdot \tau}$	$K_1 = \frac{T}{k \cdot \tau}$ $K_0 = \frac{1}{k \cdot \tau}$

У верхній частині вікна моделі встановимо час та проведемо моделювання роботи одноконтурної системи автоматичного регулювання на основі ПІ-регулятора, натисканням кнопки «Run».

Проаналізуємо якість отриманого перехідного процесу. На рис. 1.3 поданий перехідний процес нашої системи, отриманий за допомогою блоку «Scope».

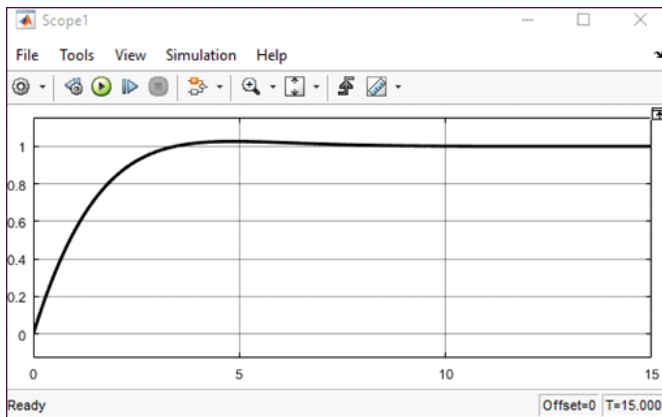


Рис. 1.3. Побудова перехідного процесу системи автоматичного регулювання на основі ПІ-регулятора

Змінюючи параметри ПІ-регулятора (K_0 , K_1), отримайте перехідні процеси вказані в табл. 1.2. Проаналізуйте отримані результати та зробіть висновки.

Зміст звіту

1. Тема та мета роботи.
2. Структурна схема отриманої системи автоматичного регулювання.
3. Рівняння та параметри налаштування ПІ-регулятора.
4. Графіки перехідних процесів.
5. Аналіз якості перехідних процесів.
6. Висновки.

Контрольні питання

1. Що таке перехідний процес?
2. Принцип дії ПІ-регулятора.
3. У чому суть інтелектуального керування?

Лабораторна робота № 2

Тема. Синтез ПІ-подібного регулятора на основі нечіткої логіки для об'єкта керування першого порядку.

Мета роботи: вивчити основні поняття теорії нечітких множин та їх додатків. Ознайомитися з пакетом математичних додатків Matlab.

Завдання

– Визначити найкращі параметри налаштування ПІ-подібного регулятора розробленого в попередній роботі (перегулювання $< 5\%$).

– За допомогою інструментів графічного інтерфейсу користувача (GUI) пакета «Fuzzy Logic Toolbox» створити нечітку систему, що реалізовує типовий аналоговий ПІ-регулятор.

– Визначити входні змінні для визначення умов регулювання та стабілізації (x_1 , x_2).

- Виконати лінгвістичний опис кожної змінної у вигляді семи трикутних термів.
- Визначити та виконати лінгвістичний опис вихідної змінної (керуючий вплив) у.
- Визначити функцію приналежності та скласти таблицю з отриманих правил.
- Реалізувати систему керування об'єктом першого порядку на базі нечіткого регулятора за допомогою Matlab.

Хід роботи

Запустити Matlab.

Зауважимо, що за допомогою пакета «Fuzzy Logic Toolbox» можна будувати нечіткі системи двох типів – Мамдані і Сугено. Зупинимося на системі типу Мамдані.

Командою «fuzzy» у вікні Matlab викликаємо вікно редактора фазі-інференційної системи, вибираємо тип системи – Мамдані (рис. 2.1).

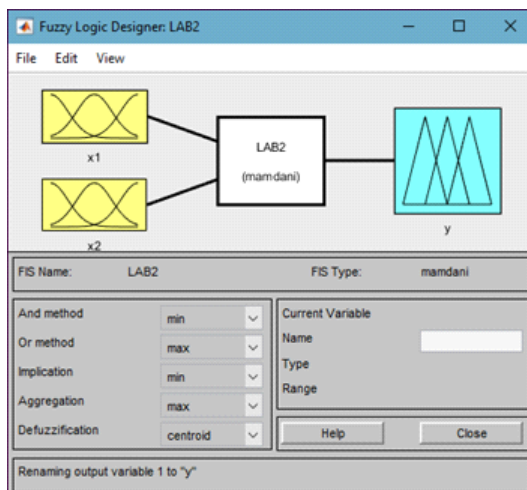


Рис. 2.1. Головне вікно редактора нечіткого регулятора

На панелі керування натискаємо «Edit» і обираємо вкладку «Add variable» та додаємо ще один вхід. Задаємо два входи – для пропорційної та інтегральної складових і називаємо вхідні змінні, наприклад, x_1 і x_2 , а вихідну – y (рис. 2.2).

У теорії нечітких множин змінні описуються лінгвістичними термами. Для лінгвістичного опису кожної змінної виберемо сім трикутних термів (NB, NM, NS, ZE, PS, PM, PB).

Зайдемо у вікно редактора першої змінної (рис. 2.2). Натиснувши «Edit», обираємо вкладку «Add MFs» та вказуємо необхідну кількість термів (у даному випадку сім).

У цьому ж вікні задаємо діапазони зміни змінних: для вхідних змінних регулятора рекомендуються симетричні діапазони зміни, при цьому $x_1 \in [-1/K_1; 1/K_1]$, $x_2 \in [-1/K_0; 1/K_0]$, де K_1, K_0 – оптимальні настройки пропорційної та інтегруючої частин ПІ-регулятора. Далі задамо форму термів «Type» – трикутна «trimf». Додамо назву кожного терма «Name» – «NB» (рис. 2.2).

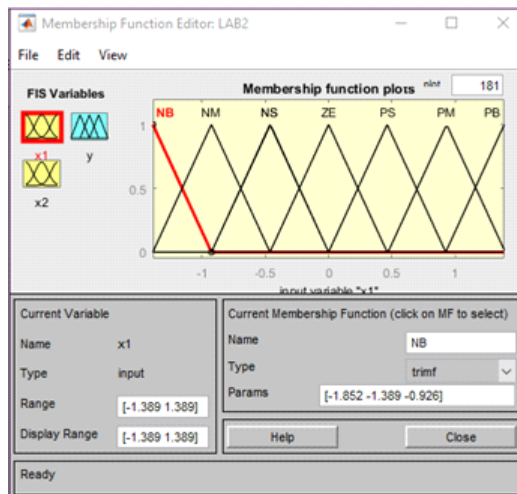


Рис. 2.2. Вікно редактора змінних нечіткої системи

Таким чином задамо другу змінну та вихідну. Терми вихідної змінної краще вибирати такі, щоб не перетиналися (рис. 2.3). Це підвищить чіткість регулювання.

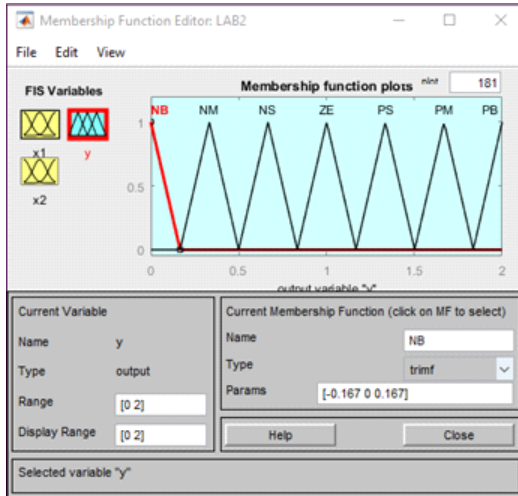


Рис. 2.3. Вікно редактора вихідної множини нечіткого регулятора

Для вихідної змінної регулятора діапазон зміни рекомендується брати у вигляді, де верхня межа C при одиничному ступінчастому впливі варіюється від 1,1 до 2, щоб вихідний сигнал регулятора міг компенсувати це збурення.

У міру збільшення значення C зменшується динамічна помилка, але зростають час регулювання та число коливань перехідного процесу. Тому рекомендується C брати рівним 2, коли спостерігається оптимальне співвідношення між величиною динамічної помилки, часу регулювання і кількістю коливань.

Наступним кроком необхідно сформувати базу правил нечіткого регулятора. В основу покладено спосіб, запропонований у літературі.

Лінійний безперервний ПІ-регулятор з диференціальним рівнянням

$$y(t) = k_D \cdot \varepsilon(t) + \frac{1}{T_E} \int_0^t \varepsilon(t) \cdot dt \quad (1)$$

можна замінити близьким за стратегією і логікою керування fuzzy-регулятором, якщо в якості його вихідної змінної розглядати прирощення керуючого впливу. Тоді закон ПІ-регулювання можна подати в наступній диференціальній формі:

$$\frac{dy(t)}{dt} = k_D \cdot \frac{d\varepsilon(t)}{dt} + \frac{1}{T_E} \cdot \varepsilon(t) \quad (2)$$

або у формі різниці (3):

$$\Delta y(k) = y(k) - y(k-1) = k_D \cdot \Delta \varepsilon(k) + \frac{\Delta t}{T_E} \cdot \varepsilon(k). \quad (3)$$

Таким чином, для вхідних змінних $\varepsilon(k)$ і $\Delta \varepsilon(k)$ та вихідної $\Delta y(k)$ може бути синтезований fuzzy-регулятор, який реалізує нелінійний закон (4)

$$\Delta y(k) = F[\Delta \varepsilon(k), \varepsilon(k)] \quad (4)$$

і еквівалентний у певному сенсі ПІ-регулятору.

Для нашого випадку x_1 відповідає сигналу неузгодженості $\varepsilon(k)$, x_2 відповідає збільшенню сигналу неузгодженості, а y відповідає $\Delta y(k)$.

Лінгвістичні правила для такого ПІ-подібного fuzzy-регулятора наведені в табл. 2.1.

Таблиця. 2.1. Лінгвістичні правила для ПІ-подібного fuzzy-регулятора

	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NB	NM	NS	ZE	PS
NS	NB	NB	NM	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PM	PB	PB
PM	NS	ZE	PS	PM	PB	PB	PB
PB	ZE	PS	PM	PB	PB	PB	PB

Щоб сформувати базу правил, відкрийте вікно редактора бази даних «Edit» – «Rule Editor» (рис. 2.4). Вихідна величина y формується як результат перетину (AND) двох вхідних величин x_1, x_2 . IF x_1 is NB AND x_2 is NB THEN y is NB.

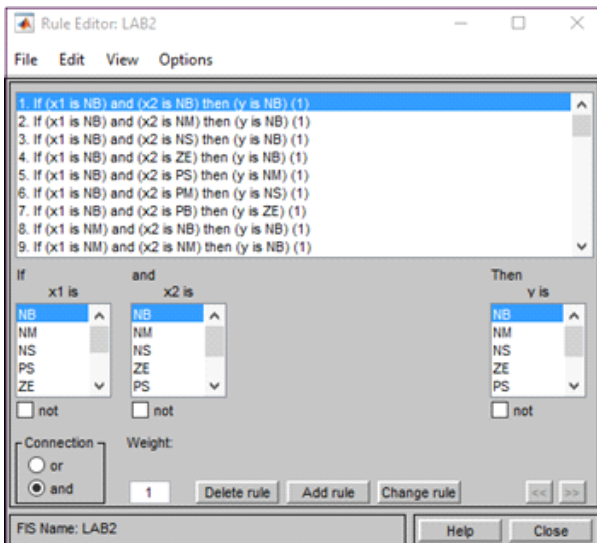


Рис. 2.4. Вікно редактора бази правил

Для збереження створеного регулятора натисніть «File» та оберіть вкладку «export to file» та «export to workspace».

Створимо модель одноконтурної системи автоматичного керування з ПІ-подібним fuzzy-регулятором (рис. 2.5).

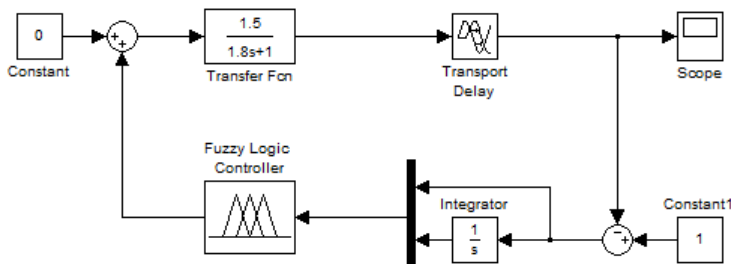


Рис. 2.5. Структурна схема моделі одноконтурної системи автоматичного регулювання на основі ПІ-подібного fuzzy-регулятора

Для додавання нечіткого регулятора до структурної схеми в Simulink оберіть вкладку Fuzzy logic toolbox та виберіть елемент Fuzzy logic controller, у ньому пропишіть ім'я файлу зі створеним нечітким регулятором.

Порівнюємо результати роботи створених систем (рис. 2.6).

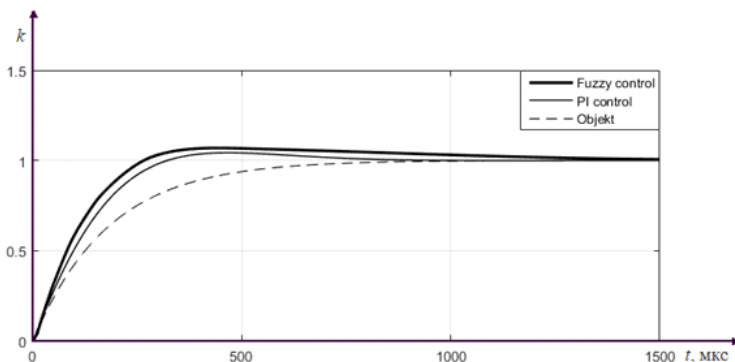


Рис. 2.6. Порівняння перехідних процесів створених систем автоматичного регулювання

Зміст звіту

1. Мета роботи.
2. Вхідні дані. Блок-схема модельованої системи, графіки перехідних процесів для аналогового і нечіткого регуляторів.
3. Параметри елементів нечіткої логіки проєктованої системи (тип функцій приналежності, спосіб реалізації логічних виразів, спосіб дефазифікації).
4. Аналіз якості перехідних процесів.
5. Висновки.

Контрольні питання

1. Що таке нечітка логіка? У яких додатках використовується?
2. Операції над нечіткими множинами.
3. Способи визначення нечіткої множини.
4. Функція приналежності. Типи функцій приналежності.

Лабораторна робота № 3

Тема. Синтез системи керування на базі нечіткої логіки для об'єкта другого порядку.

Мета роботи: закріпити навички роботи в пакеті Matlab. Практично застосувати теорії нечітких множин під час розробки регулятора стабілізації заданого параметра для об'єкта другого порядку з урахуванням похибки регулювання.

Завдання

– Визначити параметри необхідні для керування об'єктом другого порядку, що будуть використані як вхідні змінні для визначення умов регулювання та стабілізації (x_1, x_2).

– Виконати лінгвістичний опис кожної змінної у вигляді семи трикутних термів.

– Визначити та виконати лінгвістичний опис вихідної змінної (керуючий вплив) u . Терми вихідної змінної оберемо такими, що не перетинаються.

– Визначити функцію приналежності та скласти таблицю з отриманих правил.

– Реалізувати систему керування об'єктом другого порядку на базі нечіткого регулятора за допомогою Matlab.

Хід роботи

Запустимо Matlab.

За допомогою Simulink зібрати схему з системою керування на базі нечіткої логіки (рис. 3.1). Як об'єкт керування обемо об'єкт другого порядку (коливальна ланка) з передатною функцією:

$$H(p) = \frac{k}{T^2 p^2 + 2\xi Tp + 1} = \frac{0,1370}{1,4323 p^2 + 4,2 p + 1}.$$

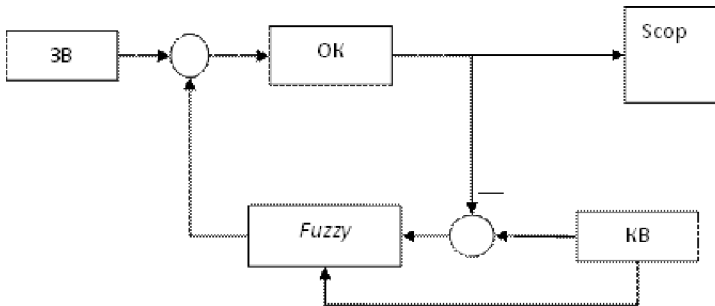


Рис. 3.1. Структурна схема системи керування:
ЗВ – зовнішній вплив; *ОК* – об'єкт керування; *KB* – керуючий вплив;
fuzzy – регулятор; *Scop* – блок отримання та збереження результатів

За допомогою пакета «Fuzzy Logic Toolbox» створимо нечітку систему, що реалізовує типовий аналоговий ПІ-регулятор. Для налаштування fuzzy-регулятора необхідно задати дві лінгвістичні функції результатом перетину яких (AND) буде вихідна функція.

Як керуючі впливи обираємо задану швидкість об'єкта та різницю між заданою та поточною швидкостями (помилку). Як вихідна функція – значення напруги, що необхідно подати на об'єкт:

$$x_1 = [-K; K]; \quad x_2 = [-K_1; K_1]; \quad y = [-U; U].$$

Функції складаються із 7 трикутних термів. Діапазони регулювання визначаються експериментальним шляхом у залежності від максимального значення вихідної змінної, визначається за варіантом ($U = N$ за списком). Для визначення значення K та K_1 необхідно провести експеримент, а саме: подати на об'єкт максимальне та мінімальне значення напруги та визначити діапазон швидкості. Знаючи максимальне та мінімальне значення швидкості, визначити діапазон можливої помилки (приклад, рис. 3.2).

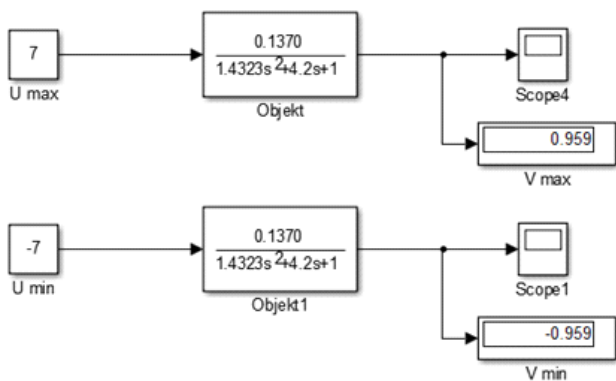


Рис. 3.2. Визначення діапазонів значень керуючих впливів x_1 та x_2 при $y_{\max} = 7$ В

Самостійно скласти таблицю з правилами перетину вхідних функцій.

Головний принцип регулювання полягає в прискоренні процесу руху, а саме: якщо помилка велика, то спочатку на об'єкт необхідно подавати максимальну напругу та зі зменшенням помилки знижувати напругу до необхідного значення.

Налаштувати регулятор згідно з отриманими даними.

Як результат необхідно навести графіки перехідного процесу роботи системи в декількох режимах (наприклад, для трьох різних швидкостей) (рис. 3.3, 3.4).

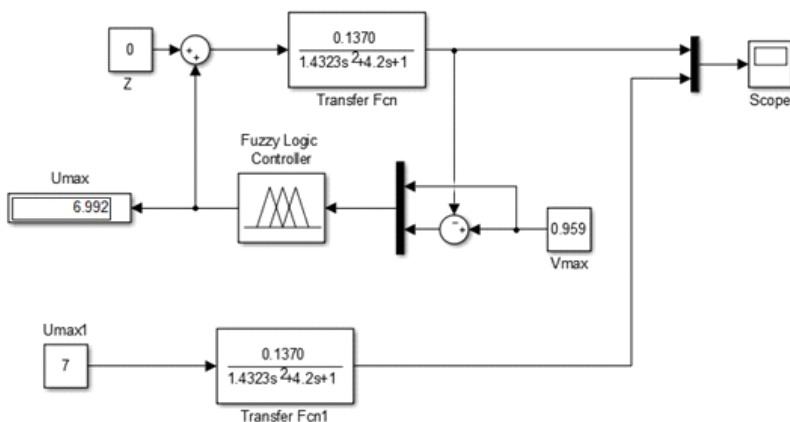


Рис. 3.3. Структурна схема fuzzy-регулятора швидкості:
 V – задане значення швидкості; U – напруга, обчислена регулятором;
 $U_{\max 1}$ – еталонне значення напруги

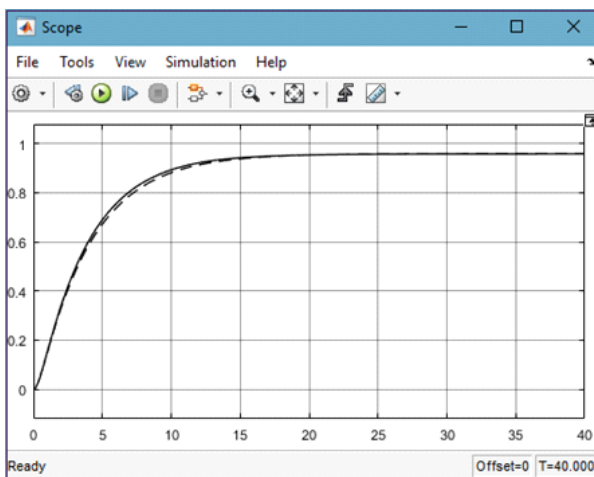


Рис. 3.4. Перевірка якості роботи розробленої системи

Контрольні питання

1. Параметри необхідні для керування об'єктом другого порядку.
2. Алгоритми нечіткого виводу. Алгоритм Мамдані.
3. Алгоритм Сугено.

Лабораторна робота № 4

Тема. Синтез стабілізуючої системи керування електродвигуном постійного струму з незалежним збудженням на базі нечіткої логіки з використанням програмного математичного пакета Matlab.

Мета роботи: отримати знання про роботу електродвигуна постійного струму за допомогою аналізу перехідного процесу при різних режимах роботи двигуна. Закріпити практичні навички під час роботи з інструментом для побудови й аналізу нечітких множин – Fuzzy Logic Toolbox.

Завдання

- Синтезувати систему керування електродвигуном постійного струму на базі нечіткої логіки за принципом ПІ-подібного регулятора.
- Визначити функцію приналежності та скласти таблицю з отриманих правил.
- За допомогою пакета «Fuzzy Logic Toolbox» створити нечітку систему керування, яка буде підтримувати постійні оберти двигуна, задані оператором, у разі зміни навантаження на валу.
- Провести моделювання роботи отриманої системи з урахуванням дії зовнішнього впливу. Отримати перехідні процеси для 2–3 значень зовнішнього впливу (наприклад, $TL = 1, 3, 5$).

Хід роботи

Створимо модель одноконтурної системи автоматичного регулювання на основі ПІ-подібного нечіткого регулятора. Як об'єкт управління обираємо електродвигун постійного струму з незалежним збудженням. Параметри обраного електродвигуна (рис. 4.1, 4.2):

HP – потужність двигуна (к. с.) – вибирається за варіантом;

U – напруга двигуна, В: 240;

$U_{\text{д}}$ – напруга збудження двигуна, В: 300;

n – частота обертання, об/хв: 1750.

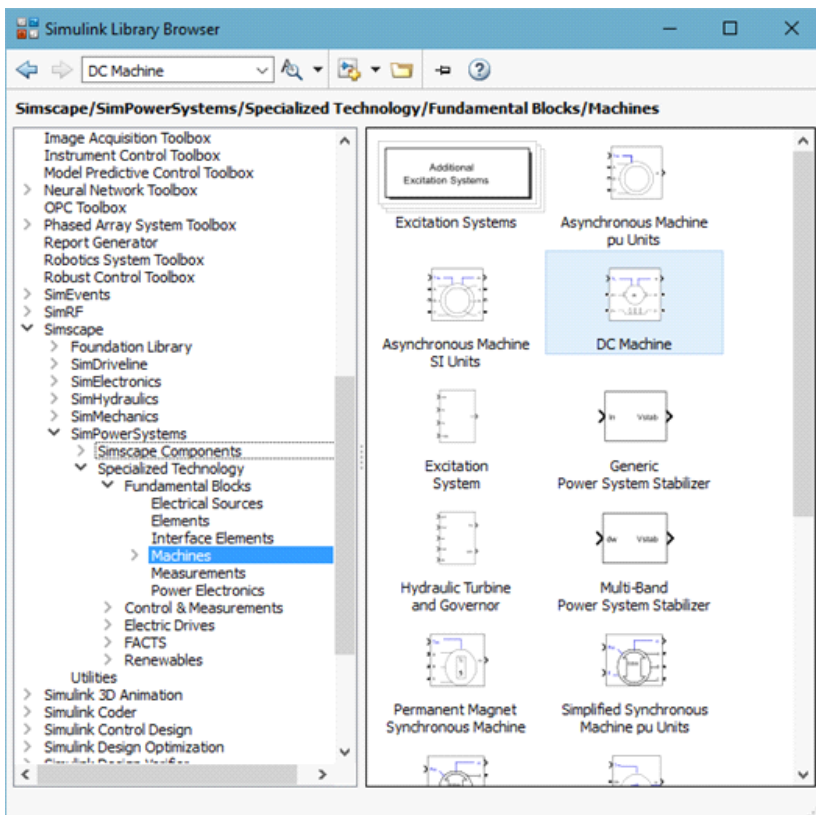


Рис. 4.1. Вікно вибору електродвигуна в системі Simulink

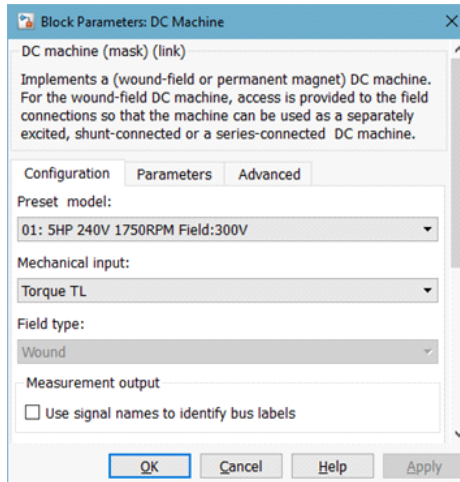


Рис. 4.2. Параметри двигуна постійного струму

Проаналізуємо робочі параметри обраного двигуна, проведемо моделювання та оцінимо якість отриманих перехідних процесів (частоту обертання, зміну сили струму та обертаючий момент) (рис. 4.3, 4.4).

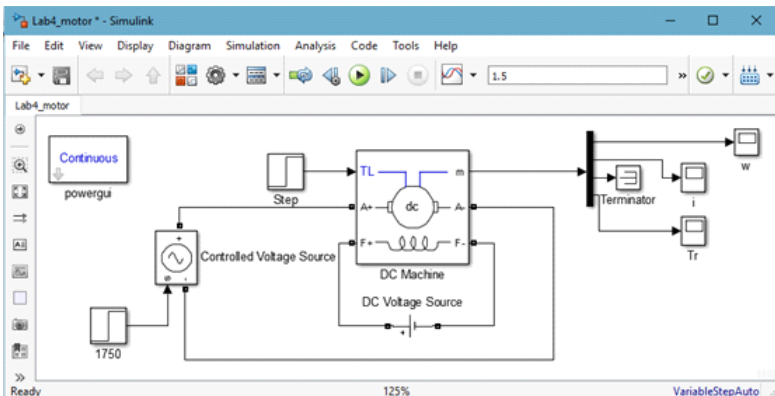


Рис. 4.3. Структурна схема для моделювання роботи електродвигуна постійного струму в системі Simulink



Рис. 4.4. Моделювання перехідного процесу частоти обертання двигуна в Simulink

За допомогою пакета «Fuzzy Logic Toolbox» створимо нечітку систему керування, яка буде підтримувати постійні оберти двигуна, задані оператором, у разі зміни навантаження на валу.

Нечіткий fuzzy-регулятор реалізовує типовий аналоговий ПІ-регулятор.

Визначимо границі регулювання даного об'єкта: x_1 – відповідає необхідній частоті обертання, визначається за варіантом, таблиця 1, x_2 – відповідає максимальній можливій похибці. Діапазон значень вихідної множини y – визначається експериментальним шляхом (таким чином, як у попередній роботі). Прийемо, що значення зовнішнього впливу $TL \in [0 \dots 5]$ Н·м. Необхідно визначити максимальне падіння частоти обертів двигуна при максимальному зовнішньому впливі. Для імітації дії сили зовнішнього впливу до створеної системи (показаної на рис. 4.3) необхідно додати блок «step» у вкладці «sources», налаштувати його так, щоб у деякий час підключалася сила зовнішнього впливу, підключити даний блок до входу TL на моделі двигуна.

1. Визначити значення обертів двигуна, необхідне для компенсації максимального зовнішнього впливу (табл. 4.1).

Таблиця 4.1.

№ вар	1	2	3	4	5	6	7	8
w , об/хв	220	240	300	320	360	400	420	460
9	10	11	12	13	14	15	16	17
500	550	600	650	700	750	800	1050	1750

2. Самостійно налаштувати нечіткий регулятор.

3. У Simulink реалізувати систему керування частотою обертів двигуна.

4. Провести моделювання роботи отриманої системи з урахуванням дії зовнішнього впливу. Отримати перехідні процеси для 2–3 значень зовнішнього впливу (наприклад, $TL = 1, 3, 5$).

5. Проаналізувати отриманий результат, зробити висновки.

Контрольні питання

1. Параметри електродвигуна постійного струму з незалежним збудженням. Які ще існують типи двигунів постійного струму?

2. Який вигляд мають перехідні процеси струму та моменту під час роботи двигуна? Укажіть перерегулювання та сталий режим роботи.

3. Загальні принципи побудови нечітких алгоритмів керування динамічними об'єктами.

Лабораторна робота № 5

Тема. Синтез інтелектуальної системи керування на базі нейронної мережі для електродвигуна постійного струму з незалежним збудженням.

Мета роботи: ознайомитися з поняттям штучної нейронної мережі. Набути практичних навичок використання

інструменту для синтезу та аналізу нейронних мереж – Neural Network Toolbox. Вивчити основні елементи управління даним інструментом.

Хід роботи

Як об'єкт керування обираємо двигун постійного струму з незалежним збудженням та наступними параметрами:

HP – потужність двигуна (к. с.) – за варіантом ($B1 = 5$);

U_d – напруга двигуна, В:240;

U_B – напруга збудження двигуна, В:300;

n – частота обертання, об/хв: 1750.

За допомогою Matlab розробимо систему керування заданим об'єктом за пропорційно-диференціальним законом керування, тобто $K_I = 0$. На рис. 5.1 подано структурну схему розробленої системи керування.

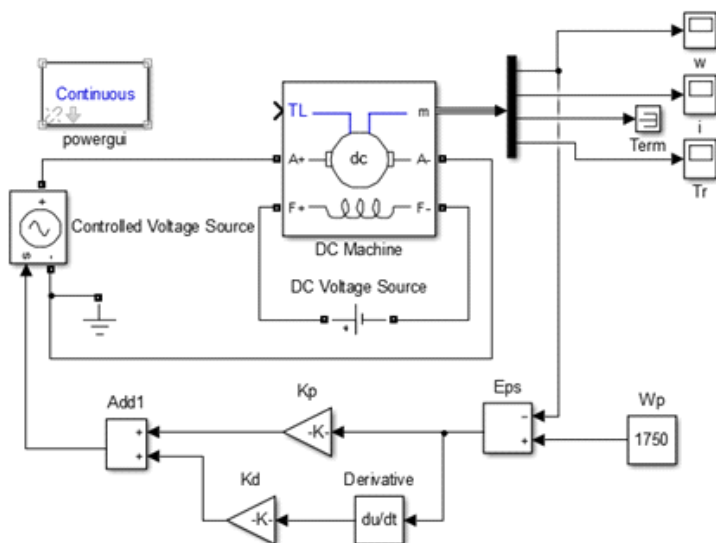


Рис. 5.1. Структурна схема моделі одноконтурної системи автоматичного керування з ПД-регулятором для електродвигуна постійного струму з незалежним збудженням

Налаштуємо отриманий регулятор за наступною методикою (метод Куна):

- нехай коефіцієнти K_p та K_d дорівнюють 0;
- поступово збільшуємо значення K_r до тих пір, поки система не вийде зі стану рівноваги й почнуться коливання, що не затухають;
- нехай K_u дорівнює отриманому значенню K_r , а T_u дорівнює періоду отриманих коливань, тоді

$$K_p = 0,6 \cdot K_u;$$

$$K_d = K_p \cdot T_u / 8.$$

На рис. 5.2 подано перехідний процес роботи налаштованого ПД-регулятора для заданого об'єкта керування. Для покращення якості регулювання параметри регулятора можуть бути відрегульовані в ручному режимі.

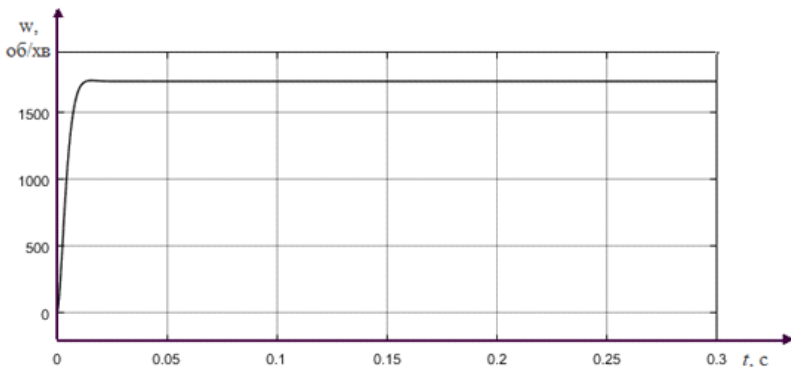


Рис. 5.2. Перехідний процес роботи налаштованого ПД-регулятора частотою обертів двигуна постійного струму

Для навчання нейронної мережі, який буде виконувати функцію ПД-регулятора необхідна вибірка даних у вигляді векторів значень коефіцієнтів підсилення пропорційної та дифе-

ренційної ланок $\overrightarrow{K_p}$ і $\overrightarrow{K_d}$ та вихідного вектора $\vec{y}$. Для отримання необхідної вибірки, як еталонну модель, оберемо розроблений вище ПД-регулятор.

У Simulink зберемо схему (рис. 5.3) та сформуємо вектор $V_{\text{вход}}$ для навчання нейронної мережі. У вкладці «sinks» оберемо елемент «To Workspace» та додамо його до схеми. У вікні налаштування елемента встановимо «save format» – «array» та надамо кожному блоку відповідне ім'я – K_d , K_p , y . Після моделювання в робочій області Matlab з'являться три масиви K_d , K_p , y .

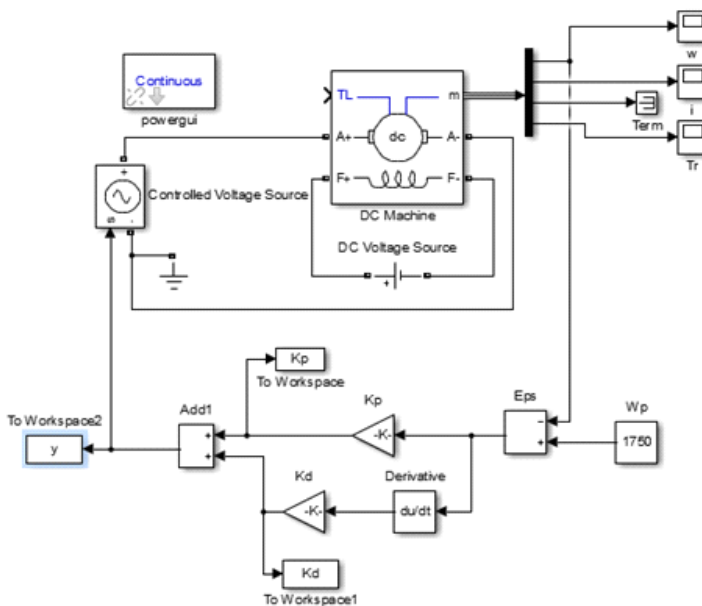


Рис. 5.3. Структурна схема необхідна для отримання масивів K_d , K_p , y

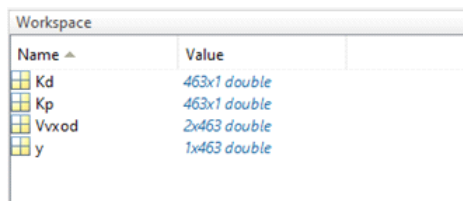
Далі з командної строки Matlab необхідно прописати наступний код:

```
Vvxod(1,:)=Kp;
```

```
Vvxod(2,:)=Kd;
```

```
y = y'
```

У результаті в робочій області Matlab з'явиться вектор Vvxod (рис. 5.4).



Name	Value
Kd	463x1 double
Kp	463x1 double
Vvxod	2x463 double
y	1x463 double

Рис. 5.4. Робочий вигляд вікна «Workspace» у Matlab

За допомогою команди `nntool` з командної строки Matlab створимо нову нейронну мережу та задамо необхідні параметри для навчання, а саме:

вектор Vvxod, що складається зі значень Kd, Kp ;

вектор у – значення керуючого впливу отриманого з еталонної моделі.

Натисканням кнопки «Import» додаємо вектор Vvxod у вікно «Input Data» та вектор у у вікно «Target Data» (рис. 5.5).

Далі натискаємо кнопку «New» та виконуємо налаштування нейронної мережі.

Тип мережі встановлюємо Feed-forward backpropagation, заповнюємо вікна «Input Data» та «Target Data». Обираємо нейронну мережу з двома прихованими шарами – «Number of layers» – 2. Перший шар містить 4 нейрони з функцією активації у вигляді тангенсальної сигмоїди: «Number of neurons» – 4, «Transfer Function» – «Tansig». Другий шар складається з одного нейрона та лінійної функції активації. Натискаємо кнопку «Create».

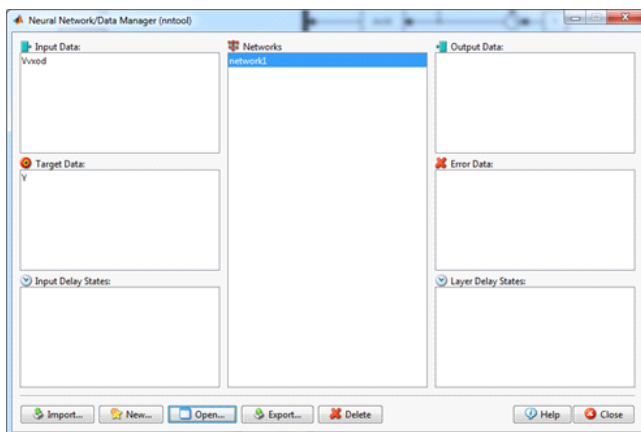


Рис. 5.5. Створення нейронної мережі в Simulink

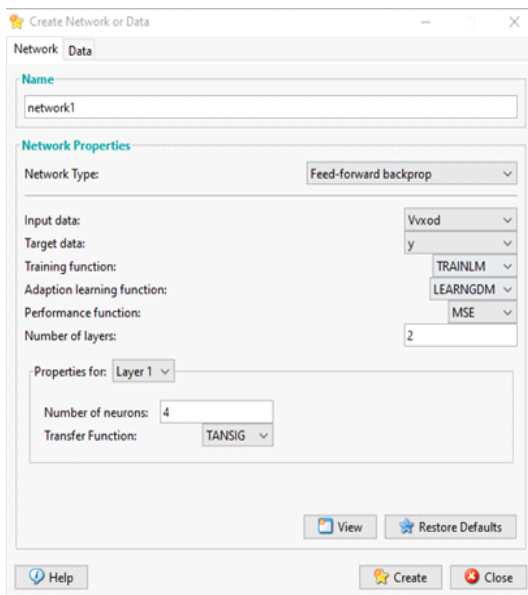


Рис. 5.6. Вікно налаштування штучної нейронної мережі в Simulink

На рис. 5.7 подано структурну схему отриманої нейронної мережі.

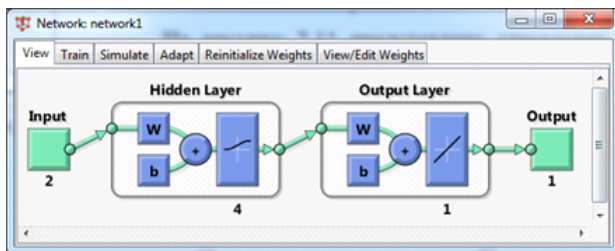


Рис. 5.7. Структурна схема створеної нейронної мережі

У вкладці Train устанавимо необхідні параметри та проведемо навчання мережі.

Установимо у вікна «Inputs» та «Targets» відповідні параметри. Натисканням кнопки «Train network» виконаємо навчання нейронної мережі (рис. 5.8).

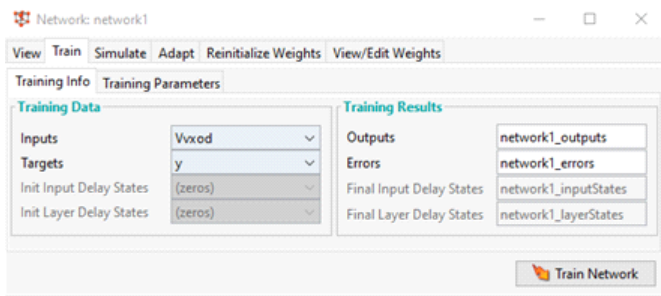


Рис. 5.8. Налаштування параметрів для навчання нейронної мережі

За допомогою команди «gensim(net, st)» з командної строки Matlab згенеруємо навчену нейронну мережу, де net – ім'я нейронмережі, st – за замовченням = 1.

Синтезуємо систему керування для заданого об'єкта управління на базі отриманої нейронної мережі методом безпосереднього включення НМ як регулятора в замкнутий контур керування об'єктом (рис. 5.9).

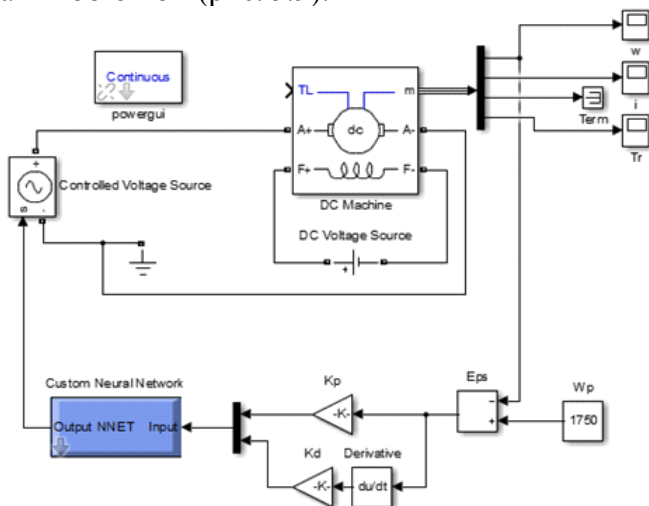


Рис. 5.9. Система керування з нейрорегулятором

На рис. 5.10 показано перехідний процес двигуна постійного струму з незалежним збудженням: 1 – із системою керування на базі нейронної мережі; 2 – без системи керування.

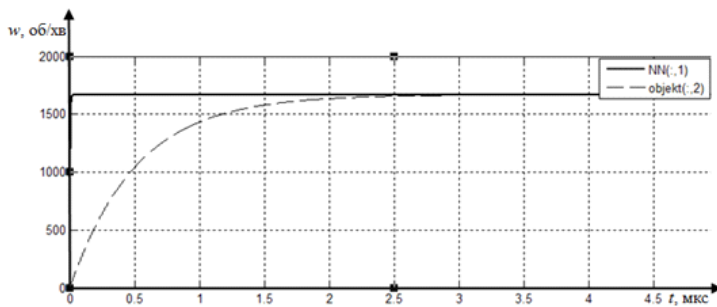


Рис. 5.10. Порівняння перехідних процесів

Контрольні питання

1. З яких елементів складається прихований шар нейронної мережі?
2. Що таке функція активації? Які види функцій існують?
3. Який метод навчання нейронної мережі, застосований у лабораторній роботі?

Лабораторна робота № 6

Тема. Розробка системи керування швидкістю горизонтального руху автономного підводного апарата на базі штучної нейронної мережі (частина перша).

Мета: набути практичних навичок синтезу математичних моделей, що реалізують у рекурентній формі рівняння динаміки об'єкта в системі Simulink. Закріпити навички використання інструменту Neural Network Toolbox при розробці регулятора на базі штучної нейронної мережі.

Хід роботи

Ознайомитися з математичною моделлю автономного підводного апарата описаного в [2]. Розібратися з реалізацією в рекурентній формі залежностей, що описують рівняння динаміки руху апарата.

Запустити Matlab. Завантажити модель руху апарата в горизонтальній площині – папка «object», файл – «ROV» (рис. 6.1).

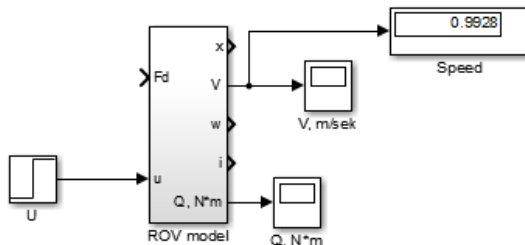


Рис. 6.1. Математична модель АПА реалізована в системі Simulink

Провести моделювання руху апарата, що розганяється до заданої швидкості (за варіантом: № Вар/10 м/с) без системи керування. Отримати перехідний процес розгону апарата (рис. 6.2).

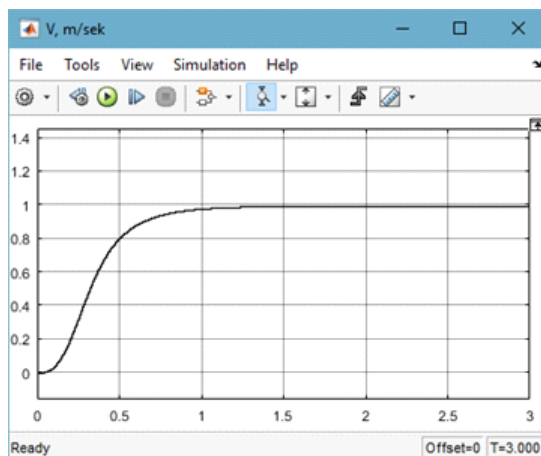


Рис. 6.2. Перехідний процес зміни швидкості АПА без системи керування в системі Simulink

Систему керування швидкістю руху АПА побудуємо методом комбінації двох регуляторів: адаптивного ПІ-регулятора та штучної нейронної мережі.

У даному випадку керування рухом АПА здійснюється подачею необхідної напруги на двигун постійного струму незалежного збудження в діапазоні від $[-10 \text{ до } 10 \text{ В}]$. Кожне значення напруги задає відповідну швидкість руху АПА від 0 до 0,996 м/с + реверсний режим роботи двигуна.

Для підтримання необхідної заданої швидкості руху апарата використаємо ПІ-регулятор, для визначення необхідного значення напруги синтезуємо штучний нейронний регулятор, як універсальний апроксиматор.

Для навчання нейрорегулятора необхідно задати вектор вхідного сигналу – значення заданої швидкості та вихідного – відповідне значення напруги (табл. 6.1).

Таблиця 6.1. Залежність швидкості руху апарата від напруги

U, В	0,5	1	1,5	2	2,5	3	3,5	4	4,5	5	5,5	6	6,5	7	7,5	8,5	9	9,5	10
V, м/с																			

Проведемо серію експериментів та внесемо необхідні дані в табл. 6.1. Поступово, із кроком 0,5 В, будемо збільшувати напругу на двигун апарата та фіксувати відповідне значення швидкості до якої розженеться апарат. Початкове значення напруги обирається за варіантом № Var/10, наприклад: Var1 = 0,1 В.

Заповнивши таблицю, створимо необхідні для навчання нейронної мережі вектори входу та виходу системи.

Для цього з командної строки Matlab необхідно прописати наступний код:

U = [значення напруги з таблиці 6.1 (через пробіл)];

V = [відповідне значення швидкості з таблиці 6.1 (через пробіл)];

Vvход(1,:) = V;

Uv(1,:) = U.

У результаті, у робочій області Matlab з'явиться вектор Vvход та Uv (рис. 6.3).



Рис. 6.3. Робочий вигляд вікна «Workspace» у Matlab

Перевірити відповідність розмірів створених векторів. Зберегти робочу область «Workspace» у відповідну папку із лабораторною роботою.

Зміст звіту

1. Тема, мета роботи.
2. Математична модель горизонтального руху АПА.
3. Перехідний процес зміни швидкості руху апарата (за варіантом).
4. Таблиця 6.1, короткий опис або алгоритм формування значень у таблиці (описати суть проведених експериментів).

Лабораторна робота № 7

Тема. Розробка системи керування швидкістю горизонтального руху автономного підводного апарата на базі штучної нейронної мережі (частина друга).

Мета: набути практичних навичок синтезу та налаштування ПІ-регуляторів для об'єктів різних типів. Закріпити навички використання інструменту Neural Network Toolbox під час розробки регулятора на базі штучної нейронної мережі.

Хід роботи

Завантажити Matlab. Завантажити збережені в попередній роботі дані в робочу область.

Для керування швидкістю руху апарата синтезуємо пропорційно-інтегральний регулятор та налаштуємо його (рис. 7.1).

Метод налаштування ПІ-регулятора.

Нехай пропорційна складова дорівнює нулю $K_p = 0$. Тоді необхідно поступово збільшувати значення інтегральної складової, доки перехідний процес не перетвориться в незатухаючі коливання (рис. 7.2). Задана швидкість обирається за варіантом (№ Var/10, м/с).

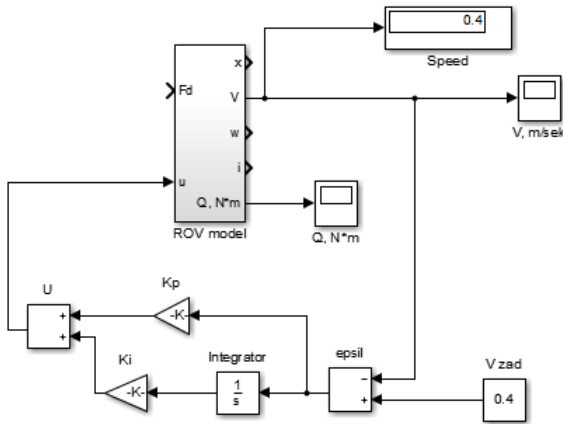


Рис. 7.1. Структурна схема ПІ-регулятора

Тоді рекомендоване значення коефіцієнта пропорційності дорівнює половині значення інтегральної складової $K_p = K_i / 2$. У разі необхідності виконати регулювання в ручному режимі.

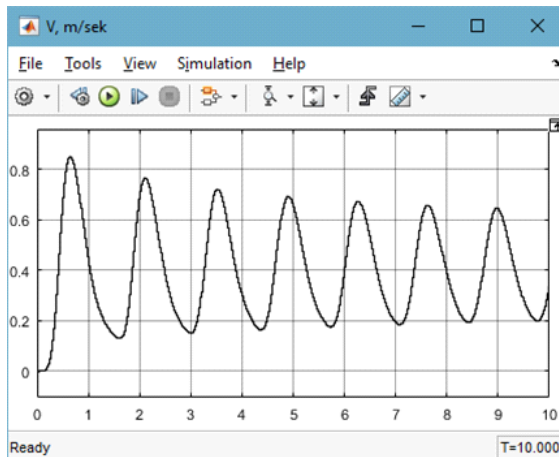


Рис. 7.2. Перехідний процес у вигляді незатухаючих коливань при налаштуванні ПІ-регулятора

За допомогою команди `nntool` перейдемо до інструменту Neural Network Toolbox та створимо новий нейрорегулятор.

Як вхідне значення (Input Data) – імпортуємо вектор $V_{\text{vход}}$, як значення ваги (Target Data) – імпортуємо вектор U_v .

Створемо нейрорегулятор типу Feed-forward-basprop., що складатиметься з двох шарів, у першому містяться 4 нейрони із функцією активації Logsig. У прихованому шарі – 1 нейрон з лінійною функцією (рис. 7.3).

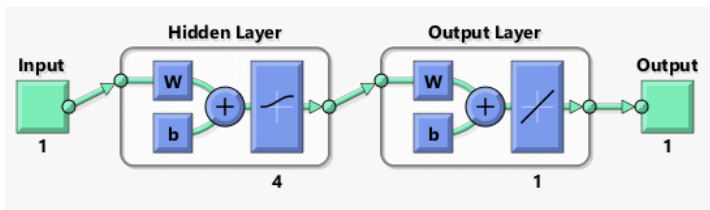


Рис. 7.3. Структура нейрорегулятора

Провести навчання регулятора та експортувати його до робочої області.

За допомогою команди `gensim(net, st)` згенеруємо створений нейрорегулятор та додамо його до розробленої системи керування, щоб визначати значення необхідної напруги для будь-якого значення заданої швидкості (рис. 7.4). Провести моделювання, отримати перехідний процес (рис. 7.5).

Провести перевірку розробленої системи керування, що найменше для двох значень заданої швидкості. Для цього як задавальний елемент встановимо блок «Step».

Перевіримо якість розробленої системи керування та отримаємо перехідний процес для двох значень заданої швидкості (за варіантом) (рис. 7.6).

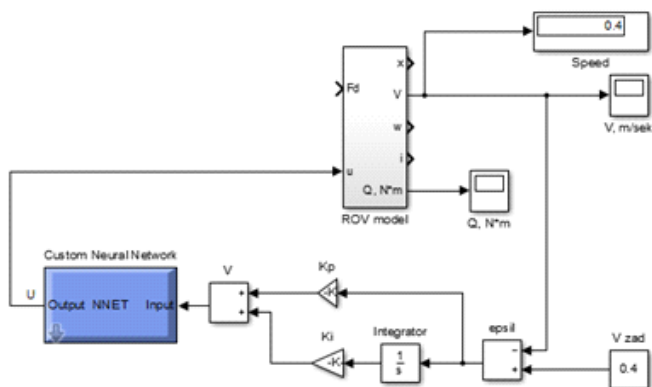


Рис. 7.4. Структура розробленої системи керування швидкістю руху АПА

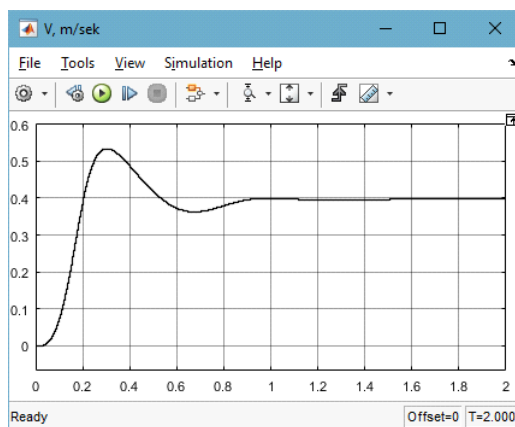


Рис. 7.5. Перехідний процес зміни швидкості руху АПА зі штучною нейронною системою керування

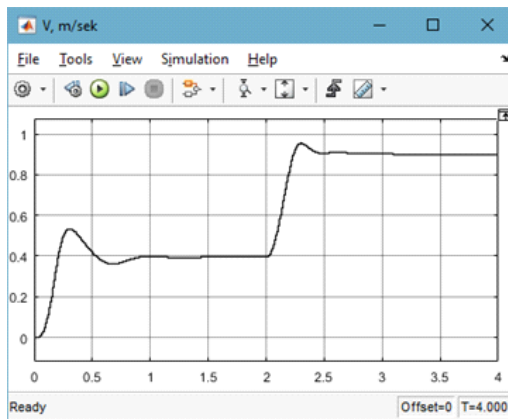


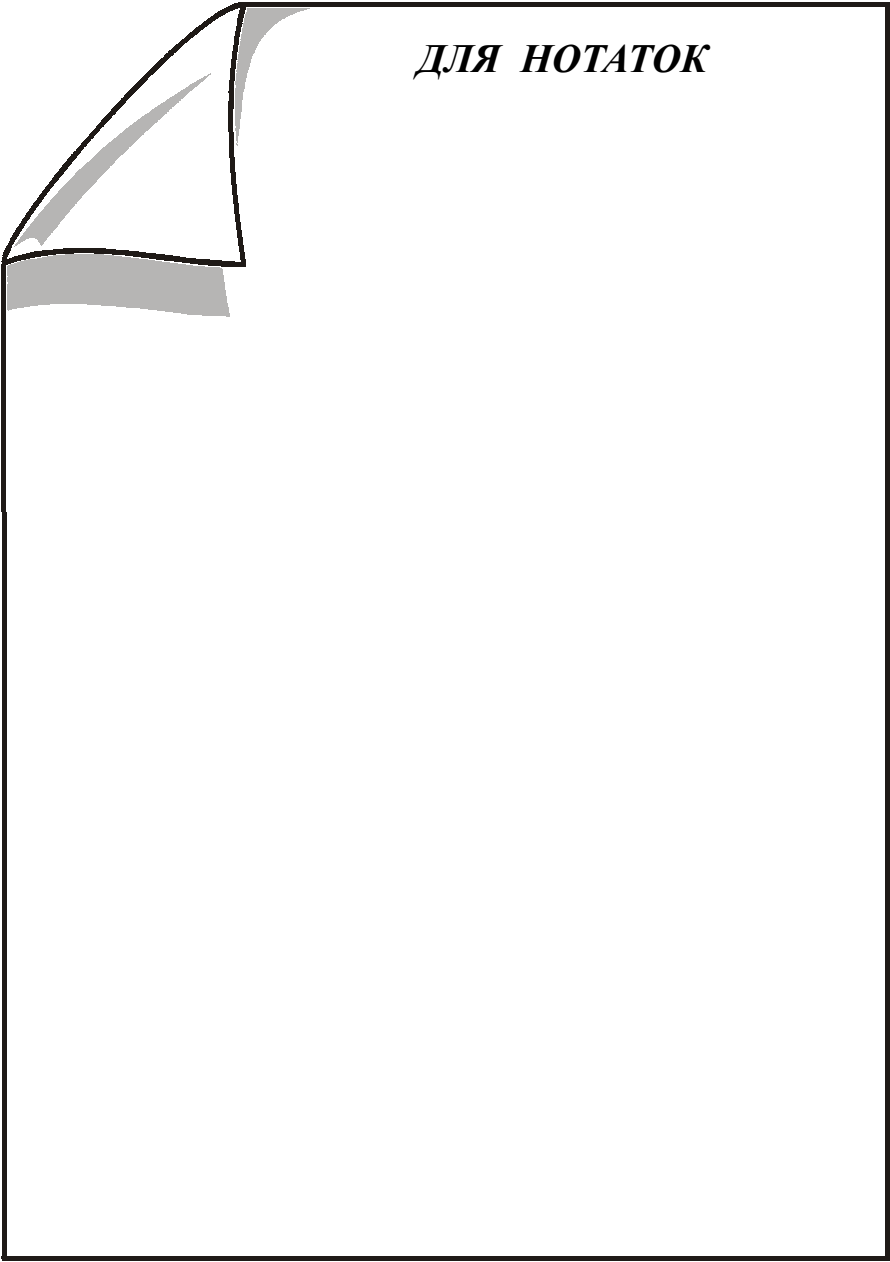
Рис. 7.6. Перехідний процес зміни швидкості руху АПА

Контрольні питання

1. З якою метою в даній роботі було застосовано ШНМ?
2. Які типи нейронних мереж Ви знаєте?
3. Зробити висновки про особливості створення інтелектуальних систем керування морськими роботизованими системами.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. **Блінцов, С. В.** Автоматичне керування автономними підводними апаратами в умовах невизначеності [Текст] / С. В. Блінцов. – Миколаїв : «Іліон», 2008. – 204 с.
2. **Блінцов, С. В.** Теоретичні основи автоматичного керування АПА [Текст] / С. В. Блінцов. – Миколаїв : НУК, 2014. – 222 с.
3. **Леоненков, А. В.** Нечёткое моделирование в среде Matlab и fuzzy TECH [Текст] / А. В. Леоненков. – Петербург : СПб: БХВ-Петербург, 2005. – 736 с.
4. Сигеру Омату, Марзуки Халид & Рубия Юсоф. Нейроуправление и его приложения [Текст] / Сигеру Омату ; пер. с англ. Н. В. Батина. – М. : ИПРЖР, 2000. – кн. 2. – 272 с.
5. **Зайченко, Ю. П.** Основи проектування інтелектуальних систем [Текст] : навч. посіб. / Ю. П. Зайченко. – К. : Видавничий Дім «Слово», 2004. – 352 с.



ДЛЯ ЗАДАТОК

Навчальне видання

БЛІНЦОВ Володимир Степанович
ГРУДІНІНА Ганна Сергіївна

**ІНТЕЛЕКТУАЛЬНЕ КЕРУВАННЯ ЗАСОБАМИ
МОРСЬКОЇ РОБОТОТЕХНІКИ**

Методичні вказівки
до виконання лабораторних робіт

Комп'ютерна правка й коректура *О. Г. Костенко*
Комп'ютерне верстання *В. В. Москаленко*

Формат 60×84/16. Ум. друк. арк. 2,6. Тираж 100 прим. Вид. № 25. Зам. № 0906-50.

Видавець і виготовник Національний університет кораблебудування
імені адмірала Макарова

просп. Героїв України, 9, м. Миколаїв, 54025

E-mail : publishing@nuos.edu.ua

Свідоцтво суб'єкта видавничої справи ДК № 6402 від 19.09.2018 р.