

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені В. Н. КАРАЗІНА

**ТЕХНОЛОГІЇ НЕЙРОННИХ МЕРЕЖ І НЕЧІТКОГО
МОДЕЛЮВАННЯ В СИСТЕМАХ УПРАВЛІННЯ**

Методичні вказівки
до виконання курсового проєкту для здобувачів вищої освіти
другого (магістерського) рівня за спеціальністю 174 «Автоматизація,
комп'ютерно-інтегровані технології та робототехніка»

Електронний ресурс

Рецензенти:

Р. М.Тріщ – доктор технічних наук, професор, завідувач кафедри мехатроніки та електротехніки Національного аерокосмічного університету ім. М. Є. Жуковського «Харківський авіаційний інститут»;

О. О. Литвин – доктор фізико-математичних наук, професор, завідувач кафедри ХТЛПід Навчально-наукового інституту «Українська інженерно-педагогічна академія».

*Затверджено до розміщення в мережі Інтернет рішенням Науково-методичної ради
Харківського національного університету імені В. Н. Каразіна
(протокол № 11 від 25 червня 2025 року)*

Технології нейронних мереж і нечіткого моделювання в системах управління :
Т 38 методичні вказівки до виконання курсового проєкту для здобувачів вищої освіти другого (магістерського) рівня за спеціальністю 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» [Електронний ресурс] / уклад. Г. І. Канюк, Т. Ю. Василець, О. О. Вірфоломієв. – Харків : ХНУ імені В. Н. Каразіна, 2025. – (PDF 73 с.)

У виданні містяться методичні вказівки до виконання курсового проєкту на тему «Синтез нейромережевої системи управління» за курсом «Технології нейронних мереж і нечіткого моделювання в системах управління». Наведено основні теоретичні положення, необхідні для виконання проєкту, дається порядок і методика виконання всіх пунктів курсового проєкту.

Видання призначене здобувачам вищої освіти освітнього ступеню «магістр», що навчаються за освітньо-професійною програмою «Автоматизація та комп'ютерно-інтегровані технології» зі спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка.

УДК 658.5.011.56:658.589(075.5)

© Харківський національний університет імені В. Н. Каразіна, 2025

© Канюк Г. І., Василець Т. Ю.,
Вірфоломієв О. О., уклад., 2025

ЗМІСТ

1. МЕТА КУРСОВОГО ПРОЄКТУ	4
2. ОРГАНІЗАЦІЯ ПРОЄКТУВАННЯ	4
3. ЗАВДАННЯ НА КУРСОВИЙ ПРОЄКТ	5
4. ЗМІСТ КУРСОВОГО ПРОЄКТУ	8
5. МЕТОДИЧНІ ВКАЗІВКИ КУРСОВОГО ПРОЄКТУ	9
5.1. Основні поняття штучних нейронних мереж	9
5.2. Розрахунок системи підлеглого регулювання швидкост	9
5.3. Дослідження динамічних характеристик двомасової системи	9
5.3.1 Розробка математичної моделі двомасової системи	10
5.3.2 Розрахунок перехідних процесів двомасової систем на ЕОМ	15
5.4 Синтез системи управління з нейрорегулятором	21
5.4.1 Регулятори реалізовані в пакеті прикладних програм Neural Network Toolbox	21
5.4.2 Принцип побудови регулятора з прогнозом NN Predictive Controller	23
5.4.3 Синтез регулятора з прогнозом NN Predictive Controller з використанням системи MATLAB	25
5.4.4 Вибір параметрів нейрорегулятора NN Predictive Controller	61
5.4.5 Моделювання нейромережевої системи управління	63
6. ОФОРМЛЕННЯ КУРСОВОГО ПРОЄКТУ	69
СПИСОК ЛІТЕРАТУРИ	70
ДОДАТОК А Файл вихідних даних	72

1. МЕТА КУРСОВОГО ПРОЄКТУ

Метою курсового проєкту є закріплення основ теоретичних знань, придбаних за курс лекцій, а також розвиток інженерного мислення та оволодіння конкретними практичними методами синтезу, аналізу і корекції систем автоматичного управління при їх розрахунку і проєктуванні, синтезу нейромережових систем управління. У курсовому проєкті розглядаються і виконуються задачі аналізу стійкості та якості систем управління, а також синтезу нейрорегуляторів з використанням системи MATLAB відносно до систем автоматичного управління.

2. ОРГАНІЗАЦІЯ ПРОЄКТУВАННЯ

Організація роботи з виконання і захисту курсового проєкту покладається на викладача, що здійснює керівництво проєктом.

У методичних вказівках формулюються назви, і розглядається основний зміст усіх розділів курсового проєкту, даються рекомендації по методиці і способах рішення розглянутих у них завданнях.

Перед тим як приступити до виконання курсового проєкту, необхідно уважно ознайомитися з даними методичних указівок, для того щоб мати уявлення щодо змісту і обсягу роботи в цілому.

Закінчений курсовий проєкт являє собою цілком оформлену і підготовлену до захисту розрахунково-пояснювальну записку, що повинна вміщувати три основних розділи:

- 1) розрахунок системи підлеглого регулювання швидкості;
- 2) дослідження характеристик двомасової системи;
- 3) синтез нейромережової системи автоматичного управління.

Закінчена робота являє собою цілком оформлену і підготовлену до захисту розрахунково-пояснювальну записку. Студенти, що не виконали роботу в термін, втрачають право на здачу екзамену з курсу. Кожний студент зобов'язаний чітко і коротко побудувати свою доповідь під час захисту і бути готовим до відповідей на кожне питання по виконанню курсового проєкту.

3. ЗАВДАННЯ НА КУРСОВИЙ ПРОЄКТ

Тема проєкту: «Синтез нейромережевої системи автоматичного управління».

У курсовому проєкті вивчаються новітні методи синтезу автоматичних систем управління з використанням нейромережевих технологій.

У курсовому проєкті об'єктом управління є електропривод за системою ТП-Д (тиристорний перетворювач - двигун постійного струму незалежного збудження). Механічна частина електроприводу містить пружні елементи і розглядається як двомасова. У завданні наведені параметри силового ланцюга електроприводу, моменти інерції двигуна і механізму, коефіцієнт жорсткості передачі.

Спочатку розглядається одномасова система (без врахування пружних елементів) і виконується розрахунок і моделювання системи підлеглого регулювання швидкості із зворотним зв'язком по ЕРС двигуна, що знаходить широке застосування в промисловості.

Далі проводиться дослідження динамічних характеристик двомасової системи. Перехідні процеси двомасової системи зазвичай мають коливальний характер.

Для покращення характеристик двомасової системи у курсовому проєкті виконується синтез нейромережевого регулятора, що забезпечує високі показники якості функціонування двомасової системи.

Завдання на курсовий проєкт наведені в таблиці 3.1. Номер варіанта видається викладачем, що веде заняття по даній дисципліні.

В окремих випадках у якості курсового проєкту може бути узята розробка нейромережевої системи управління, що представляє інтерес для підприємства, на якому працює студент, або кафедри АМЕТ ННІ «УПА» ХНУ ім. В.Н.Каразіна. У таких випадках тема курсового проєкту, а також його об'єм і зміст, повинні бути додатково погоджені з завідуючим кафедрою.

Таблиця 3.1 - Вихідні дані до курсового проєкту

№ з/п	Параметри	Номер варіант					
		1	2	3	4	5	6
1	Номінальна напруга двигуна, U_n, B	440	440	220	440	440	220
2	Номінальний струм двигуна, I_n, A	92	69,5	57	168	57	21
3	Номінальна швидкість двигуна, $n_n, об/хв$	570	720	800	560	800	1080
4	Сумарний активний опір силового ланцюга електроприводу, $R_\Sigma, Ом$	0,183	0,5935	0,4145	0,1568	0,413	1,36
5	Опір якірного ланцюга електродвигуна, $R_{як}, Ом$	0,136	0,221	0,036	0,0707	0,38	1,278
6	Електромагнітна постійна часу силового ланцюга електроприводу, T_e, c	0,014	0,044	0,036	0,034	0,037	0,022
7	Електромагнітна постійна часу якірного ланцюга електродвигуна, T_a, c	0,017	0,057	0,036	0,033	0,036	0,021
8	Електромеханічна постійна часу електроприводу, T_m, c	0,027	0,185	0,21	0,068	0,42	0,82
9	Коефіцієнт посилення двигуна, k_d	0,13	0,18	0,43	0,137	0,21	0,584
10	Коефіцієнт посилення тиристорного перетворювача, $k_{тп}$	34,2	21,4	27,5	51,3	18,3	11,46
11	Жорсткість передачі, c_{12}	5	10	5	5	60	5
12	Момент інерції двигуна і жорстко пов'язаних з ним елементів $J_{д\Sigma}, кг \cdot м^2$	8,4	0,96	0,51	12,24	1,02	0,37
13	Момент інерції механізму, $J_m, кг \cdot м^2$	0,17	3,66	2,23	0,18	11,4	1,34
14	Наявність контуру напруги	-	-	-	+	+	+
15	Тип регулятора контуру швидкості	ПІ	ПІ	П	ПІ	ПІ	П

Продовження таблиці 3.1

№ з/п	Номер варіанту														
	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	440	220	440	220	440	220	440	220	440	440	220	440	220	440	220
2	112	84	138	57	84	280	84	51	168	57	21	175	132	69,5	57
3	630	710	560	800	710	560	710	800	560	800	1080	540	565	720	800
4	0,22	0,12	0,0911	0,526	0,36	0,022	0,36	0,633	0,1751	0,413	1,358	0,138	0,0778	0,4715	0,531
5	0,1469	0,073	0,07	0,38	0,2954	0,018	0,1477	0,38	0,071	0,38	1,2784	0,0441	0,0468	0,22	0,38
6	0,041	0,079	0,043	0,027	0,061	0,071	0,061	0,025	0,038	0,037	0,023	0,031	0,041	0,065	0,028
7	0,059	0,066	0,056	0,036	0,071	0,08	0,071	0,04	0,023	0,036	0,022	0,08	0,073	0,073	0,036
8	0,12	0,34	0,072	0,34	0,08	0,093	0,08	0,25	0,084	0,42	0,79	0,048	0,051	0,147	0,27
9	0,31	0,347	0,136	0,42	0,18	0,27	0,1786	0,42	0,137	0,21	0,588	0,13	0,144	0,185	0,43
10	40,3	27	21,4	27	59,8	59,8	59,8	29,9	21	18,3	18,3	56	34,2	34,2	34,3
	80	5	4	10	30	4	30	10	4	10	1,4	4	4	10	11
12	4,8	1,2	12,3	0,51	2,4	10,25	2,4	0,49	12,22	2,02	0,37	19,56	8,4	1,92	1,09
13	16,2	4,58	0,18	2,95	5,19	0,45	18,4	1,72	0,17	8,72	1,34	0,38	0,17	7,32	1,848
14	-	-	+	+	+	-	-	-	+	+	+	-	-	-	-
15	ПІ	П	ПІ	П	ПІ	ПІ	ПІ	П	ПІ	ПІ	П	ПІ	ПІ	ПІ	П

4 ЗМІСТ КУРСОВОГО ПРОЄКТУ

Розрахунково-пояснювальна записка повинна містити титульний лист, завдання, зміст, основну частину, список літератури.

Основна частина пояснювальної записки повинна включати наступне.

1. У вступі до проєкту дається коротка характеристика основної задачі, що повинна бути вирішена в результаті виконання проєкту, вихідні дані.

2. Основні поняття штучних нейронних мереж.

3 Розрахунок системи підлеглого регулювання швидкості із зворотним зв'язком по ЕДС двигуна.

3.1. Синтез послідовного коригувального пристрою контуру струму (регулятора струму) (вибір типового регулятора, розрахунок параметрів типового регулятора і параметрів ланцюга зворотного зв'язку по струму).

3.2. Синтез послідовного коригувального пристрою контуру швидкості із зворотним зв'язком по ЕРС двигуна (вибір типового регулятора, розрахунок параметрів типового регулятора і параметрів контуру швидкості із зворотним зв'язком по ЕРС).

3.3. Графіки перехідних процесів швидкості і струму по задаючому і збурюючому впливах.

4. Дослідження динамічних характеристик двомасової системи.

4.1 Розробка математичної моделі двомасової системи.

4.2 Розрахунок перехідних процесів двох масової систем на ЕОМ.

5. Синтез системи управління з нейрорегулятором

5.1 Опис регуляторів, реалізовані в пакеті прикладних програм Neural Network Toolbox системи MATLAB.

5.2 Принцип побудови регулятора з прогнозом NN Predictive Controller

5.3 Порядок синтезу нейрорегулятора.

5.4 Вибір параметрів нейрорегулятора NN Predictive Controller

5.5 Моделювання нейромережевої системи управління

5 МЕТОДИЧНІ ВКАЗІВКИ ПО ВИКОНАННЮ РОЗДІЛІВ ПРОЄКТУ

У вступі визначають роль автоматизованих систем управління в рішенні задачі автоматизації виробничих процесів. Приводять конкретне завдання на курсовий проєкт із докладними технічними даними електричного двигуна, робочого механізму і короткої характеристики основної задачі, що повинна бути вирішена в результаті виконання проєкту.

5.1 Основні поняття штучних нейронних мереж.

Основні поняття штучних нейронних мереж і їх застосуванні в системах управління детально розглянуті в літературі [1-11].

У пояснювальній записці необхідно навести: галузі застосування штучних нейронних мереж, структуру штучного нейрона, структуру і властивості нейронних мереж, методи навчання нейронних мереж, питання, пов'язані з використанням нейронних мереж в системах управління.

5.2 Розрахунок системи підлеглого регулювання швидкості.

Для синтезу послідовних коригувальних пристроїв системи підлеглого регулювання із зворотним зв'язком по ЕДС двигуна (пункт 3 курсового проєкту) варто скористатися літературою [12-23].

5.3 Дослідження динамічних характеристик двомасової системи.

У даному параграфі наведені методичні вказівки по виконанню пункту 4 курсового проєкту). Можна також скористатися літературою [12-25].

5.3.1 Розробка математичної моделі двомасової системи

З курсу теорії електропривода відомо, що представлення механічної частини електропривода жорсткою приведеною ланкою відбиває дійсний характер руху мас у системі тільки в середньому, тому що не враховує впливу пружкості реальних механічних зв'язків. Через кінцеву жорсткість цих зв'язків механічна частина електропривода являє собою пружну систему, додаток до якої керуючого (момент двигуна) або збурюючого (навантаження) впливів викликає коливання зв'язаних мас, що збільшують максимальні навантаження зв'язків і ускладнюють точності відпрацювання необхідних траєкторій руху робочого органа машини.

Як відомо, кінематична схема більшості механізмів, приведена до двомасової системи, в загальному випадку з урахуванням моменту внутрішнього в'язкого тертя, описується рівняннями:

$$\begin{cases} J_{\text{м}} \frac{d\omega_{\text{м}}}{dt} = M_{\Sigma} - M_{\text{ст}}; \\ \frac{dM_{\text{пр}}}{dt} = c_{12}(\omega_{\text{д}} - \omega_{\text{м}}); \\ J_{\text{д}\Sigma} \frac{d\omega_{\text{д}}}{dt} = M_{\text{д}} - M_{\Sigma}. \end{cases} \quad (5.1)$$

де M_{Σ} – сумарний момент, що передається пружною передачею, який рівний сумі пружного моменту $M_{\text{пр}}$ і моменту в'язкого тертя $M_{\text{вт}} = \beta(\omega_{\text{д}} - \omega_{\text{м}})$:

$$M_{\Sigma} = M_{\text{пр}} + \beta(\omega_{\text{д}} - \omega_{\text{м}}), \quad (5.2)$$

β – коефіцієнт внутрішнього в'язкого тертя; у розрахунках приймаємо $\beta = 0$.

Розглянемо двомасову систему підлеглого регулювання швидкості з контуром струму, настроєним по модульному критерію і контуро ЕРС, настроєним по симетричному критерію.

Запишемо систему (5.1) з урахуванням (5.2) у формі Коші

$$\begin{cases} \frac{d\omega_{\text{м}}}{dt} = \frac{1}{J_{\text{м}}} M_{\text{пр}} + \frac{\beta}{J_{\text{м}}} \omega_{\text{д}} - \frac{\beta}{J_{\text{м}}} \omega_{\text{м}} - \frac{1}{J_{\text{м}}} M_{\text{ст}}; \\ \frac{dM_{\text{пр}}}{dt} = c_{12} \omega_{\text{д}} - c_{12} \omega_{\text{м}}; \\ \frac{d\omega_{\text{д}}}{dt} = \frac{1}{J_{\text{д}\Sigma}} M_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{пр}} - \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{д}} + \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{м}}. \end{cases} \quad (5.3)$$

Одномасова система підлеглого регулювання швидкості з контуром струму,

настроєним по модульному критерію і контуром ЕРС, настроєним по симетричному критерію описується наступною системою диференціальних рівнянь, записаних у формі Коші:

$$\left\{ \begin{array}{l} \frac{d\omega_{\text{д}}}{dt} = \frac{1}{J_{\text{д}\Sigma}} M_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{ст}}; \\ \frac{dM_{\text{д}}}{dt} = \frac{1}{k_{\text{д}}} P; \\ \frac{dP}{dt} = \frac{k_{\text{пе}}}{2T_{\mu\Gamma}^2 k_{\Gamma}} E_{\text{зе}} - \frac{k_{\text{пе}}}{2T_{\mu\Gamma}^2 k_{\Gamma}} U_{\text{ззе}} - \frac{k_{\text{д}}}{2T_{\mu\Gamma}^2} M_{\text{д}} - \frac{1}{T_{\mu\Gamma}} P + \frac{1}{T_{\mu\Gamma}^2 k_{\Gamma}} U_{\text{іе}}; \\ \frac{dE_{\text{зе}}}{dt} = \frac{1}{T_{\text{а}}} U_{\text{зе}} - \frac{1}{T_{\text{а}}} E_{\text{зе}}; \\ \frac{dU_{\text{ззе}}}{dt} = \frac{k_{\text{н}}}{T_{\text{а}} k_{\text{д}}} \omega_{\text{д}} - \frac{1}{T_{\text{а}}} U_{\text{ззе}}; \\ \frac{dU_{\text{іе}}}{dt} = k_{\text{іе}} E_{\text{зе}} - k_{\text{іе}} U_{\text{ззе}}. \end{array} \right. \quad (5.4)$$

При записі системи (5.4) прийняті наступні позначення.

$U_{\text{іе}}$ – напруга на виході інтегральної частини регулятора; P – ривок; $k_{\text{пе}}$, $k_{\text{іе}}$ – коефіцієнти підсилення пропорційної і інтегральної частин регулятора ЕРС; значення указаних параметрів можуть бути отримані з виразу передатної функції регулятора ЕРС:

$$W_{\text{пе}}(p) = \frac{T_{\text{м}} k_{\Gamma}}{2T_{\mu\text{е}} k_{\text{н}} R_{\Sigma}} \cdot \frac{4T_{\mu\text{е}} p + 1}{4T_{\mu\text{е}} p} = \frac{T_{\text{м}} k_{\Gamma}}{2T_{\mu\text{е}} k_{\text{н}} R_{\Sigma}} + \frac{T_{\text{м}} k_{\Gamma}}{8T_{\mu\text{е}}^2 k_{\text{н}} R_{\Sigma}} \frac{1}{p} = k_{\text{пе}} + \frac{k_{\text{іе}}}{p}, \quad (5.5)$$

$$k_{\text{пе}} = \frac{T_{\text{м}} k_{\Gamma}}{2T_{\mu\text{е}} k_{\text{н}} R_{\Sigma}}, \quad k_{\text{іе}} = \frac{T_{\text{м}} k_{\Gamma}}{8T_{\mu\text{е}}^2 k_{\text{н}} R_{\Sigma}}. \quad (5.6)$$

При записі системи (5.4) замість $T_{\text{м}}$ записано відповідний вираз

$$T_{\text{м}} = \frac{R_{\Sigma} \cdot J_{\text{д}\Sigma}}{c^2} = R_{\Sigma} J_{\text{д}\Sigma} \cdot k_{\text{д}}^2; \quad k_{\text{д}} = 1/c.$$

Об'єднаємо системи (5.3) і (5.4), одержимо наступну систему рівнянь стану двомасової системи підлеглого регулювання швидкості із зворотним зв'язком по ЕРС двигуна:

$$\left\{ \begin{array}{l} \frac{d\omega_{\text{м}}}{dt} = \frac{1}{J_{\text{м}}} M_{\text{пр}} + \frac{\beta}{J_{\text{м}}} \omega_{\text{д}} - \frac{\beta}{J_{\text{м}}} \omega_{\text{м}} - \frac{1}{J_{\text{м}}} M_{\text{ст}}; \frac{dM_{\text{пр}}}{dt} = c_{12} \omega_{\text{д}} - c_{12} \omega_{\text{м}}; \\ \frac{d\omega_{\text{д}}}{dt} = \frac{1}{J_{\text{д}\Sigma}} M_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{пр}} - \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{д}} - \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{м}}; \\ \frac{dM_{\text{д}}}{dt} = \frac{1}{k_{\text{д}}} P; \\ \frac{dP}{dt} = \frac{k_{\text{пе}}}{2T_{\mu T}^2 k_{\text{т}}} E_{\text{зе}} - \frac{k_{\text{пе}}}{2T_{\mu T}^2 k_{\text{т}}} U_{\text{ззе}} - \frac{k_{\text{д}}}{2T_{\mu T}^2} M_{\text{д}} - \frac{1}{T_{\mu T}} P + \frac{1}{T_{\mu T}^2 k_{\text{т}}} U_{\text{іе}}; \\ \frac{dE_{\text{зе}}}{dt} = \frac{1}{T_{\text{а}}} U_{\text{зе}} - \frac{1}{T_{\text{а}}} E_{\text{зе}}; \\ \frac{dU_{\text{ззе}}}{dt} = \frac{k_{\text{н}}}{T_{\text{а}} k_{\text{д}}} \omega_{\text{д}} - \frac{1}{T_{\text{а}}} U_{\text{ззе}}; \\ \frac{dU_{\text{іе}}}{dt} = k_{\text{іе}} E_{\text{зе}} - k_{\text{іе}} U_{\text{ззе}}. \end{array} \right. \quad (5.7)$$

Алгоритмічна схема двомасової системи, що відповідає рівнянням (5.7), показана на рис. 5.1. На схемі оптимізований контур струму представлений у вигляді послідовного з'єднання аперіодичної і інтегруючої ланки, охоплених одиничним зворотним зв'язком. Сигнал на виході першої ланки пропорційний ривку електроприводу P .

Запишемо результуючу систему у векторному – матричній формі, доповнивши її рівнянням виходу системи.

Введемо вектор стану системи $\vec{X}(t)$ і вектор управління $\vec{U}(t)$

$$\vec{X}(t) = \{ \omega_{\text{м}}(t), M_{\text{пр}}(t), \omega_{\text{д}}(t), M_{\text{д}}(t), P(t), E_{\text{зе}}(t), U_{\text{ззе}}(t), U_{\text{іе}}(t) \}^T, \quad (5.8)$$

$$\vec{U}(t) = \{ U_{\text{зе}}(t), M_{\text{ст}}(t) \}^T, \quad (5.9)$$

де T – знак транспонування.

Рівняння стану системи і рівняння виходу має вигляд

$$\begin{cases} \dot{\vec{X}}(t) = \mathbf{A}\vec{X}(t) + \mathbf{B}\vec{U}(t); \\ \vec{Y}(t) = \mathbf{C}\vec{X}(t), \end{cases} \quad (5.10)$$

де $\mathbf{A}$ – матриця стану; $\mathbf{B}$ – матриця управління; $\mathbf{C}$ – матриця виходу; $\vec{Y}(t)$ – вектор виходу.

Матриці стану $\mathbf{A}$ і управління $\mathbf{B}$ мають вид, показаний нижче.

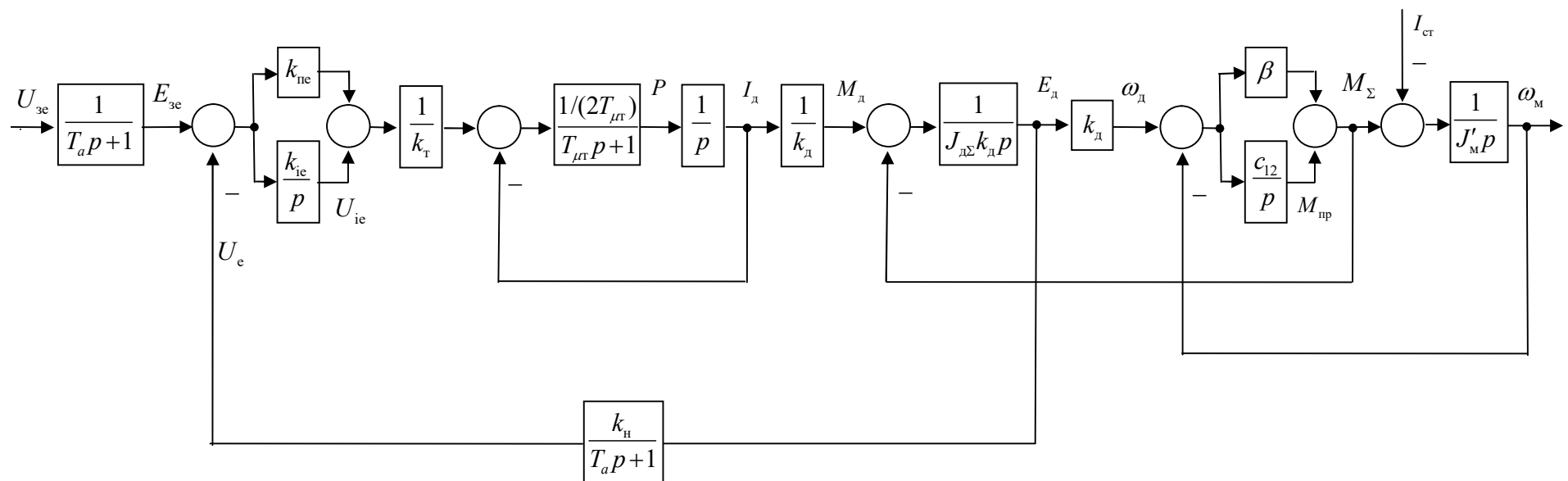


Рисунок 5.1 – Спрощена алгоритмічна схема двомасової системи

$$\mathbf{A} = \begin{array}{c} \begin{array}{cccccccc} \omega_{\text{м}} & M_{\text{ип}} & \omega_{\text{д}} & M_{\text{д}} & P & E_{\text{з\text{е}}} & U_{\text{е}} & U_{\text{и\text{е}}} \end{array} \\ \begin{array}{|c|c|c|c|c|c|c|c|} \hline -\frac{\beta}{J_{\text{м}}} & \frac{1}{J_{\text{м}}} & \frac{\beta}{J_{\text{м}}} & 0 & 0 & 0 & 0 & 0 \\ \hline -c & 0 & c & 0 & 0 & 0 & 0 & 0 \\ \hline \frac{\beta}{J_{\text{д}}} & -\frac{1}{J_{\text{д}}} & -\frac{\beta}{J_{\text{д}}} & \frac{1}{J_{\text{д}}} & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \frac{1}{k_{\text{д}}} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \frac{-k_{\text{д}}}{2T_{\text{мт}}^2} & -\frac{1}{T_{\text{мт}}} & \frac{k_{\text{п\text{е}}}}{2T_{\text{мт}}^2 k_{\text{т}}} & -\frac{k_{\text{п\text{е}}}}{2T_{\text{мт}}^2 k_{\text{т}}} & \frac{1}{2T_{\text{мт}}^2 k_{\text{т}}} \\ \hline 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_a} & 0 & 0 \\ \hline 0 & 0 & \frac{k_{\text{н}}}{T_a k_{\text{д}}} & 0 & 0 & 0 & -\frac{1}{T_a} & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & k_{\text{и\text{е}}} & -k_{\text{и\text{е}}} & 0 \\ \hline \end{array} \end{array}$$

$$\mathbf{B} = \begin{array}{c} \begin{array}{cc} U_{\text{з\text{е}}} & M_{\text{с\text{т}}} \end{array} \\ \begin{array}{|c|c|} \hline 0 & -\frac{1}{J_{\text{м}}} \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline \frac{1}{T_a} & 0 \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline \end{array} \end{array}$$

5.3.2 Розрахунок перехідних процесів двомасової систем на ЕОМ

Розрахунок перехідних процесів слід виконувати з використанням пакету прикладних програм MATLAB. Дана система дозволяє проводити розрахунок перехідних процесів по передавальних функціях системи, по векторно-матричному рівнянню, з використанням схеми моделі системи, показаної на рис. 5.2.

У додатку А приведений приклад програми, в якому містяться початкові дані і матриці **A**, **B** і **C** двомасової системи.

Як приклад, на рис.5.2-5.5 показані графіки перехідних процесів перших чотирьох змінних стану двомасової системи по задаючому і збурюючому впливах.

У курсовому проєкті слід проаналізувати отримані графіки і зробити висновки.

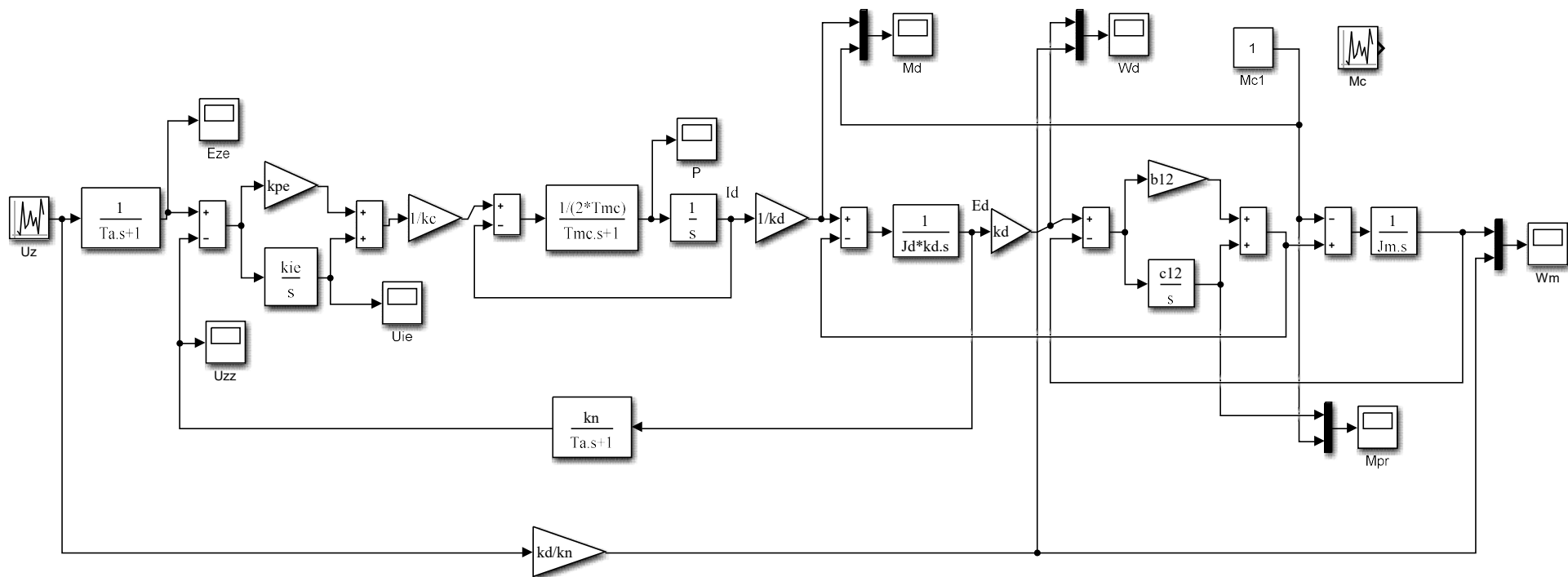
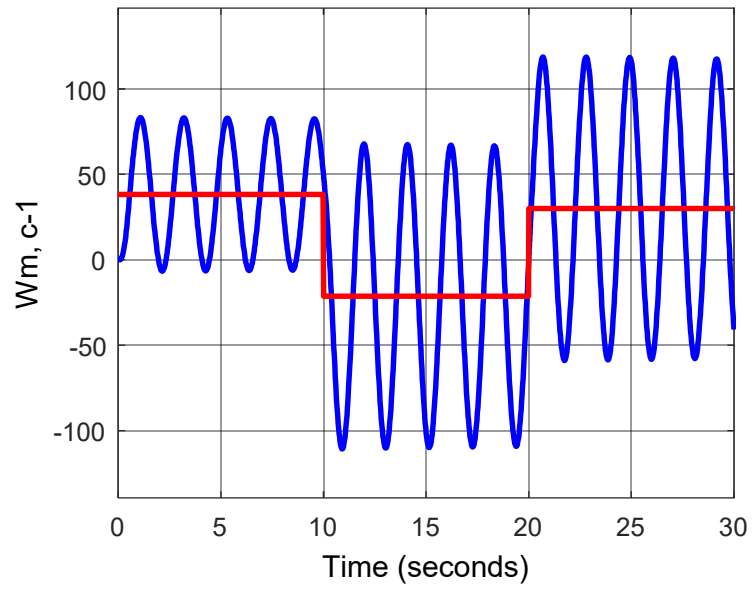
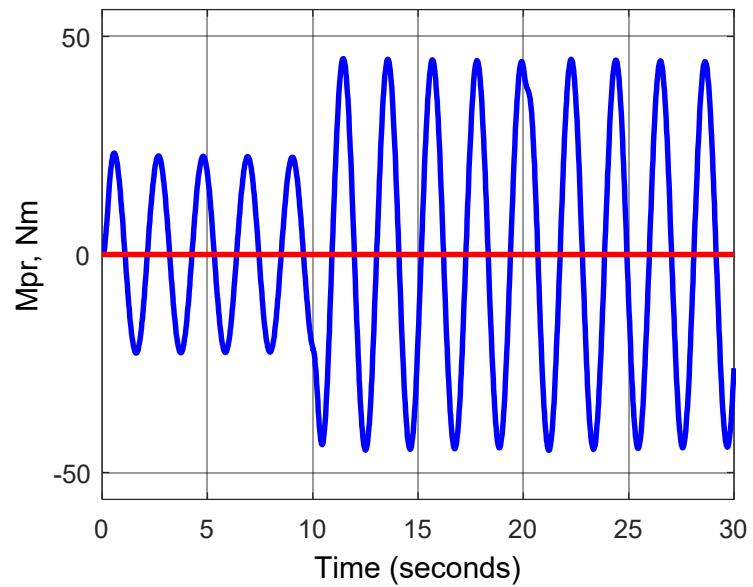


Рисунок 5.2 – Схема моделі двомасової системи, розроблена на ЕОМ

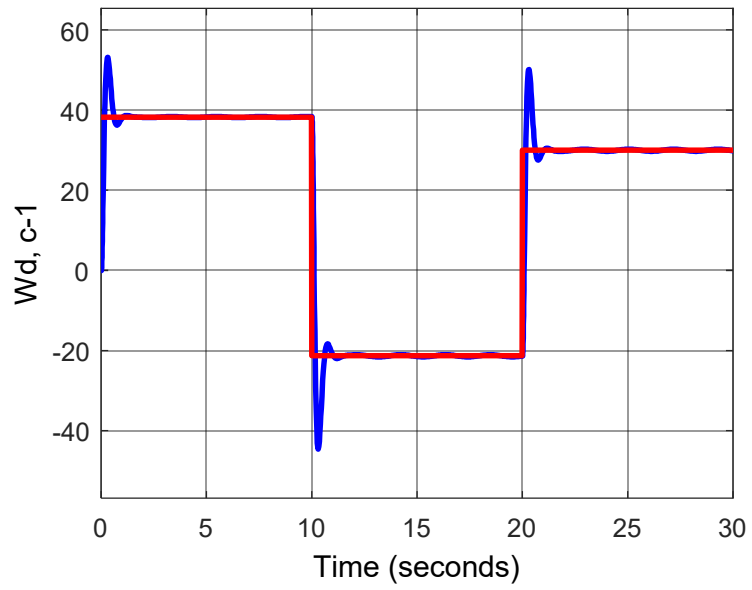


a)

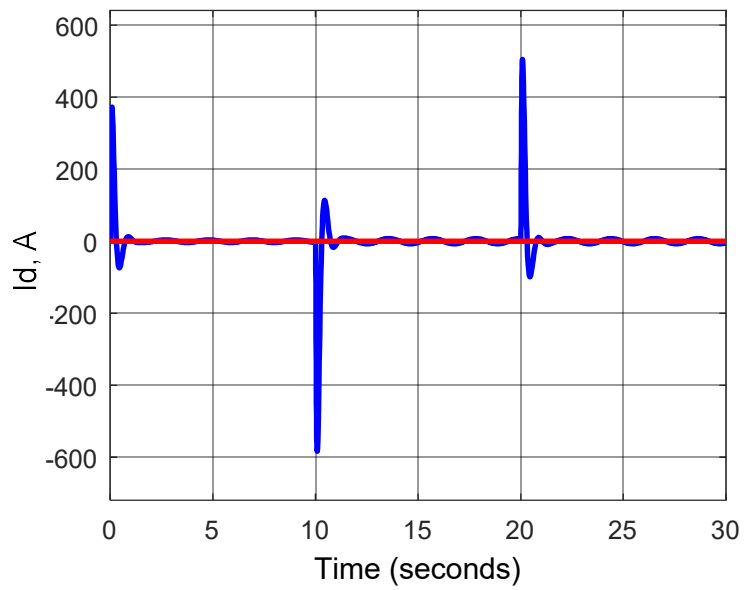


б)

Рисунок 5.3 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи по задаючій дії

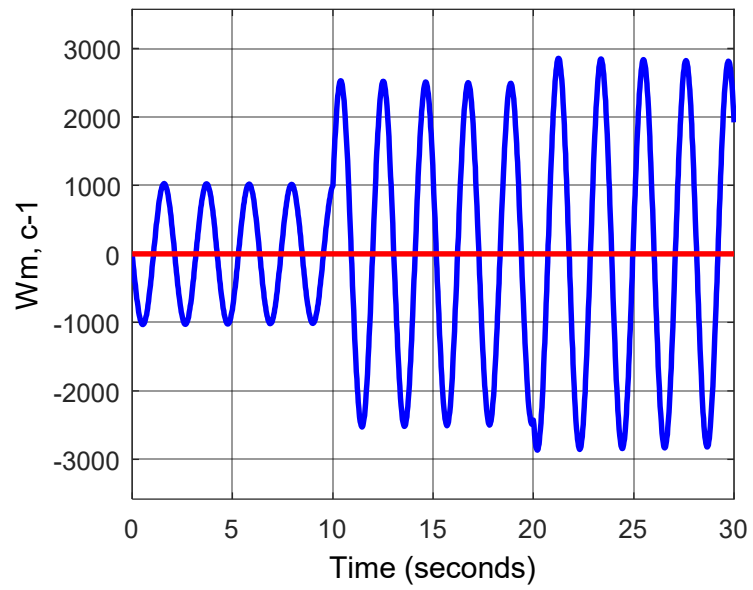


a)

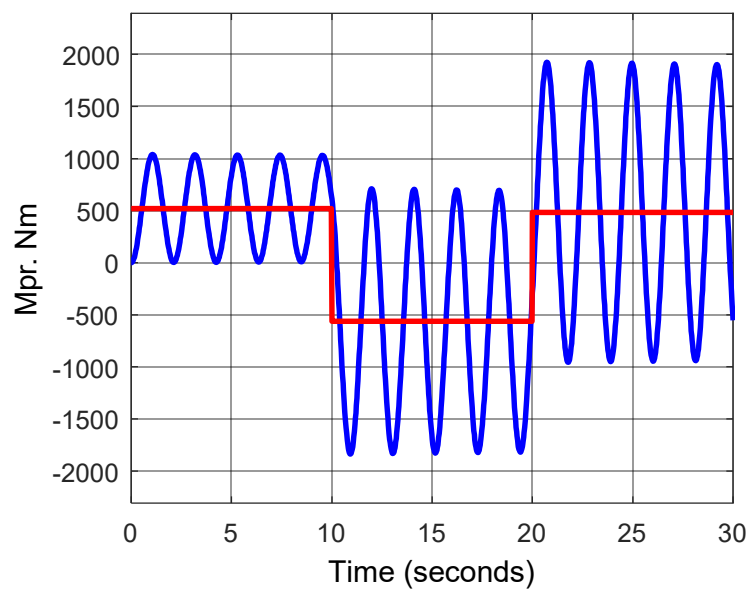


б)

Рисунок 5.4 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і струму двигуна $I_d = f(t)$ (б) двомасової системи по задаючій дії

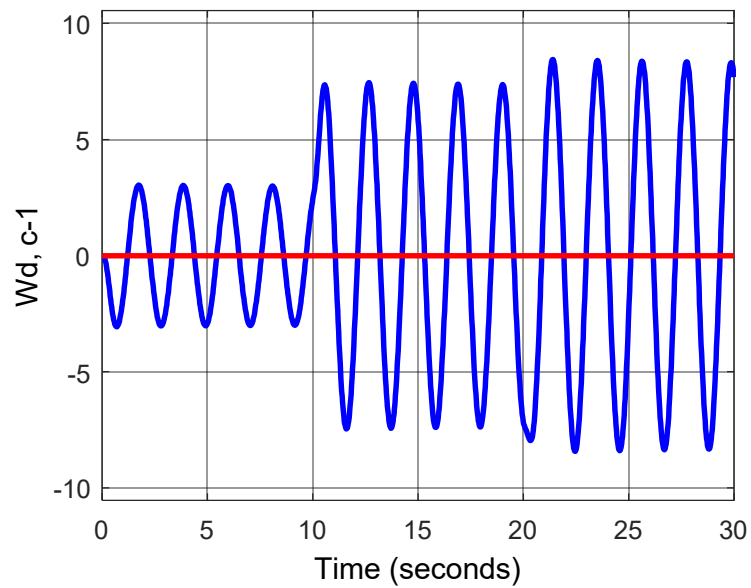


a)

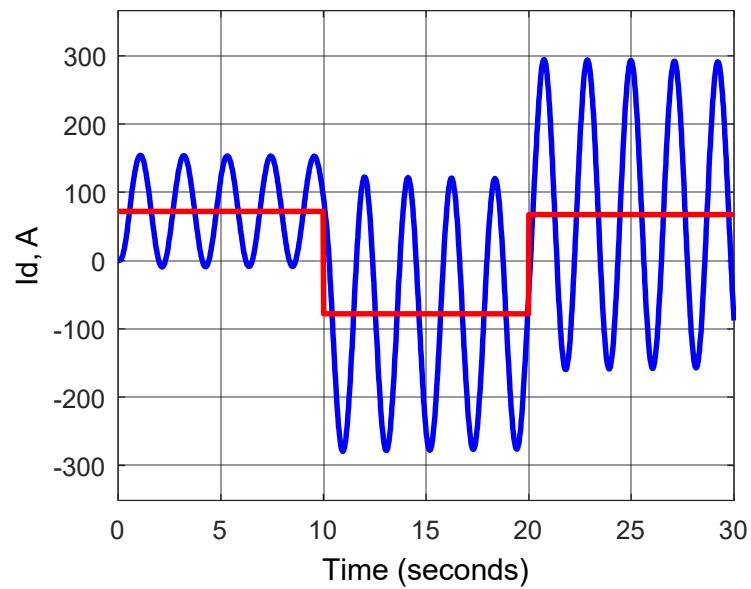


б)

Рисунок 5.5 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи по збурюючій дії



a)



б)

Рисунок 5.6 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і струму двигуна $I_d = f(t)$ (б) двомасової системи по збурюючій дії

5.4 Синтез системи управління з нейрорегулятором

5.4.1 Регулятори реалізовані в пакеті прикладних програм Neural Network Toolbox

Синтез нейромережевої системи управління проводиться з допомогою пакета прикладних програм Neural Network Toolbox системи MATLAB (або Deep Learning Toolbox починаючи з версії MATLAB R2017a). Далі приводиться коротка характеристика пакету і порядок синтезу нейрорегулятора, а також результати моделювання нейромережевої системи управління.

У [26-28] описано 3 архітектуру нейронних мереж, які реалізовані в пакеті прикладних програм (ППП) Neural Network Toolbox системи MATLAB і викаористовуються в наступних регуляторах:

- регулятора з прогнозом (NN Predictive Controller);
- регулятора на основі моделі авторегресії з ковзаючим середнім (NARMA-L2 Controller);
- регулятора на основі еталонної моделі (Model Reference Controller).

Застосування нейронних мереж для вирішення завдань управління дозволяє виділити 2 етапи проєктування:

- етап ідентифікації керованого об'єкту;
- етап синтезу закону управління.

На етапі ідентифікації розробляється модель керованого об'єкту у вигляді нейронної мережі, яка на етапі синтезу використовується для синтезу регулятора. Для кожної з трьох архітектури використовується одна і та ж процедура ідентифікації, проте етапи синтезу істотно розрізняються.

При управлінні з прогнозом моделі керованого об'єкту використовується для того, щоб передбачити його майбутню поведінку, а алгоритм оптимізації застосовується для розрахунку такого управління, яке мінімізує різницю між бажаними і дійсними змінами виходу моделі.

При управлінні на основі моделі авторегресії з ковзаючим середнім регулятор є достатньо простою реконструкцією моделі керованого об'єкту.

При управлінні на основі еталонної моделі регулятор – це нейронна мережа, яка навчена управляти об'єктом так, щоб він відстежував поведінку еталонної моделі. При цьому модель керованого об'єкту активно використовується при настройці параметрів самого регулятора.

Динамічні моделі систем управління з нейромережевими регуляторами розміщені в спеціальному розділі Control Systems набору блоків Neural Network Toolbox (рис.5.7).

Регулятор з прогнозом. Цей регулятор використовує модель керованого процесу у вигляді нейронної мережі для того, щоб передбачити майбутню реакцію процесу на випадкові сигнали управління. Алгоритм оптимізації обчислює керуючі сигнали, що мінімізують різницю між бажаною і дійсною зміною сигнальна виході моделі і таким чином оптимізують керований процес. Побудова моделі керованого процесу виконується автономно з використанням нейронної мережі, яка навчається в груповому режимі з використанням одного з алгоритмів навчання. Регулятор, що реалізує такий регулятор, потребує значного об'єму обчислень, оскільки для розрахунку оптимального закону управління оптимізація виконується на кожному такті управління.

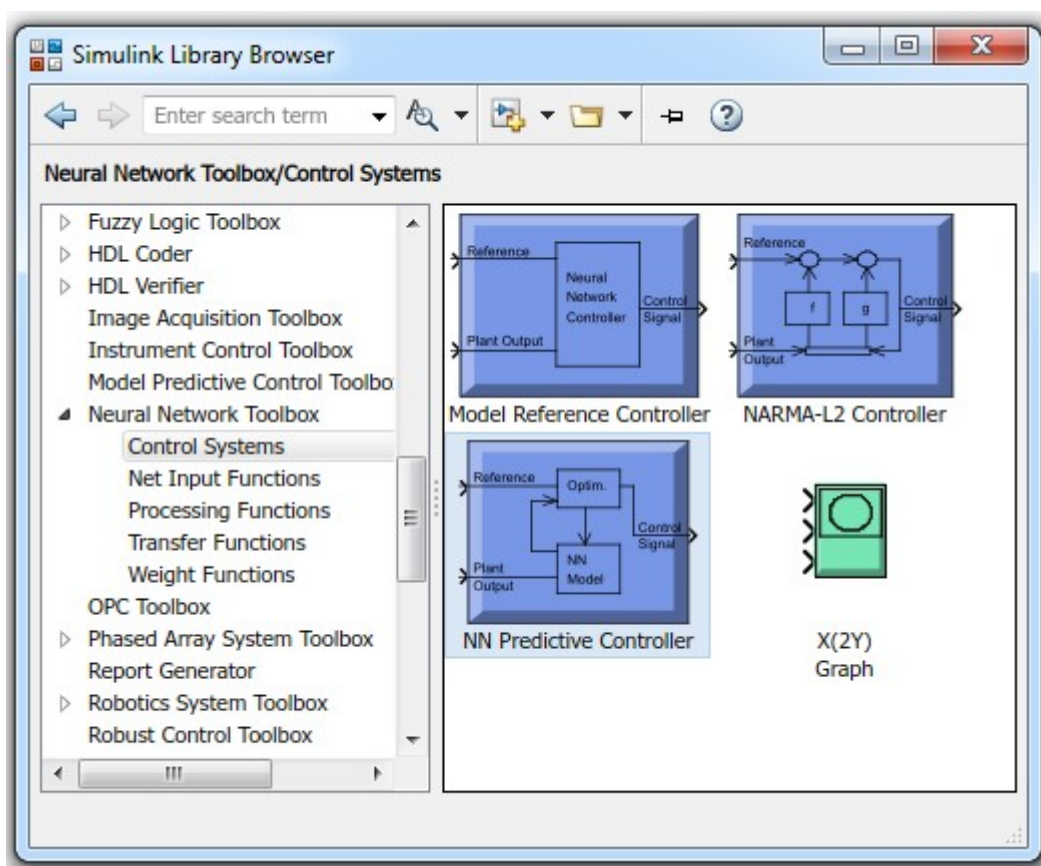


Рисунок 5.7 – Блоки Neural Network Blocksets розділу Control Systems системи SIMULINK

Регулятор на основі моделі авторегресії з ковзаючим середнім. Зі всієї архітектури цей регулятор вимагає найменшого об'єму обчислень. Даний регулятор – це просто деяка реконструкція нейромережевої моделі керованого об'єкту, отриманої на етапі автономної ідентифікації. Недолік методу полягає в тому, що модель об'єкту повинна бути задана в канонічній формі простору стану, якій відповідає супроводжуюча матриця, що може приводити до обчислювальних погрешностей.

Регулятор на основі еталонної моделі. Необхідний об'єм обчислень для цього регулятора порівнянний з попереднім. Проте архітектура регулятора з еталонною

моделлю вимагає навчання нейронної мережі об'єкту управління і нейронної мережі регулятора. При цьому навчання регулятора виявляється достатньо складним, оскільки навчання засноване на динамічному варіанті методу зворотного розповсюдження помилки. Достоїнством регуляторів на основі еталонної моделі є те, що вони застосовні до різних класів об'єктів управління.

Для системи, що розглядається в даній роботі, використовуємо регулятор з прогнозом, оскільки застосування регулятора на основі моделі авторегресії з ковзаючим середнім і регулятора на основі еталонної моделі не дало позитивних результатів.

5.4.2 Принцип побудови регулятора з прогнозом NN Predictive Controller

Регулятор з прогнозом NN Predictive Controller, реалізований в ППП Neural Network Toolbox, використовує модель нелінійного керованого об'єкту у вигляді нейронної мережі для того, щоб передбачати його майбутню поведінку. Крім того, регулятор обчислює сигнал управління, який оптимізує поведінку об'єкту на заданому інтервалі часу.

5.4.2.1 Ідентифікація об'єкту управління

Схема підсистеми ідентифікації показана на рисунку 5.8. Вона включає модель об'єкту управління у вигляді нейронної мережі, яка повинна бути навчена в автономному режимі так, щоб мінімізувати помилку між реакціями об'єкту і моделі $e = y - y_n$ на послідовність пробних сигналів u .

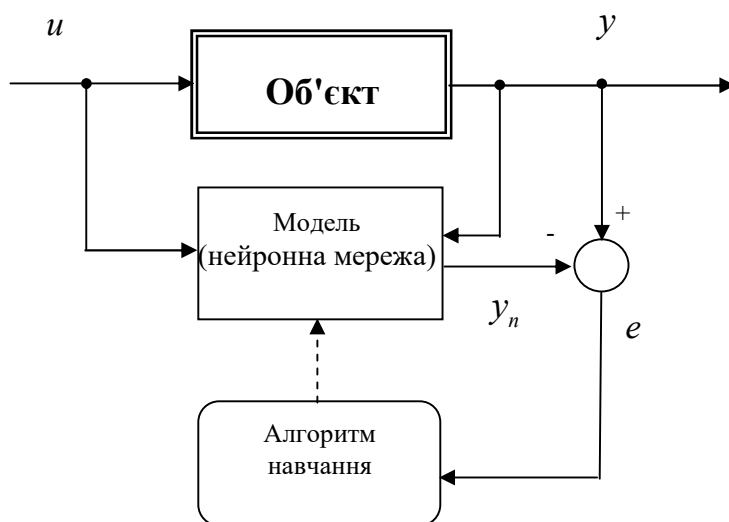


Рисунок 5.8 – Схема підсистеми ідентифікації.

Нейронна мережа регулятора об'єкту управління представлена на рис.5.9. Вона має 2 шари нейронів і використовує лінії затримки (ЛЗ), щоб запам'ятати попередні значення входів і виходів об'єкту і передбачити майбутнє значення виходу.

Настройка параметрів цієї мережі виконується автономно методом групового навчання, використовуючи дані, отримані при випробуваннях реального об'єкту. Для навчання мережі може бути використаний будь-якому з навчальних алгоритмів для нейронних мереж. У даній роботі використаний алгоритм Левенберга-Марквардта. У ППП Neural Network Toolbox даний алгоритм реалізований у вигляді М-функції `trainlm`.

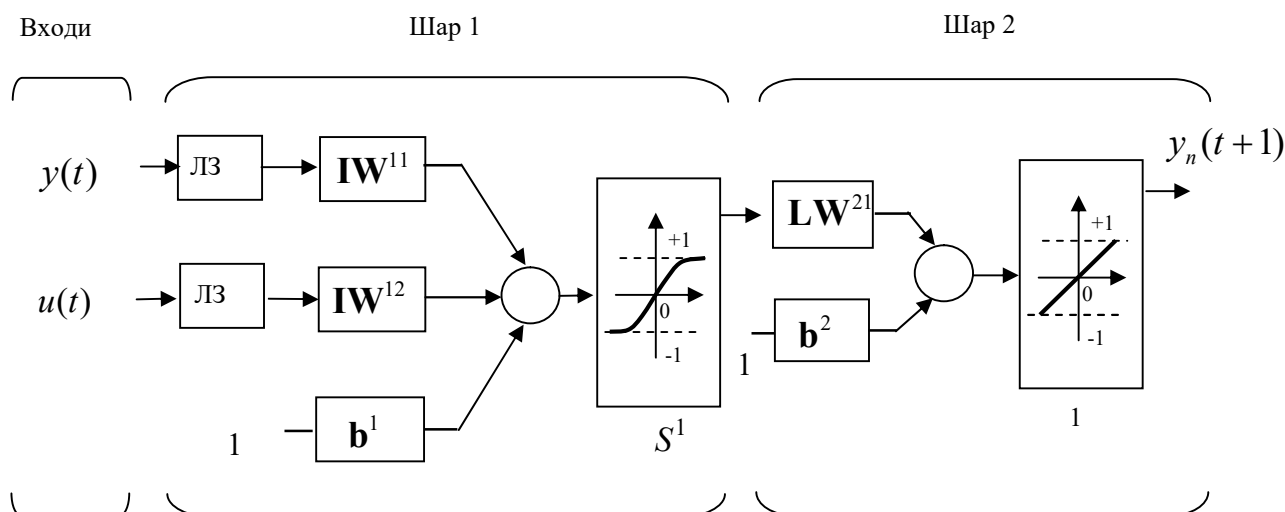


Рисунок 5.9 – Нейронна мережа регулятора керованого об'єкту

5.4.2.2 Схема управління з прогнозом

Управління з прогнозом використовує принцип горизонту, що віддаляється, коли нейрмережева модель керованого об'єкту передбачає реакцію об'єкту на певному інтервалі часу в майбутньому. Прогноз використовується програмою чисельної оптимізації для того, щоб обчислити сигнал, що управляє, який мінімізує критерій якості

$$J = \sum_{j=N_1}^{N_2} [y_r(t+j) - y_m(t+j)]^2 + \rho \sum_{j=1}^{N_4} [u'(t+j-1) - u'(t+j-2)]^2, \quad (5.1)$$

де константи N_1 , N_2 і N_4 задають межі, усередині яких обчислюються помилки стеження і потужність сигналу, що управляє. Змінна u' описує пробний сигнал, що управляє; y_r – очікувана, а y_m – дійсна реакція моделі даної системи. Величина ρ визначає внесок, який вносить потужність управління в критерій якості. Структурна схема на рис. 5.10 ілюструє процес управління з прогнозом. Регулятор складається з нейрмережевої моделі керованого об'єкту і блоку оптимізації. Блок оптимізації визначає значення u' , які мінімізують критерій якості управління, а відповідний управляючий сигнал управляє процесом.

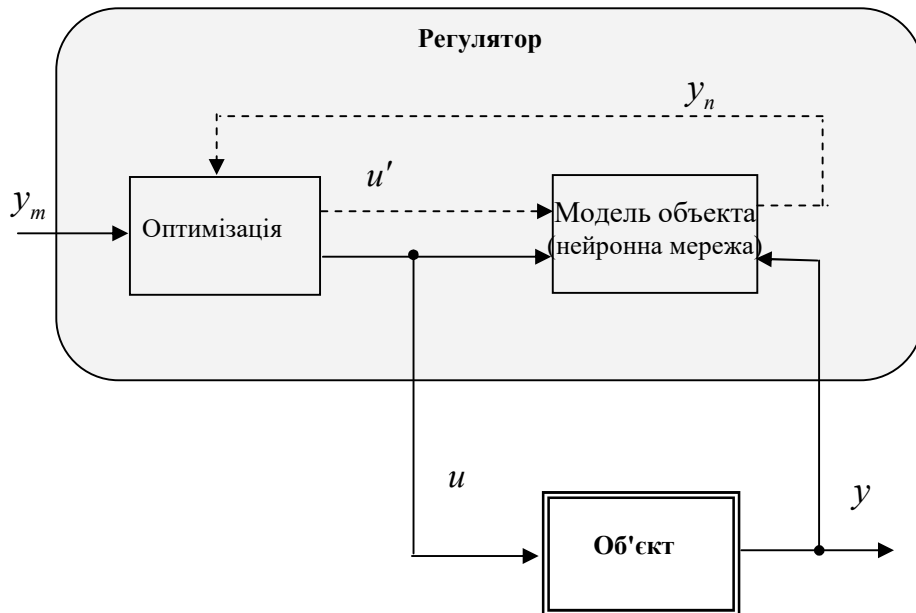


Рисунок 5.10 – Структурна схема системи з регулятором, що використовує принцип прогнозу

5.4.3 Синтез регулятора з прогнозом NN Predictive Controller з використанням системи MATLAB

При синтезі нейрорегулятора NN Predictive Controller використовуються наступні файли, розміщені в каталозі toolbox/nnet/nncontrol системи Simulink.

Функції одновимірної оптимізації:

Csrchbas – пошук із зворотним прогоном;

Csrchbre – метод Брента, об'єднуючий методи золотого перетину і квадратичної інтерполяції;

Csrchcha – метод кубічної інтерполяції Чараламбуса;

Csrchgol – метод золотого перетину;

Csrchhyb – гібридний метод бісекції і кубічної інтерполяції.

Функції для синтезу управління з прогнозом:

Calcjddj – обчислення функціонала якості і його градієнта;

Predopt – оптимізація регулятора з прогнозом;

Dyduvar – обчислення приватних похідних виходу по входу.

Моделі Simulink.

Predcstr – GUI – додаток для регулятора з прогнозом;

Prest3sim2 – нейромережева модель управління використовується М-функцією deport для прогнозу процесу в майбутньому.

Допоміжні функції:

Nncontrolutil – підтримка, що забезпечує можливість звернення до приватних функцій з системи SIMULINK;

Nnident.m – основна функція, використовується при ідентифікації об'єкту, зна-

ходиться в каталозі private. Забезпечує GUI користувача, генерацію навчальної вибірки, створення і тренування мережі.

У вікні системи Simulink формується схема системи із структурою, показаною на рисунку 5.11.

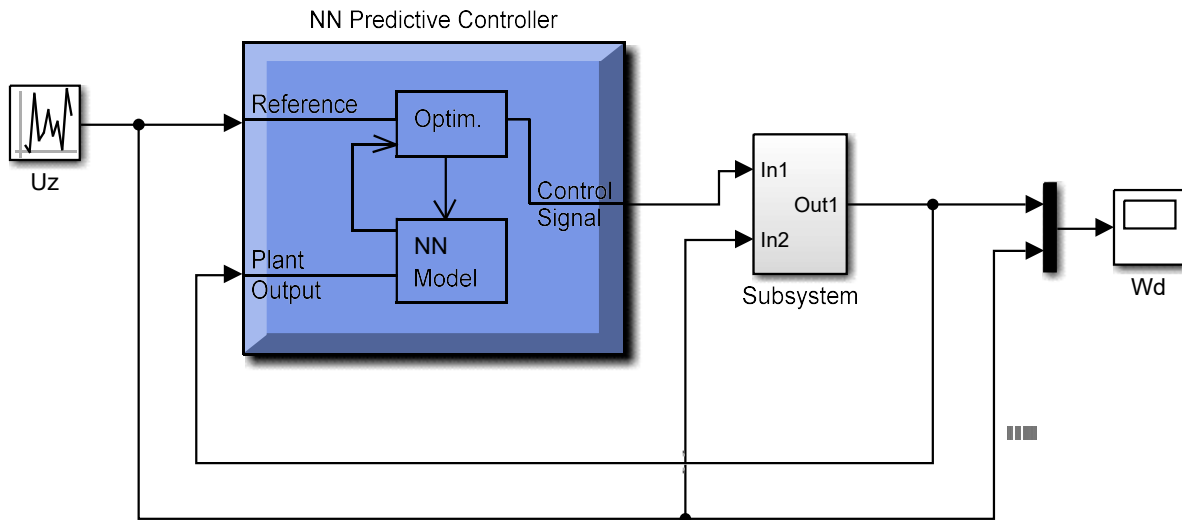


Рисунок 5.11 – Схема системи управління з нейрорегулятором NN Predictive Controller

Ця структура включає блок керованого об'єкту **Subsystem** і блок регулятора **NN Predictive Controller**, а також блоки генерації еталонного ступінчастого сигналу з випадковою амплітудою **Random Reference**, блок побудови графіків. Особливість цієї структури полягає в тому, що вона виконується не тільки функції блок-схеми

системи SIMULINK, але і функції графічного інтерфейсу користувача GUI.

Структурна схема нейрорегулятора NN Predictive Controller показана на рис. 5.12. Дана схема з'являється в окремому вікні при виборі пункту меню Look Under Mask блоку NN Predictive Controller

Процес синтезу нейрорегулятора починається шляхом активізації блоку **NN Predictive Controller**. З'являється вікно, показане на рис. 5.13. Воно виконує функції графічного інтерфейсу користувача.

В області фрейма вікна рис. 5.13 у вигляді повідомлення є інформація, яка керує подальшими діями. «Perform plant

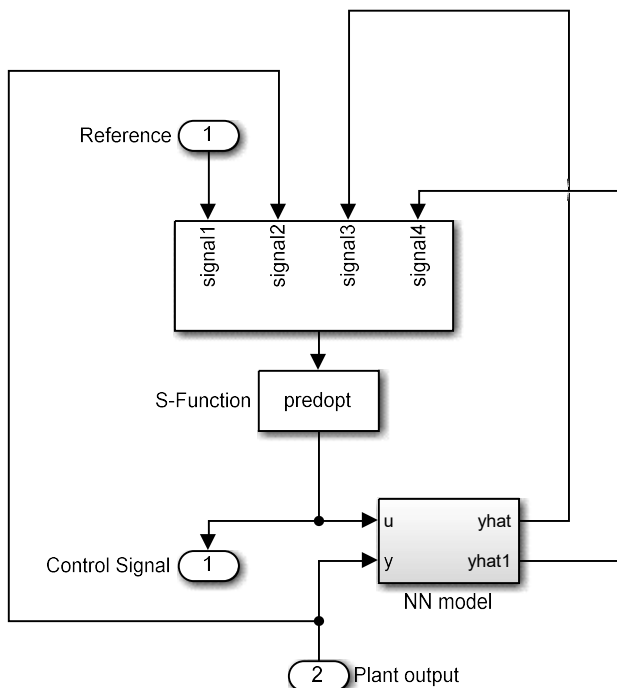


Рисунок 5.12 – Структурна схема нейрорегулятора

identification before controller configuration». Це означає, що перш за все необхідне виконати ідентифікацію керованого об'єкту, тобто побудувати його нейромережеву модель, скориставшись спеціальною процедурою **Plant Identification**.

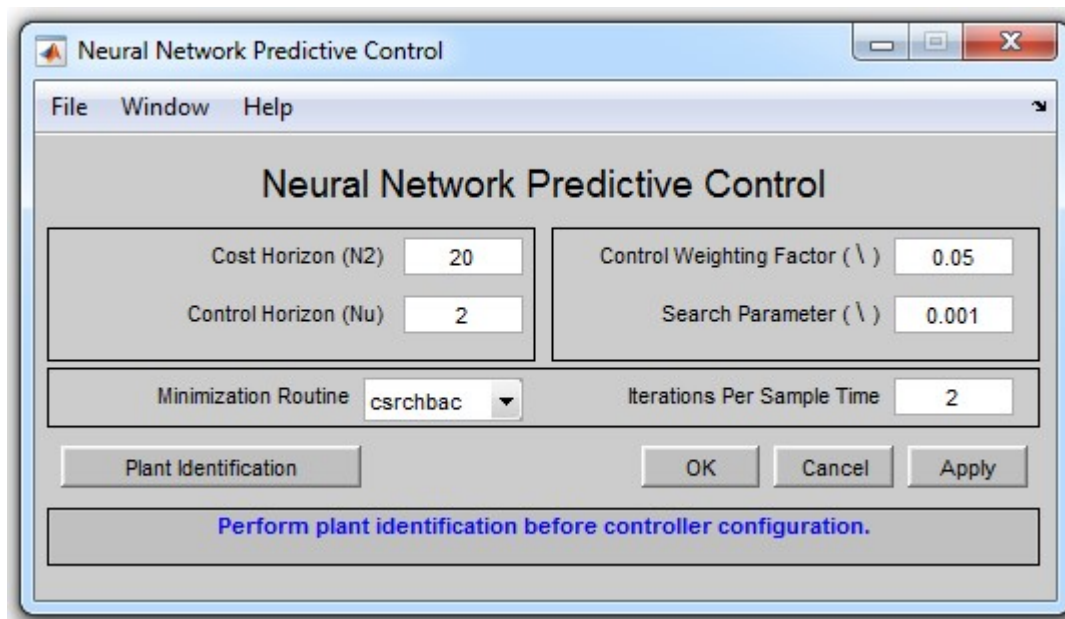


Рисунок 5.13 – Вікно завдання параметрів регулятора

Вид вікна **Plant Identification** проведений на рис. 5.14. Це вікно універсальне і може бути використано для побудови нейромережевих моделей для будь-якого динамічного об'єкту, який описаний моделлю SIMULINK. У розглянутому випадку такою моделлю є модель двомасової системи.

Процедура ідентифікації дозволяє побудувати нейронну мережу, яка моделюватиме динаміку керованого об'єкту. Оскільки модель повинна використовуватися при настройці регулятора, то її слід створити перш, ніж почнеться розрахунок регулятора.

Процедура ідентифікації вимагає завдання ряду параметрів, які задаються з використанням вікна Plant Identification. Дане вікно містить кілька секцій.

Опис вікна Plant Identification

Секція Network Architectuer (Архітектура нейронної мережі). Параметри, що адаються в цій секції визначають архітектуру нейронної мережі, яка використовується для ідентифікації і моделювання динаміки системи. Вибір правильних параметрів секції Network Architecture є ключовим етапом в ідентифікації системи, оскільки від них залежить здатність моделі адекватно відтворювати динаміку досліджуваної системи.

Size of Hidden Layer (розмір прихованого шару). Параметр Size of Hidden Layer вказує на кількість нейронів у прихованому шарі нейронної мережі, яку ви використовуєте для моделювання або ідентифікації об'єкта або системи. Вибір розміру

прихованого шару є важливим параметром при проєктуванні нейронної мережі, оскільки він впливає на потужність і здатність мережі моделювати складність взаємозв'язків у ваших даних. Зазвичай, збільшення розміру прихованого шару може допомогти нейронній мережі навчатися більш складним залежностям у даних, але воно також може призвести до перенавчання (overfitting), тобто погіршення загальної здатності мережі генералізувати результати на нові дані.

Розмір прихованого шару часто є гіперпараметром, який може бути експериментально налаштований для досягнення найкращих результатів у конкретній задачі.

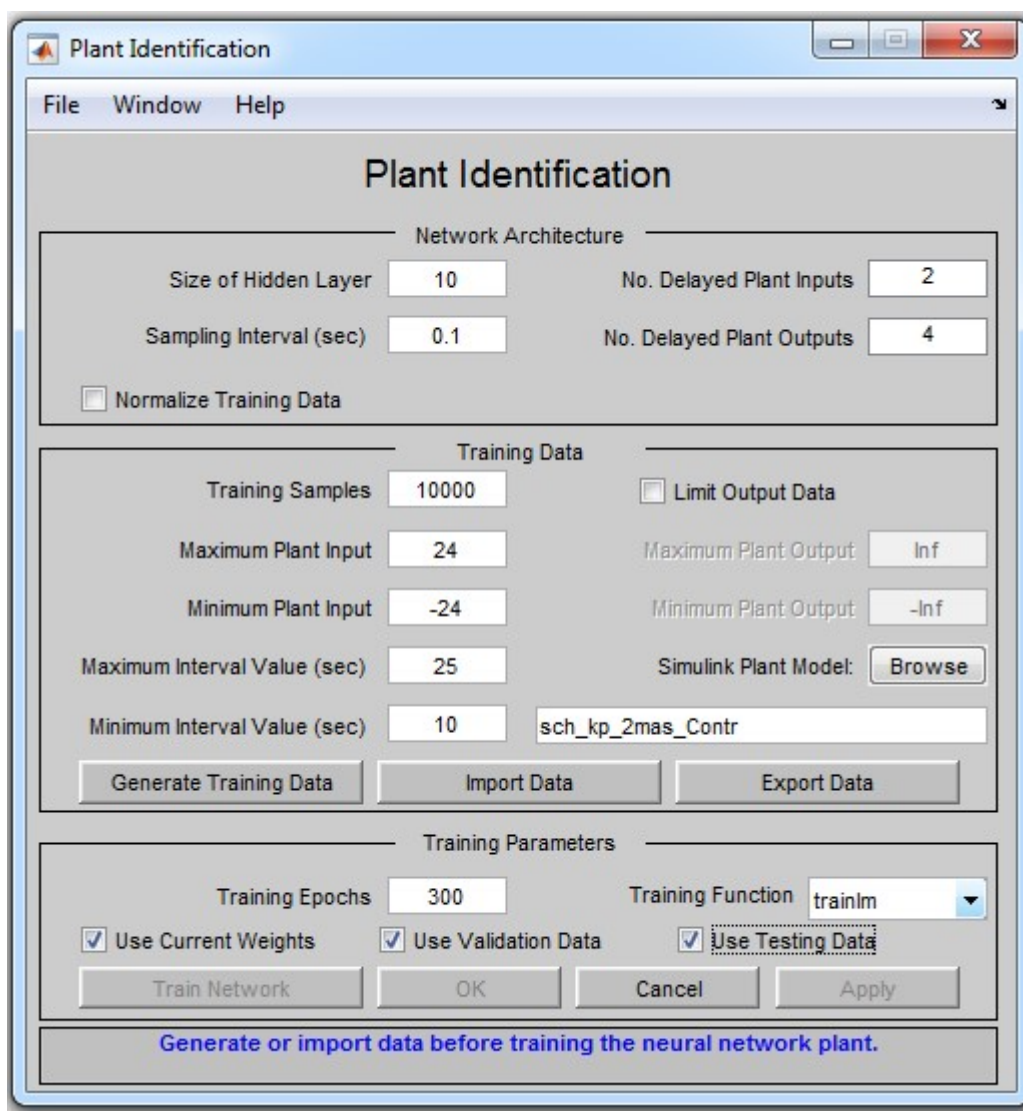


Рисунок 5.14 – Вікно ідентифікації керованої системи

Sampling Interval (sec) (інтервал вибірки в секундах). Параметр Sampling Interval (sec) визначає інтервал між двома послідовними моментами знімання даних. Параметр важливий при зборі даних для моделювання або ідентифікації системи, оскільки впливає на точність та роздільну здатність зібраних даних. Для точної ро-

боти моделі ідентифікації важливо визначити правильний інтервал вибірки, який враховує динаміку системи і вимоги до точності результатів ідентифікації.

No. Delayed Plant Inputs (кількість затриманих входів системи). Параметр No. Delayed Plant Inputs вказує на кількість запізнених (затриманих) входів системи, які враховуються при ідентифікації динаміки системи. Запізнені входи враховують затримки у вхідних сигналах системи і допомагають створити більш точну модель системи.

Запізнені входи особливо важливі в ідентифікації систем, де існують затримки між вхідними сигналами та відповідними виходами. Наприклад, в автоматичному керуванні часто є затримки в активних процесах, і ідентифікація таких систем може вимагати врахування цих затримок.

Кількість No. Delayed Plant Inputs вказує, скільки вхідних сигналів системи вважаються запізненими і будуть використовуватися при створенні моделі системи. Від правильного визначення цього параметра залежить точність ідентифікації системи, особливо в умовах існування затримок між сигналами.

Цей параметр важливий для налаштування ідентифікаційного процесу та допомагає створити більш адекватну модель системи, яка може враховувати реальні умови та характеристики системи керування.

No. Delayed Plant Outputs (кількість затриманих виходів системи). Параметр No. Delayed Plant Outputs вказує на кількість запізнених (затриманих) виходів системи, які враховуються при ідентифікації динаміки системи. Запізнені виходи враховують затримки у вихідних сигналах системи і допомагають створити більш точну модель системи.

Цей параметр дозволяє визначити, скільки затриманих виходів системи вважаються при ідентифікації і впливає на точність та адекватність створеної моделі. Затримані виходи дозволяють враховувати затримки в вихідних сигналах, які можуть бути важливими для ефективної ідентифікації та моделювання системи.

Вибір кількості No. Delayed Plant Outputs залежить від конкретних характеристик досліджуваної системи та вимог до точності ідентифікації.

Normalize Training Data (нормалізація даних навчання). Вікна контролю нормування навчальних даних. Normalize Training Data вказує на необхідність нормалізації даних під час навчання моделі системи. Нормалізація даних - це процес масштабування вхідних та вихідних даних системи до певного діапазону або стандартної форми перед подачею їх на вхід моделі для навчання.

Якщо вікно увімкнено, то вхідні та вихідні дані будуть нормалізовані перед початком процесу навчання моделі. Нормалізація даних може включати в себе масштабування даних до діапазону від 0 до 1, використання стандартного відхилення та середнього значення для центрування даних, або інші методи нормалізації.

Якщо вікно не увімкнено, то нормалізація даних не буде застосована, і дані будуть використовуватися без змін.

Мета нормалізації даних під час навчання полягає в тому, щоб забезпечити стабільність та швидкість навчання моделі, особливо коли вхідні дані мають різний масштаб або діапазон значень. Нормалізація може полегшити збіжність навчання та поліпшити точність ідентифікації системи.

Вибір варіанта Normalize Training Data повинен здійснюватися відповідно до характеру та потреб конкретної задачі ідентифікації системи

Секція Training Data (дані навчання). Training Data - це набір даних, які використовуються для навчання моделі ідентифікації системи. Ці дані представляють собою вхідні та вихідні сигнали системи, і їх використовують для створення математичної моделі системи.

Основна мета Training Data полягає в навчанні моделі системи таким чином, щоб вона могла адекватно відтворювати динаміку реальної системи. Дані включають в себе виміряні значення входів та виходів системи протягом певного періоду часу або при певних умовах.

Зазвичай Training Data містять дані з різних режимів роботи системи, щоб модель була здатна відтворювати різноманітні аспекти системи. Важливо, щоб ці дані були репрезентативними та відображали різноманітні умови функціонування системи.

На основі цих даних проводиться процес ідентифікації, під час якого створюється модель, яка намагається знайти математичні зв'язки між вхідними та вихідними сигналами системи. Точність та адекватність цієї моделі залежать від якості та репрезентативності Training Data.

Training samples (навчальні зразки). Довжина навчальної вибірки (кількість точок знімання інформації). Навчальна вибірка включає в себе вхідні та вихідні дані, які використовуються для створення моделі системи. Довжина навчальної вибірки грає важливу роль у процедурі ідентифікації об'єкта і створенні адекватної моделі системи.

Maximum Plant Input (максимальне значення вхідного сигналу). Параметр Maximum Plant Input вказує на максимальне значення вхідного сигналу (величини, яка керує системою) під час ідентифікації динаміки системи. Цей параметр встановлює верхню межу діапазону значень вхідного сигналу, які будуть використовуватися під час навчання моделі.

Максимальне значення вхідного сигналу важливо вказати для того, щоб навчальний процес був обмежений реалістичними значеннями вхідного сигналу. Вказане значення може відображати фізичні обмеження системи або діапазони значень, які найчастіше використовуються в реальних умовах.

Важливо правильно налаштувати цей параметр, оскільки надмірно велике значення може призвести до надмірного навчання моделі на діапазонах вхідних сигна-

лів, які надто віддалені від реальних умов. Це може призвести до некоректних прогнозів та погіршення якості моделі системи.

Minimum Plant Input (мінімальне значення вхідного сигналу). Параметр Minimum Plant Input вказує на мінімальне значення вхідного сигналу (величини, яка керує системою) під час ідентифікації динаміки системи. Цей параметр встановлює нижню межу діапазону значень вхідного сигналу під час навчання моделі.

Максимальне і мінімальне значення вхідного сигналу важливо вказати для того, щоб навчальний процес був обмежений реалістичними значеннями вхідного сигналу. Вказане значення може відображати фізичні обмеження системи або діапазони значень, які найчастіше використовуються в реальних умовах. Неправильне встановлення цих параметрів може призвести до некоректних прогнозів та погіршення якості моделі системи.

Maximum Interval Value (sec) (максимальне значення інтервалу). Параметр Maximum Interval Value (sec) вказує на максимальну довжину інтервалу часу в секундах, протягом якого використовуються вхідні та вихідні дані для ідентифікації динаміки системи, тобто максимальний інтервал ідентифікації. Цей параметр дозволяє обмежити часовий період, на який поширюється навчальний набір даних, використовуваний під час ідентифікації.

Вказане максимальне значення в секундах служить для обмеження часового діапазону аналізу інформації. Це корисно в ситуаціях, коли вхідні та вихідні дані мають значення протягом певного часового інтервалу, і деякий період часу може бути менш важливим для ідентифікації системи.

Зазвичай, під час встановлення параметру Maximum Interval Value, враховуються характеристики конкретної ідентифікаційної задачі. Навчальний набір даних може включати в себе декілька циклів або періодів роботи системи, і цей параметр допомагає обмежити аналіз до конкретного періоду часу.

Minimum Interval Value (sec) (мінімальне значення інтервалу). Параметр Minimum Interval Value (sec) вказує на мінімальну довжину інтервалу часу в секундах, протягом якого використовуються вхідні та вихідні дані для ідентифікації динаміки системи, тобто мінімальний інтервал ідентифікації. Цей параметр дозволяє обмежити часовий період, на який поширюється навчальний набір даних, використовуваний під час ідентифікації.

Мінімальне значення вказує на мінімальний період, який вважається допустимим для аналізу даних. Налаштування Minimum Interval Value (sec) допомагає контролювати, які частини часового ряду вважати допустимими для використання під час ідентифікації, і може допомогти враховувати тільки ті інтервали, які мають певну мінімальну тривалість або важливість.

Цей параметр важливий для налаштування ідентифікаційного процесу відповідно до конкретних вимог і особливостей досліджуваної системи.

Limit Output Data (обмеження вихідних даних). Вікно контролю, що дозволяє обмежити об'єм вихідних даних, тільки при включеному вікні будуть доступні 2 наступних вікна редагування тексту.

Maximum Plant Output. Максимальне значення вихідного сигналу.

Minimum Plant Output. Мінімальне значення вихідного сигналу.

Параметри Maximum Plant Output і Minimum Plant Output вказують на обмеження вихідних даних, які використовуються під час ідентифікації динаміки системи. Ці параметри можуть бути використаними для вибору конкретного діапазону вихідних даних для аналізу та навчання моделі системи.

Параметри Maximum Plant Output і Minimum Plant Output корисні, коли потрібно провести аналіз лише на певному діапазоні вихідних даних, або коли дослідника цікавить певний аспект системи, що відбувається в певний період часу.

Параметри Maximum Plant Output і Minimum Plant Output дозволяють зменшити обсяг даних, які використовуються для ідентифікації, і спростити процес аналізу та навчання моделі системи.

Simulink Plant Model (Simulink модель об'єкту). Завдання моделі Simulink з вказівкою вхідних і вихідних портів, використовуваних при побудові нейромережевої моделі керованої системи. Вказується модель об'єкта або системи, яка використовується для ідентифікації.

За допомогою кнопки **Browser** вибирається модель Simulink об'єкту управління; в даній роботі це схема моделі, показана на рис. 5.15.

Generate Training Data (генерація навчальних даних). Кнопка запуску процесу генерації навчальної послідовності, тобто створення навчальних даних для ідентифікації динаміки системи на основі параметрів та моделі системи. Ця функція дозволяє симулювати дані для навчання моделі, коли реальні навчальні дані не доступні.

Import Data. Імпорт навчальної послідовності з робочої області або файлу даних.

Export Data. Експорт даних, що згенерували, в робочу область або MAT – файл.

Секція Training Parameters (параметри навчання). Training Parameters - це набір налаштувань, які визначають, як саме буде проводитися процес навчання моделі системи на основі навчальних даних. Ці параметри дозволяють керувати процесом навчання ідентифікаційної моделі та впливати на його результати.

Training Epochs (епохи навчання). Training Epochs вказують на кількість ітерацій (повних циклів навчання) для навчання моделі або алгоритму ідентифікації. Кожна епоха представляє собою одну ітерацію, під час якої модель навчається на підготовчих даних та оновлює свої внутрішні параметри, щоб покращити свою точність.

Кількість Training Epochs визначає, скільки разів модель або алгоритм пройде через навчальні дані для покращення своєї здатності до відтворення динаміки системи. Зазвичай навчання включає багато епох, і після кожної епохи модель оцінюється на тестових даних, щоб визначити, наскільки добре вона справляється з завданням ідентифікації.

Вибір кількості Training Epochs може бути важливим параметром при налаштуванні процесу навчання. Занадто мала кількість епох може призвести до недоосвіченої моделі, яка не здатна адекватно відтворити динаміку системи, тоді як занадто велика кількість епох може призвести до перенавчання (overfitting) моделі на навчальних даних, що погіршує її здатність до узагальнення на нові дані. Точна кількість епох може варіюватися в залежності від конкретної задачі ідентифікації.

Training function (навчальна функція). Training Function вказує на функцію, яка використовується для навчання нейромережевої моделі об'єкта або системи. Ця функція визначає, яким чином модель навчається на навчальних даних та які параметри моделі оптимізуються під час навчання.

Вибір правильної Training Function важливий для успішного навчання моделі. У MATLAB і Neural Network Toolbox, доступні різні Training Functions для нейронних мереж та інших моделей ідентифікації, наприклад: `traingd` - градієнтний спуск; `traingdm` - градієнтний спуск з моментом; `trainlm` - метод найменших квадратів Левенберга-Марквардта; `trainbr` - байєсівський регулятор.

Кожна з цих функцій має свої особливості та параметри, які можна налаштувати для досягнення найкращих результатів в конкретній задачі ідентифікації. Training Function визначає, як модель або алгоритм оновлює внутрішні параметри під час навчання та яким чином він збільшує точність та здатність моделі до адекватного відтворення динаміки системи.

Use Current Weights (використовувати поточні ваги). Вікно контролю, що дозволяє підтвердити використання поточних вагів нейронної мережі.

Параметр Use Current Weights вказує, чи слід використовувати поточні ваги для ідентифікації системи під час навчання моделі. Цей параметр впливає на початкову точку або ініціалізацію процесу навчання.

Якщо Use Current Weights встановлено в «Так» (або включено), то попередні ваги (параметри моделі) вважаються початковими значеннями для нового процесу навчання. Це означає, що процес навчання розпочнеться з вагами, які вже були встановлені або навчені під час попередніх ітерацій.

Якщо Use Current Weights встановлено в «Ні» (або виключено), то навчання розпочнеться зі звичайними початковими значеннями, які зазвичай є випадковими або ініціалізованими за допомогою певних методів.

Використання поточних ваг при навчанні може бути корисним, якщо необхідно покращити або доповнити існуючу модель, або якщо відомо, що поточні ваги вже

набули певного рівня адекватності. Важливо розуміти, як цей параметр впливає на процес навчання та на результати ідентифікації системи.

Use Validation Data (використовувати дані валідації). Вікно контролю, що дозволяє вказати, чи слід використовувати окремий набір даних для валідації моделі під час навчання ідентифікаційної системи. Валідація - це процес оцінки якості побудованої моделі на основі навчальних даних.

Якщо Use Validation Data встановлено в «Так» (або включено), то під час навчання моделі буде використовуватися окремий набір даних, який не брав участь в процесі навчання. Ці дані використовуються для оцінки того, наскільки добре модель узгоджується з даними, які вона раніше не бачила. Використання валідаційних даних допомагає виявити ознаки перенавчання моделі та визначити, наскільки добре модель узгоджується з новими вхідними даними.

Якщо Use Validation Data встановлено в «Ні» (або виключено), то валідація не буде використовуватися, і модель буде оцінюватися лише на основі навчальних даних. Використання валідаційних даних рекомендується для кращої оцінки якості моделі та попередження перенавчання, але варіант без валідації може бути використаний, коли недоступні додаткові дані для цієї цілі.

Use Testing Data (використовувати тестові дані). Вікно контролю, що дозволяє вказати, чи слід використовувати окремий набір тестових даних для оцінки якості побудованої моделі системи після її навчання. Тестові дані - це дані, які модель не бачила під час навчання або валідації.

Якщо Use Testing Data встановлено в «Так» (або включено), то після завершення навчання і валідації моделі буде використовуватися окремий набір тестових даних для оцінки якості моделі. Ці дані допомагають визначити, наскільки добре модель здатна передбачати реальну динаміку системи, якщо у неї немає попередніх знань про ці дані.

Якщо Use Testing Data встановлено в «Ні» (або виключено), то тестові дані не використовуються, і модель оцінюється лише на основі навчальних і, можливо, валідаційних даних.

Використання тестових даних важливе для об'єктивної оцінки якості моделі та підтвердження її ефективності на нових, незалежних вибірках даних. Тестові дані допомагають уникнути перенавчання та підтвердити, наскільки добре модель може працювати на нових вхідних даних.

Вибір процедури **Generate Training Data** приведе до того, що буде запущена програма генерації навчальної послідовності. Програма генерує навчальні дані шляхом дії ряду випадкових ступінчастих сигналів на модель Simulink керованого об'єкту. Графіки вхідного і вихідного сигналів об'єкту управління виводиться на екран (рис.5.16). Після закінчення генерації навчальної послідовності передбачається або прийняти дані (**Accept Data**), що згенерували, або відмовитися від них(**Reject Data**).

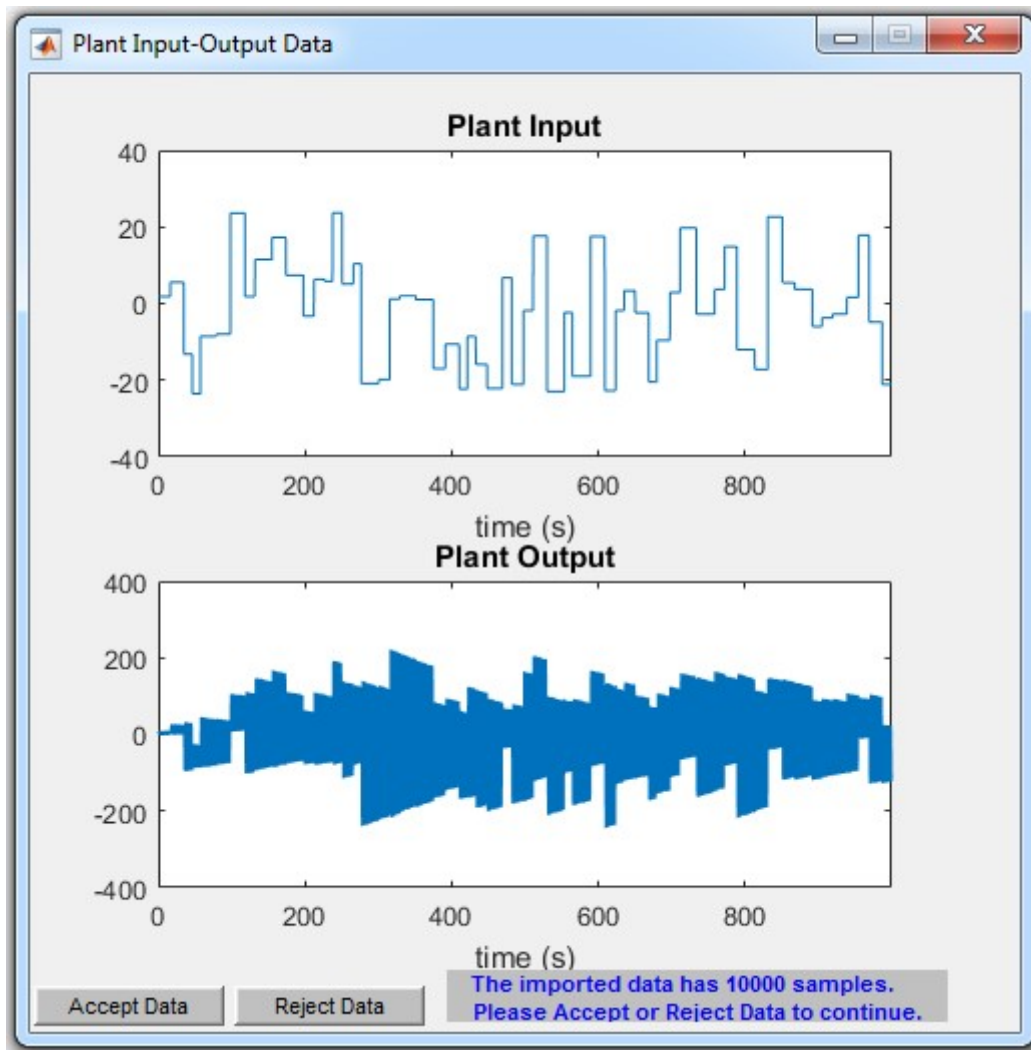


Рисунок 5.16 – Графіки вхідного і вихідного сигналів при генерації навчальної послідовності

Якщо приймаються дані, програма повертає до декілька зміненого вікна Plant Identification (рис. 5.17). Тут частина вікон недоступні, а кнопка Generate Training Data замінена на кнопку **Erase Generate Data**, що дозволяє видалити дані, що згенерували.

Після натиснення на кнопку **Train Network** відбувається створення мережі з прямою передачею сигналу за допомогою М-функції newff. Дана функція не тільки створює мережу з ім'ям netn, але і ініціалізує її ваги і зсуви і готує нейронну мережу до навчання. Модель нейронної мережі може бути побудована в системі Simulink за допомогою оператора gensim(netn) (рис.5.18).

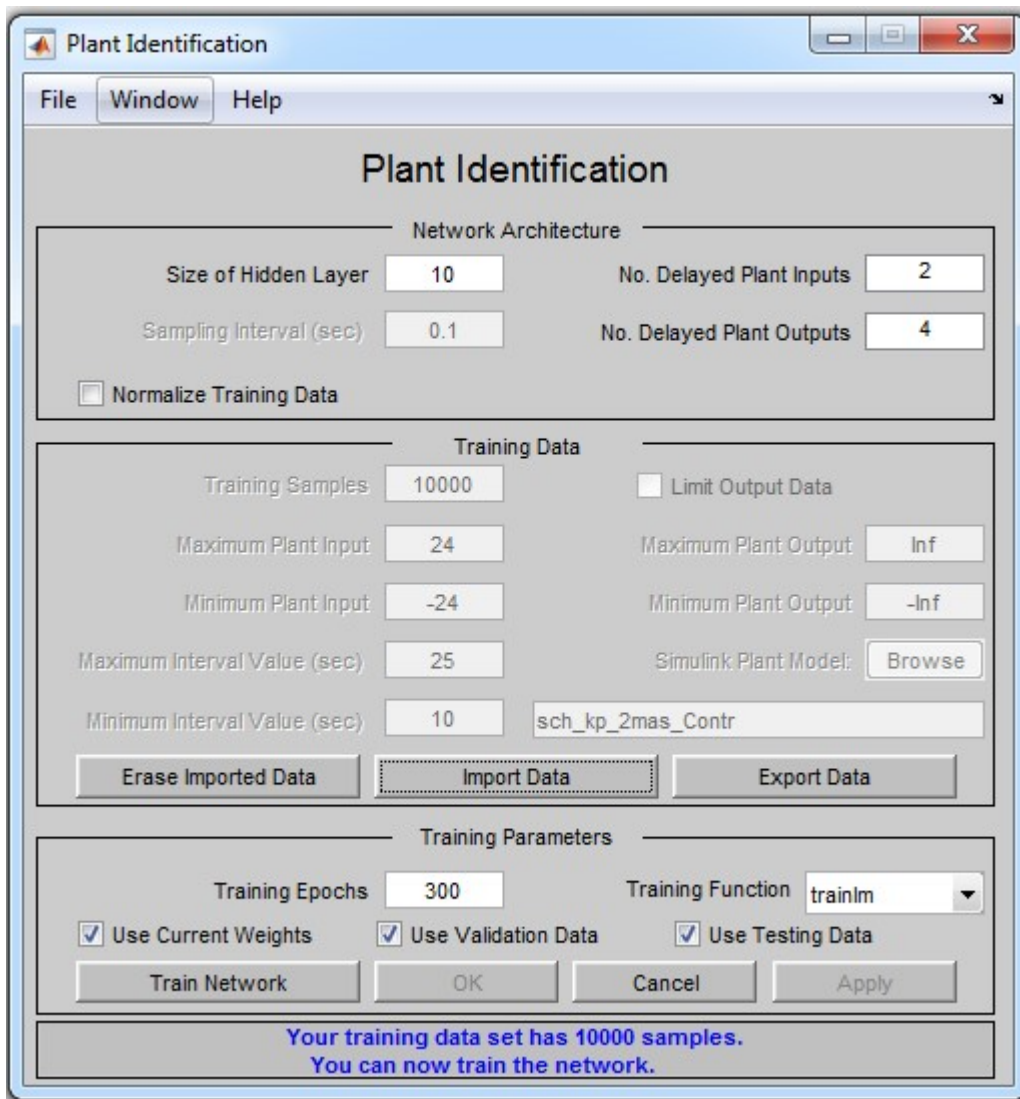


Рисунок 5.17 – Вид вікна Plant Identification після генерації навчальної послідовності

Елементи нейронної мережі, відповідають наступним параметрам: розмір прихованого шару $S=12$, кількість елементів запізнювання на вході моделі $N_i=2$, кількість елементів запізнювання на виході моделі $N_j=4$. Кожен подальший елемент з'являється в окремому вікні при активізації попереднього подвійним клацанням миші. З даних елементів в системі Simulink може бути побудована схема мережі, показана на рис.5.19.

Дана мережа є статичною (статична мережа характеризується тим, що в її складі немає елементів запізнювання і зворотних зв'язків). Мережа використовує 1 вектор входу з 6 елементами, має 2 шари з 12 нейронами в першому (прихованому) шарі і 1 нейроном в другому (вихідному) шарі. Використовувані функції активації: гіперболічного тангенса (tansig) – в першому шарі, лінійна (purelin) – в другому шарі.

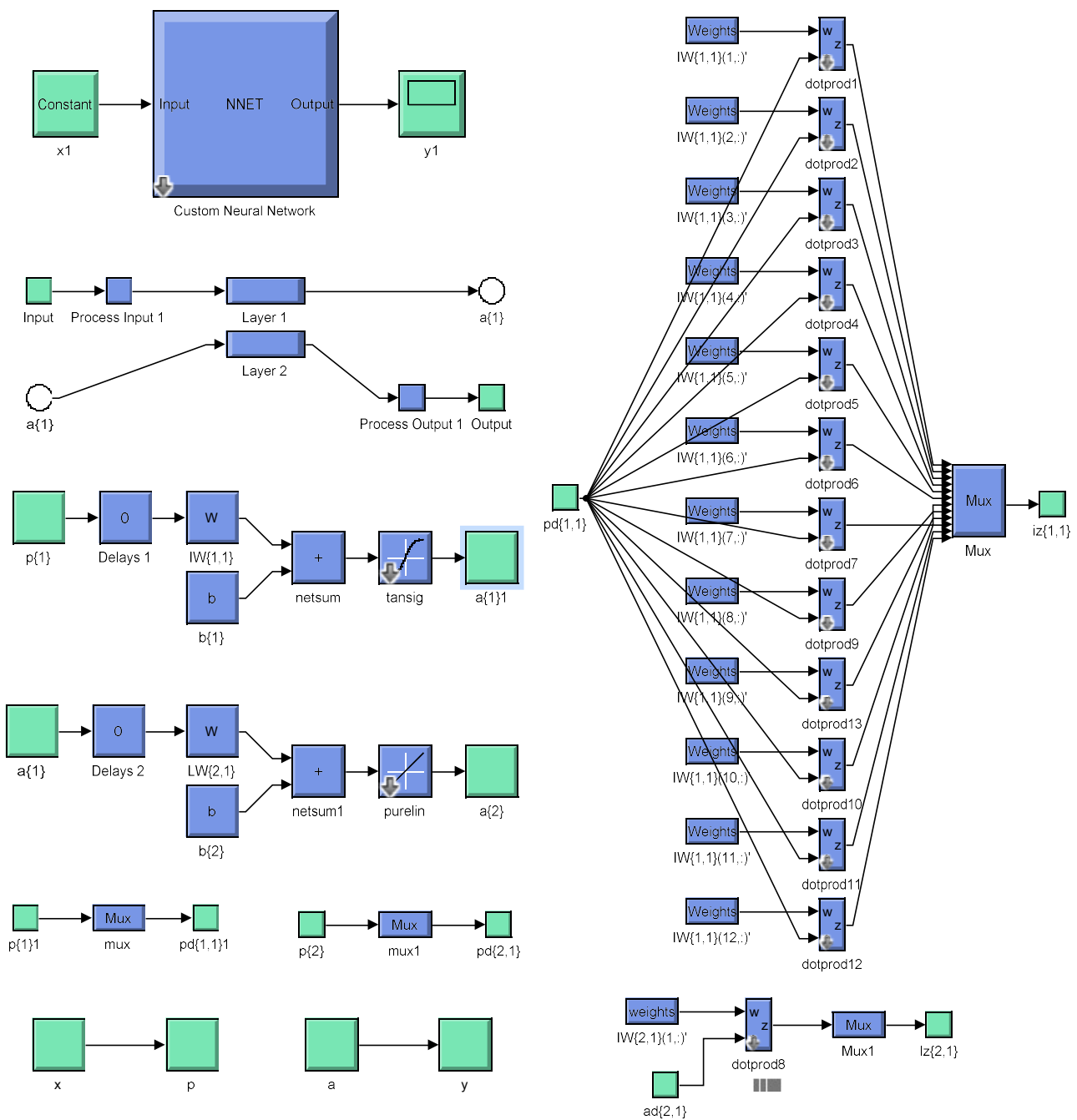


Рисунок 5.18 – Моделі елементів мережі з прямою передачею сигналу, реалізовані в Simulink

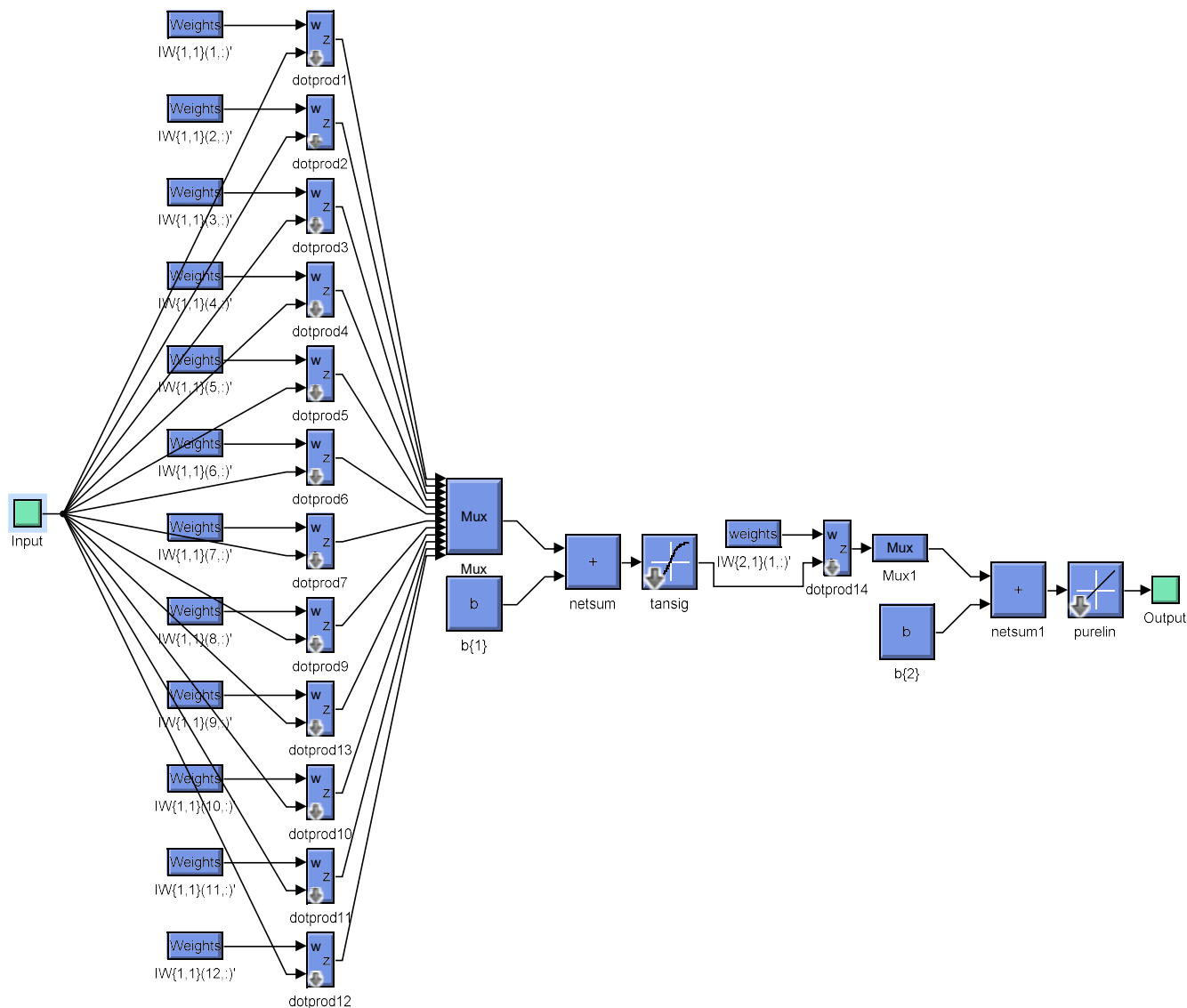


Рисунок 5.19 – Модель статичної мережі з прямою передачею сигналу, побудована в Simulink

Після створення мережі починається процес її навчання. Процес та результати навчання наочно демонструються за допомогою діалогового вікна **Neural Network Training (nntraintool)**, показаного на рис.5.20. Наведене вікно - це інтерфейс для навчання нейронних мереж у Neural Network Toolbox в MATLAB. Воно є потужним інструментом для навчання і налаштування нейронних мереж з візуалізацією та можливістю моніторингу процесу навчання.

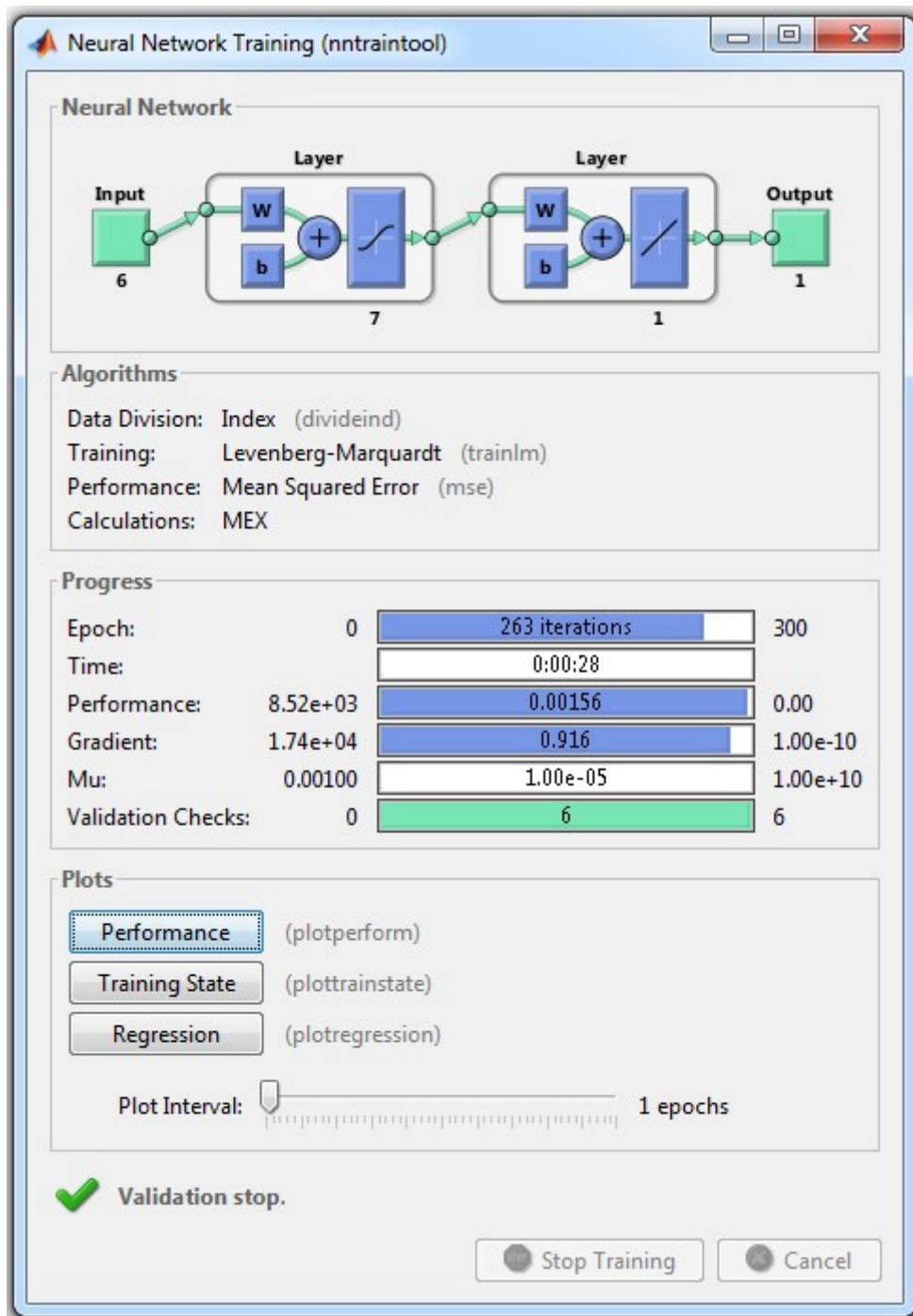


Рисунок 5.20 - Головне вікно, що відображає хід навчання та результати навчання нейронної мережі

Онус вікна Neural Network Training (nntraintool)

Вікно Neural Network Training (nntraintool) має наступні секції.

Секція Neural Network (нейронна мережа). У секції наводиться структура згенерованої нейронної мережі.

Секція Algorithms (алгоритми). Секція призначена для відображення різних алгоритмів та параметрів, які використовуються під час навчання нейронної мережі.

Data Division (ділення даних). Параметр Data Division: Index (поділ даних: ін-

декс) вказує на конкретні індекси прикладів даних, які використовуються для навчання, перевірки та тестування нейронної мережі. Цей параметр використовується для визначення, як саме дані розділяються на навчальні, валідаційні та тестові набори під час навчання мережі. Якщо ділення даних не відбувається, то рядок Data Division у вікні Neural Network Training (nntraintool) відсутній.

Training (навчання). Функція навчання, яка визначає, як нейронна мережа оновлює свої ваги та зсуви під час навчання. Популярні функції включають в себе Backpropagation, Levenberg-Marquardt, Bayesian Regularization та інші.

Встановлення Training: Levenberg-Marquardt (trainlm) вказує на використання алгоритму тренування, відомого як алгоритм Левенберга-Марквардта (Levenberg-Marquardt, позначається як trainlm), для навчання нейронної мережі в середовищі MATLAB. Алгоритм Левенберга-Марквардта - це широко використовуваний метод оптимізації для навчання нейронних мереж. Цей метод комбінує в собі ідеї з методу найменших квадратів і методу Ньютона для оптимізації функції втрати мережі. Основна ідея алгоритму Левенберга-Марквардта полягає в швидкому спаді значення функції втрати і пошуку локального мінімуму шляхом оновлення вагових коефіцієнтів мережі. Він добре справляється зі складними задачами навчання, де інші методи можуть справлятися гірше.

Performance (виконання). Термін Performance вказує на показник якості навчання нейронної мережі на основі обраної функції втрат (loss function) та даних для перевірки (validation dataset). Цей показник відображає, наскільки добре нейронна мережа працює під час навчання на навчальних та валідаційних даних і допомагає визначити, коли навчання мережі має бути завершено.

Параметр Performance: Mean Squared Error (mse) вказує на використання середньоквадратичної помилки (MSE) для оцінки точності нейронної мережі під час навчання. Середньоквадратична помилка (Mean Squared Error, MSE) є однією з найпоширеніших метрик у задачах регресії. Вона обчислюється як середнє значення квадратів різниці між прогнозованими значеннями моделі (у цьому випадку нейронної мережі) та відомими або фактичними значеннями у наборі даних. Формула MSE виглядає наступним чином:

$$MSE = \frac{\sum_{i=1}^N (y_{actual_i} - y_{predicted_i})^2}{N},$$

де N - кількість прикладів даних в наборі; y_{actual_i} - фактичні значення; $y_{predicted_i}$ - прогнозовані значення, отримані від нейронної мережі.

MSE вимірює середню квадратичну відстань між фактичними і прогнозованими значеннями. Чим менше значення MSE, тим краще модель адаптується до даних. Під час навчання нейронної мережі, оптимізаційний алгоритм намагається мінімізу-

вати MSE, тобто підганяти прогнозовані значення, щоб вони були ближчі до фактичних.

Calculations (розрахунки). Параметр відноситься до кількості обчислень або розрахунків, які виконуються під час навчання нейронної мережі. Цей параметр може бути важливим для моніторингу продуктивності та визначення часу, який потрібен для завершення процесу навчання.

Параметр **Calculations: MEX** вказує на використання MEX-функцій під час навчання нейронної мережі в середовищі MATLAB. MEX (MATLAB Executable) - це механізм, який дозволяє виконувати оптимізований обчислювальний код, написаний на C, C++ або Fortran, безпосередньо в середовищі MATLAB.

В контексті навчання нейронних мереж Calculations: MEX вказує, чи використовувати MEX-функції для підвищення швидкості обчислень під час тренування. MEX-функції можуть бути більш швидкими за інтерпретовані мови, такі як MATLAB, і зазвичай вони використовуються для виконання обчислень на рівні мови програмування C або C++.

Секція Progress (прогрес). У секції вказується на інформація про прогрес навчання нейронної мережі під час процесу навчання. Містить інформацію, яка дозволяє в режимі реального часу відстежувати прогрес та якість навчання моделі. Це допомагає контролювати процес навчання та вживати необхідні дії для поліпшення якості моделі.

Epoch (епоха). Вказує на один повний прохід через весь навчальний набір даних під час навчання нейронної мережі. Під час кожної епохи навчання всі приклади з навчального набору даних проходять через нейронну мережу один раз, і ваги мережі оновлюються відповідно до результатів навчання на цій епохі. Епохи допомагають відстежувати процес навчання та роблять його більш керованим.

Time (час). Вказує на час, який вже було витрачено на процес навчання нейронної мережі. Цей параметр служить для відстеження тривалості процесу навчання і може бути корисним для оцінки часових аспектів навчання моделі.

Performance (виконання). Вказує на показник якості навчання нейронної мережі на основі обраної функції втрат (loss function) та даних для перевірки (validation dataset). Цей показник відображає, наскільки добре нейронна мережа працює під час навчання на навчальних та валідаційних даних. Для задач регресії зазвичай використовується середньоквадратична помилка (Mean Squared Error, MSE), яка визначає середню квадратичну помилку між прогнозованими значеннями і справжніми значеннями цільової змінної.

Gradient (градієнт). Вказує на градієнт функції втрат (loss function) відносно параметрів нейронної мережі під час навчання. Градієнт - це вектор, який вказує на те, як потрібно змінювати ваги та зсуви (biases) нейронної мережі, щоб мінімізувати функцію втрат і покращити її ефективність.

Під час навчання нейронної мережі використовується метод зворотного поширення помилки (backpropagation), і градієнт відіграє важливу роль у цьому процесі. Градієнт обчислюється для кожного параметра (ваги та зсуви) і показує, в якому напрямку і наскільки слід змінити ці параметри, щоб зменшити значення функції втрат.

Mu. Відноситься до тау-параметра, який використовується в алгоритмах оптимізації для регулювання швидкості навчання нейронної мережі. Тау-параметр (Mu) використовується в контексті адаптивних методів оптимізації, таких як метод зворотного розповсюдження і метод Бройдена-Флетчера-Гольдфарба-Шанно (BFGS), для регулювання швидкості навчання. Він дозволяє алгоритму оптимізації адаптуватися до поточної динаміки процесу навчання та забезпечити збіжність навчання навіть при складних функціях втрат або підвищених градієнтах.

Зазвичай значення Mu обчислюється алгоритмом оптимізації під час навчання і може змінюватися на кожній ітерації. Цей параметр є однією з ключових частин процесу оптимізації нейронної мережі та допомагає покращити збіжність та стабільність навчання.

Validation Checks (перевірки підтвердження). Вказується кількість перевірок (checks) якості моделі, які виконуються під час навчання нейронної мережі з використанням набору даних для перевірки (validation dataset). Validation Checks допомагає контролювати процес навчання та визначити, коли навчання слід припинити або зупинити, щоб уникнути перенавчання.

Основна ідея полягає в тому, що під час навчання нейронної мережі може виникнути перенавчання (overfitting), коли модель вивчає «на пам'ять» навчальний набір даних і втрачає здатність узагальнювати на нові дані. Для запобігання перенавчанню, використовуються перевірки якості моделі на окремому наборі даних, який не використовується під час навчання, а називається набором даних для перевірки (validation dataset).

Validation Checks вказують, як часто виконувати перевірки якості моделі під час навчання. Наприклад, якщо встановлено 6 перевірок, то під час навчання нейронної мережі програма буде регулярно перевіряти, наскільки добре модель працює на наборі даних для перевірки після кожних 6 епох навчання (де епоха - це один прохід через весь навчальний набір даних). Якщо якість моделі на наборі даних для перевірки перестає покращуватися або погіршується, то це може бути індикатором перенавчання, і навчання може бути припинене.

Секція Plots (графіки). Містить графіки та візуалізацію прогресу навчання нейронної мережі під час процесу навчання. Ця секція допомагає візуально аналізувати якість та хід навчання мережі.

Кнопка **Performance** використовується для відображення показників якості під час навчання нейронної мережі. При натисканні на цю кнопку відкривається вікно **Neural Network Training Performance**, показане на рис. 5.21.

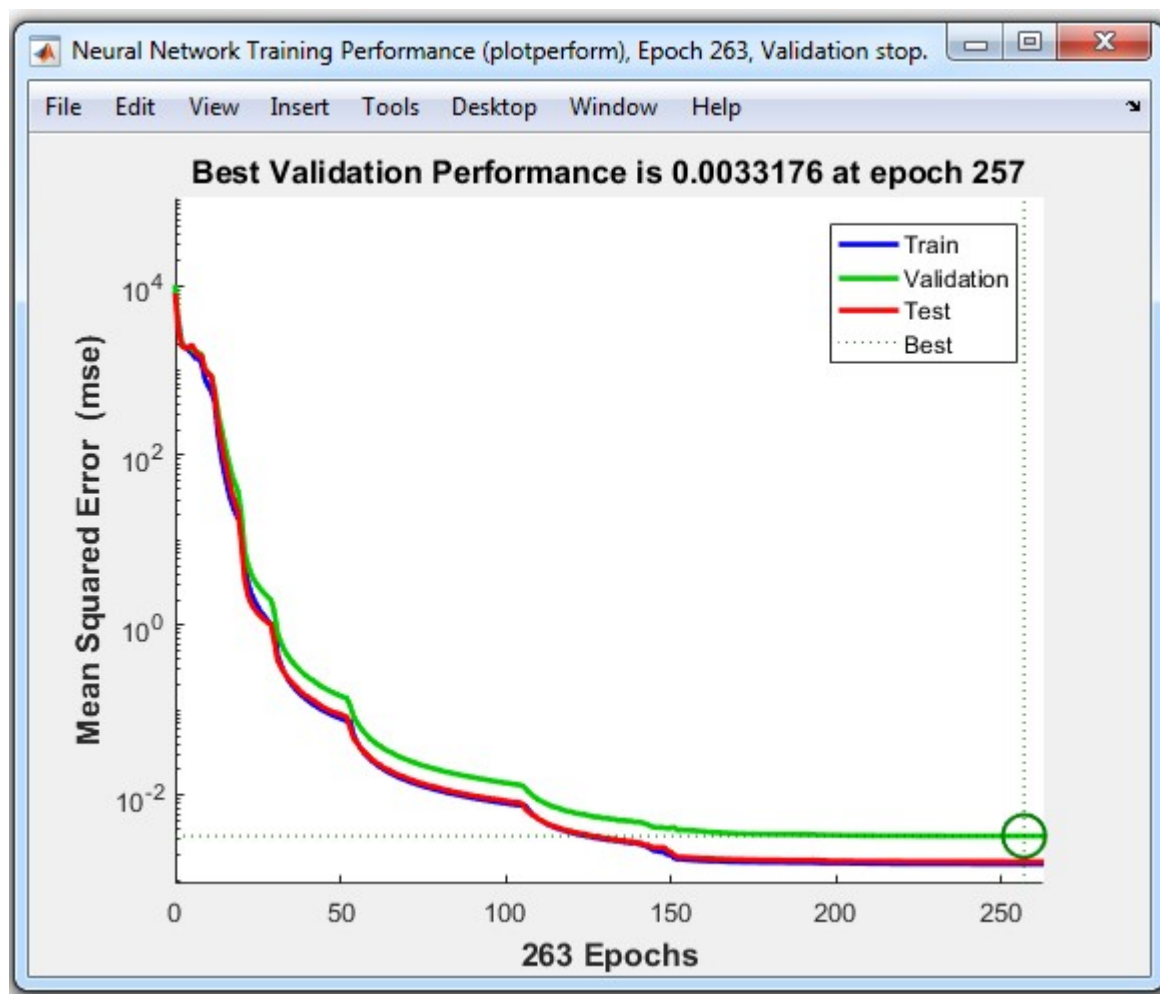


Рисунок 5.21 – Вікно контролю процесу навчання

У вікні Neural Network Training Performance відображаються графіки **Mean Squared Error (mse)**, які показують, як змінюється середньоквадратична помилка на навчальних, валідаційних (перевірочних) та тестових наборах даних під час кожної ітерації навчання. Графіки MSE показують, як змінюється точність мережі під час навчання і чи є ознаки перенавчання (overfitting), коли помилка на валідаційних та тестових даних починає зростати. Метою є мінімізувати MSE на усіх наборах даних для досягнення найкращої моделі.

Кнопка **Training State (стан навчання)**. При натисканні на кнопку Training State відкривається вікно **Neural Network Training State** (Рис. 5.22) в якому відображається інформацію про стан навчання нейронної мережі на певний момент часу під час процесу навчання.

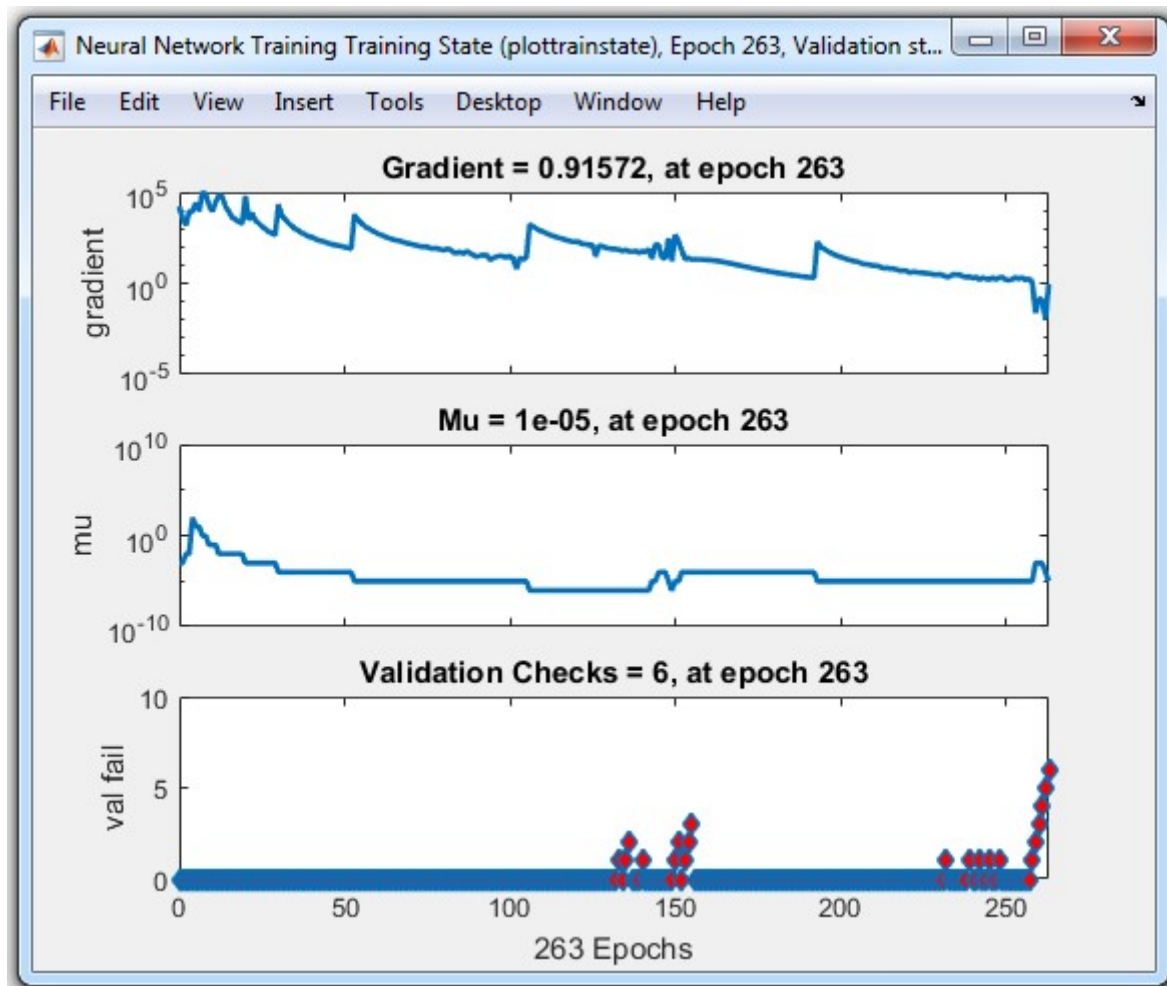


Рисунок 5.22 - Вікно Neural Network Training State

Опис вікна Neural Network Training State

У вікні Neural Network Training State відображаються наступні графіки.

Графік Gradient показує зміну значення градієнта під час навчання нейронної мережі. Градієнт в контексті навчання нейронних мереж є вектором, який вказує напрямок та міру найшвидшого зростання функції втрати (або помилки) мережі в точці навчання. Градієнт використовується для оновлення вагових коефіцієнтів мережі під час оптимізації.

Основна ідея використання градієнта полягає в тому, що шляхом зміни вагових коефіцієнтів мережі в напрямку, протилежному до градієнта, можна найбільше зменшити значення функції втрати. Градієнт допомагає визначити, в якому напрямку і наскільки потрібно вносити зміни в вагові коефіцієнти для покращення продуктивності мережі.

На графіку Gradient відображається, як змінюється величина градієнта під час процесу навчання. Зазвичай, мережа спочатку навчається з великими градієнтами, і з часом ці градієнти зменшуються, коли мережа наближається до оптимальних вагових коефіцієнтів.

Моніторинг графіка Gradient може бути корисним для визначення, чи існують проблеми з градієнтним зникненням або вибуванням (gradient vanishing/exploding), а також для визначення, чи виникають проблеми зі збіжністю навчання. Великі або надто малі градієнти можуть вказувати на необхідність оптимізації параметрів навчання.

Графік μ показує зміну значення параметра μ під час навчання нейронної мережі. Параметр μ відноситься до швидкості навчання та регуляризації мережі, і його зміна вказує на те, як адаптується швидкість навчання під час тренування.

Зазвичай, під час навчання нейронної мережі використовуються методи оптимізації, такі як стохастичний градієнтний спуск (SGD) або його модифікації. Параметр μ в цьому контексті часто вказує на швидкість навчання (learning rate) та регуляризацію.

Зміна μ показує, як модифікується швидкість навчання під час процесу навчання. Зазвичай, μ може збільшуватися або зменшуватися в залежності від реакції нейронної мережі на дані. Наприклад, велике значення " μ " може вказувати на те, що швидкість навчання збільшилася для прискорення збіжності, але при цьому може виникнути ризик нестабільності. Зменшення μ може вказувати на те, що навчання сталося занадто швидко або навіть зупинилося.

Моніторинг графіка μ може бути корисним для контролю за швидкістю навчання та стабільністю процесу навчання. Великий вміст інформації в графіку μ може вказувати на те, що вибрані параметри навчання потребують подальшої налаштування.

Графік Validation Failure (або val fail) (помилка підтвердження) показує, як змінюється величина, що відображає кількість невдач (помилки) під час перевірки (валідації) нейронної мережі під час навчання. Головною метою валідації є визначення, наскільки добре навчена мережа узгоджується з набором даних, які не використовувалися під час навчання, і чи існують ознаки перенавчання. Зазвичай, мінімізація цього показника є однією з цілей тренування мережі. Якщо кількість помилок на валідації росте, це може бути ознакою перенавчання, і може бути необхідно прийняти заходи для покращення продуктивності мережі, такі як зупинка навчання раніше або зміна гіперпараметрів.

Продовження опису вікна Neural Network Training (nntraintool)

Кнопка **Regression (регресія)** використовується для вибору та налаштування параметрів навчання нейронної мережі для завдань регресії. Як відомо, завдання регресії - це один з типів завдань в машинному навчанні, де основною метою є передбачення числових значень або кількісних величин на основі вхідних даних. У регресійних завданнях модель намагається знайти функціональну залежність між вхідними ознаками (параметрами) і вихідними числовими значеннями, які зазвичай називаються цільовими або відгуком.

При натисканні на кнопку Regression відкривається вікно **Neural Network Training Regression** (рис. 5.23), у якому показані графіки та інформація, які допомагають в оцінці та аналізі продуктивності нейронної мережі під час навчання. Для наочної оцінки якості навчання приведені графіки регресії для Training, Validation та Test наборів даних.

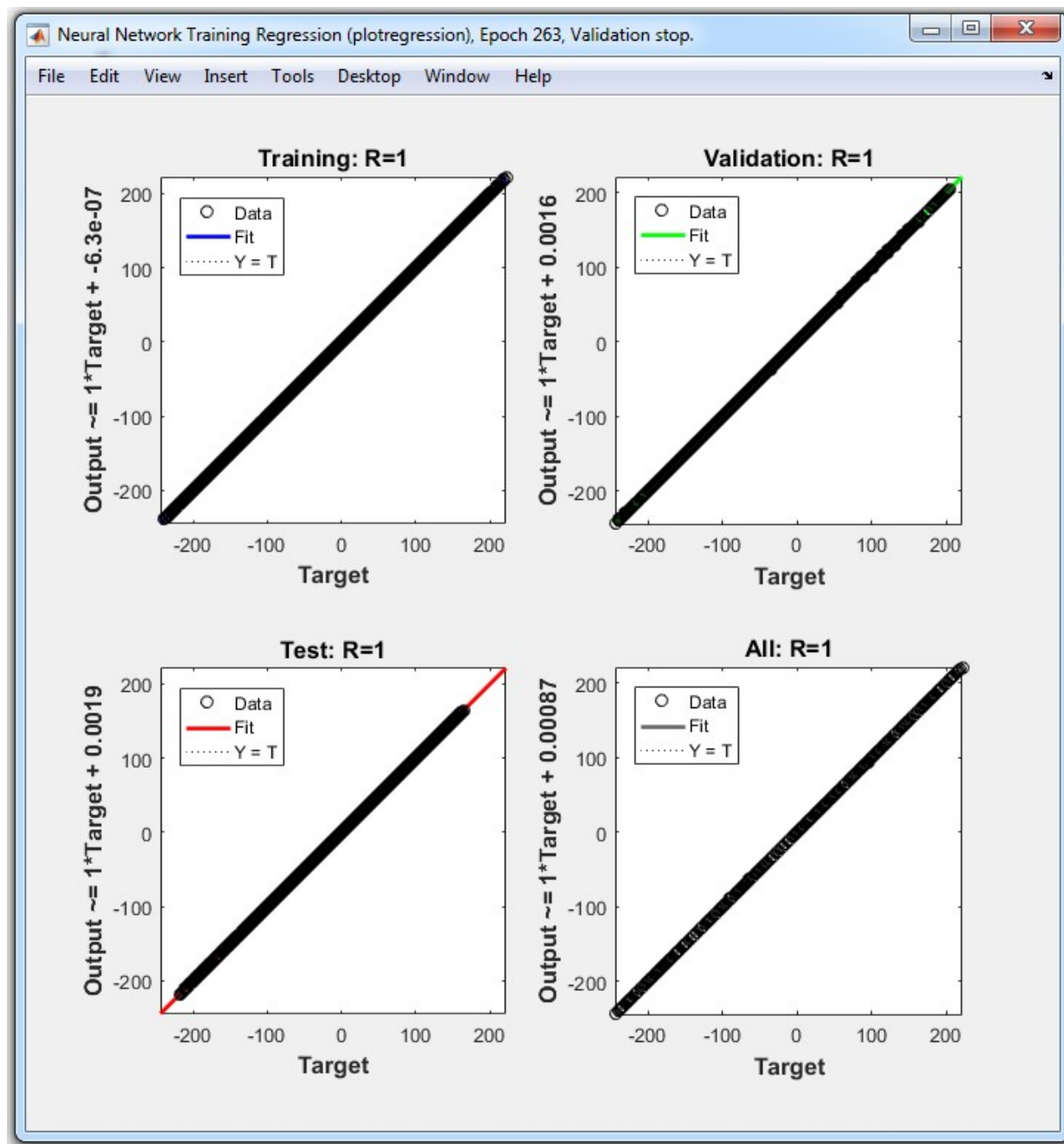


Рисунок 5.23 - Вікно Neural Network Training Regression

Опис вікна Neural Network Training Regression

Графіки **Training** показують, наскільки точно модель відповідає навчальним даним, тобто даним, на яких вона була навчена.

Графіки **Validation** демонструють, наскільки точно модель працює на наборі даних для перевірки. Дані для перевірки використовуються для валідації результатів моделі та визначення, наскільки вона загальніє на нових даних.

Графіки **Test** включають тестові дані, які незалежно від використовувалися для оцінки точності моделі на даних, які модель раніше не бачила. Тестові дані надають об'єктивну оцінку роботи моделі на нових даних.

Графіки **All** показують регресійні результати для всіх трьох наборів даних: Training, Validation та Test. Графіки All об'єднують і відображають прогнози моделі та фактичні значення для всіх трьох наборів даних на одному графіку з метою порівняння. Ці графіки забезпечують користувача загальним зоровим порівнянням точності моделі на різних наборах даних і допомагають визначити, чи існують різниці в її поведінці на тренувальних, валідаційних та тестових даних.

Графіки **Data (дані)** для Training, Validation, Test та All даних у вікні Neural Network Training Regression представляють собою графічну інформацію про те, як навчальна модель нейронної мережі відповідає даним на різних наборах даних. Ці графіки показують на горизонтальній вісі фактичні значення (дані з набору даних), а на вертикальній вісі - відповідні прогнозовані значення, отримані від моделі нейронної мережі, тобто кожній парі з тренувальних, валідаційних та тестових даних відповідає одна точка (у вікні Neural Network Training Regression позначено кружечками) з координатою по осі абсцис, яка дорівнює бажаному значенню, а по осі ординат – значенню на виході штучної нейронної мережі. Якщо всі точки лежать близько до ідеальної (діагональної) лінії (це графіки $Y = T$, позначені точками у вікні Neural Network Training Regression), то це означає, що модель добре передбачає дані. Якщо є велика розсіюваність точок, це може свідчити про недосконалість моделі або шум у даних.

Графіки **dat** допомагають візуально оцінити точність моделі на трьох різних наборах даних і визначити, наскільки добре модель може узгоджуватися з фактичними даними. Вони можуть бути корисними для виявлення аномалій, оцінки загальної продуктивності моделі і визначення необхідності подальшої оптимізації.

Графіки **Fit (прогноз)** – це графіки лінійної регресії, які відображають прогнозовані значення, які надає нейронна мережа на основі вхідних даних. Графіки Fit демонструють відповідність прогнозованих значень, отриманих від моделі, фактичним значенням з наборів даних. Вони допомагають візуально оцінити, наскільки точно модель регресії передбачає числові значення на основі вхідних даних. Якщо прогнозована лінія добре відповідає фактичним даним і проходить через багато точок, це свідчить про високу точність моделі. Напротив, якщо лінія регресії відхиляється від фактичних значень, це може вказувати на неадекватність моделі і необхідність налаштування параметрів нейронної мережі.

Вирази для лінійної регресії, а також значення коефіцієнта кореляції R наведені у вікні Neural Network Training Regression для кожного набору даних:

Training, Validation, Test та All. Якщо графік Fit добре узгоджується з графіком $Y=T$, то це свідчить про високу точність моделі регресії.

Коефіцієнт кореляції R у вікні Neural Network Training Regression (plotregression) є мірою кореляції між фактичними та прогнозованими значеннями у завданні регресії. Цей коефіцієнт допомагає визначити, наскільки добре модель регресії відповідає фактичним даним і наскільки точним є її прогноз.

Зазвичай, коефіцієнт кореляції R може приймати значення в діапазоні від -1 до 1. Його інтерпретація наступна:

- Коефіцієнт кореляції R дорівнює 1: Це означає, що прогнозовані значення і фактичні значення абсолютно лінійно залежні одне від одного, і модель регресії ідеально передбачає фактичні значення.
- Коефіцієнт кореляції R дорівнює 0: Це вказує на жодну лінійну залежність між прогнозованими і фактичними значеннями, і модель не може передбачити дані з жодною точністю.
- Коефіцієнт кореляції R дорівнює -1: Це означає, що прогнозовані значення і фактичні значення є лінійно залежними, але зворотним способом, і модель регресії передбачає їх ідеально, але з зміненням напрямком.

Кореляційний коефіцієнт R близький до 1 вказує на високу точність моделі регресії, тоді як значення близьке до 0 або від'ємне вказує на низьку точність. Цей коефіцієнт дуже корисний для візуальної оцінки точності моделі та порівняння прогнозованих та фактичних значень у завданнях регресії.

Продовження опису вікна Neural Network Training (nntraintool)

Plot Interval (інтервал графіка). Цей параметр контролює частоту оновлення графіків та відображення результатів під час навчання нейронної мережі для задачі регресії. Plot Interval визначає, через скільки ітерацій навчання (або епох) має бути відображена інформація на графіках.

Зазвичай вибір Plot Interval допомагає контролювати кількість інформації, яка відображається під час навчання, особливо якщо навчання триває багато епох і графіки можуть бути дуже завантажені. Встановлення меншого значення Plot Interval означатиме, що графіки оновлюватимуться частіше, що може бути корисним для більш детального аналізу, але збільшить обсяг відображеної інформації. Навпаки, встановлення більшого значення Plot Interval зменшить частоту оновлень графіків і зменшить обсяг відображеної інформації.

Зазвичай, налаштування Plot Interval є важливою опцією під час навчання моделі для того, щоб зручно аналізувати процес навчання та вплив варіацій параметрів моделі на результати.

Кнопки **Stop Training** і **Cancel** у вікні Neural Network Training (nntraintool) в середовищі MATLAB використовуються для керування процесом навчання нейронної мережі, але вони мають різні функції:

Кнопка **Stop Training** (зупинка навчання) призначена для негайної зупинки процесу навчання нейронної мережі. Вона припиняє навчання без можливості продовження або збереження поточного стану моделі. Використовується, коли ви бачите, що навчання триває дуже довго, ви визнали, що навчання не дає задовільних результатів, або ви просто хочете негайно припинити процес навчання з будь-якої причини.

Кнопка **Cancel** (скасувати) використовується для припинення процесу навчання, але зазвичай вона надає можливість підтвердження та збереження поточного стану моделі. Вона дозволяє вам контролювати ітерації навчання, перевіряти результати, змінювати параметри та вирішувати, коли припинити навчання з можливістю збереження результатів навчання.

Після завершення навчання результати відображаються на графіках у вікні **Training data for NN Predictive Control**, яке показано на рис.5.24.

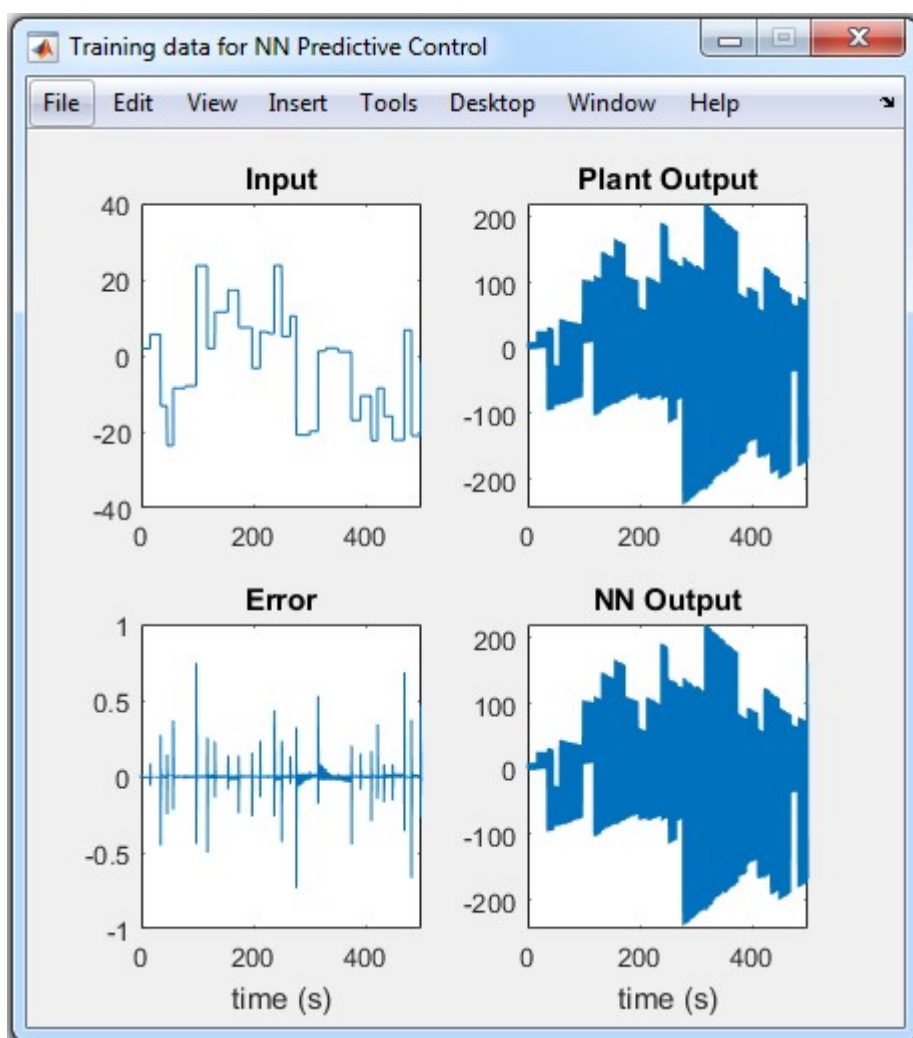


Рисунок 5.24 – Результати тренування мережі

Опис вікна Training data for NN Predictive Control

Input (вхід). Графік Input відображає вхідні сигнали, які подаються на систему

і нейромережеву модель під час навчання для генерації відповіді.

Plant Output (вихід системи). Графік Plant output відображає фактичні вихідні дані системи (Plant) під час навчання. Ці дані представляють реальну динаміку системи під впливом вхідних сигналів.

NN Output (вихід нейромережі). Графік NN Output відображає вихідні дані, що згенеровані нейромережевою моделлю під час навчання. Ці дані представляють прогнозовану відповідь моделі, яка намагається відтворити динаміку системи.

Error (Помилка). Графік Error показує різницю між прогнозованими відповідями нейромережевої моделі (NN Output) і фактичними відповідями системи (Plant Output). Ця помилка вказує на те, наскільки добре модель відтворює динаміку системи під час навчання.

На рис. 5.25, 5.26 показані вікна **Validation data for NN Predictive Control** і **Testing data for NN Predictive Control**, в яких наведені аналогічні графіки, побудовані в результаті перевірки на контрольній і текстовій множині.

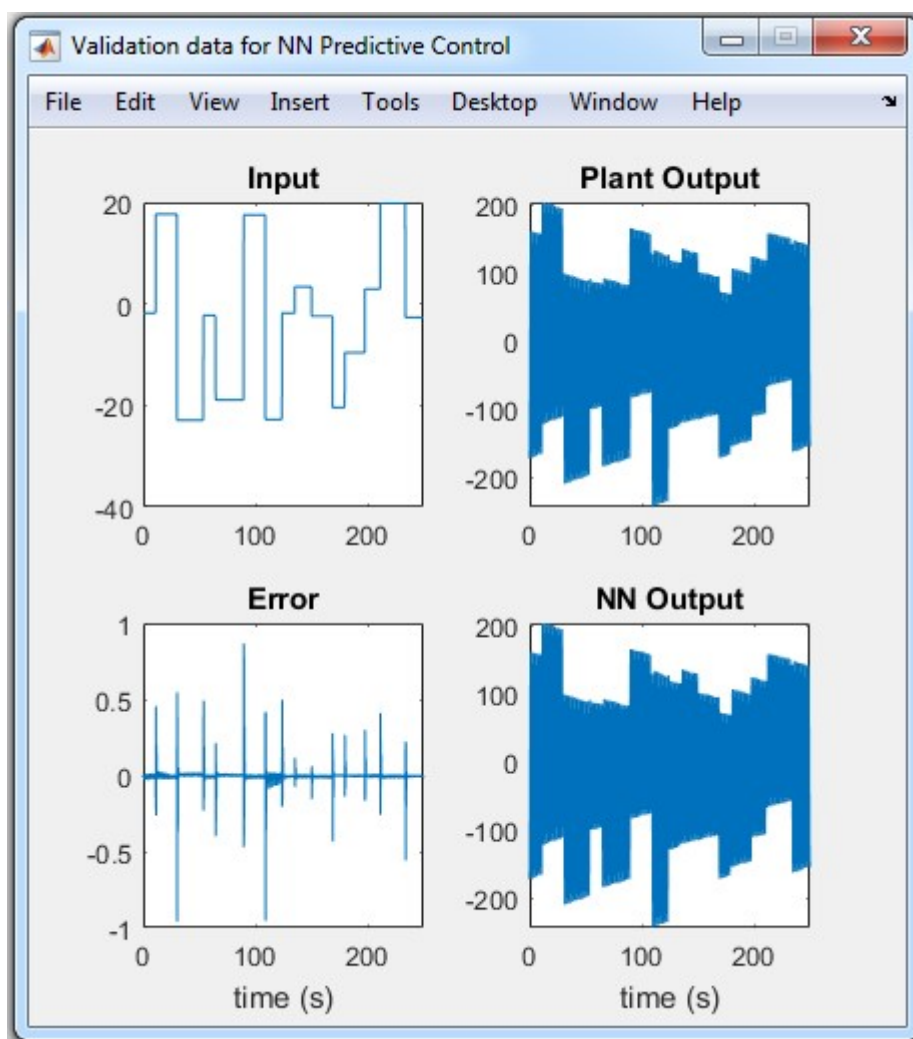


Рисунок 5.25 – Результати перевірки мережі на контрольній множині

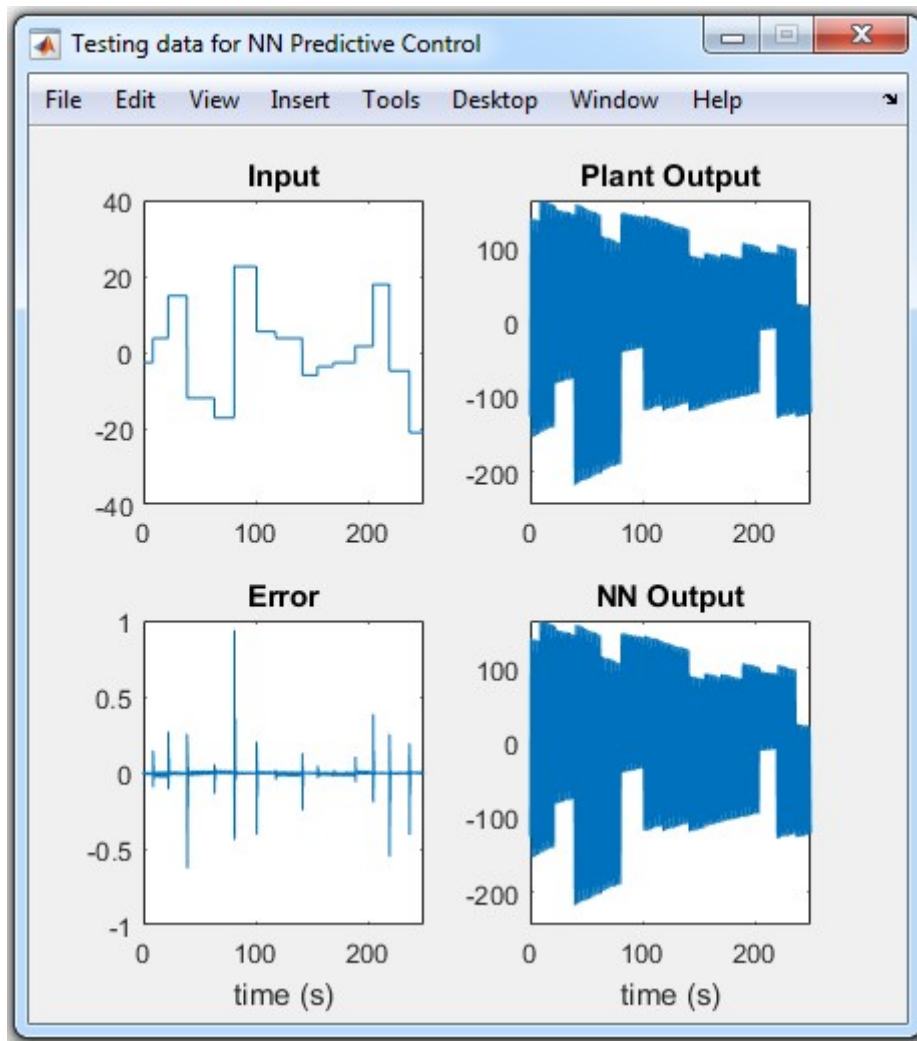


Рисунок 5.26 – Результати перевірки мережі на тестовій множині

Продовження опису вікна *Plant Identification*.

Поточний стан відмічений у вікні *Plant Identification* повідомленням «*Training complete. You can generate or import new data, continue training or save results by selecting OK or Apply*» (Навчання завершено. Можна згенерувати або імпортувати нові дані, продовжити навчання або зберегти отримані результати, вибравши кнопки *OK* або *Apply*).

Після завершення навчання М-функція *Nnident* виконує побудову динамічної мережі *netn2* із заданим числом затримок по входу і виходу моделі, не змінюючи при цьому набутих значень вагів і зсувів нейронів шарів. Згорнута схема динамічної мережі і моделі елементів мережі показані на рис.5.27.

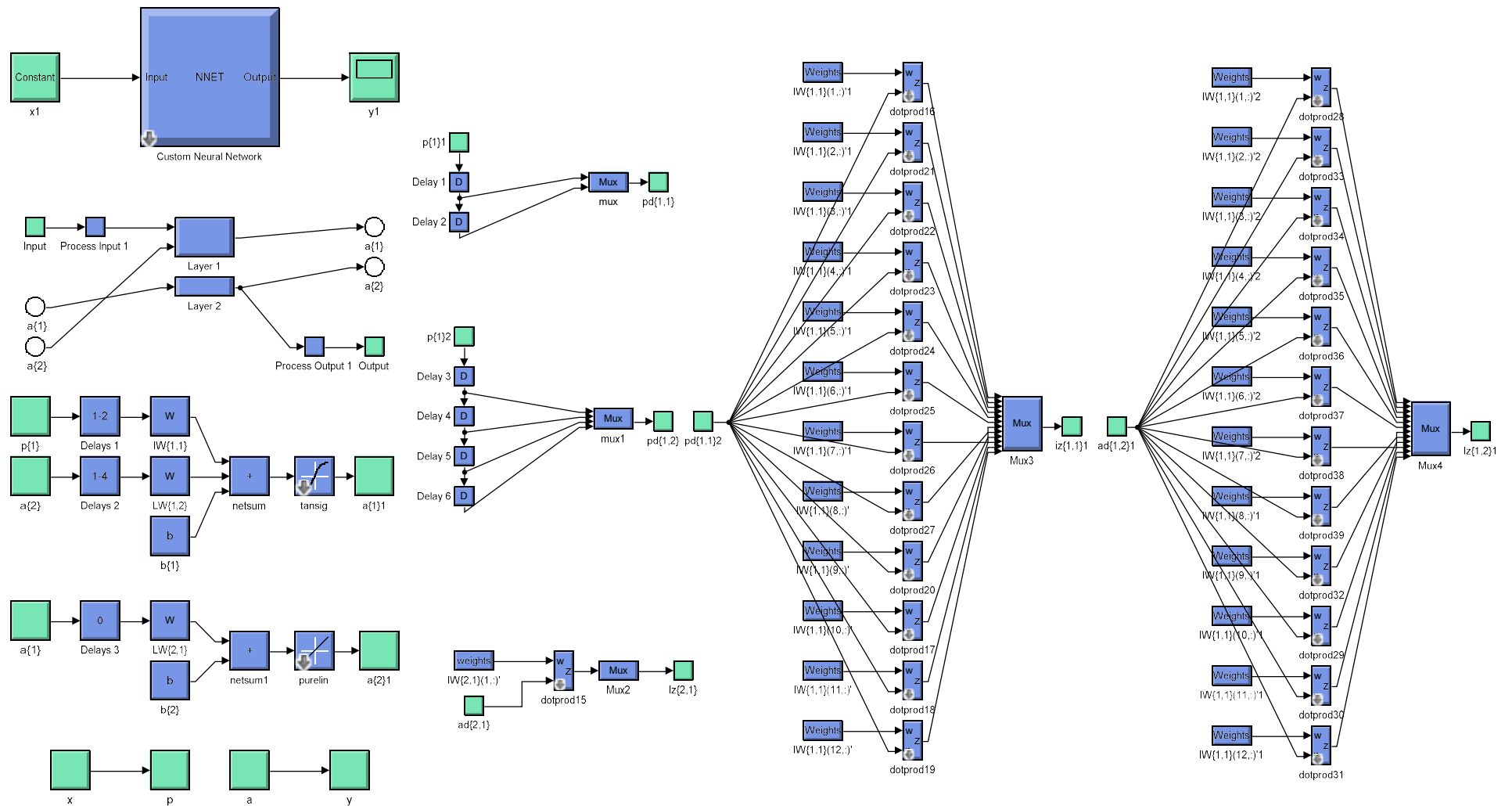


Рисунок 5.27 – Моделі елементів динамічної мережі з елементами затримок, реалізовані в Simulink

Модель динамічної нейронної побудована в системі Simulink за допомогою оператора gensim(netn2). Кожен подальший елемент з'являється в окремому вікні при активізації попереднього подвійним клацанням миші. З даних елементів в системі Simulink побудована схема динамічної мережі, показана на рис.5.28.

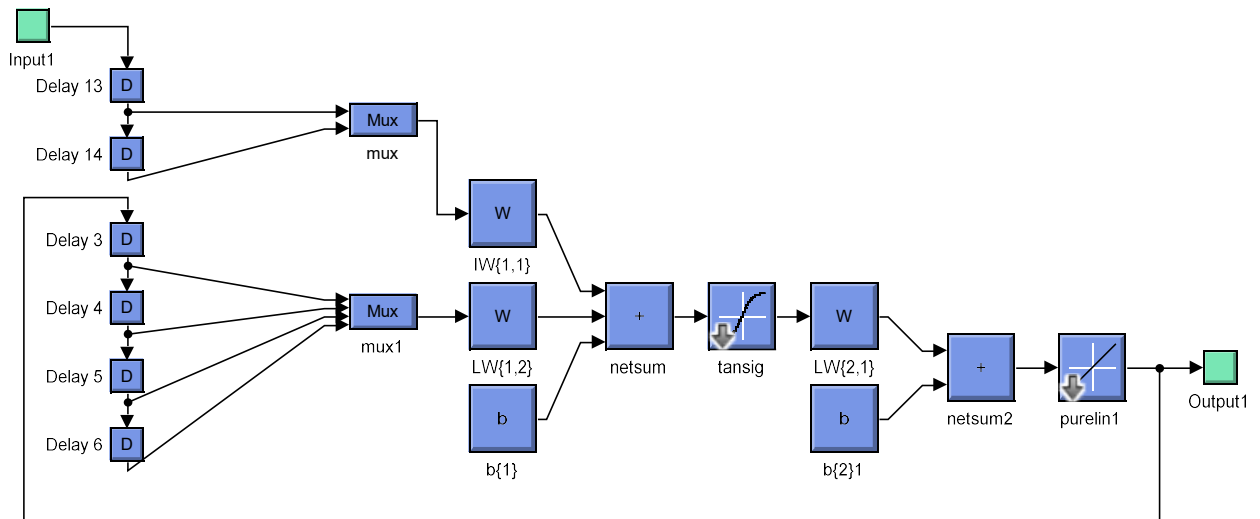


Рисунок 5.28 – Модель динамічної мережі з елементами затримок, побудованої в Simulink

Слід зазначити, що для побудови моделей мереж netn і netn2 треба перервати виконання програми шляхом попереднього встановлення **Breakpoint** у відповідному місці функції Nnident.m, оскільки після закінчення виконання зазначеної функції побудова мереж стає неможливою.

В результаті параметри нейромережевої моделі керованого об'єкту вводяться в блок NN Predictive Controller системи Simulink. У системі Simulink дана мережа представляється у вигляді структурної схеми, показаної на рис.5.29.

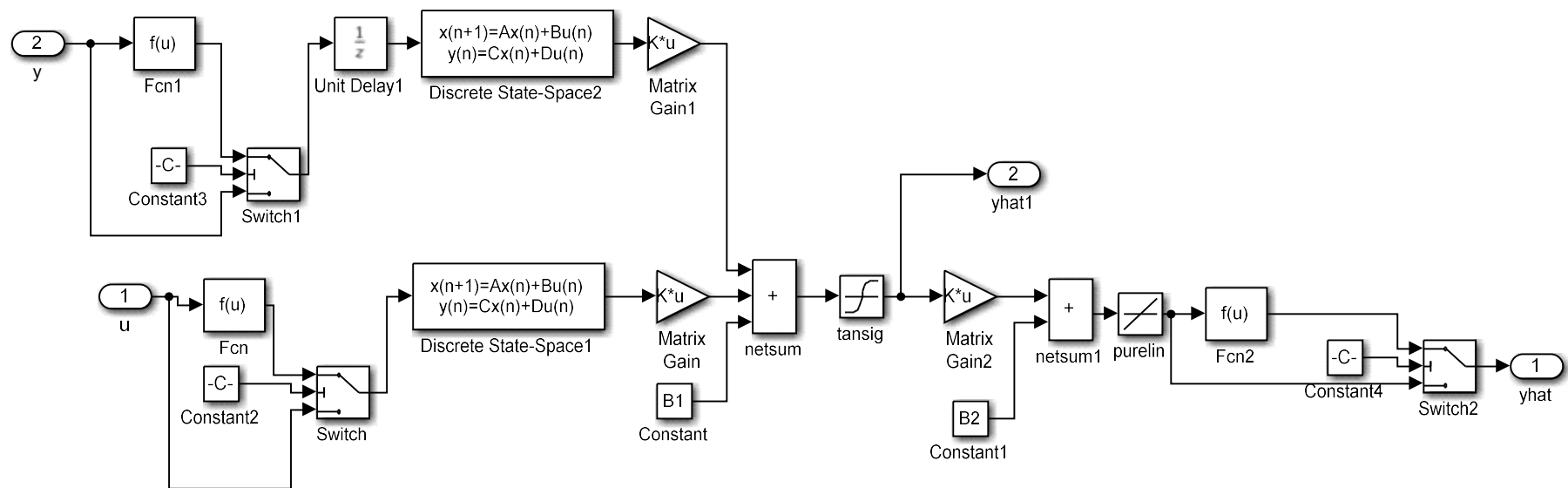


Рисунок 5.29 – Структурна схема нейронмережевої моделі об'єкту регулювання

Блоки Matrix Gain і Matrix Gain 1 відповідають матрицям $IW\{1,1\}$ і $LW\{1,2\}$ відповідно. Блоки Constant (B1) Constant1 (B2) відносяться до зсувів нейронів першого і другого шарів. Елементи затримок моделюються за допомогою блоків Discrete State Space 1 і Discrete State Space 2

$$\mathbf{y}(n) = \mathbf{C}\mathbf{x}(n) + \mathbf{D}\mathbf{u}(n) ;$$

$$\mathbf{x}(n+1) = \mathbf{A}\mathbf{x}(n) + \mathbf{B}\mathbf{u}(n).$$

Для випадку $N_i = 2$ і $N_j = 5$ чисельні значення матриць $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ і $\mathbf{D}$ вказаних блоків наступні

Discrete State Space 1

$$\mathbf{A} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Discrete State Space 2

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

У системі Simulink формується також схема ptest3sim2, показана на рис.5.30. Після активізації блоку Subsystem відкривається вікно з схемою рис.5.31. Дана схема теж є нейромережевою моделлю об'єкту управління, що має додаткові виходи, і використовується М-функцією predort для прогнозу процесу в майбутньому.

Перетворена структурна схема нейрорегулятора, отримана в результаті об'єднання схем рис.5.12 і рис.5.29 показана на рис. 5.32.

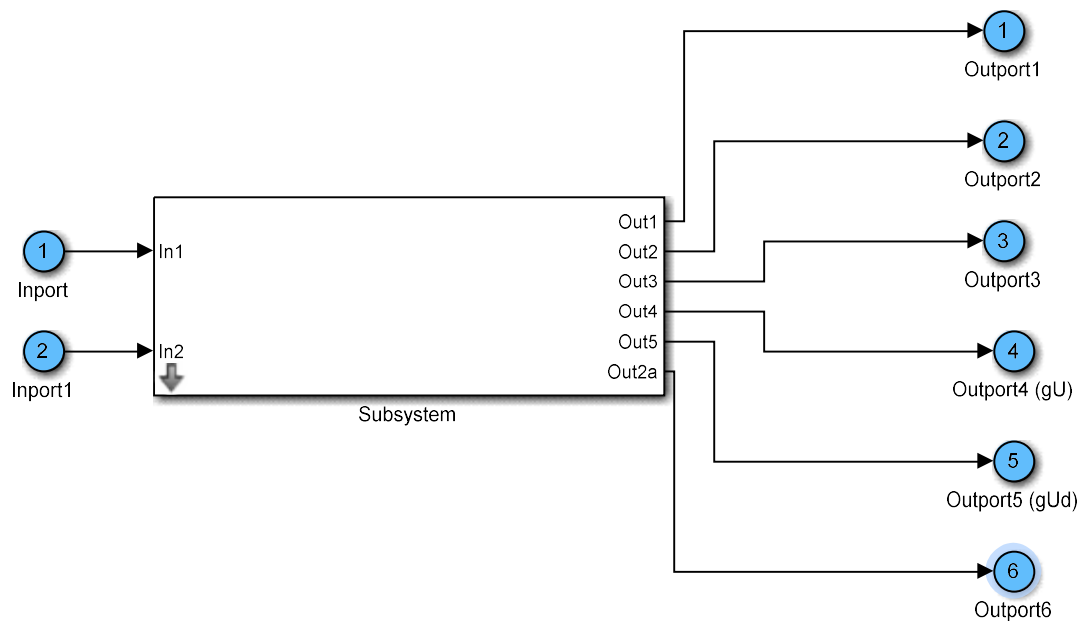


Рисунок 5.30 – Схема ptest3sim2, використовувана М-функцією predopt для прогнозу процесу в майбутньому

Після завершення побудови нейромережевої моделі керованого об'єкту відбувається повернення до вікна Neural Network Predictive Controller (рис.5.7) і встановлюються параметри оптимізації.

Продовження опису вікна Neural Network Predictive Controller

Const Horizon (N2). Верхня межа підсумовування в показнику якості (нижня межа N_1 фіксована і рівна 1). Цей параметр визначає горизонт прогнозу, тобто кількість кроків у майбутньому, на які алгоритм прогнозованого керування буде робити прогнози. Вибір значення Const Horizon (N2) важливий, оскільки він впливає на те, наскільки точно і передбачувано система буде керована за допомогою алгоритму NNPC.

Зазвичай, збільшення значення Const Horizon (N2) дозволяє отримати більший горизонт прогнозу, що означає більш далекі випереджальні прогнози, але водночас може збільшити обчислювальну складність і затримки у системі керування. Обране значення "Const Horizon" повинно бути ретельно підібрано з урахуванням конкретних вимог і характеристик вашого керувального завдання і системи.

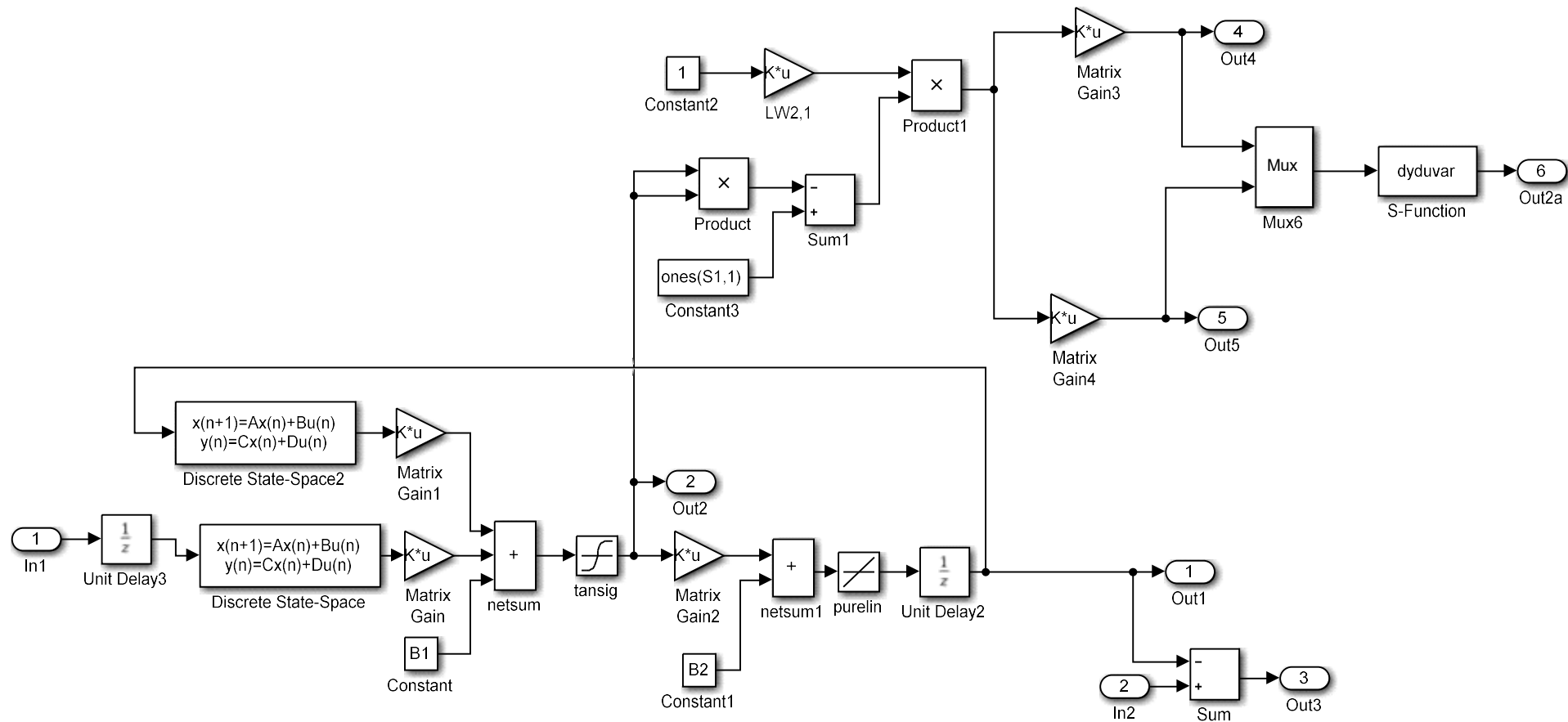


Рисунок 5.31 – Схема блоку Subsystem системи ptest3sim2

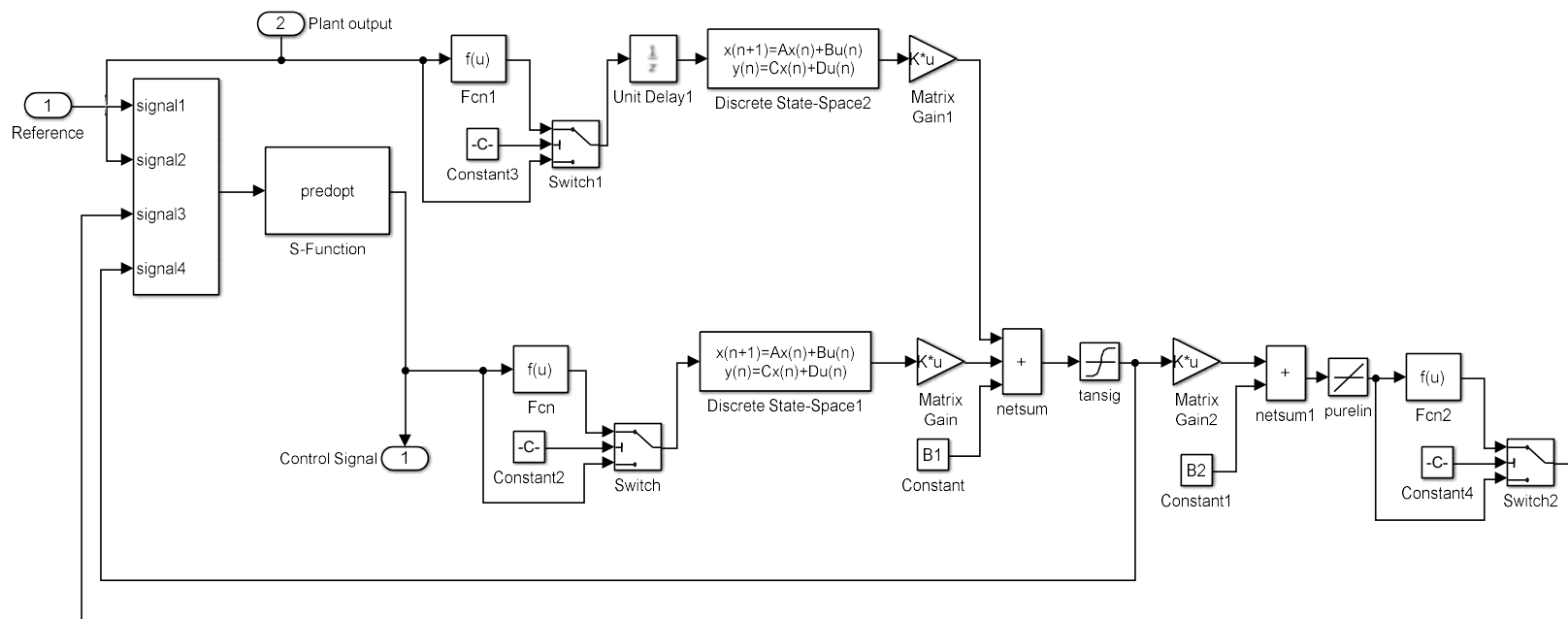


Рисунок 5.32 – Перетворена структурна схема нейрорегулятора

Control Horizon (Nu). Верхня межа підсумовування при оцінці потужності управління N_u . Цей параметр грає важливу роль при плануванні оптимального керування системою, і він визначає кількість кроків у майбутньому, на які буде розрахована потужність управління в рамках контролю.

Основна ідея цього параметру полягає в тому, що велике значення Control Horizon (Nu) дозволяє оптимізаційному алгоритму керування урахувати більшу кількість кроків у майбутньому при плануванні керування. Це може бути корисним для досягнення бажаного стану системи та забезпечення стабільності, а також враховувати обмеження на керування.

Однак велика кількість кроків в Control Horizon (Nu) також може призвести до складніших обчислювальних задач і збільшити вимоги до ресурсів, тому вибір значення параметра Control Horizon (Nu) повинен бути обґрунтованим і базуватися на конкретних вимогах та характеристиках системи керування.

Control Weighting Factor (ρ). Коефіцієнт ваги для потужності управління. Цей параметр дозволяє вам налаштувати важливість мінімізації керуючих сигналів в обчисленнях контролера.

Контролер має оптимізувати вихідні сигнали системи, забезпечуючи бажаний режим роботи. Однак інколи деякі сигнали керування можуть бути дорогою або необхідною величиною. Наприклад, в деяких ситуаціях, зниження витрат на керування може бути важливим завданням.

Control Weighting Factor визначає, як обчислювати вартість або функцію витрат управління під час оптимізації. Якщо цей фактор великий, то контролер надає велику вагу мінімізації витрат на керування, і, отже, старається робити керуючі сигнали якнайбільш економічними. Зменшення фактора зробить керуючі сигнали менш важливими у функції витрат, і контролер буде акцентуватися на досягненні бажаного режиму роботи системи.

Користувачі можуть регулювати Control Weighting Factor в залежності від своїх конкретних вимог та цілей управління. Цей параметр дозволяє балансувати між досягненням бажаного результату і вартістю управління в контексті нейронного мережевого прогнозного управління.

Search parameter (α). Параметр одновимірного пошуку, задаючий поріг зменшення показника якості. Цей параметр використовується для контролю процесу оптимізації і для визначення, коли оптимізація може бути завершена, оскільки зменшення показника якості досягло заданого порогу.

Одновимірний пошук (або одновимірна оптимізація) - це метод оптимізації, де здійснюється пошук оптимального значення одного параметра, тримаючи інші параметри незмінними. В контексті контролера з використанням нейронної мережі це може бути важливим для налаштування певних параметрів контрольної стратегії, які впливають на якість та ефективність керування.

Поріг зменшення показника якості вказує на те, наскільки мінімальне зменшення величини, яку слід оптимізувати, вважати прийнятним для завершення процесу оптимізації. Якщо зменшення показника якості стає меншим за встановлений поріг, то оптимізація завершується, і визначені параметри вважаються оптимальними.

Цей параметр дозволяє контролеру зупинити оптимізацію, коли досягнуто задовільного рівня покращення показника якості, що може бути корисним для зкономлення обчислювальних ресурсів та забезпечення швидкої збіжності оптимізації.

Minimization Routine. Вибір процедури одновимірного пошуку.

Цей параметр визначає, який алгоритм оптимізації буде використовуватися для налаштування параметрів контрольної стратегії. Вибір підходящого алгоритму оптимізації може суттєво впливати на якість та ефективність контрольної системи. Вибір конкретного "Minimization Routine" залежить від характеристик завдання керування, обмежень, доступності обчислювальних ресурсів та інших факторів.

Iterations Per Sample Time. Параметр визначає кількість ітерацій оптимізації, які виконуються протягом кожного відрізка часу (Sample Time) при розрахунку оптимального керування системою. Цей параметр дозволяє встановити, скільки ітерацій оптимізації виконується в межах кожного відрізка часу, і впливає на швидкість та точність оптимізації контролера. Зазвичай, більша кількість ітерацій дозволяє отримати більш точну оптимальну стратегію керування, але водночас може збільшити обчислювальну складність та вимоги до ресурсів.

Важливо збалансувати кількість ітерацій Iterations Per Sample Time так, щоб контролер зберігав ефективність та швидкість реакції на зміни у системі керування. Занадто багато ітерацій може призвести до затримок в реакції контролера, тоді як занадто мало ітерацій може призвести до недостатньої точності оптимізації.

Загальною практикою є експериментальне налаштування цього параметра в залежності від конкретних вимог та обмежень системи керування для досягнення оптимального балансу між точністю та швидкістю оптимізації.

Після установки параметрів оптимізації вони вводяться в блок NN Predictive controller системи Simulink.

5.4.4 Вибір параметрів нейрорегулятора NN Predictive Controller

При синтезі регулятора варіюються величини N_2 , N_u і ρ (величина N_1 фіксована і рівна 1), а також параметр одновимірного пошуку α , задаючий поріг зменшення показника якості і число ітерацій на 1 такт дискретності γ . Задається також процедура одновимірного пошуку.

Як показали дослідження, величини N_u , ρ і α не значно впливають на результати синтезу і прийняті: $N_u=2$, $\rho=0,05$, $\alpha=0,001$. Як процедура одновимірного пошуку прийнята `csrchbas`.

Значення N_2 і γ роблять істотний вплив на роботу регулятора. При їх збільшенні точність зростає, проте об'єм обчислень на кожному такті дискретності істотно збільшується. Для вирішуваної задачі оптимальні значення знаходяться в межах $N_2 = 15 \div 25$, $\gamma = 2 \div 3$.

При ідентифікації об'єкту управління найбільш важливим питанням є вибір кількості нейронів прихованого шару N . При малій кількості нейронів мережа не може виконувати поставлене завдання, а при великому спостерігається явище перенавчання і зростає об'єм обчислень.

Для даного завдання оптимальні значення $N = 7 \div 12$ при цьому помилка навчання, а також помилка на контрольній і тестовій множині ε не перевищують $10^{-3} \div 10^{-5}$.

Успіх тренування мережі в значній мірі залежить від довжини навчальної вибірки N_b і такту дискретності Δt , що визначає інтервал між двома послідовними моментами знімання даних. Оптимальними у вирішуваній задачі є: $N_b=10000$, $\Delta t=0,1$ с. При збільшенні Δt знижується точність обчислення і різниця між помилкою навчання і помилкою, отриманою на контрольній і тестовій множині. Зменшення Δt викликає необхідність відповідного збільшення N_b і, як наслідок, значно збільшується час тренування мережі, при цьому істотного зниження помилки не спостерігається.

Для отримання представницької вибірки необхідно правильно задати максимальне і мінімальне значення інтервалу ідентифікації. Величина їх залежить від параметрів об'єкту Subsystem; у даній роботі прийнято $t_{\min} = 4 \div 5$ с, $t_{\max} = 10 \div 20$ с.

При синтезі нейромережевої моделі системи задається кількість елементів запізнювання на вході z_1 і виході z_2 моделі. Якнайкращі результати отримані при $z_1 = 2$, $z_2 = 2 \div 5$.

Результат тренування мережі залежить від початкового значення вагів нейронної мережі w_{ij} і кількості циклів навчання $N_{\text{ц}}$. Для досягнення глобального мінімуму процес навчання необхідно повторювати багато разів при різних початкових

значеннях w_{ij} і величині $N_{\text{ц}}$. У даній зробіті для кожного варіанту мережі вибиралося декілька десятків початкових точок розрахунку, кількість циклів навчання, після закінчення яких помилка навчання переставала зменшуватися, складало $300 \div 600$.

Як навчальна функція вибрана `trainlm`.

5.4.5 Моделювання нейромережевої системи управління

Для побудови графіків перехідних процесів двомасової системи з нейрорегулятором NN Predictive Controller слід використати моделі Simulink, приведені на рис. 5.11 або 5.33.

Як приклад, на рис.5.34-5.37 приведені результати моделювання нейромережевої системи.

У курсовому проєкті слід проаналізувати отримані графіки змінних стану нейромережевої системи і зробити висновки.

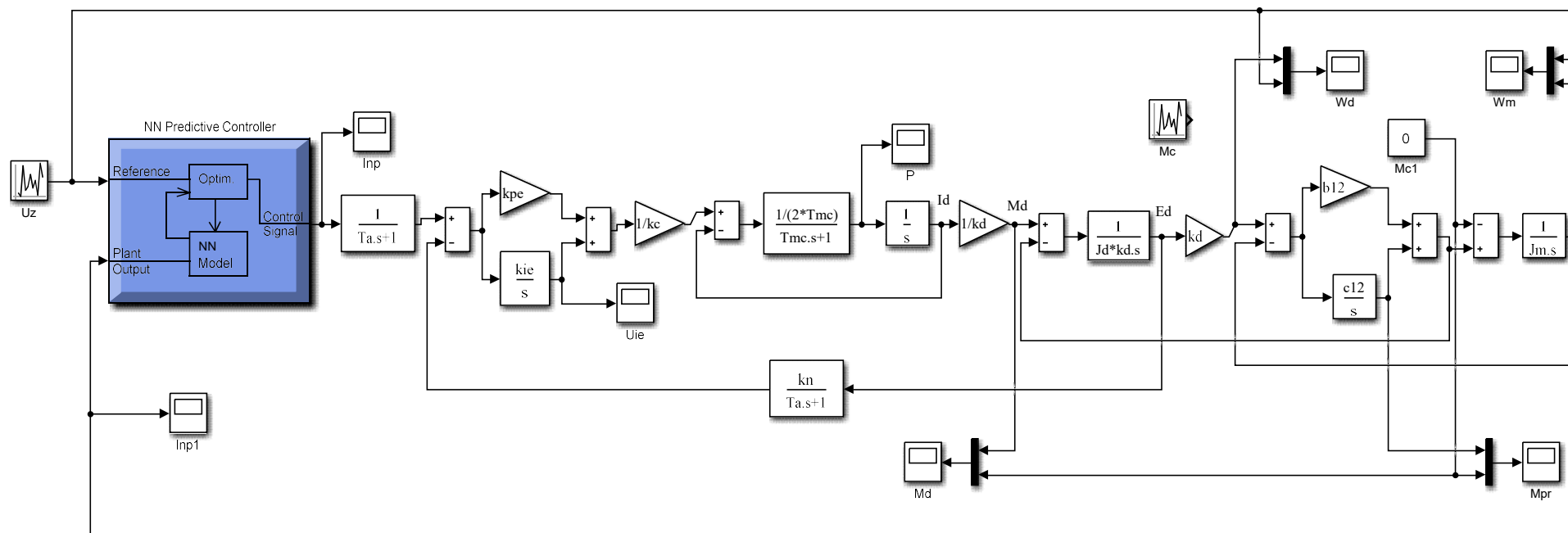


Рисунок 5.33 – Схема моделі двомасової системи з нейрорегулятором

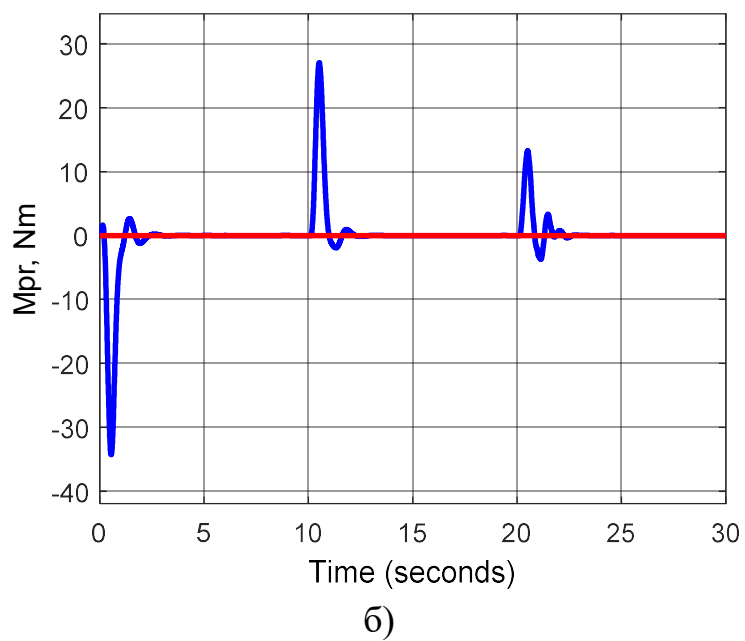
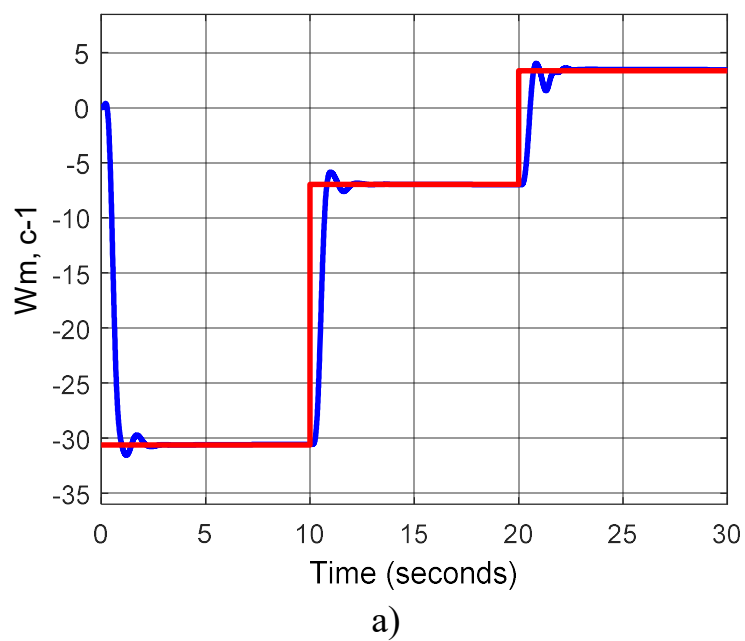


Рисунок 5.34 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи з нейрорегулятором по задаючій дії

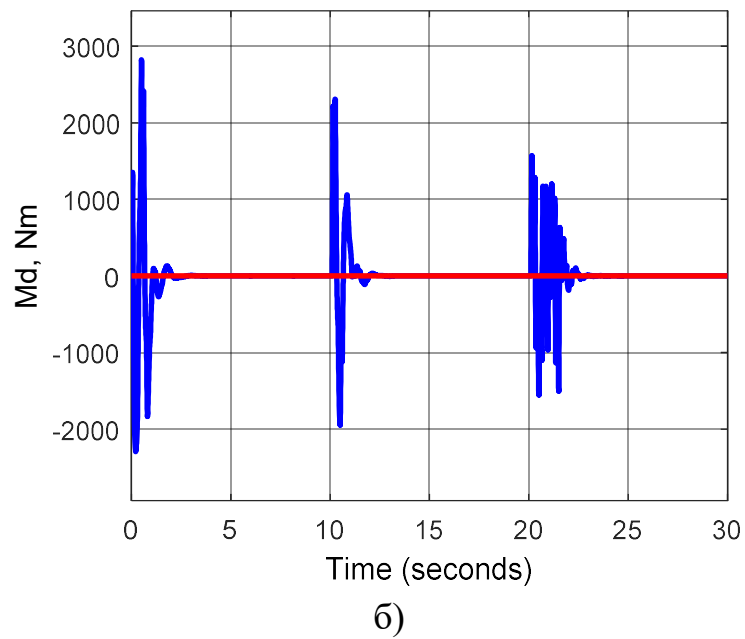
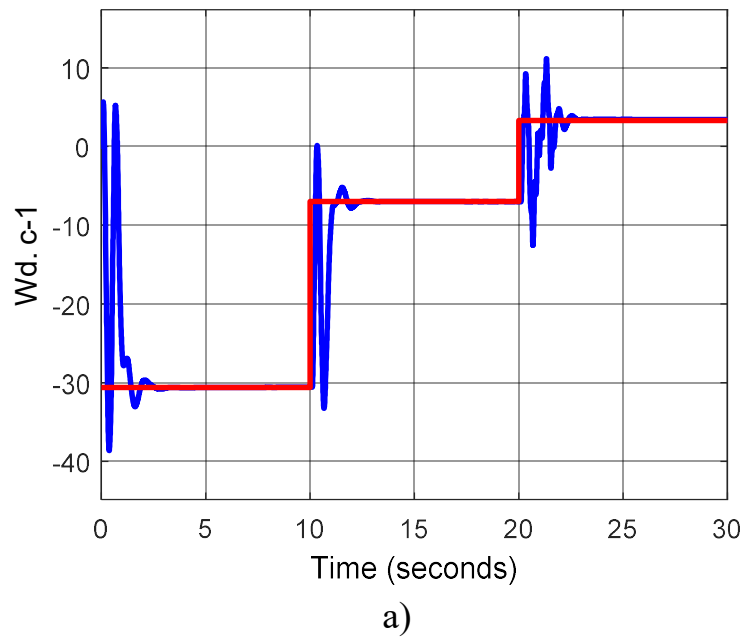


Рисунок 5.35 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи з нейрорегулятором по задаючій дії

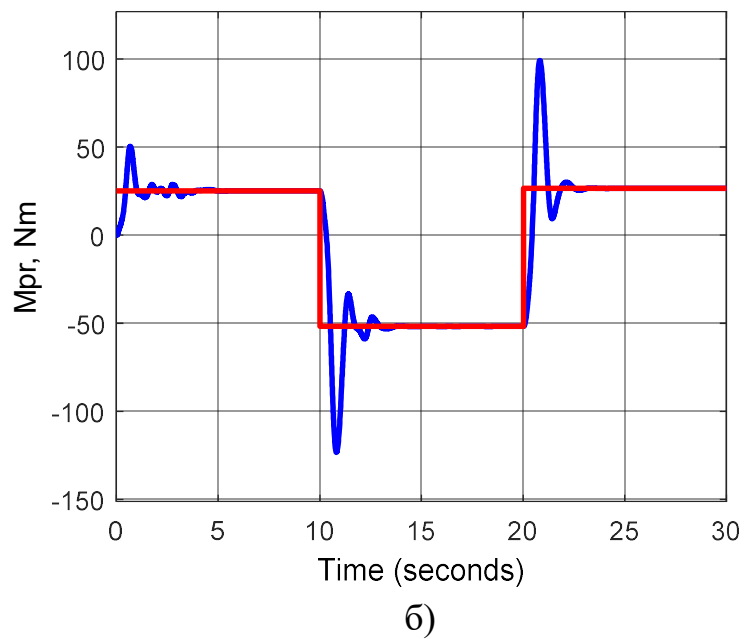
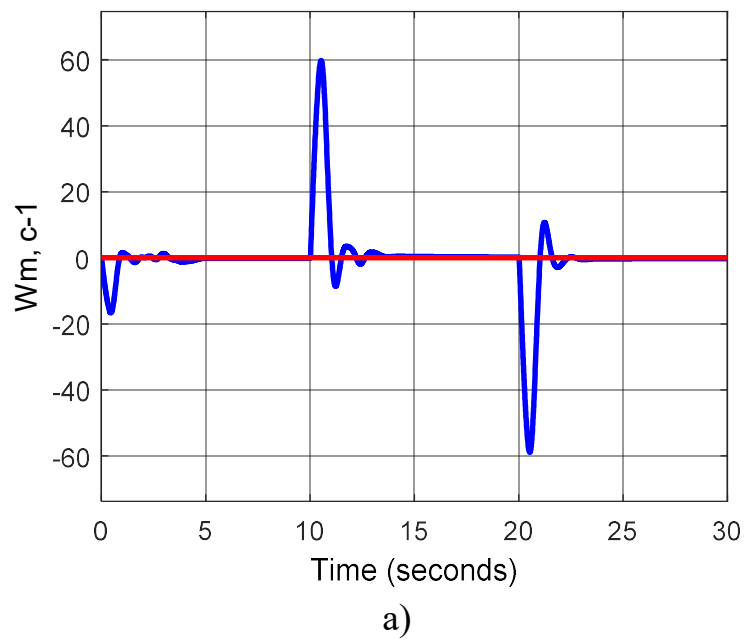


Рисунок 5.36 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи з нейрорегулятором по обурюючій дії

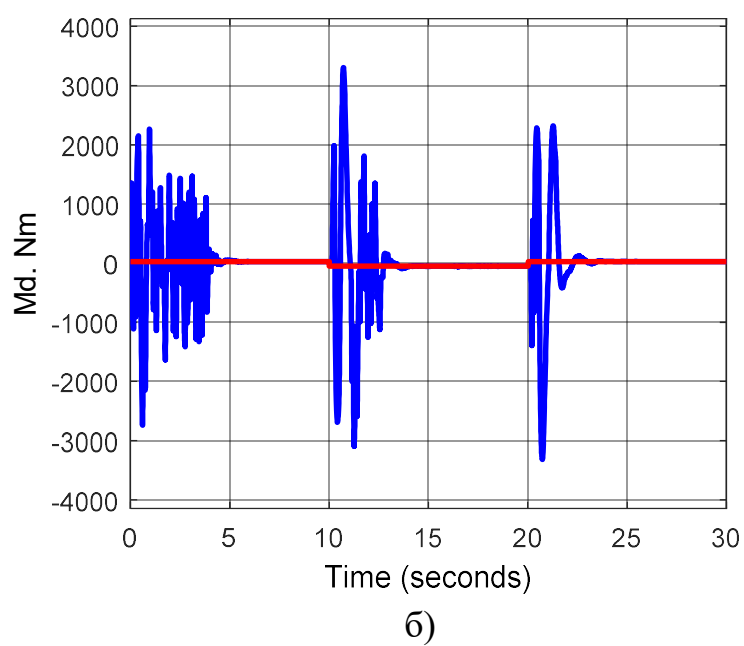
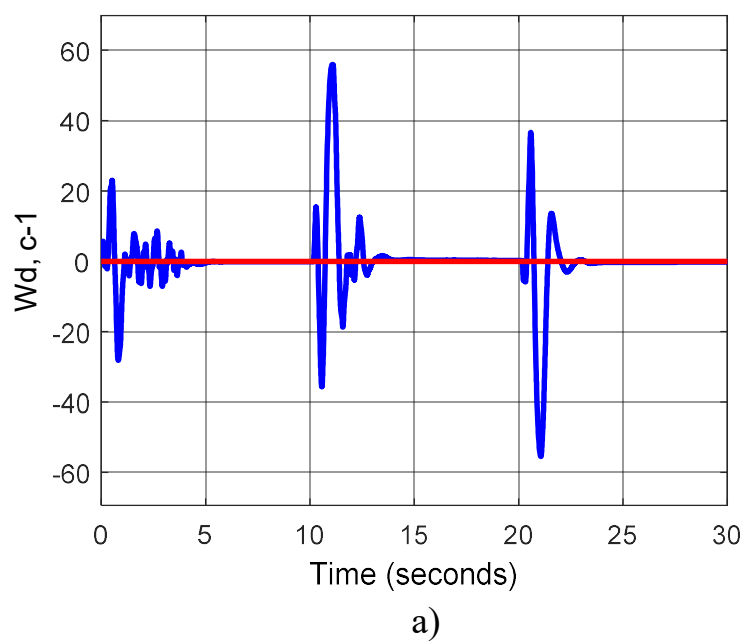


Рисунок 5.37 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи з нейрорегулятором по обурюючій дії

6. ОФОРМЛЕННЯ КУРСОВОГО ПРОЄКТУ

Пояснювальну записку слід виконувати на листах паперу формату А-4 із розмірами 297х210 мм, на всіх сторонах листа треба залишати поля. При виконанні пояснювальної записки рекомендується набирати текст через півтора інтервали на одному боці аркуша з полями, застосовуючи шрифт Times New Roman 14 розміру. Сторінка повинна містити 38-40 рядків тексту.

Формули, таблиці, рисунки треба нумерувати арабськими цифрами в межах розділу з указівкою номера розділу і порядкового номера формули, рисунка, таблиці в даному розділі. Наприклад, формула (7.3) – третя формула сьомого розділу. Номер формули необхідно вказувати в круглих дужках. До всіх рисунків варто приводити підрисункові підписи. (*Приклад підпису:* Рисунок 7.5 – Функціональна схема системи). Назву таблиці треба розміщати над таблицею, посередині, починаючи з номера таблиці, (*Наприклад:* Таблиця 7.1 – Розрахунок навантажувальної діаграми). Схеми виконувати відповідно до вимог стандартів. У тексті посилання на літературу вказувати в квадратних дужках, наприклад: [2].

СПИСОК ЛІТЕРАТУРИ

1. Технології нейронних мереж і нечіткого моделювання в системах управління : підруч. для здобувачів вищої освіти спец. 151 Автоматизація та комп'ютерно-інтегровані технології / Г.І. Канюк, Б.І. Кузнецов, Т.Ю. Василець, А.Ю. Мезеря, О.О. Варфоломієв. – Харків : Друкарня Мадрид, 2020. – 306 с.
2. Нейромережеві технології в системах управління: Підручник для вузів./ Б. І. Кузнецов, Т.Ю. Василець, Т.Б. Нікітіна, В. В. Коломиєць, О.О. Варфоломієв; Укр. інж.-пед. акад.. - Харків: УПА, 2014. - 232 с.
3. С.В.Ткаліченко. Штучні нейронні мережі: Навчальний посібник. – Кривий Ріг: Державний університет економіки і технологій, 2023. –150 с.
4. Кирик В. В. Математичний апарат штучного інтелекту в електроенергетичних системах:підручник.– Київ: КПП ім. Ігоря Сікорського, Вид-во «Політехніка», 2019. –224 с.
5. Зайченко Ю.П. Основи проектування інтелектуальних систем.. – К.: Слово, 2004. –353 с.
6. Штучні нейронні мережі: навчальний посібник для студентів вищих навчальних закладів / О.Г. Руденко, Є.В. Бодянський. – К: Компанія СМІТ, 2006. - 404 с.
7. Субботін С. О. Нейронні мережі : теорія та практика: навч. посіб. / С. О. Субботін. – Житомир : Вид. О. О. Євенок, 2020. – 184 с
8. Тимошук П.В. Штучні нейронні мережі; Навч. посібн. - Львів: Львівська політехніка, 2011. -444 с.
9. Кононюк, А. Ю. .Нейронні мережі і генетичні алгоритми. - Київ: Корнійчук, 2008. - 468 с.
- 10.Глибовець М.М., Отецький О.В. Штучний інтелект. – К.: Вид. дім «КМ Академія», 2002. – 366 с.
- 11.Ямпольський Л. С. Системи штучного інтелекту в плануванні, моделюванні та управлінні : підруч. для студ. вищ. навч. закл. / Л. С. Ямпольський, Б. П. Ткач, О. І. Лісови-ченко. - К. : ДП «Вид. дім «Персонал», 2011. — 544 с.
1. Попович М. Г., Ковальчук О. В. Теорія автоматичного керування: Підручник. – 2-ге вид., перероб. і доп. – К. : Либідь, 2007. – 656 с.
2. Голюк П.Ф., Гречин Т.М. Теорія автоматичного керування: Навчальний посібник. – Л: Видавництво Львівської політехніки, 2012. – 280 с.
3. Головінський Б.Л., Шуруб Ю.В., Лисенко В.П. Теорія автоматичного управління: навч. посіб. – К.: Вид. Центр НУБіП України, 2012. – 240 с.
4. Ладанюк А.П. Теорія автоматичного керування технологічними об'єктами : навч. посіб. / Ладанюк А.П., Архангельська К.С., Власенко Л.О. – К.: НУХТ, 2014 – 274 с.
5. Сорока К.О.Теорія автоматичного керування і комп'ютерне моделювання (неперервні лінійні системи). Частина перша. Основи теорії систем автоматичного керування. : навч. посібник. – Харків : ФОП Тімченко, 2010. – 218 с.

6. Сорока К.О. Теорія автоматичного керування і комп'ютерне моделювання (неперервні лінійні системи). Частина друга. Аналіз систем автоматичного керування засобами комп'ютерного моделювання : навч. посібник. – Харків : ФОП Тімченко, 2010. – 156 с.
7. І.Г. Абраменко, Д. І. Абраменко Теорія автоматичного керування. – Харків: ХНАМГ, 2008. – 178 с.
8. Автоматизований електропривод ч. 2 [Електронний ресурс]: навчальний посібник для студентів освітньої програми «Електромеханічні системи автоматизації, електропривод та електромобільність» спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» / В.І. Теряєв. – Київ: КПІ ім. Ігоря Сікорського, 2022.– 204 с
9. Попович М.Г., Лозинський О.Ю., Клепиков В.Б. та інші. Електромеханічні системи автоматичного керування та електроприводи. Навч. посіб. за напрямом «Електромеханіка» / М.Г.Попович, О.Ю. Лозинський, В.Б. Клепиков та інші. – К.: Либідь, 2005. – 680 с. Ч1.
10. Попович М.Г., Лозинський О.Ю., Клепиков В.Б. та інші. Електромеханічні системи автоматичного керування та електроприводи. Навч. посіб. за напрямом «Електромеханіка» / М.Г. Попович, О.Ю. Лозинський, В.Б. Клепиков та інші. – К.: Либідь, 2005. – 680 с. Ч2.
11. Системи керування електроприводами. Видання 2: Навч. посібник з дисципліни «Системи керування електроприводами» (для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» денної і заочної форми навчання) / – Краматорськ: ДДМА, 2018. – 225 с.
12. Гурко О.Г. Аналіз та синтез систем автоматичного управління у MATLAB: Навчальний посібник / О.Г. Гурко, І.Ф. Єрмоменко. Харків, ХНАДУ, 2012. – 284 с.
13. Моделювання систем управління в SIMULINK : навч. посібник / В. О. Богомолів, О. Г. Гурко, В. І. Клименко, Д. М. Леонтьєв, О. М. Красюк. Харків ХНАДУ, 2018. – 220 с.

ДОДАТОК А

Файл вихідних даних

```
clear;clc;echo on;

% Вихідні дані

kd=0.139;    CF=1/kd;
In=92;       Mn=In/kd;
ktp=34.2;    kn=0.054;
kc=0.13;     Tmc=0.005;
Ta=0.035;    Te=0.028;
Re=0.183;
b12=0;       c12=5;
Jd=8.4;      Jm=0.17;
Js=Jd+Jm;    Uz=24;

w0=3;
c12=(w0^2)*(Jd*Jm)/(Jd+Jm);

Tme=2*Tmc+Ta;
kic=Re/(2*Tmc*kc*ktp);
kpc=kic*Te;
kpe=Js*kd^2*kc/(2*Tme*kn);
kie=kpe/(4*Tme);

% Матриці
A=[-b12/Jm 1/Jm b12/Jm zeros(1,5);
-c12 0 c12 zeros(1,5);
b12/Jd -1/Jd -b12/Jd 1/Jd zeros(1,4);
zeros(1,4) 1/kd 0 0 0;
0 0 0 -kd/(2*Tmc^2) -1/Tmc kpe/(2*Tmc^2*kc) -kpe/(2*Tmc^2*kc)
1/(2*Tmc^2*kc);
zeros(1,5) -1/Ta 0 0;
zeros(1,2) kn/(Ta*kd) zeros(1,3) -1/Ta 0
zeros(1,5) kie -kie 0];
B=[0 -1/Jm;
zeros(4,2);
1/Ta 0;
0 0
0 0];
C=zeros(6,8);
C(1,1)=1; C(2,2)=1; C(3,3)=1; C(4,4)=1; C(5,5)=1; C(6,8)=1;
D=zeros(6,2);
```

Електронне навчальне видання комбінованого використання
Можна використовувати в локальному мережному режимі

Канюк Геннадій Іванович
Василець Тетяна Юхимівна
Варфолмієв Олексій Олексійович

ТЕХНОЛОГІЇ НЕЙРОННИХ МЕРЕЖ І НЕЧІТКОГО МОДЕЛЮВАННЯ В СИСТЕМАХ УПРАВЛІННЯ

Методичні вказівки
до виконання курсового проєкту для здобувачів вищої освіти
другого (магістерського) рівня за спеціальністю 174 «Автоматизація,
комп'ютерно-інтегровані технології та робототехніка»

В авторській редакції

Підписано до розміщення 25.06.2025. Гарнітура Times New Roman.
Ум. друк. арк. 3,31. Обсяг. 3,09. Мб. Зам. № 296/25.

Харківський національний університет імені В. Н. Каразіна,
61022, м. Харків, майдан Свободи, 4.
Свідоцтво суб'єкта видавничої справи ДК № 3367 від 13.01.2009
Видавництво ХНУ імені В. Н. Каразіна