

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

З В І Т

**Лабораторна робота №3
з дисципліни
«Сучасні методи та моделі
інтелектуальних систем керування»**

Виконавець:

аспірант групи АКІТР-23-1а

Косей М.П.

Керівник:

викладач

Тиханський М. П.

2024

Лабораторна робота №2

Тема: Ознайомлення з принципами роботи нейронних мереж у складі систем керування

Мета: Дослідити систему керування з нейромережевими регуляторами

ХІД РОБОТИ

1) Ознайомитись з теоретичними відомостями до лабораторної роботи

Neural Network Toolbox дозволяє проектувати, моделювати та навчати нейронні мережі для задач обробки сигналів, керування, фінансового моделювання тощо. У пакеті реалізовані три типи регуляторів:

1. Регулятор із прогнозом (Predictive Controller):

Використовує нейронну мережу для прогнозування майбутньої поведінки процесу. Оптимізація керуючих сигналів виконується на кожному кроці, що вимагає значних обчислювальних ресурсів.

2. Регулятор NARMA-L2:

Найменш ресурсозатратний. Використовує реконструкцію моделі процесу, але вимагає його представлення у канонічній формі, що може спричиняти похибки.

3. Регулятор на основі еталонної моделі (Model Reference Controller):

Потребує навчання двох нейронних мереж (процесу та регулятора). Використовується для різних типів процесів, але має складний механізм навчання.

Проектування систем включає два етапи:

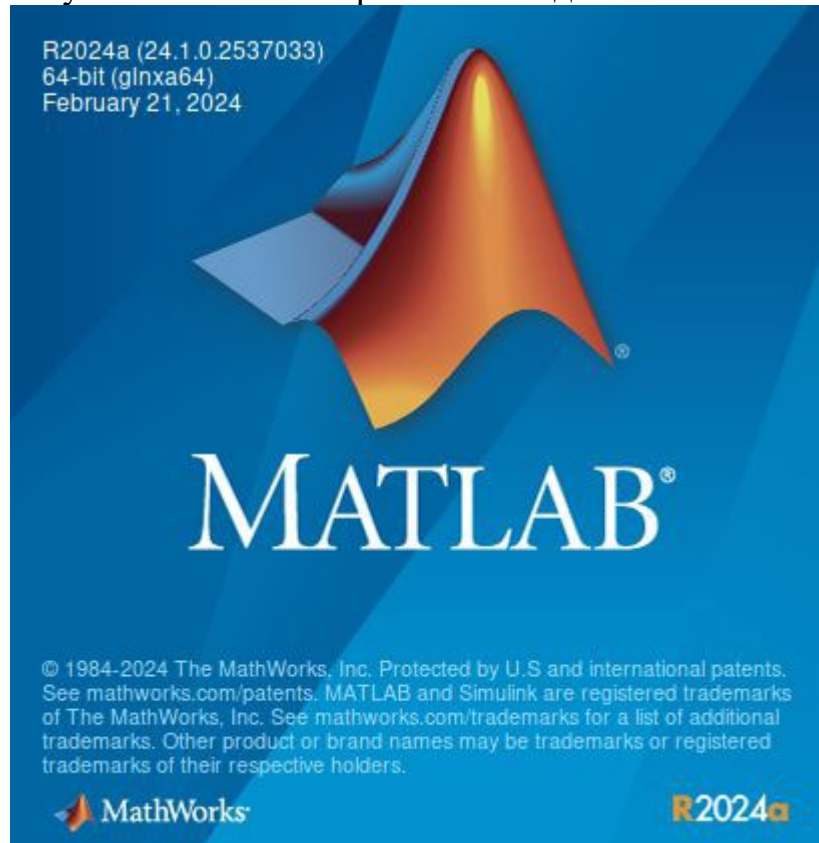
- Ідентифікація: створення нейромережевої моделі процесу.
- Синтез керування: побудова регулятора, який керує процесом.

Особливості регуляторів:

- Прогнозний регулятор забезпечує точність, але потребує значних обчислень.
- NARMA-L2 є простішим, але менш універсальним.
- Регулятор на основі еталонної моделі адаптивний, але складний у навчанні.

2) Практична частина

Використовуємо **MATLAB** версія R2024a для Linux.

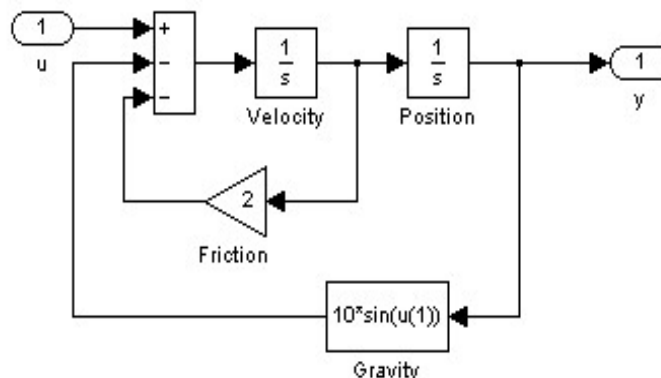


Реалізовуємо систему керування, яка б керувала рухом ланки промислового робота, відслідковуючи вихід еталонної моделі:

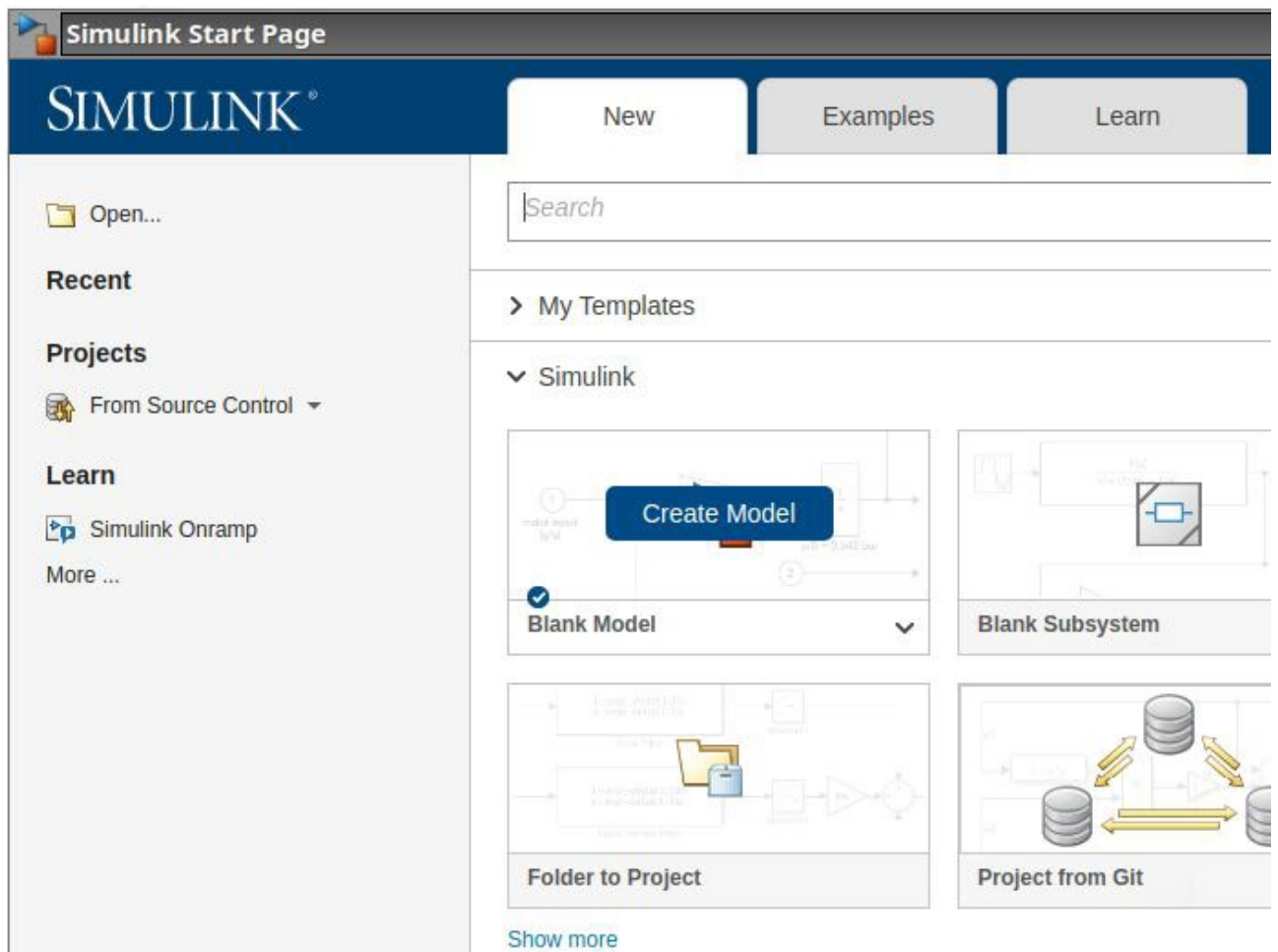
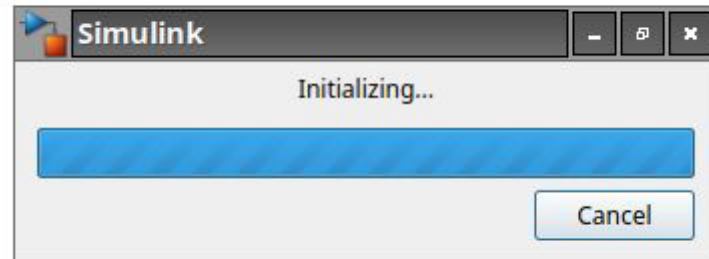
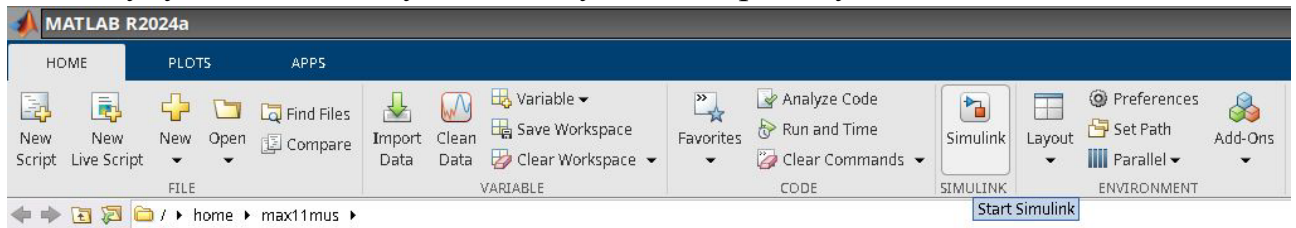
$$\frac{d^2 y_r}{dt^2} = -9y_r - 6 \frac{dy_r}{dt} + 9r$$

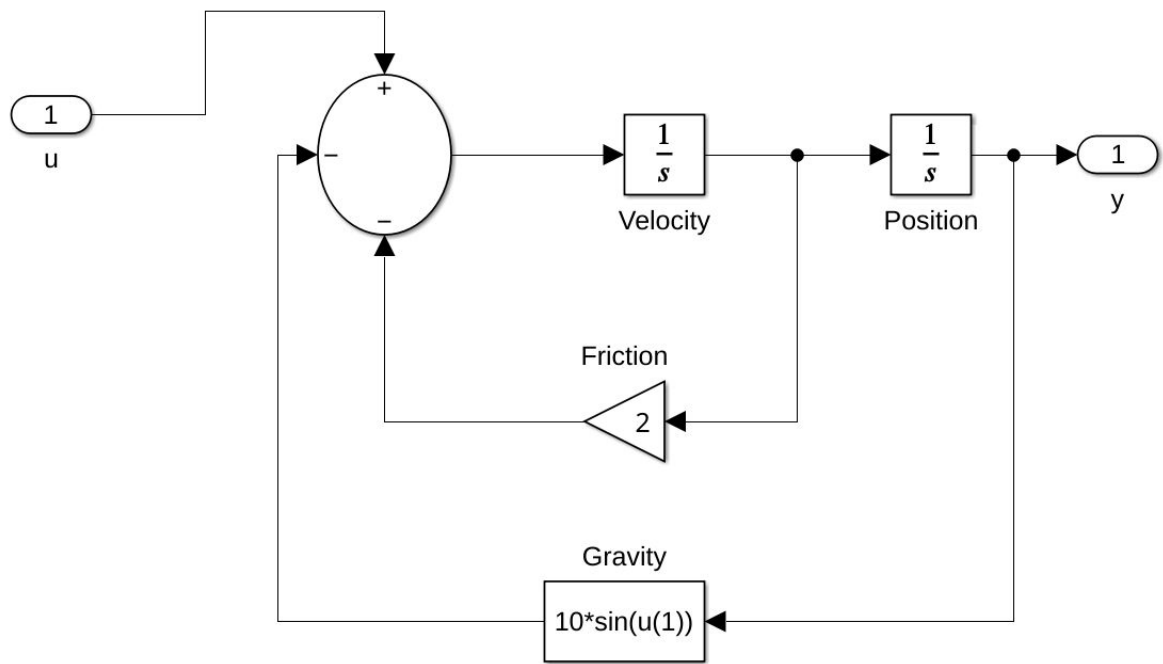
де y_r – вихід еталонної моделі; r – задає сигнал на вході моделі.

Для системи керування необхідна динамічна модель, реалізована в Simulink, що відповідає рівнянню руху ланки, має вигляд

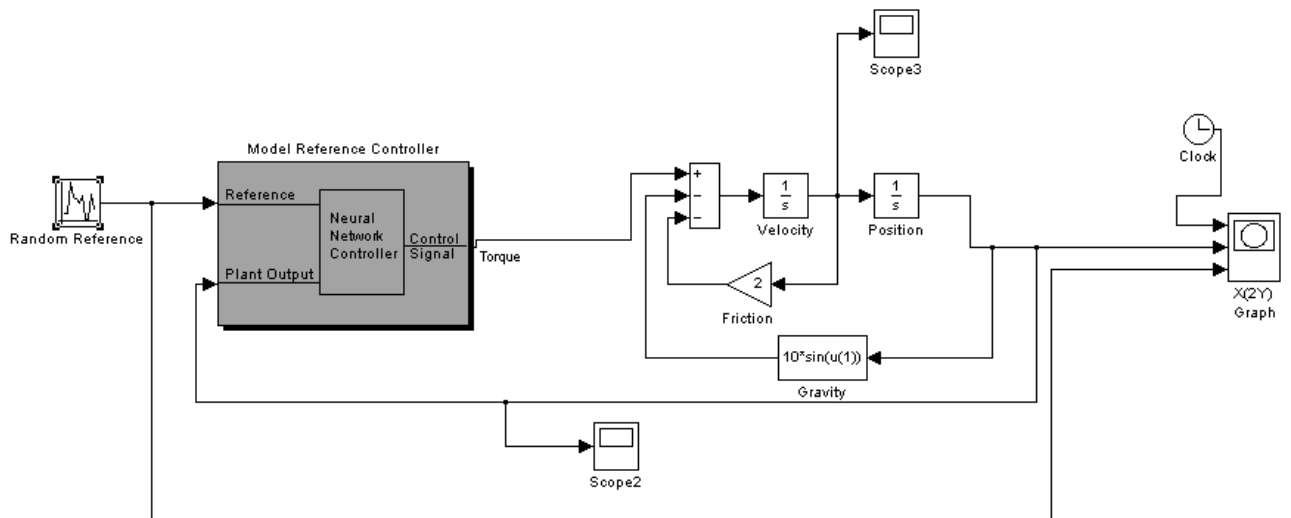


Будуємо відповідну динамічну модель, реалізувавши її в **Simulink**:



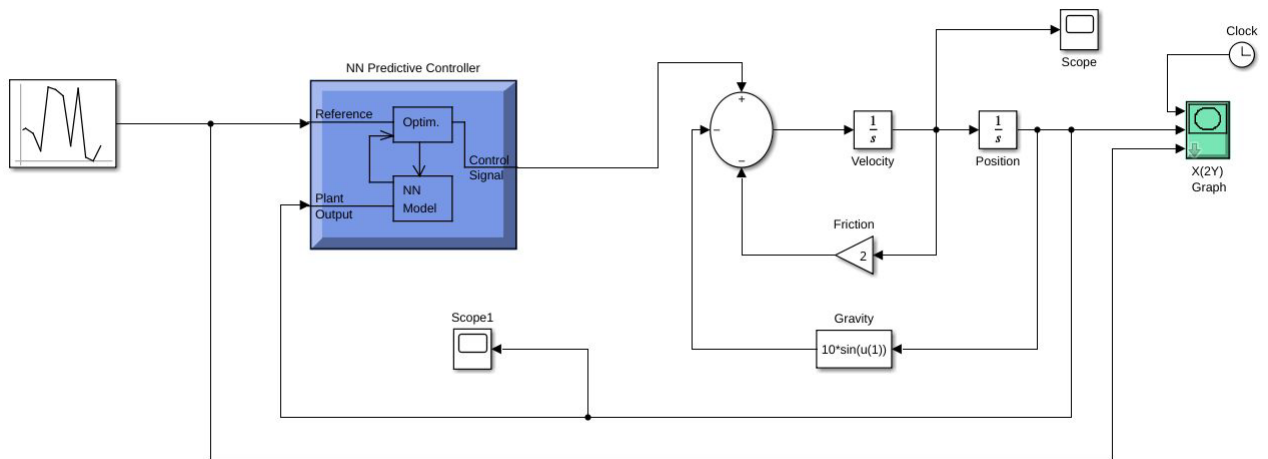
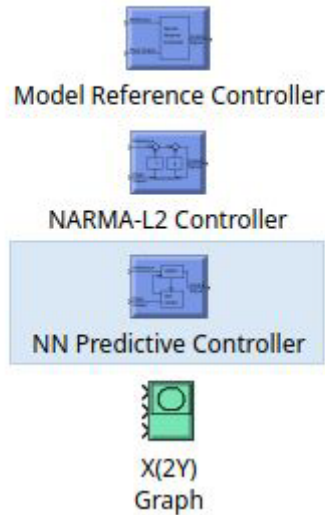


Будуєм відповідну динамічну модель, реалізувавши її в Simulink



Згідно варіанту №11 : тип нейрорегулятора - NN Predictiv Control, кількість слоїв 5, кількість епох -10.

- ▼ Shallow Neural Networks
- ▼ Control Systems



Налаштовуємо NN Predictiv Control та робимо Plant Identification

Plant Identification

File Window Help

Network Architecture

Size of Hidden Layer: 15

Sampling Interval (sec): 0.05

No. Delayed Plant Inputs: 2

No. Delayed Plant Outputs: 2

☐ Normalize Training Data

Training Data

Training Samples: 10000

Maximum Plant Input: 15

Minimum Plant Input: -15

Maximum Interval Value (sec): 2

Minimum Interval Value (sec): 0.1

☒ Limit Output Data

Maximum Plant Output: 3.1

Minimum Plant Output: -3.1

Simulink Plant Model: robotarm

Generate Training Data Import Data Export Data

Training Parameters

Training Epochs: 10

Training Function: trainlm

☒ Use Current Weights ☒ Use Validation Data ☒ Use Testing Data

Train Network OK Cancel Apply

Generate or import data before training the neural network plant.

Neural Network Predictive Control

File Window Help

Cost Horizon (N2): 0

Control Horizon (Nu): 0

Control Weighting Factor (V): 0

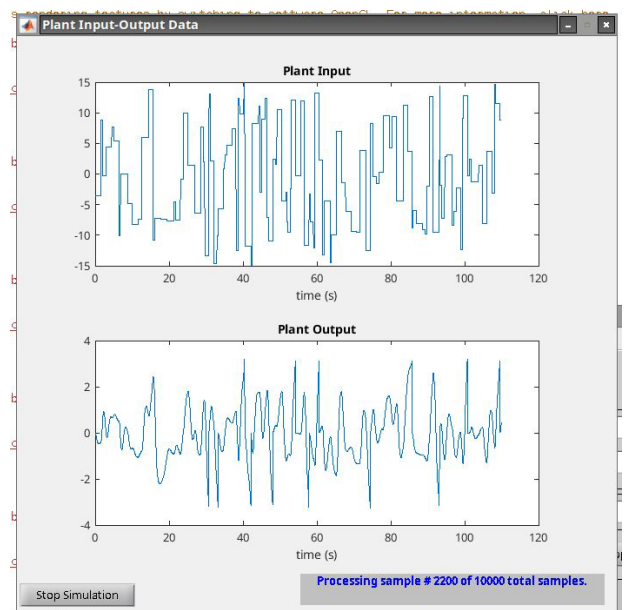
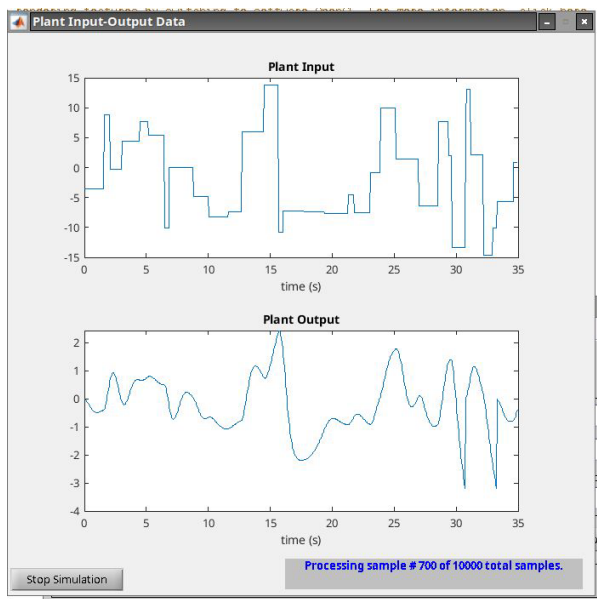
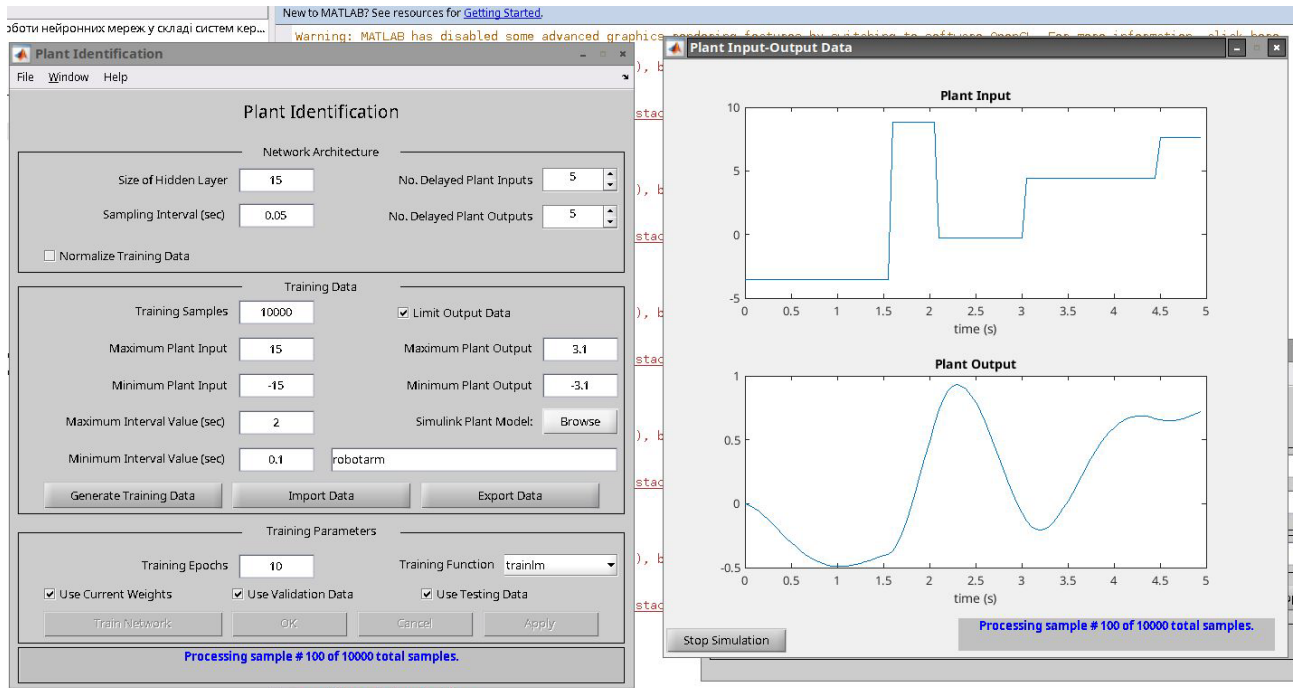
Search Parameter (λ): 0

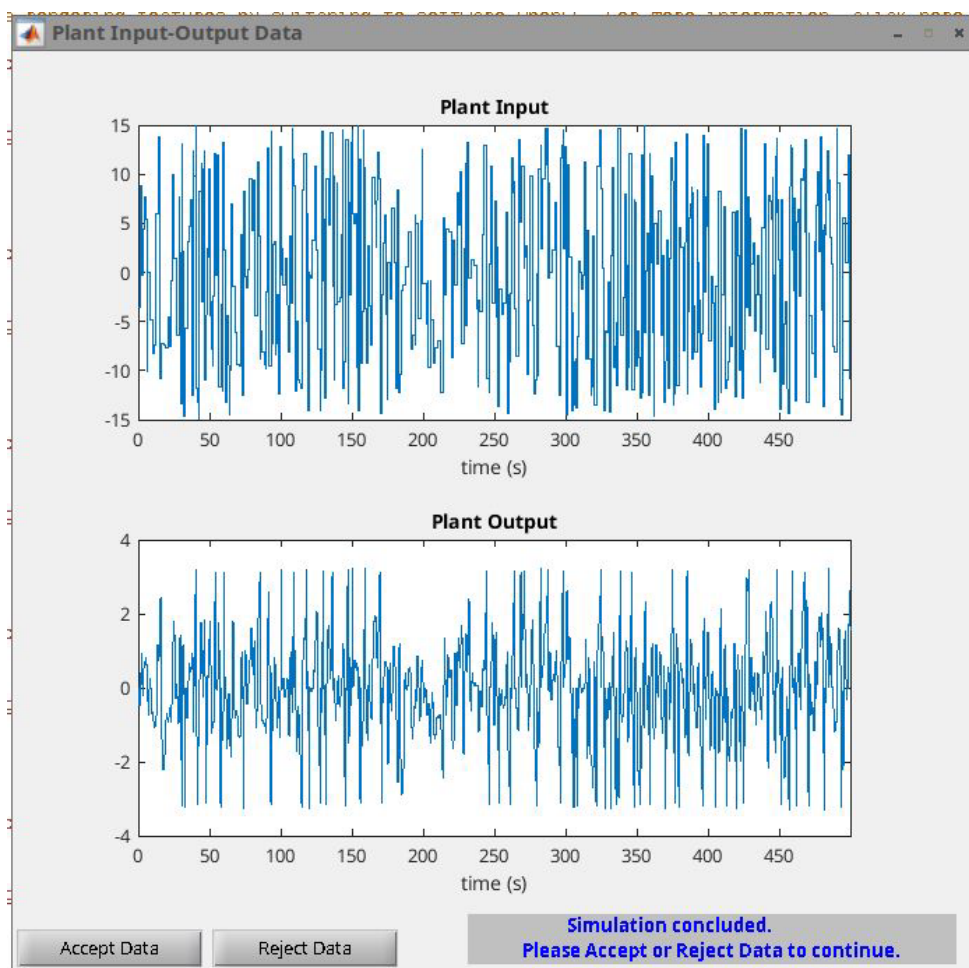
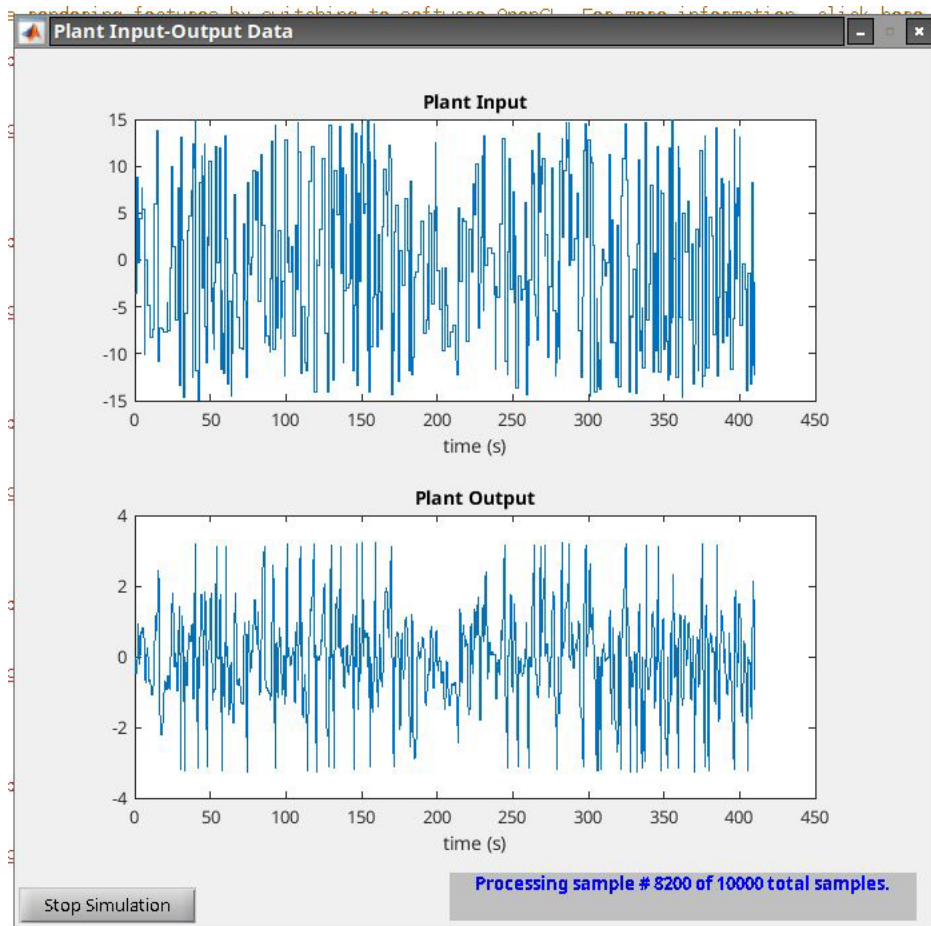
Minimization Routine: csrchbc

Iterations Per Sample Time: 0

Plant Identification OK Cancel Apply

Perform plant identification before controller configuration.





Тренуємо нейромережу.

Plant Identification

File Window Help

Plant Identification

Network Architecture

Size of Hidden Layer: 15
Sampling Interval (sec): 0.05
No. Delayed Plant Inputs: 5
No. Delayed Plant Outputs: 5

☐ Normalize Training Data

Training Data

Training Samples: 10000
Maximum Plant Input: 15
Minimum Plant Input: -15
Maximum Interval Value (sec): 2
Minimum Interval Value (sec): 0.1
Limit Output Data: ☒
Maximum Plant Output: 3.1
Minimum Plant Output: -3.1
Simulink Plant Model: Browse

Erase Generated Data Import Data Export Data

Training Parameters

Training Epochs: 10
Training Function: trainlm
☒ Use Current Weights ☒ Use Validation Data ☒ Use Testing Data

Train Network OK Cancel Apply

Your training data set has 10000 samples.
You can now train the network.

Plant Identification

File Window Help

Plant Identification

Network Architecture

Size of Hidden Layer: 15
Sampling Interval (sec): 0.05
No. Delayed Plant Inputs: 5
No. Delayed Plant Outputs: 5

☐ Normalize Training Data

Training Data

Training Samples: 10000
Maximum Plant Input: 15
Minimum Plant Input: -15
Maximum Interval Value (sec): 2
Minimum Interval Value (sec): 0.1
Limit Output Data: ☒
Maximum Plant Output: 3.1
Minimum Plant Output: -3.1
Simulink Plant Model: Browse

Erase Generated Data Import Data Export Data

Training Parameters

Training Epochs: 10
Training Function: trainlm
☒ Use Current Weights ☒ Use Validation Data ☒ Use Testing Data

Train Network OK Cancel Apply

Training Neural Network

Neural Network Train... (24-Nov-2024 13:35:27)

Network Diagram

Training Results

Training finished: Reached maximum number of epochs ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	10	10
Elapsed Time	-	00:00:09	-
Performance	2.67	6.05e-06	0
Gradient	45.5	0.0163	1e-10
Mu	0.001	1e-05	1e+10
Validation Checks	0	0	6

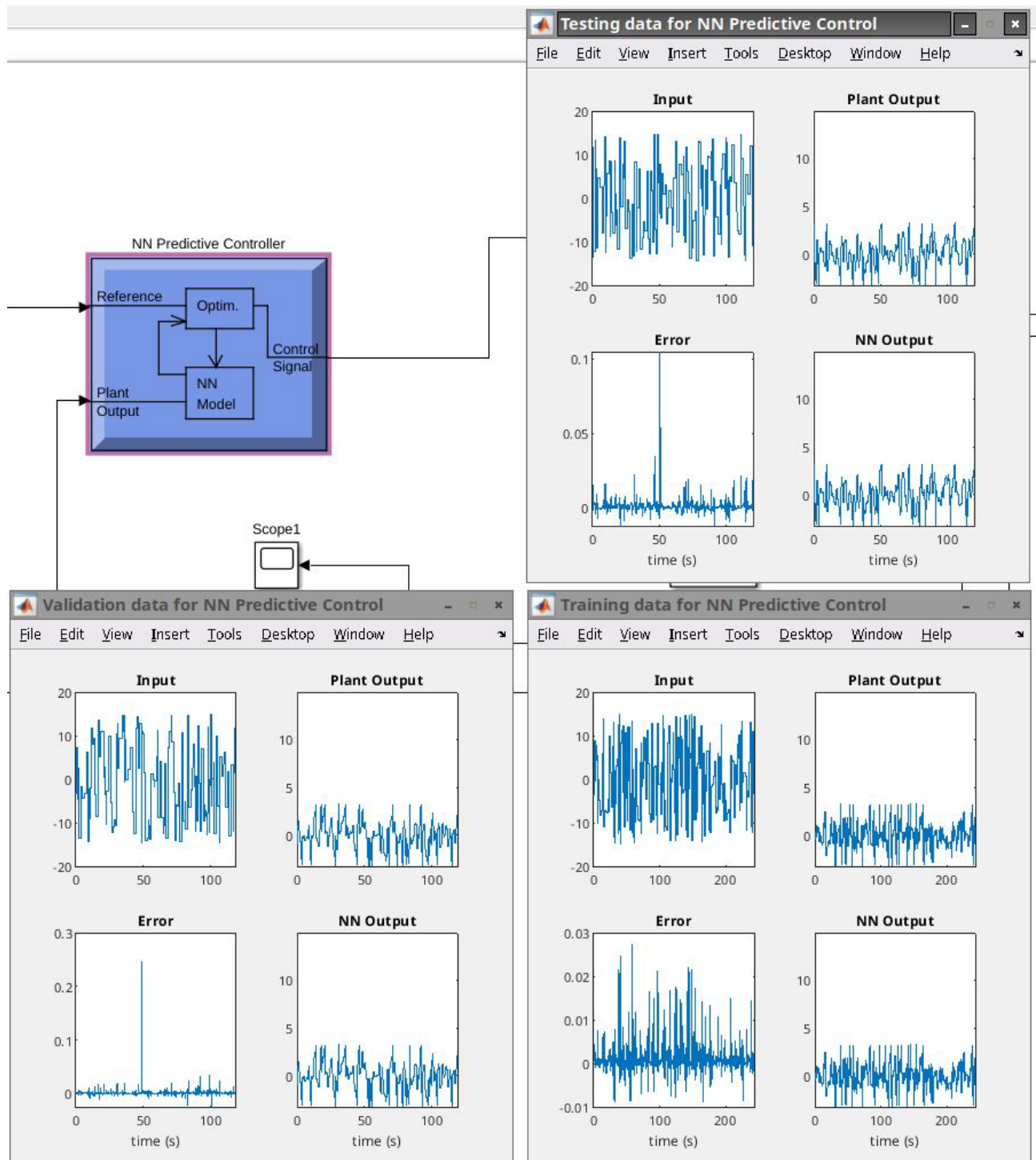
Training Algorithms

Data Division: Index divideind
Training: Levenberg-Marquardt trainlm
Performance: Mean Squared Error mse
Calculations: MEX

Training Plots

Performance Training State

Regression



ВИСНОВКИ

В результаті виконаної лабораторної роботи ознайомилися з принципами роботи нейронних мереж у складі систем керування.

Усі матеріали викладенні у репозиторії GitHub, за посиланням <https://github.com/Max11mus/LAB3-Modern-Methods-and-Models-of-Intelligent-Control-Systems>.