

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ, ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΤΗΛΕΠΟΙΚΟΙΝΩΝΙΩΝ  
Π.Μ.Σ. ΣΤΗΝ ΡΟΜΠΟΤΙΚΗ  
P202 – ΜΗΧΑΝΙΚΗ ΕΥΦΥΙΑ

## “NNArm”

Το NNArm είναι ένα σύστημα ελέγχου ενός ρομποτικού βραχίονα, βασισμένο σε τεχνικές ελέγχου με νευρωνικά δίκτυα και υλοποιημένο στο περιβάλλον MATLAB/Simulink.

Εργασία στην Ενότητα Νευρωνικά Δίκτυα του Μεταπτυχιακού Φοιτητή  
Κωστελίδη Ιορδάνη  
ΑΕΜ: 146 – EMAIL: [iordkost@ihu.gr](mailto:iordkost@ihu.gr)

INTERNATIONAL HELLENIC UNIVERSITY  
POLYTECHNIC SCHOOL  
DEPARTMENT OF COMPUTER, INFORMATICS AND TELECOMMUNICATIONS ENGINEERING  
M.Sc. in ROBOTICS  
P202 – MACHINE INTELLIGENCE

## “NNArm”

NNArm is a MATLAB/Simulink-based project for controlling a robotic arm using advanced neural network control techniques.

Semester Project on Neural Networks of the Postgraduate Student  
Kostelidis Iordanis  
Student ID: 146 – EMAIL: [iordkost@ihu.gr](mailto:iordkost@ihu.gr)

## Περιγραφή

Το NNArm είναι ένα σύστημα ελέγχου ενός ρομποτικού βραχίονα, βασισμένο σε τεχνικές ελέγχου με νευρωνικά δίκτυα και υλοποιημένο στο περιβάλλον MATLAB/Simulink. Στόχος του συστήματος είναι η υλοποίηση, προσομοίωση και σύγκριση διαφορετικών στρατηγικών νευρωνικού ελέγχου.

## Abstract

NNArm is a MATLAB/Simulink-based project for controlling a robotic arm using advanced neural network control techniques. The aim of the project is to implement, simulate, and compare multiple neural control strategies.

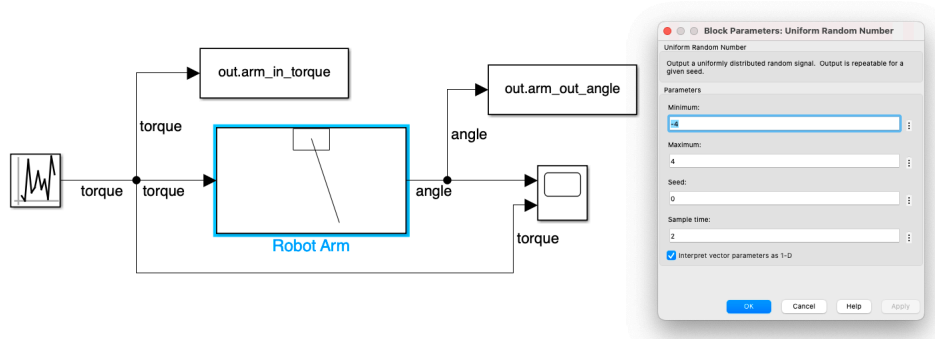
## Περιεχόμενα / Table of Contents

<b>Περιγραφή.....</b>	<b>3</b>
<b>Abstract.....</b>	<b>3</b>
<b>Συστήματα Δειγματοληψίας .....</b>	<b>5</b>
Σύστημα Robot Arm .....	5
Σύστημα Reference.....	5
<b>Neural Network Predictive Controller.....</b>	<b>6</b>
<b>NARMA-L2 Neural Controller.....</b>	<b>11</b>
<b>Model Reference Controller.....</b>	<b>16</b>

# Συστήματα Δειγματοληψίας

## Σύστημα Robot Arm

Έγινε σχεδιασμός στο Simulink ενός συστήματος, για την δημιουργία dataset (input, output) για χρήση από τους νευρικούς ελεγκτές.

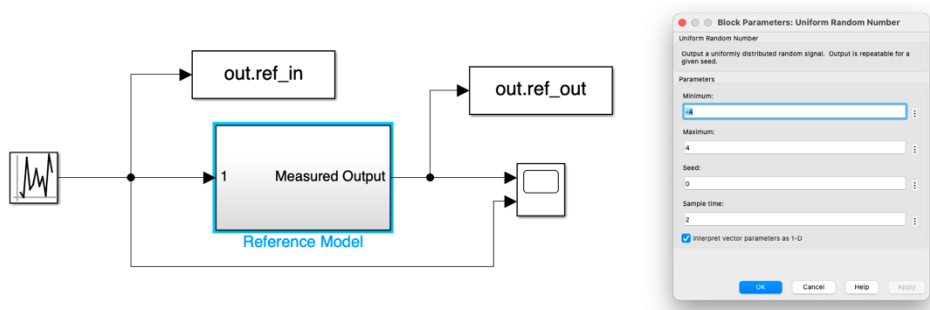


Εικόνα 1

Ορίσαμε ως όρια του URN (Uniform Random Number) από -4 έως 4 με sample time 2

## Σύστημα Reference

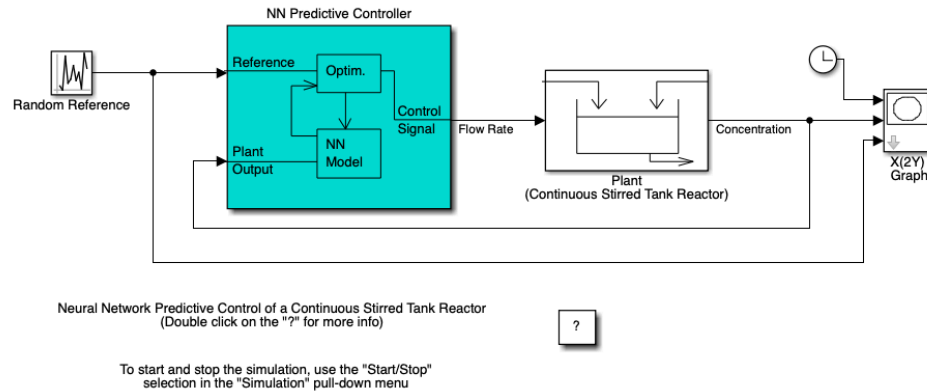
Έγινε σχεδιασμός στο Simulink ενός συστήματος, για την δημιουργία reference dataset (input, output) για χρήση από τον ελεγκτή τύπου Model-Reference.



Εικόνα 2

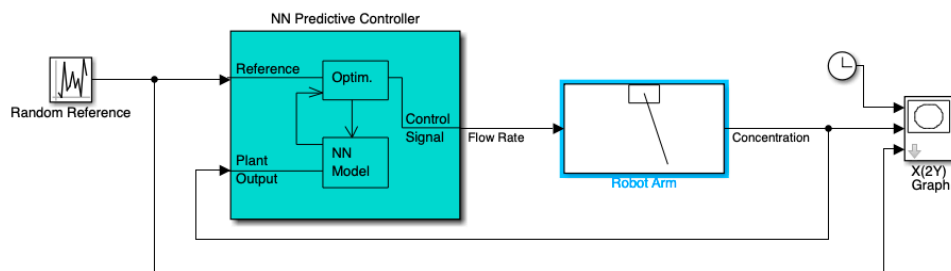
# Neural Network Predictive Controller

Για την δημιουργία συστήματος με «Neural Network Predictive Controller», χρησιμοποιούμε ως βάση το παράδειγμα «predcstr».



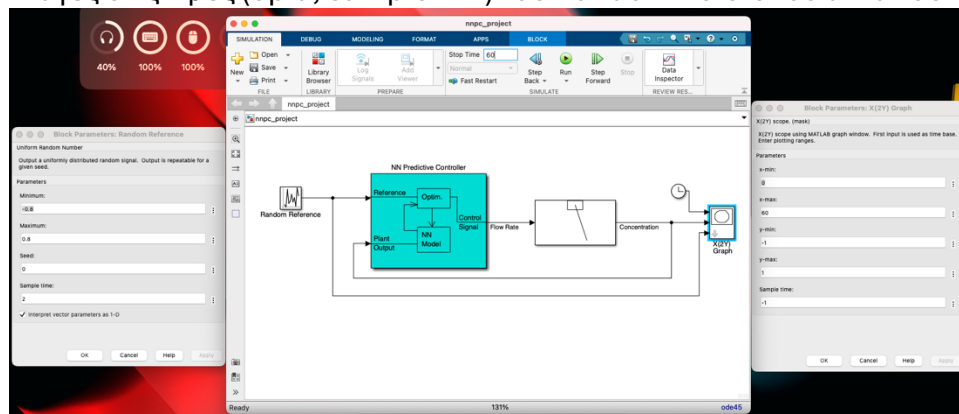
Εικόνα 3

Αφαιρούμε ότι δεν χρειαζόμαστε και αντικαθιστούμε το Plant με το δικό μας σύστημα (Robot Arm).



Εικόνα 4

Κάνουμε αλλαγές στις τιμές (όρια, sample κλπ) του Random Reference αλλά του X(2Y) Graph



Εικόνα 5

Καθώς ξεκινάμε με βάση ένα example άλλου συστήματος, αλλάζουμε τις βασικές παραμέτρους, φορτώνουμε τα δεδομένα μας από το workspace (arm\_in\_torgue, arm\_out\_angle) και εκτελούμε την εκπαίδευση.

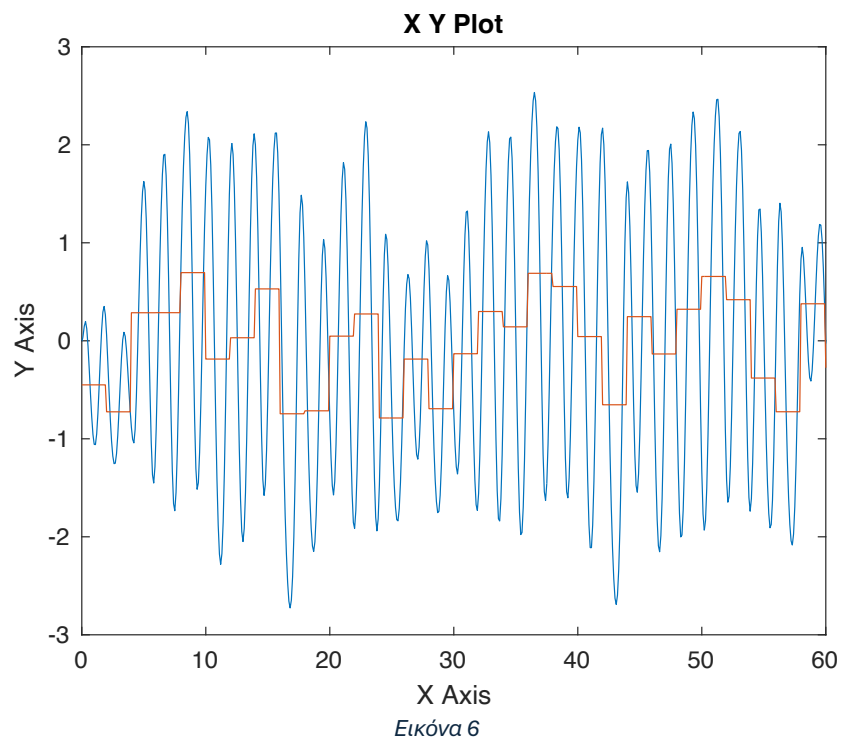
The screenshot shows the 'Plant Identification' dialog box with the following settings:

- Network Architecture:**
  - Size of Hidden Layer: 2
  - Sampling Interval (sec): 0.1
  - No. Delayed Plant Inputs: 1
  - No. Delayed Plant Outputs: 1
  - ☐ Normalize Training Data
- Training Data:**
  - Training Samples: 10000
  - Maximum Plant Input: 4
  - Minimum Plant Input: -4
  - Maximum Interval Value (sec): 3
  - Minimum Interval Value (sec): 2
  - Limit Output Data: ☐
  - Maximum Plant Output: 23
  - Minimum Plant Output: 20
  - Simulink Plant Model: Browse
  - Model Name: cstr
- Training Parameters:**
  - Training Epochs: 200
  - Training Function: trainlm
  - ☒ Use Current Weights
  - ☒ Use Validation Data
  - ☐ Use Testing Data

Buttons at the bottom: Erase Imported Data, Import Data, Export Data, Train Network, OK, Cancel, Apply.

Message at the bottom: Your training data set has 10000 samples. You can now train the network.

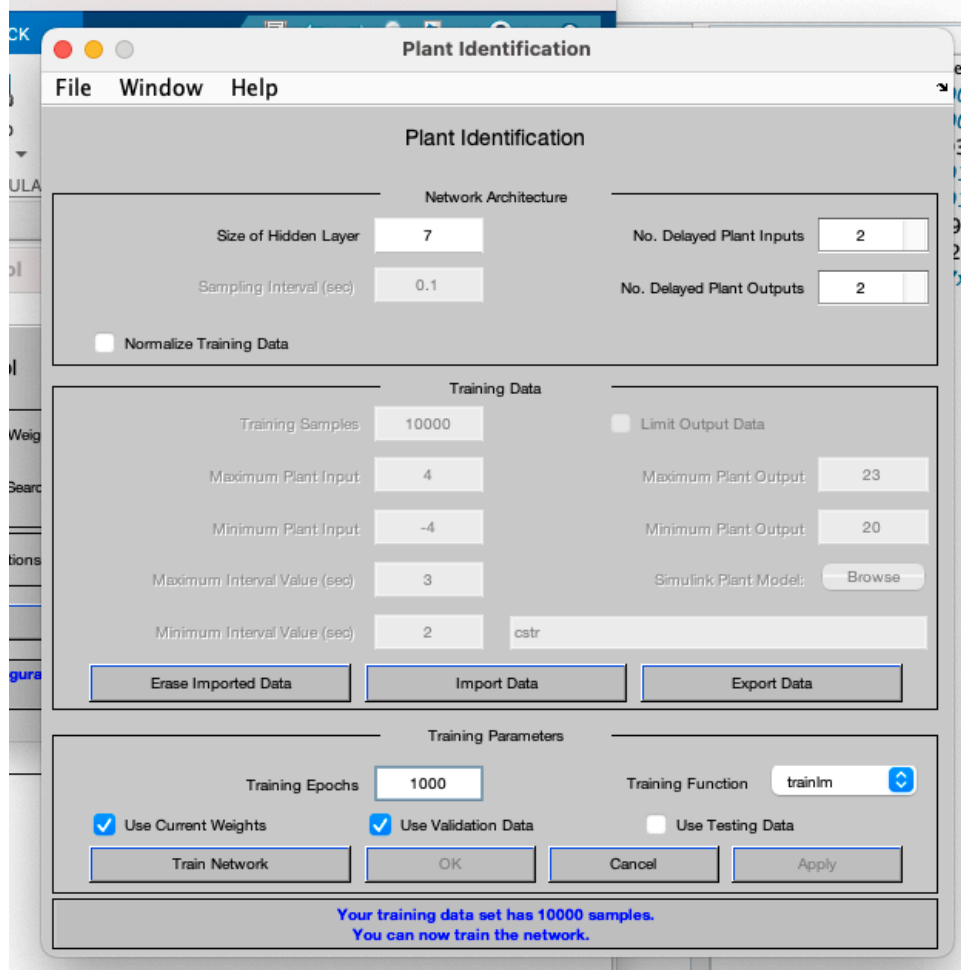
Μετά την εκπαίδευση, τρέχουμε το πείραμα για 60s για να δούμε πως συμπεριφέρεται.



Παρατηρούμε ότι έχει πολύ μεγάλες αποκλίσεις.

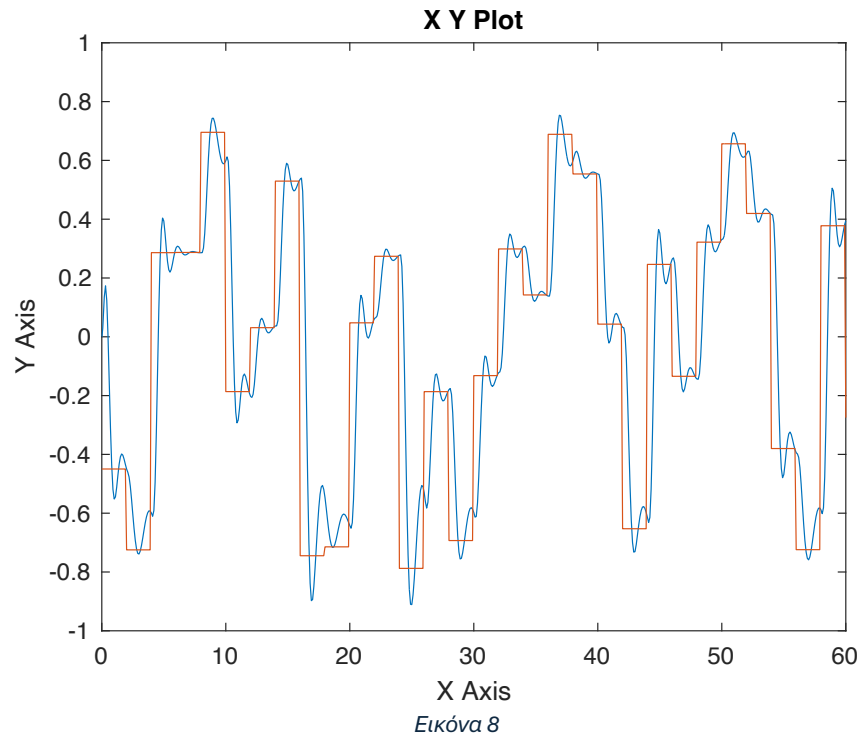
Ξαναεκπαιδεύουμε το μοντέλο μας, με τις παρακάτω ρυθμίσεις.





Εικόνα 7

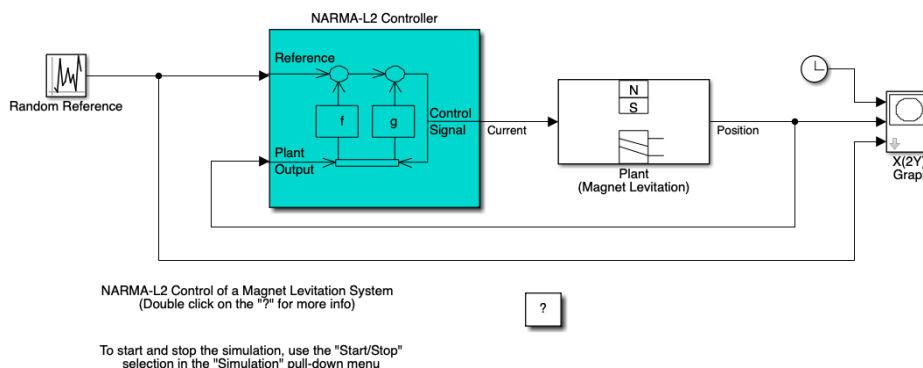
Στην συνέχεια τρέχουμε το πείραμα για 60s για να δούμε πως συμπεριφέρεται με τις νέες ρυθμίσεις.



Παρατηρούμε ότι έχει λιγότερες και μικρότερες αποκλίσεις.

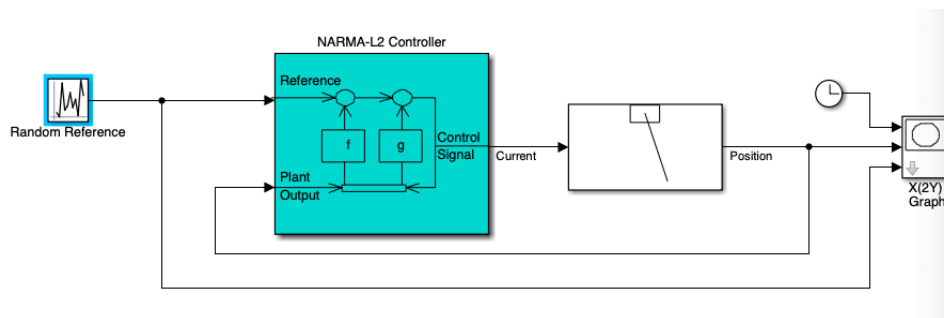
# NARMA-L2 Neural Controller

Για την δημιουργία συστήματος με «NARMA-L2 Neural Controller», χρησιμοποιούμε ως βάση το παράδειγμα «narmamaglev».



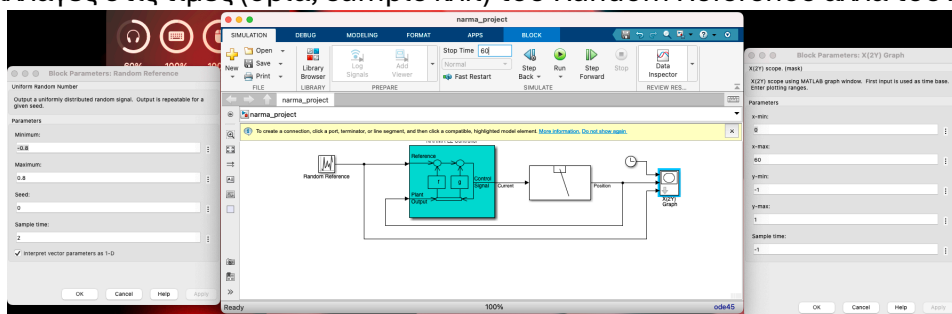
Εικόνα 9

Αφαιρούμε ότι δεν χρειαζόμαστε και αντικαθιστούμε το Plant με το δικό μας σύστημα (Robot Arm).



Εικόνα 10

Κάνουμε αλλαγές στις τιμές (όρια, sample κλπ) του Random Reference αλλά του X(2Y) Graph



Εικόνα 11

Καθώς ξεκινάμε με βάση ένα example άλλου συστήματος, αλλάζουμε τις βασικές παραμέτρους, φορτώνουμε τα δεδομένα μας από το workspace (arm\_in\_torgue, arm\_out\_angle) και εκτελούμε την εκπαίδευση.

Plant Identification - NARMA-L2

File Window Help

Plant Identification - NARMA-L2

Network Architecture

Size of Hidden Layer: 2

Sampling Interval (sec): 0.1

No. Delayed Plant Inputs: 1

No. Delayed Plant Outputs: 1

☐ Normalize Training Data

Training Data

Training Samples: 10000

Maximum Plant Input: 4

Minimum Plant Input: -4

Maximum Interval Value (sec): 3

Minimum Interval Value (sec): 2

Limit Output Data: ☐

Maximum Plant Output: Inf

Minimum Plant Output: 0

Simulink Plant Model: Browse

ballrepel0

Erase Imported Data Import Data Export Data

Training Parameters

Training Epochs: 200

Training Function: trainlm

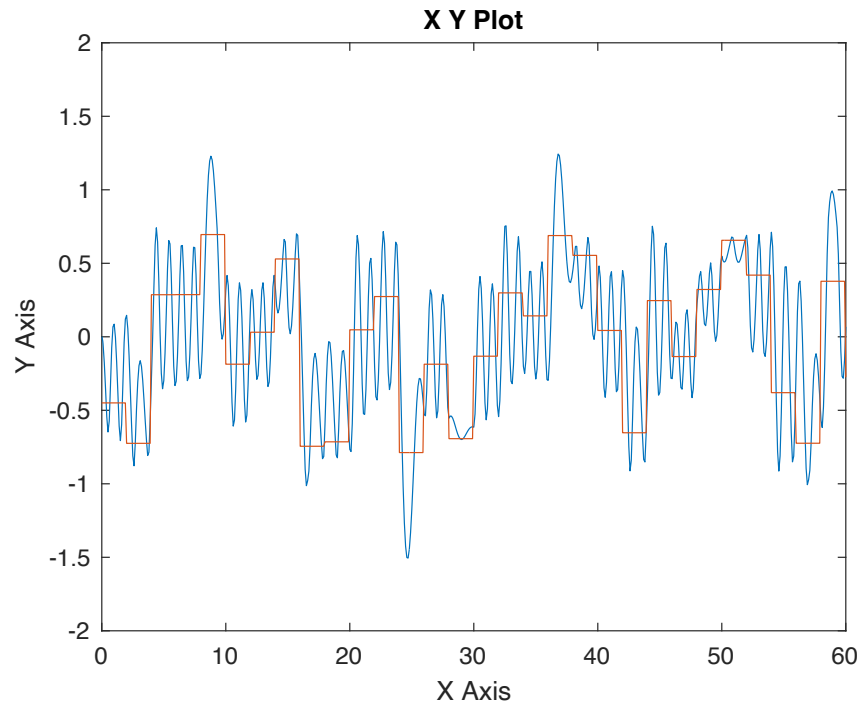
☒ Use Current Weights ☒ Use Validation Data ☐ Use Testing Data

Train Network OK Cancel Apply

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

Εικόνα 12

Μετά την εκπαίδευση, τρέχουμε το πείραμα για 60s για να δούμε πως συμπεριφέρεται.



Εικόνα 13

Παρατηρούμε ότι έχει μεγάλες αποκλίσεις.

Ξαναεκπαιδεύουμε το μοντέλο μας, με τις παρακάτω ρυθμίσεις.

Plant Identification - NARMA-L2

File Window Help

Plant Identification - NARMA-L2

Network Architecture

Size of Hidden Layer: 4

Sampling Interval (sec): 0.1

No. Delayed Plant Inputs: 3

No. Delayed Plant Outputs: 5

☐ Normalize Training Data

Training Data

Training Samples: 10000

Maximum Plant Input: 4

Minimum Plant Input: -4

Maximum Interval Value (sec): 3

Minimum Interval Value (sec): 2

Limit Output Data: ☐

Maximum Plant Output: Inf

Minimum Plant Output: 0

Simulink Plant Model: Browse

ballrepel0

Erase Imported Data Import Data Export Data

Training Parameters

Training Epochs: 1000

Training Function: trainlm

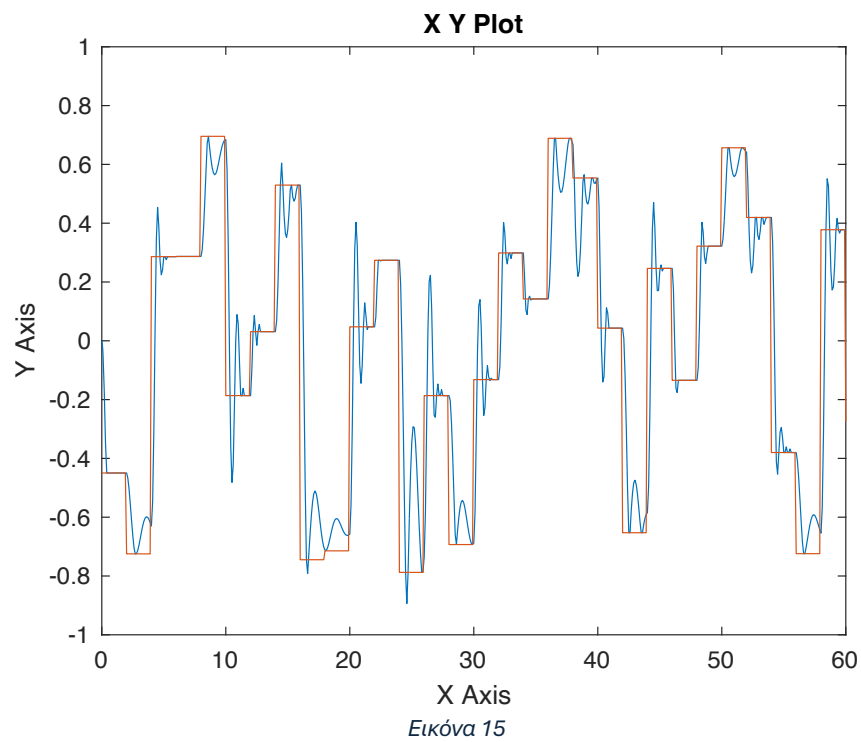
☒ Use Current Weights ☒ Use Validation Data ☒ Use Testing Data

Train Network OK Cancel Apply

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

Εικόνα 14

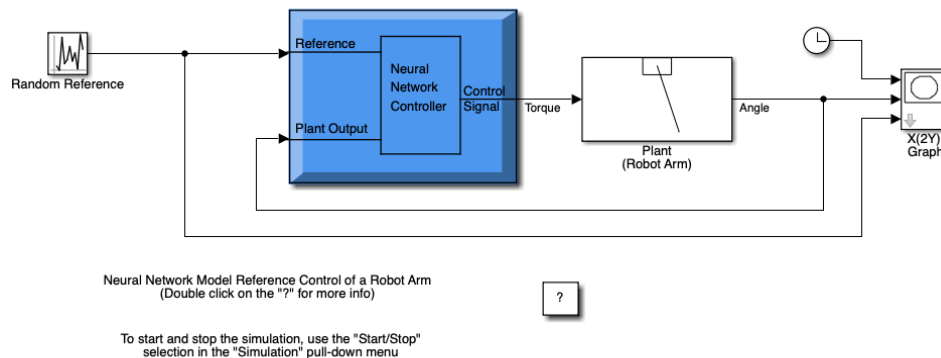
Στην συνέχεια τρέχουμε το πείραμα για 60s για να δούμε πως συμπεριφέρεται με τις νέες ρυθμίσεις.



Παρατηρούμε ότι έχει λιγότερες και μικρότερες αποκλίσεις.

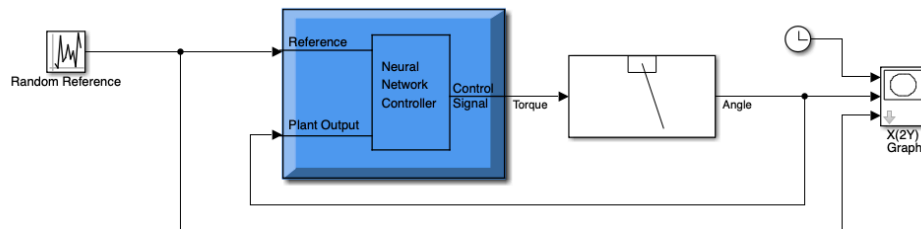
# Model Reference Controller

Για την δημιουργία συστήματος με « Model Reference Controller», χρησιμοποιούμε ως βάση το παράδειγμα «mrefrobotarm».



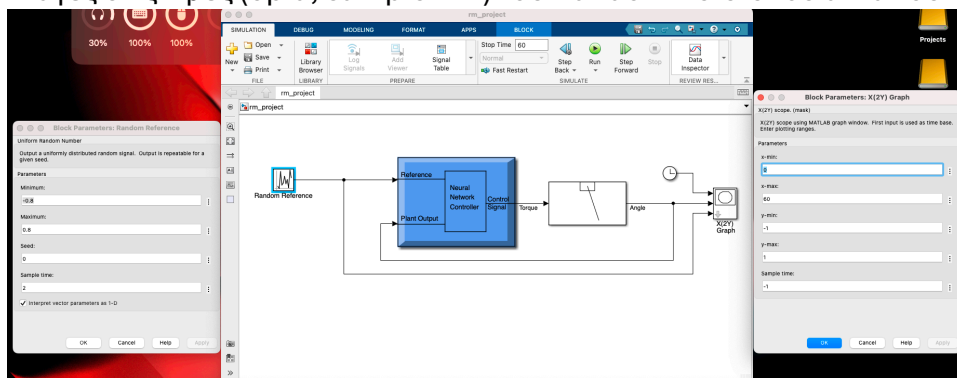
Εικόνα 16

Αφαιρούμε ότι δεν χρειαζόμαστε και αντικαθιστούμε το Plant με το δικό μας σύστημα (Robot Arm).



Εικόνα 17

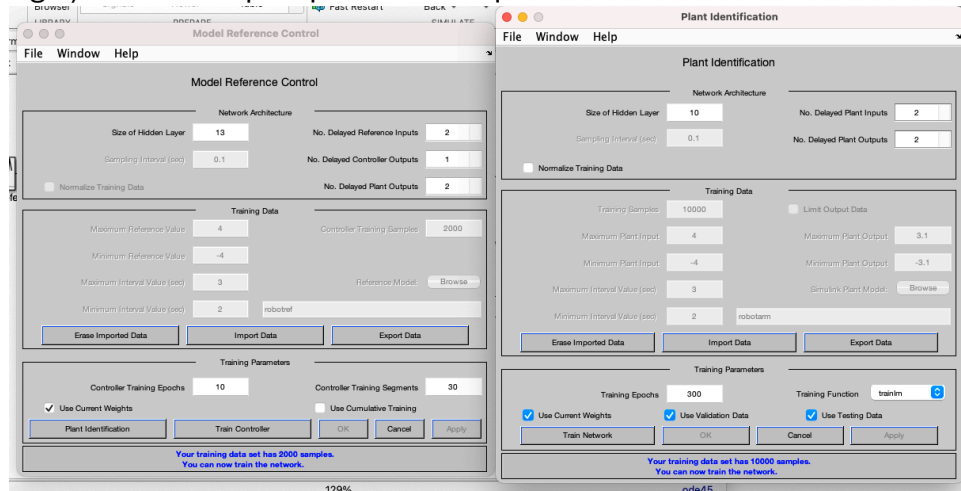
Κάνουμε αλλαγές στις τιμές (όρια, sample κλπ) του Random Reference αλλά του X(2Y) Graph



Εικόνα 18

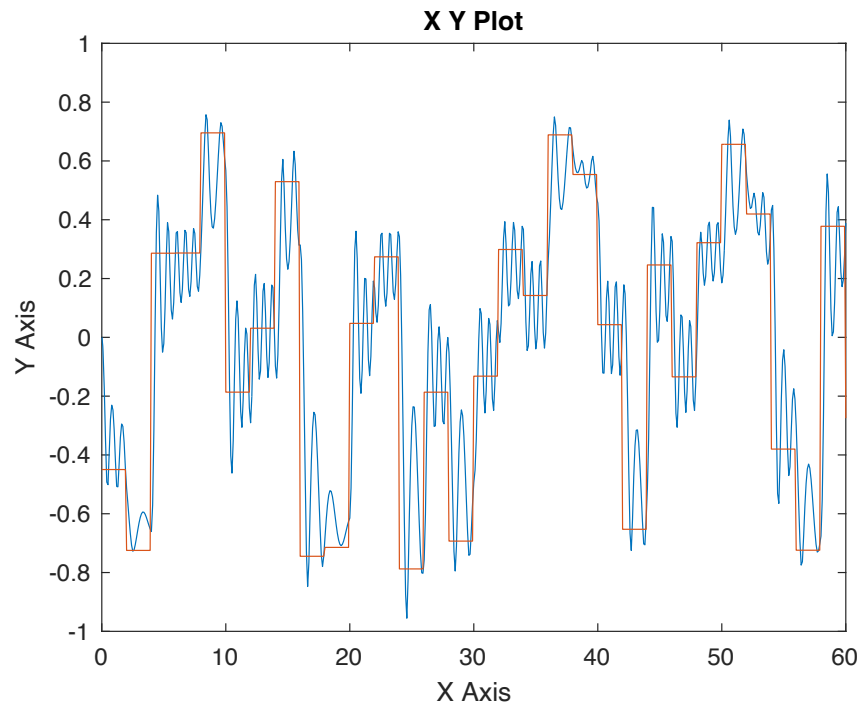


Καθώς ξεκινάμε με βάση ένα example άλλου συστήματος, αλλάζουμε τις βασικές παραμέτρους, φορτώνουμε τα δεδομένα μας από το workspace (arm\_in\_torgue, arm\_out\_angle) και εκτελούμε την εκπαίδευση.



Εικόνα 19

Μετά την εκπαίδευση, τρέχουμε το πείραμα για 60s για να δούμε πως συμπεριφέρεται.



Εικόνα 20

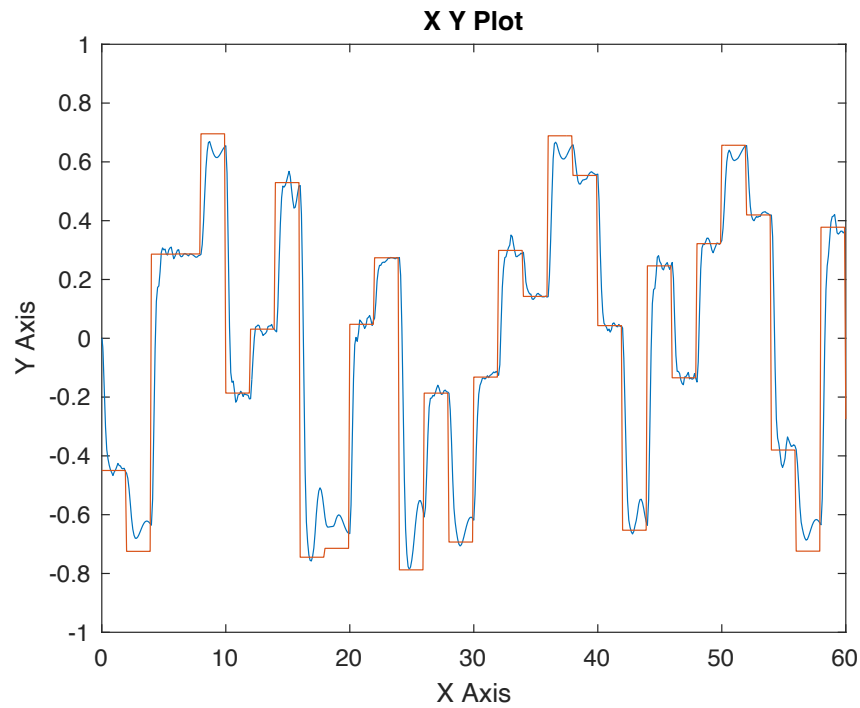
Παρατηρούμε ότι έχει μεγάλες αποκλίσεις.

Ξαναεκπαιδεύουμε το μοντέλο μας, με τις παρακάτω ρυθμίσεις.



Εικόνα 21

Στην συνέχεια τρέχουμε το πείραμα για 60s για να δούμε πως συμπεριφέρεται με τις νέες ρυθμίσεις.



Εικόνα 22

Παρατηρούμε ότι έχει λιγότερες και μικρότερες αποκλίσεις.