

POLITECHNIKA POZNAŃSKA

WYDZIAŁ AUTOMATYKI, ROBOTYKI I ELEKTROTECHNIKI

INSTYTUT ROBOTYKI I INTELIGENCJI MASZYNOWEJ

ZAKŁAD STEROWANIA I ELEKTRONIKI PRZEMYSŁOWEJ



NEURONOWY REGULATOR ADAPTACYJNY Z  
MODELEM ODNIESIENIA.

METODY INTELIGENCJI MASZYNOWEJ W  
AUTOMATYCE

MATERIAŁY DO ZAJĘĆ LABORATORYJNYCH

MGR INŻ. ADRIAN WÓJCIK

ADRIAN.WOJCIK@PUT.POZNAN.PL

## I. CEL ZAJĘĆ

### WIEDZY

Celem zajęć jest zapoznanie z:

- strukturą wewnętrzną neuronowego regulatora z modelem odniesienia,
- procedurą akwizycji danych uczących oraz treningu (adaptacji) neuronowego regulatora z modelem odniesienia.

### UMIEJĘTNOŚCI

Celem zajęć jest nabycie umiejętności w zakresie:

- wykorzystania modeli symulacyjnych obiektów dynamicznych w procesie uczenia neuronowego regulatora z modelem odniesienia,
- generowania sygnałów wejściowych dla obiektu sterowania adekwatnych, użytecznych w zadaniu uczenia modelu neuronowego,
- generowania sygnałów wejściowych dla modelu odniesienia zamkniętej pętli regulacji, użytecznych w zadaniu uczenia regulatora neuronowego,
- walidacji neuronowego regulatora z modelem odniesienia w otwartej i zamkniętej pętli sterowania.

### KOMPETENCJI SPOŁECZNYCH

Celem zajęć jest kształtowanie właściwych postaw w zakresie:

- prawidłowego zarządzania bazą kodu i danych,
- prawidłowej prezentacji i dokumentacji sztucznych sieci neuronowych.
- skutecznej komunikacji w kontekście metod inteligencji maszynowej.

## II. POLECENIA KOŃCOWE

Wykonaj [zadania laboratoryjne](#) zgodnie z poleceniami i wskazówkami prowadzącego. Zaprezentuj rozwiązania prowadzącemu. Zrealizowane zadania są oceniane zero-jedynkowo.

W przypadku niezrealizowania wszystkich zadań w trakcie zajęć możliwe jest wykonanie raportu w celu uzupełnienia punktacji uzyskanej na zajęciach. Raport oceniany jest dodatkowo pod względem redakcyjnym. Wymogi redakcyjne oraz szablon raportu laboratoryjnego dostępne są w na portalu *eKursy* w katalogu „Wymogi raportu laboratoryjnego”. Raport należy przesłać jako rozwiązanie zadania na portalu *eKursy* w terminie do 7 dni od realizacji zajęć.

## III. PRZYGOTOWANIE DO ZAJĘĆ

### A) ZAPOZNANIE Z PRZEPISAMI BHP

Wszystkie informacje dotyczące instrukcji BHP laboratorium są zamieszczone w sali laboratoryjnej oraz na stronie Zakładu [1]. Wszystkie nieścisłości należy wyjaśnić z prowadzącym laboratorium. Wymagane jest zaznajomienie i zastosowanie do regulaminu.

Na zajęcia należy przyjść przygotowanym zgodnie z tematem zajęć. Obowiązuje również materiał ze wszystkich odbytych zajęć.

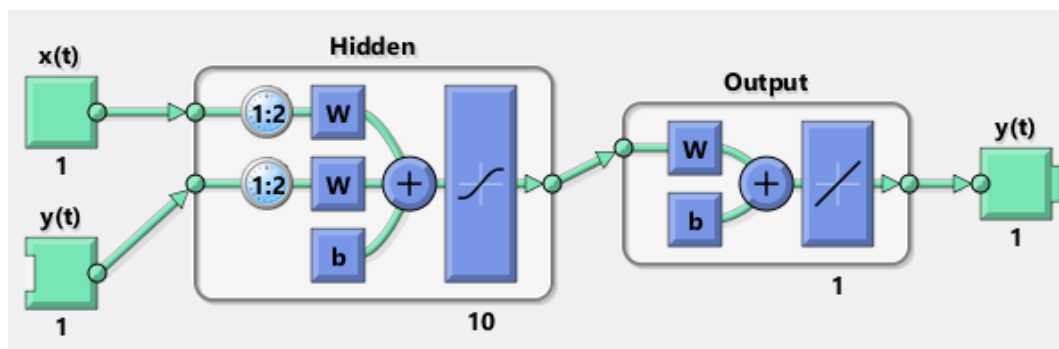
## B) REKURENCYJNE SZTUCZNE SIECI NEURONOWE

Nieliniowy model autoregresywny z zewnętrznym wejściem (ang. *nonlinear autoregressive exogenous model*, NARX) to klasa dyskretnych, rekurencyjnych modeli wejściowo-wyjściowych dla procesów stochastycznych. Modele NARX stanowią uogólnienie struktury ARX; ich ogólna postać algebraiczna może zostać zapisana jako (1).

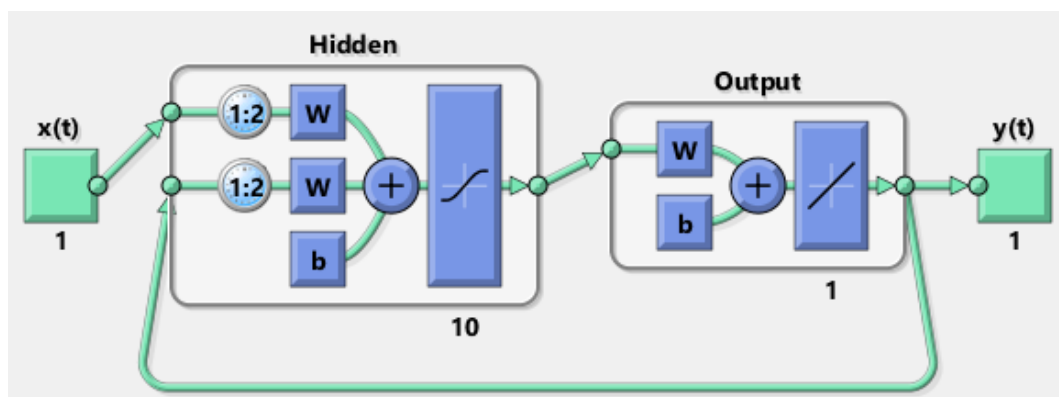
$$\underline{y}_k = F\left(\underline{y}_{k-1}, \underline{y}_{k-2}, \dots, \underline{y}_{k-N}, \underline{u}_k, \underline{u}_{k-1}, \dots, \underline{u}_{k-M}\right) + \varepsilon_k \quad (1)$$

gdzie:  $\underline{y}_k$  -  $k$ -ta próbka sygnału wyjściowego,  
 $\underline{u}_k$  -  $k$ -ta próbka sygnału wejściowego,  
 $\varepsilon_k$  -  $k$ -ta próbka szumu / błędu pomiarowego,  
 $N$  - liczba historycznych próbek wyjściowych,  
 $M$  - liczba historycznych próbek wejściowych,  
 $F$  - funkcja nieliniowa, np. w formie wielomianu, szeregu harmonicznego, **sztucznej sieci neuronowej** etc,

Podstawowym zastosowaniem NARX jest modelowanie układów dynamicznych – w tym urządzeń technicznych. Parametry nieliniowej funkcji  $F$  dobierane są na podstawie szeregów czasowych wejść i wyjść modelowanego obiektu na drodze optymalizacji. Funkcją celu tego procesu optymalizacji celu jest stopień dopasowania danych wyjściowych obiektu do odpowiedzi modelu dla tego samego wymuszenia. Ze względu na to, że sztuczne sieci neuronowe są uniwersalnym aproksymatorem funkcji oraz dostępność szerokiej gamy algorytmów optymalizacji ich parametrów (metod *uczenia*) modele NARX w wielu praktycznych zastosowaniach przyjmują postać sztucznych sieci neuronowych.



(a) Sieć NARX z otwartym sprzężeniem zwrotnym (dodatkowe wejście).



(b) Sieć NARX z zamkniętym sprzężeniem zwrotnym.

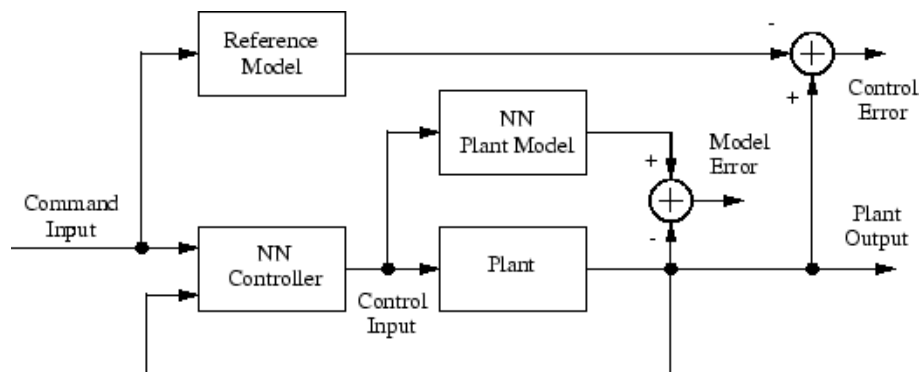
Rys. 1. Schemat rekurencyjnych sztucznych sieci neuronowych w środowisku MATLAB.

W środowisku MATLAB modele NARX stanowią szczególny wariant obiektów klasy `network` z pakietu Deep Learning Toolbox. Do tworzenia obiektu tego typu można zastosować funkcję (konstruktor) `narxnet`. Na rys. 1. przedstawiono schemat rekursywnej sztucznej sieci neuronowej w środowisku MATLAB: (a) bez zamkniętego sprzężenia zwrotnego – tę wersję wykorzystuje się do uczenia off-line; (b) z zamkniętym sprzężeniem zwrotnym – tę wersję wykorzystuje się do ewaluacji oraz w finalnym układzie regulacji. Aby uzyskać sieć z zamkniętą pętlą sprzężenia zwrotnego należy wykorzystać funkcję `closeloop`.

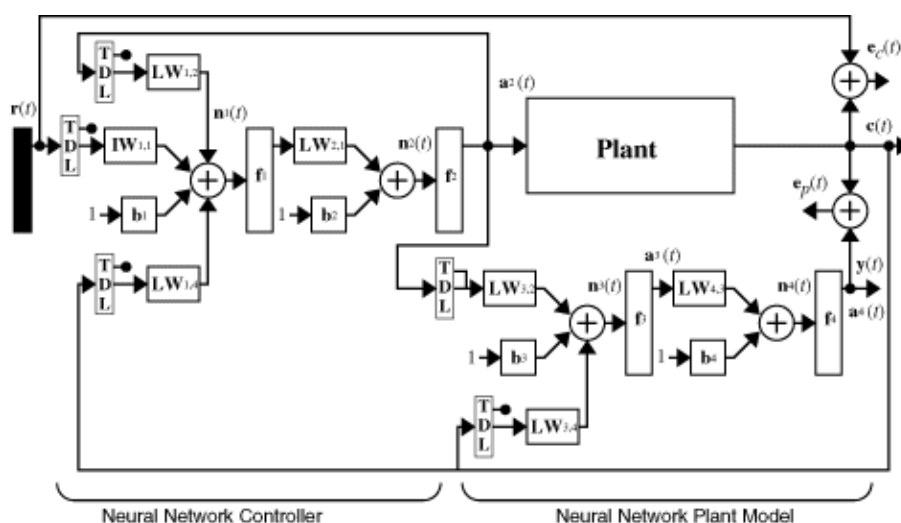
**Uwaga: w środowisku MATLAB sztuczne sieci neuronowe typu NARX mają wbudowane bloki opóźnień próbek wejściowych!**

### c) WPROWADZENIE DO NEURONOWEGO REGULATORA Z MODELEM ODNIESIENIA

Sterowanie adaptacyjne z modelem odniesienia (ang. *model reference adaptive control*, MRAC) to kategoria sterowania adaptacyjnego, w którym pożądane, docelowe działanie zamkniętej pętli regulacji definiowane jest za pomocą modelu odniesienia (modelu referencyjnego). Proces adaptacji ma na celu taki dobór parametrów regulatora, aby odpowiedź zamkniętego układu regulacji była możliwie najbardziej zbliżona do odpowiedzi modelu referencyjnego na to samo wymuszenie. Na rys. 2. przedstawiono ogólny schemat blokowy układu z modelem referencyjnym wykorzystujący neuronowy regulator (NN Controller) oraz neuronowy model obiektu sterowania (NN Plant Model).



Rys. 2. Schemat blokowy adaptacyjnego układu sterowania z modelem odniesienia.



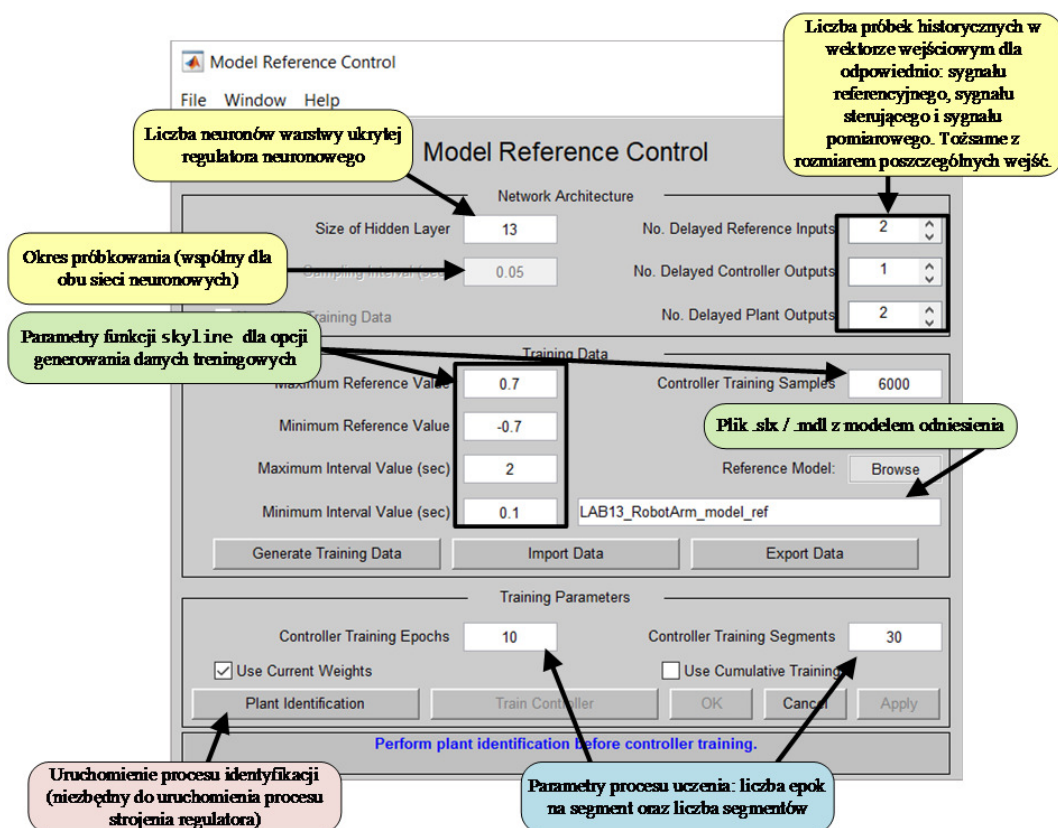
wania, tj. uczenie neuronowego modelu obiektu sterowania za pomocą wstecznej propagacji błędów na podstawie błędu modelowania (**Model Error**). Gdy proces uczenia/identyfikacji zakończy się sukcesem, wagi modelu neuronowego zostają zablokowane (nie podlegają aktualizacji). Następnie przeprowadzany jest proces strojenia regulatora neuronowego, tj. uczenie układu *regulator–model neuronowy* za pomocą wstecznej propagacji błędów na podstawie różnicy względem odpowiedzi modelu odniesienia (**Control Error**). Na rys. 3. przedstawiona jest kompletna struktura neuronowa wraz z obiektem sterowania.

Model neuronowy nie wchodzi bezpośrednio w strukturę układu sterowania, stanowi jedynie element układu adaptacji wag regulatora neuronowego. Jego rolą w systemie jest jedynie umożliwienie przeprowadzenia wstecznej propagacji uchybu regulacji do wyjścia regulatora.

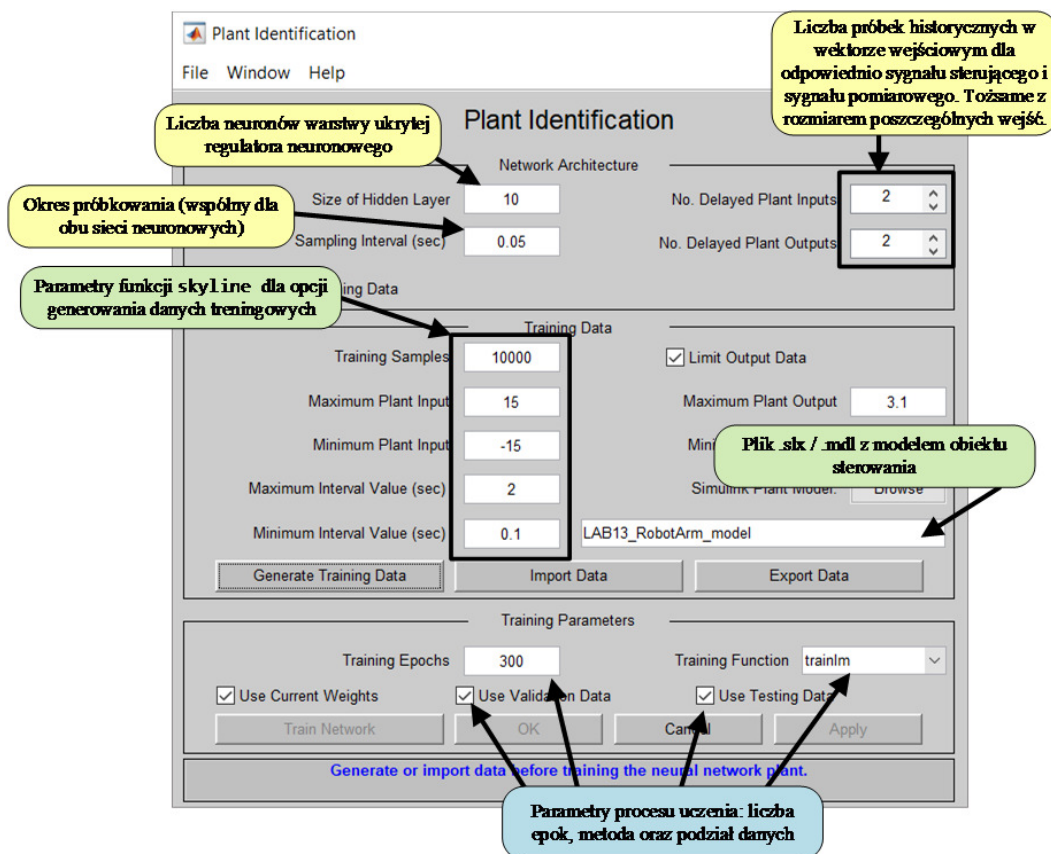
#### D) OBSŁUGA BLOKU MODEL REFERENCE CONTROL

**Uwaga:** Oryginalne demo neuronowego układu MRAC można uruchomić wpisując w konsoli środowiska MATLAB polecenie `mrefrobotarm`.

Blok *Model Reference Control*, stanowiący element dostarczonego demo układu regulacji, jest implementacją dwóch sieci neuronowych (regulator i model) [2]. Za pomocą graficznego interfejsu użytkownika (ang. *graphical user interface*, GUI) bloku możliwe jest także przeprowadzenie adaptacji układu sterowania bezpośrednio z poziomu środowiska Simulink z wykorzystaniem Deep Learning Toolbox. Proces uczenia podzielony jest na dwa etapy – identyfikację obiektu sterowania oraz strojenie regulatora. Interfejs bloku również podzielony jest na dwa elementy: okno identyfikacji obiektu sterowania (rys. 4.) oraz okno strojenia regulatora (rys. 5.).



Rys. 4. Interfejs graficzny bloku Model Reference Control #1: Strojenie regulatora neuronowego



Rys. 5. Interfejs graficzny bloku Model Reference Control #2: Identyfikacja obiektu sterowania

#### E) ŚRODKI DYDAKTYCZNE

- Sprzętowe: • komputer,  
Programowe: • MATLAB (Deep Learning Toolbox).

#### F) ZADANIA DO REALIZACJI

- Przygotowanie danych uczących do **procesu identyfikacji obiektu sterowania**.
  - Zapoznaj się z dostarczonym modelem neuronowego systemu MRAC. Uruchom symulację dla domyślnych parametrów i przeanalizuj przebiegi sygnałów. Zapoznaj się z interfejsem graficznym bloku Model Reference Control [2].
  - Zapoznaj się dostarczonym skrypcem realizującym trening regulatora neuronowego. Uruchom skrypt. Przeanalizuj powstałe struktury neuronowe oraz wyniki procesu uczenia [3].
  - Stwórz model symulacyjny obcowzbudnego silnika prądu stałego w oparciu o klasę `state space` oraz blok `LTI`. Przyjmij wartości znamionowe oraz parametry modelu podane przez prowadzącego (np.  $\omega_N = 157$  rad/s;  $T_N = 525$  Nm;  $u_N = 420$  V;  $i_N = 127$  A;  $R = 0.472$   $\Omega$ ;  $L = 7.85$  mH;  $k_\Phi = 4.13$  Wb;  $J = 0.32$  kg·m<sup>2</sup>).
  - Należy przygotować minimum dwa osobne zestawy danych: jeden zestaw do treningu (optymalizacji) modelu, drugi do jego walidacji (sprawdzenia).
  - Układ powinien logować niezbędne dane identyfikacyjne: sygnał wejściowy (*napięcie*) oraz sygnał wyjściowy (*prędkość kątowa*).
  - Sygnały referencyjne powinny obejmować cały założony zakres sterowania.
  - Zaprezentuj zebrane dane w formie graficznej. Możesz dokonać również wstępnej obróbki danych, np. normalizacji.

2. Na podstawie uzyskanych danych przeprowadź uczenie **modelu neuronowego**.
  - (a) Zaproponuj topologie sieci: rozmiar wejść, liczbę i rozmiar warstw ukrytych, liczbę sprzężeń zwrotnych.
  - (b) Przeprowadź uczenie modelu neuronowego z użyciem bloku **Model Reference Control**. Alternatywnie wykorzystaj bezpośrednio dostępne narzędzia z Deep Learning Toolbox pisząc skrypty lub wykorzystując dostępne aplikacje okienkowe.
  - (c) Zaprezentuj odpowiedź modelu neuronowego dla danych uczących.
3. Dokonaj walidacji uzyskanego **modelu neuronowego** za pomocą osobnego zestawu danych.
  - (a) Przeprowadź symulację z użyciem uzyskanego modelu neuronowego i danych walidacyjnych.
  - (b) Zaprezentuj odpowiedź modelu neuronowego dla danych walidacyjnych. Porównaj wynik z odpowiedzią układu / modelu symulacyjnego.
  - (c) W razie potrzeby powtórz czynności iteracyjnie modyfikując strukturę sieci oraz parametry uczenia do osiągnięcia dobrych rezultatów w procesie walidacji.
4. Przygotowanie danych uczących do **optymalizacji regulatora neuronowego**.
  - (a) Zaproponuj model odniesienia zamkniętego układu regulacji prędkości napędu z silnikiem obcowzbudnym. Przyjmij prostą dynamikę układu, np. obiekt inercyjny II rzędu.
  - (b) Przeprowadź symulację zaproponowanego modelu logując jego wejścia (*referencyjną prędkość kątową*) oraz wyjście (*mierzoną prędkość kątową*). Należy przygotować minimum dwa osobne zestawy danych: jeden zestaw do treningu (optymalizacji) regulatora, drugi do jego walidacji (sprawdzenia).
  - (c) Sygnały referencyjne powinny obejmować cały dostępny zakres sterowania.
  - (d) Zaprezentuj zebrane dane w formie graficznej. Możesz dokonać również dodatkowej obróbki danych, np. normalizacji.
5. Na podstawie uzyskanych danych przeprowadź uczenie **regulatora neuronowego** z użyciem uzyskanego wcześniej modelu neuronowego.
  - (a) Zaproponuj topologie sieci: rozmiar wejść, liczbę i rozmiar warstw ukrytych, liczbę sprzężeń zwrotnych.
  - (b) Dołącz uzyskany model neuronowy do struktury NARX. Pamiętaj o wyłączeniu zezwolenia na uczenie dla tej sekcji sieci.
  - (c) Przeprowadź uczenie modelu neuronowego z użyciem bloku **Model Reference Control**. Alternatywnie wykorzystaj bezpośrednio dostępne narzędzia z Deep Learning Toolbox pisząc skrypty lub wykorzystując dostępne aplikacje okienkowe.
  - (d) Zaprezentuj odpowiedź modelu neuronowego dla danych uczących.
6. Dokonaj walidacji uzyskanego **regulatora neuronowego** za pomocą osobnego zestawu danych.
  - (a) Przeprowadź symulacje z użyciem uzyskanego regulatora neuronowego i danych walidacyjnych.
  - (b) Zaprezentuj odpowiedź neuronowego układu regulacji dla danych walidacyjnych. Porównaj wynik z odpowiedzią modelu referencyjnego.
  - (c) W razie potrzeby powtórz czynności iteracyjnie modyfikując strukturę sieci oraz parametry uczenia do osiągnięcia dobrej jakości sterowania w procesie walidacji.

---

## BIBLIOGRAFIA

1. *Regulaminy porządkowe, instrukcje BHP* [online]. [B.d.] [udostępniono 2019-09-30]. Dostępne z: <http://zsep.cie.put.poznan.pl/materialy-dydaktyczne/MD/Regulaminy-porz%C4%85dkowe-instrukcje-BHP/>.
2. *Design Model-Reference Neural Controller in Simulink - MATLAB & Simulink* [online] [udostępniono 2022-05-30]. Dostępne z: <https://www.mathworks.com/help/deeplearning/ug/design-model-reference-neural-controller-in-simulink.html>.
3. *Create Reference Model Controller with MATLAB Script - MATLAB & Simulink* [online] [udostępniono 2022-05-30]. Dostępne z: <https://www.mathworks.com/help/deeplearning/ug/create-reference-model-controller-with-matlab-script.html>.