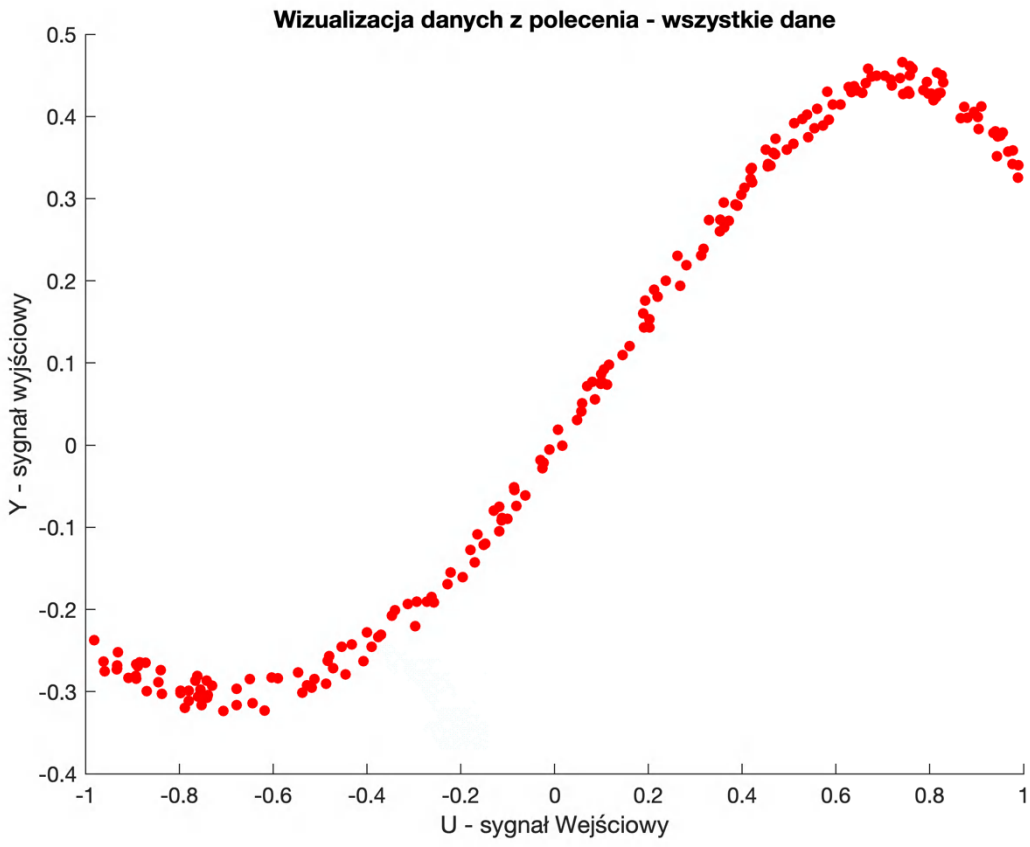
Sprawozdanie z projektu nr 2
Paweł Kutyła 318 384

Podział danych na zbiór uczący i weryfikujący.

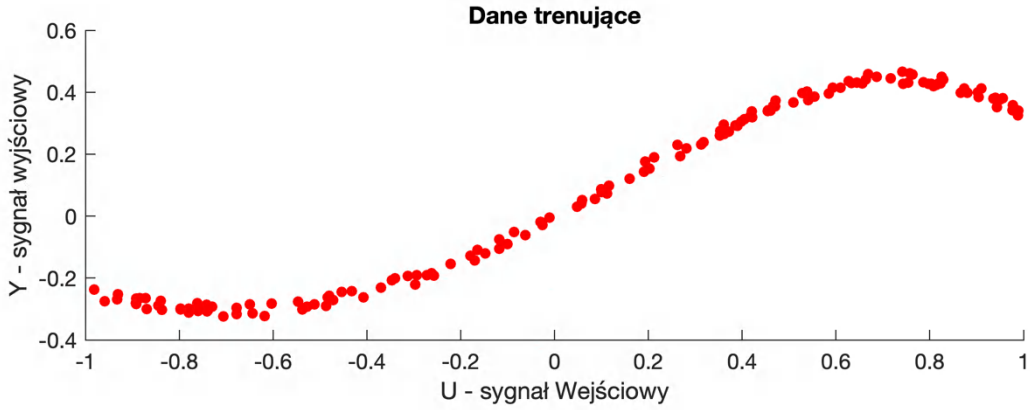
Realizację tego zadania przeprowadzę używając funkcji randperm generującej losową permutację liczb w zakresie od 0 do parametru, który owa funkcja przyjmuje. Dodatkowo zastosuję funkcję rng, która jest odpowiednikiem dobrze znanej funkcji random seed z nowoczesnych języków programowania. Owa funkcja pozwala na osiągnięcie pseudolosowych permutacji przy użyciu randperm'a, która w owym przypadku zwróci losowy wektor permutacji, jednak za każdym razem będzie on taki sam, co pozwala otrzymywać powtarzalne efekty i ułatwia analizę problemu. Zbiory zostały podzielone w następujący sposób:

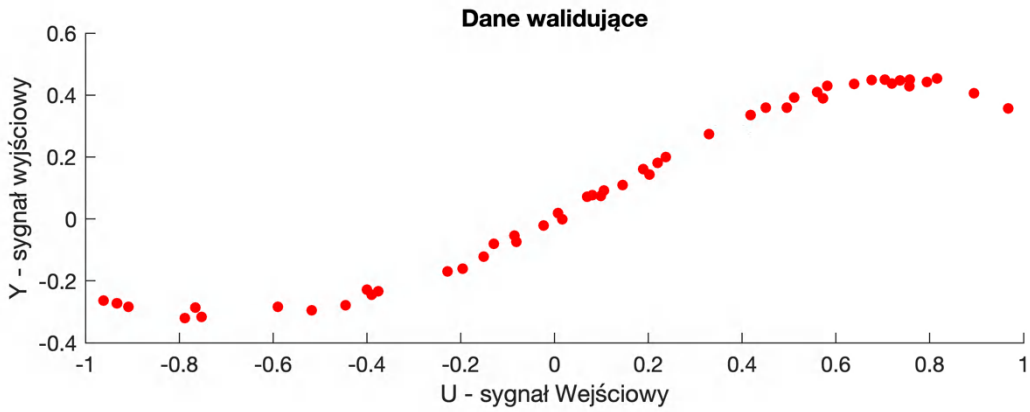
- Zbiór uczący – 75%
- Zbiór weryfikujący – 25%

Pozostałe czynności wykonywane przeze mnie są trywialne. Poniżej przedstawiam wykres zawierający wszystkie dane z pliku podanego w poleceniu, jeszcze przed podziałem na poszczególne zbiory.



Następnie wizualizuję dane po podziale według wcześniej wspomnianych reguł:





Liniowy model statyczny i jego analiza.

W tym zadaniu badam zasadność zastosowania liniowego modelu statycznego i jego potencjalne działanie. Model jest postaci:

$$y(u) = a_0 + a_1 u$$

Rozwiązywanie tego zadania zacznę od utworzenia macierzy Y, M, W, które posłużą mi do wyznaczenia optymalnych współczynników a_0 i a_1.

Wektor Y wygląda w następujący sposób:

$$Y = \begin{bmatrix} y(1) \\ \vdots \\ y(n) \end{bmatrix}$$

Macierz M jest postaci:

$$M = \begin{bmatrix} 1 & u(1) \\ \vdots & \vdots \\ 1 & u(n) \end{bmatrix}$$

A wektor W:

$$W = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}$$

Równanie opisujące zależności między danymi wypisanymi powyżej jest przedstawione na poniższych

$$Y = M \cdot W$$
$$W = M \setminus Y$$

W ten sposób wyznaczam parametry a_0 i a_1

$$a_0 = 0.043190164643173$$
$$a_1 = 0.465952963314842$$

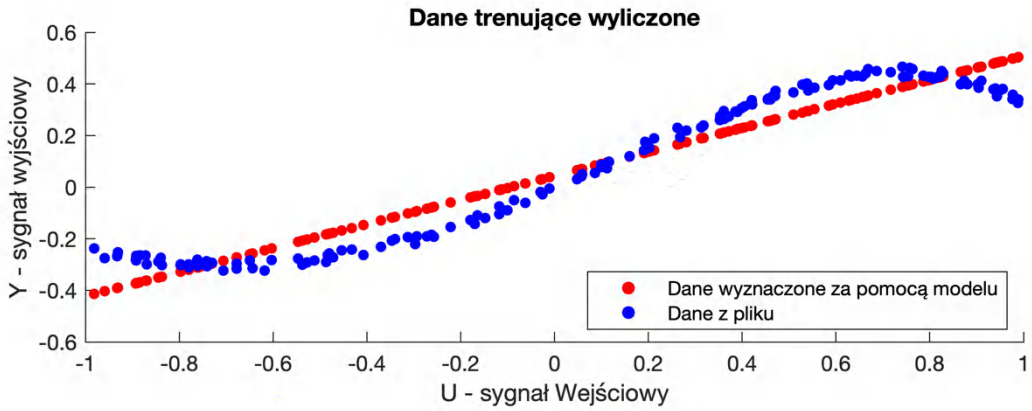
Następnie obliczam błędy dla tak uzyskanych danych trenujących i walidacyjnych ze wyliczając $Y_{trenujące} i\ Y_{walidacyjne}$ za pomocą wzoru

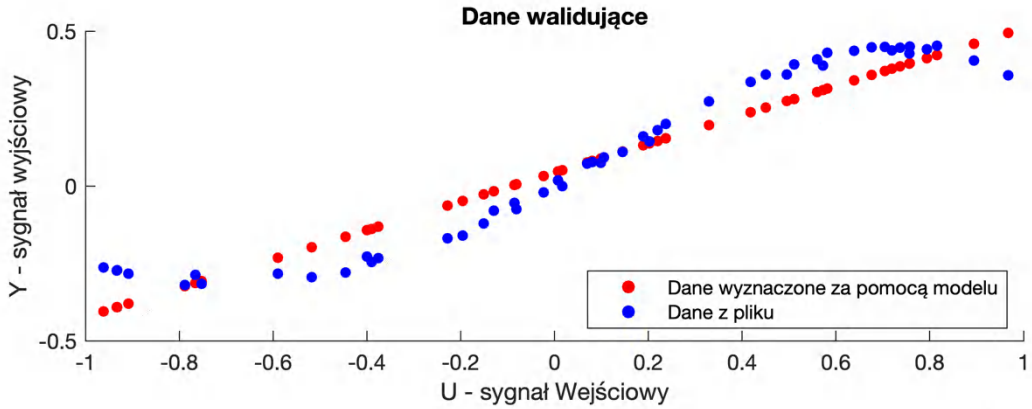
$$Y = M \cdot W$$

Oraz robię wizualizację danych otrzymywanych za pomocą modeli wraz z porównaniem dostarczonym z pliku.

$$E_{trenujące} = 0.929246757219909$$
$$E_{walidacyjne} = 0.305006364388444$$

Błędy liczone są jako suma błędów najmniejszych kwadratów z czego podział próbek jest równy 3:1. Proporcjonalnie wynika więc, że średni błąd na każdej z próbek jest mniejszy dla danych trenujących, co jest zgodne z założeniem, gdyż na tych danych dobieramy nasze parametry.





3. Wyznaczanie statycznych modeli nieliniowych postaci $y(u) = a_0 + \sum_{i=1}^{N} a_i u^i$ dla różnych stopni wielomianu$N$.

Dla stopnia $N = 1$ otrzymujemy model z poprzedniego zadania, który jest modelem niewłaściwym.
Dla stopnia $N = 2$ otrzymujemy model, który jest prezentowany równaniem :

$$y(u) = a_0 + a_1 u + a_2 u^2$$

Macierz Y :

$$Y = \begin{bmatrix} y(1) \\ \vdots \\ y(n) \end{bmatrix}$$

Macierz M:

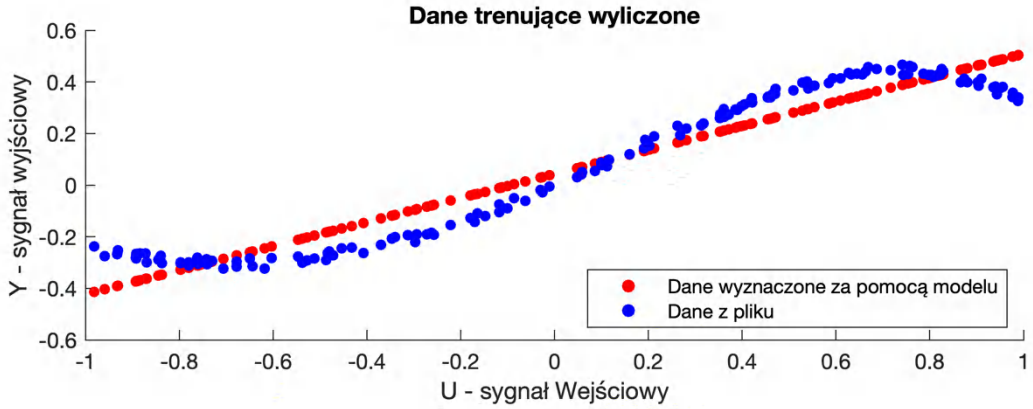
$$M = \begin{bmatrix} 1 & u(1) & u^2(1) \\ : & : & : \\ 1 & u(n) & u^2(n) \end{bmatrix}$$

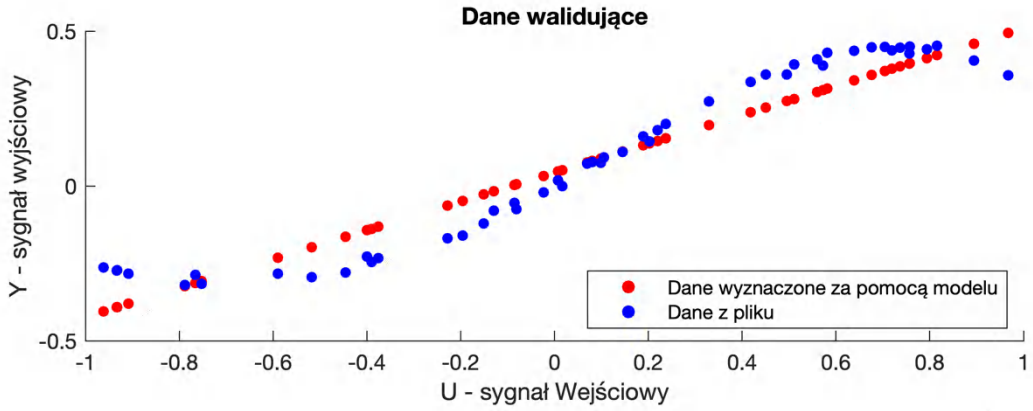
Macierz W:

$$W = \begin{bmatrix} a_o \\ a_1 \\ a_2 \end{bmatrix}$$

Wyliczona macierz W przez mój program:

$$W = \begin{bmatrix} 0.033907834833117 \\ 0.464768249864130 \\ 0.024472583705938 \end{bmatrix}$$





Wyznaczanie statycznych modeli nieliniowych postaci $y(u) = a_0 + \sum_{i=1}^{N} a_i u_i$ dla różnych stopni wielomianu.

Dla stopnia $N = 1$ otrzymujemy model z poprzedniego zadania, który jest modelem niewłaściwym.

Dla stopnia $N = 2$ otrzymujemy model, który jest reprezentowany równaniem :

$$y(u) = a_0 + a_i u + a_2 u^2$$

Macierz Y:

$$Y = \begin{bmatrix} y(1) \\ : \\ y(n) \end{bmatrix}$$

Macierz M:

$$M = \begin{bmatrix} 1 & u(1) & u^2(1) \\ : & : & : \\ 1 & u(n) & u^2(n) \end{bmatrix}$$

Macierz W:

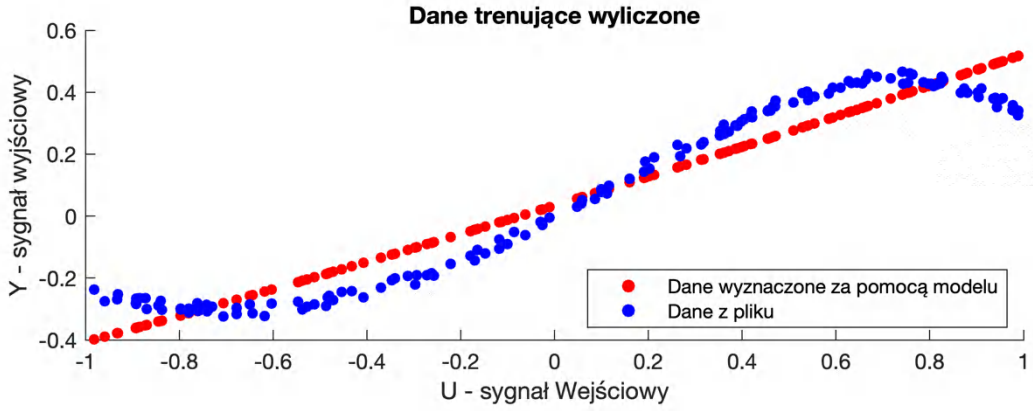
$$W = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$

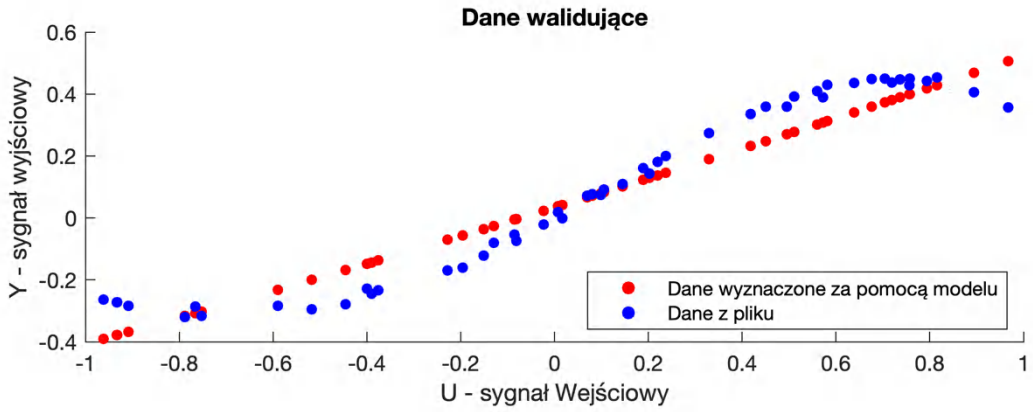
Wyliczona macierz W przez mój program:

$$W = \begin{bmatrix} 0.033907834833117 \\ 0.464768249864130 \\ 0.024472583705938 \end{bmatrix}$$

W ten sposób wyliczona macierz W pozwala nam na wyznaczenie błędu dla danych trenujących jak i walidacyjnych, a następnie przeprowadzenie ich wizualizacji na wykresach.

$$E_{trenujące} = 0.920952132663390$$
$$E_{walidacyjne} = 0.288660002729244$$





Zbudowany model nie jest wystarczający, a zmniejszenie błędu jest niewystarczające. Konieczne jest więc dalsze testowanie modelu i poszukiwanie innych stopni wielomianu zapewniających większą dokładność modelu. Analizuję więc wielomian opisany równaniem dla stopnia wielomianu $N = 3$:

$$y = a_0 + a_1 u + a_2 u^2 + a_3 u^3$$

Macierz Y :

$$Y = \begin{bmatrix} y(1) \\ : \\ y(n) \end{bmatrix}$$

Macierz M :

$$\mathrm{M} = \begin{bmatrix} 1 & u(1) & u^2(1) & u^3(1) \\ : & : & : & : \\ 1 & u(n) & u^2(n) & u^3(n) \end{bmatrix}$$

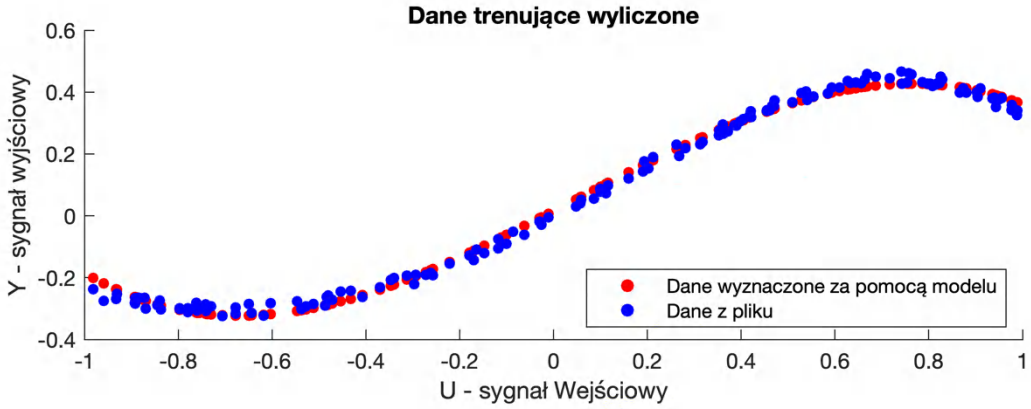
Macierz W:

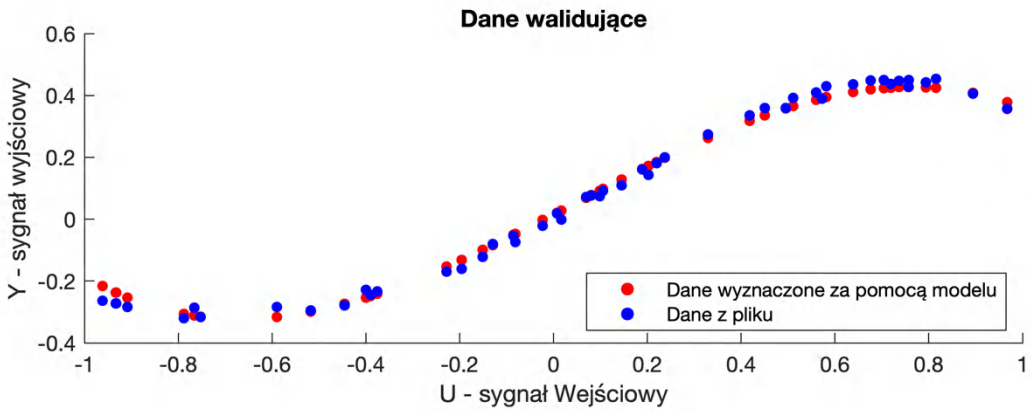
$$\mathrm{W} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Wyliczona macierz W przez mój program:

$$W = \begin{bmatrix} 0.015154720980673 \\ 0.781596070255800 \\ 0.072554476967790 \\ 0.509318605632862 \end{bmatrix}$$

Poniżej przedstawiam wizualizację wykresów wraz z wcześniej obliczonymi błędami:





$$E_{trenujące} = 0.053977161649822$$

$$E_{walidacyjne} = 0.021608078496034$$

Jak widzimy odnotowujemy ogromny spadek błędu sugerujący, że model jest opisany równaniem co najmniej trzeciego stopnia. Przechodzimy dalej, aby sprawdzić jak zachowuje się on dla wyższych potęg wielomianu np. N =4.

$$y(u) = a_0 + a_1 u + a_2 u^2 + a_3 u^3 + a_4 u^4$$

Macierz Y :

$$Y = \begin{bmatrix} y(1) \\ : \\ y(n) \end{bmatrix}$$

Macierz M:

$$\mathrm{M} = \begin{bmatrix} 1 & u(1) & u^2(1) & u^3(1) & u^4(1) \\ : & : & : & : & : \\ : & : & : & : & : \\ 1 & u(n) & u^2(n) & u^3(n) & u^4(n) \end{bmatrix}$$

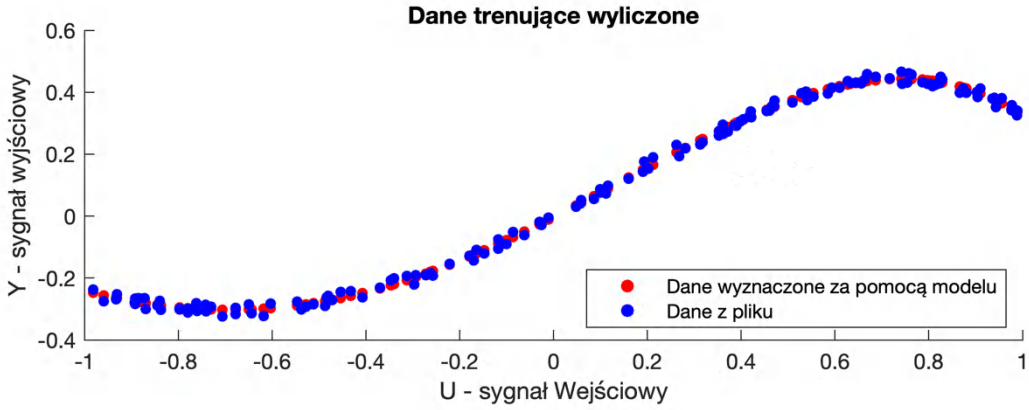
Macierz W:

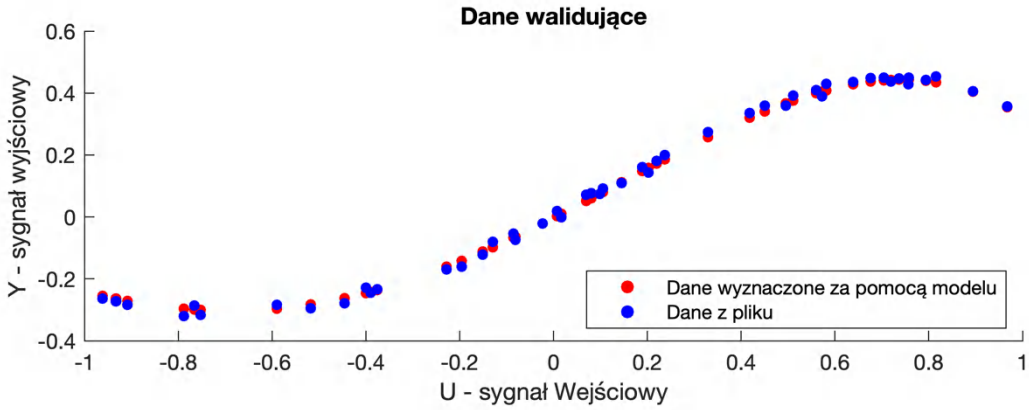
$$\mathrm{W} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}$$

Wyliczone współczynniki macierzy W wynoszą:

$$\mathrm{W} = \begin{bmatrix} -0.003151505460229 \\ 0.769853615177817 \\ 0.246036032775070 \\ -0.487134024748421 \\ -0.201280921334497 \end{bmatrix}$$

Wizualizację przedstawiam poniżej:





I konsekwentnie wyznaczam błędy:

$$E_{trenujące} = 0.021846077159569$$
$$E_{walidacyjne} = 0.008168133340443$$

W dalszym stopniu możemy odnotować poprawę działania modelu zarówno obserwując jego charakterystykę na wykresie zarówno jak i odnotowując wielkość błędu:
Kolejnym więc krokiem będzie przeanalizowanie modelu dla parametru $N = 5$

$$y = a_0 + a_1 u + a_2 u^2 + a_3 u^3 + a_4 u^4 + a_5 u^5$$

Macierz Y:

$$Y = \begin{bmatrix} y(1) \\ : \\ y(n) \end{bmatrix}$$

Macierz M:

$$\mathrm{M} = \begin{bmatrix} 1 & u(1) & u^2(1) & u^3(1) & u^4(1) & u^5(1) \\ : & : & : & : & : & : \\ 1 & u(n) & u^2(n) & u^3(n) & u^4(n) & u^5(n) \end{bmatrix}$$

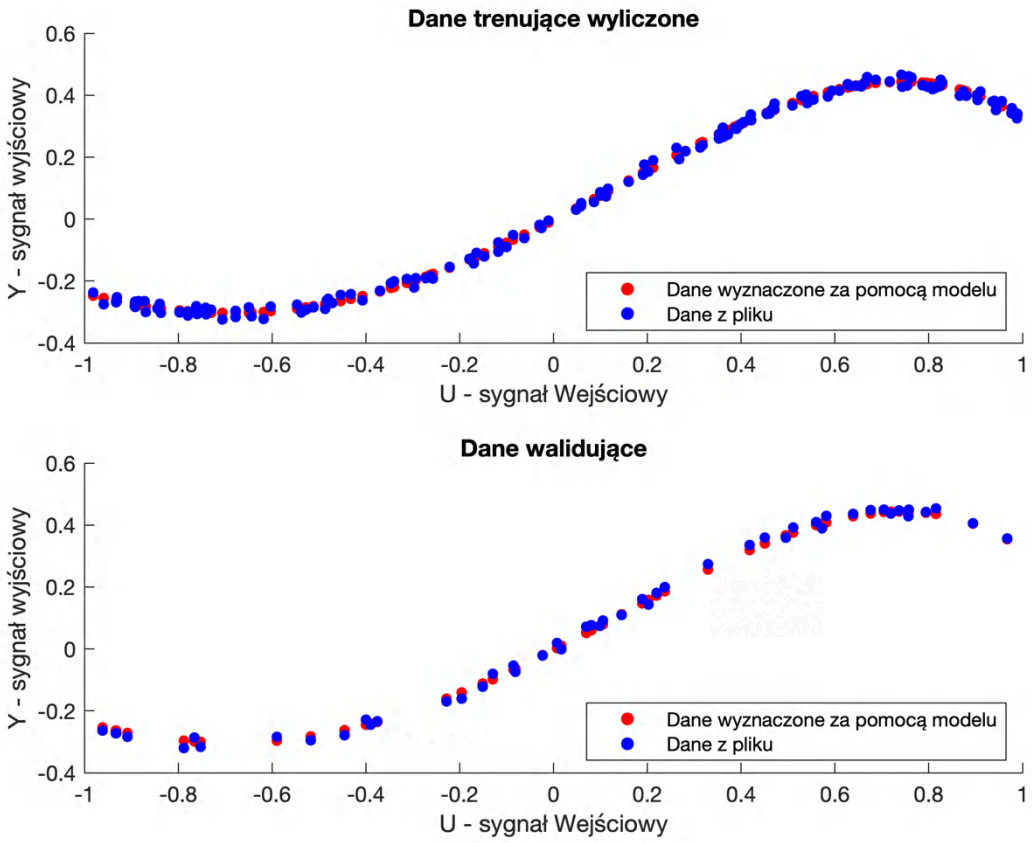
Macierz W:

$$W = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix}$$

Wyliczone współczynniki macierzy W wynoszą:

$$W = \begin{bmatrix} -0.003093338659975 \\ 0.767906828440693 \\ 0.245501543100280 \\ -0.478186503678169 \\ -0.200526392842303 \\ -0.008125993685350 \end{bmatrix}$$

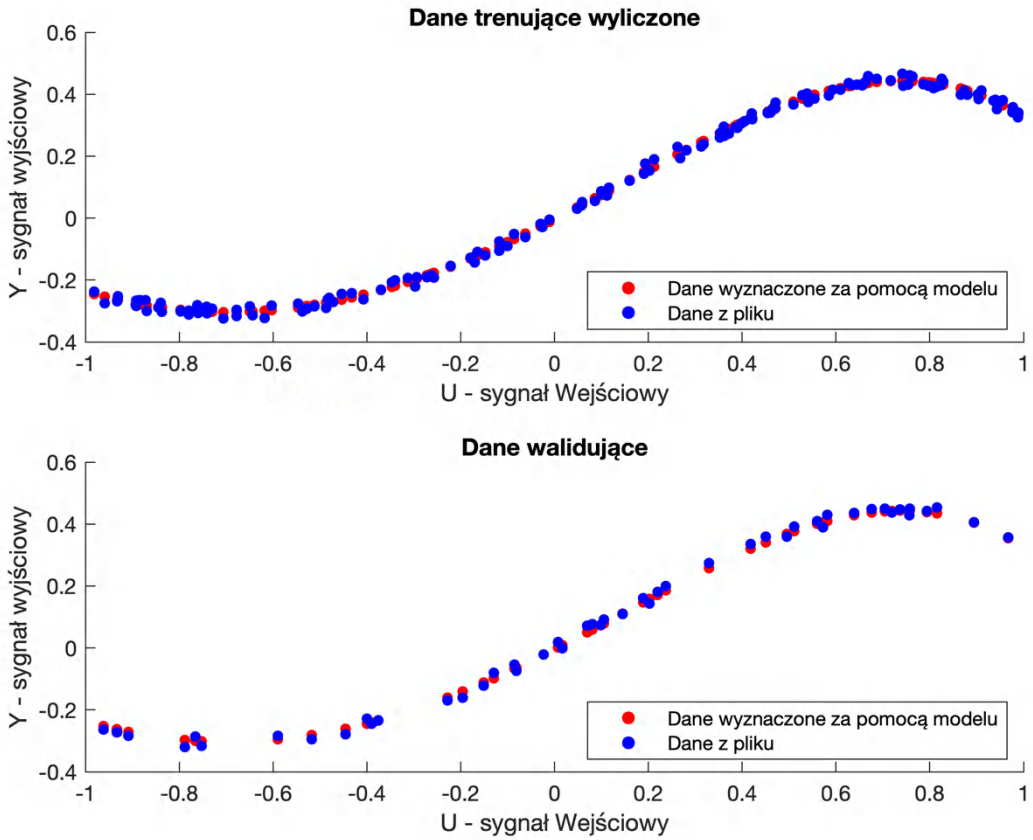
Wizualizacja na wykresach:



$$E_{trenujące} = 0.021831967360376$$
$$E_{walidacyjne} = 0.008232987318106$$

Jak możemy zauważyć doświadczamy delikatnego zwiększenia się wartości błędu dla walidacyjnego zestawu danych, co jest związane ze zjawiskiem overfittingu. Model jest zbyt dokładnie wyznaczony dla danych trenujących zatracając tym samym swoją uniwersalność. Dla pewności sprawdzimy model dla poziomu wielomianu , aby upewnić się, że wysnuty wcześniej wniosek jest zasadny. Pomijam tym razem wyprowadzenia macierzy z

wcześniejszych podpunktów, gdyż są trywialne, a zastosowanie w owym przypadku modelu wielomianu ma charakter testowy.



$$E_{trenujące} = 0.021770804948412$$
$$E_{walidacyjne} = 0.008322794956722$$

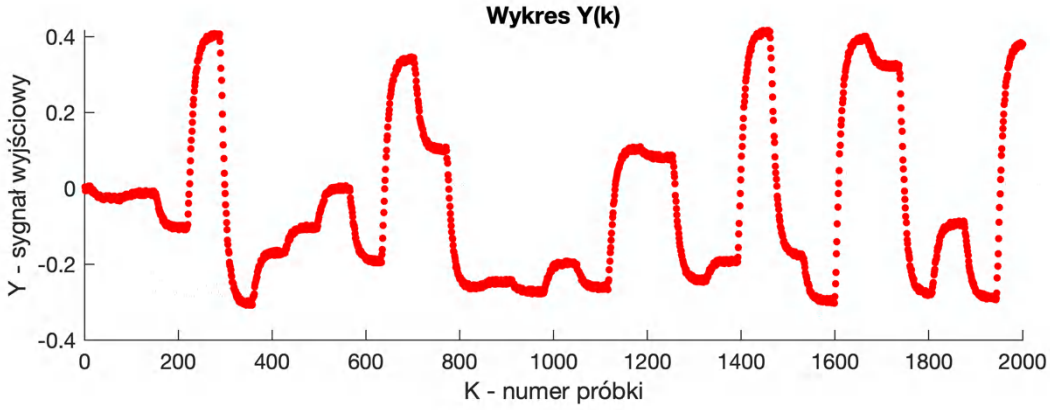
Jak możemy zauważyć trend zwiększania się błędu dla danych walidacyjnych się zwiększa, co utwierdza nas w przekonaniu o słuszności wniosku wysnutego na podstawie wcześniejszych obserwacji i analizowaniu zachowania błędu. Gdybyśmy dalej przeprowadzali operacje zwiększania stopnia wielomianu zaobserwowalibyśmy poprawę działania modelu dla danych trenujących, jednak odnotowalibyśmy znaczny wzrost wartości błędu dla danych walidacyjnych, co sprawiłoby że nasz model stałby się bezużyteczny w rzeczywistym zastosowaniu.

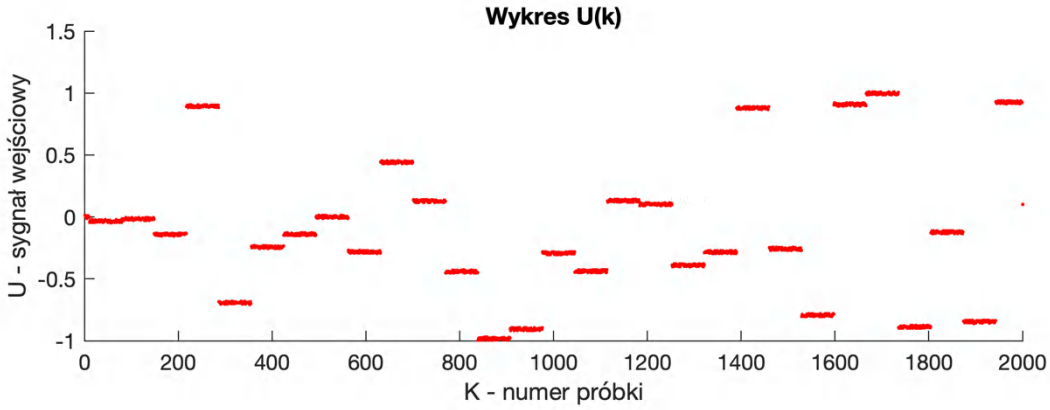
Najlepszym modelem będzie model stworzony dla $N = 5$, ponieważ to właśnie dla niego odnotowujemy minimum błędy. Dla kolejnych wartości N błąd będzie rósł i wystąpi zjawisko overfittingu, dlatego też najlepszym modelem będzie ten stworzony dla $N = 5$.

2.

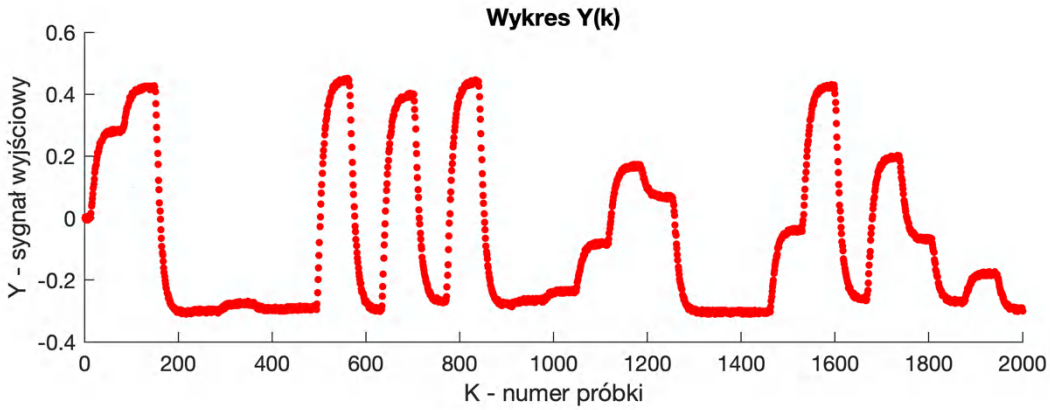
Wizualizacja danych uczących i walidacyjnych

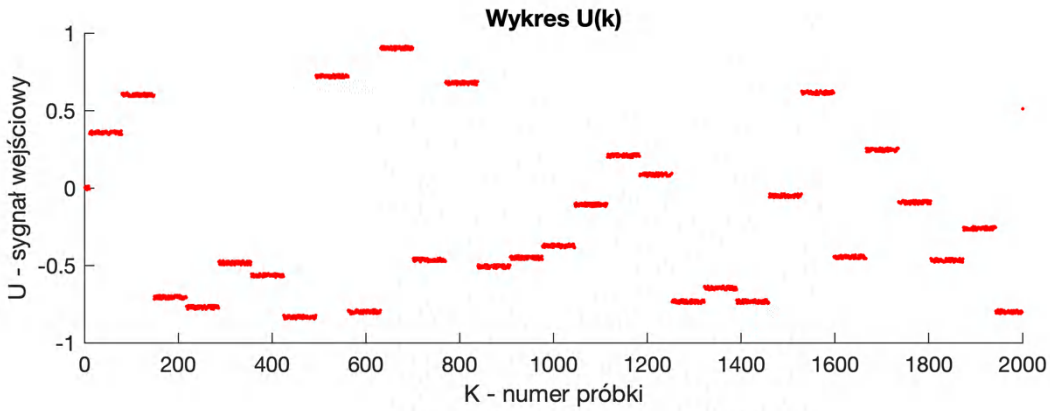
Dane uczące:





Dane walidacyjne:





Dynamiczne modele liniowe postaci $\boldsymbol{y(k) = \sum_{i=1}^{n_B} b_i u(k-i) + \sum_{i=1}^{n_A} a_i y(k-i)}$ pierwszego, drugiego i trzeciego rzędu.

Na początku rozważę model pierwszego rzędu opisanego równaniem:

$$W = b_1 u(k-1) + a_1 y(k-1)$$

Dla zgodności rozpoczynania próbkowania dla każdego z modelów określę parametr od którego startujemy . W kodzie wartością tej zmiennej będzie 10, ale zmiana tego jest trywialna i nie powoduje drastycznych zmian zarówno w obliczanym błędzie jak i wyglądzie charakterystyki.

Wektor Y wygląda w następujący sposób:

$$Y = \begin{bmatrix} y(st) \\ \vdots \\ y(n) \end{bmatrix}$$

Macierz M jest postaci:

$$M = \begin{bmatrix} u(st-1) & y(st-2) \\ \vdots & \vdots \\ u(n-1) & y(n-1) \end{bmatrix}$$

A wektor W:

$$W = \begin{bmatrix} b_1 \\ a_1 \end{bmatrix}$$

Po podstawieniu i zaimplementowaniu powyższych równań w MATLABIE'e mogę swobodnie przejść do kluczowej operacji wyznaczenie poszczególnych parametrów macierzy W . Następnie dokonując operacji lewego dzielenia wyznaczam parametry wektora W.

$$W = \begin{bmatrix} 0.020148671414205 \\ 0.956209213022186 \end{bmatrix}$$

Przy użyciu parametrów macierzy W wyznaczam Y_{mod} reprezentującą wartości Y obliczane rekurencyjnie. Na początku muszę zainicjalizować $Y_{mod}(1)$:

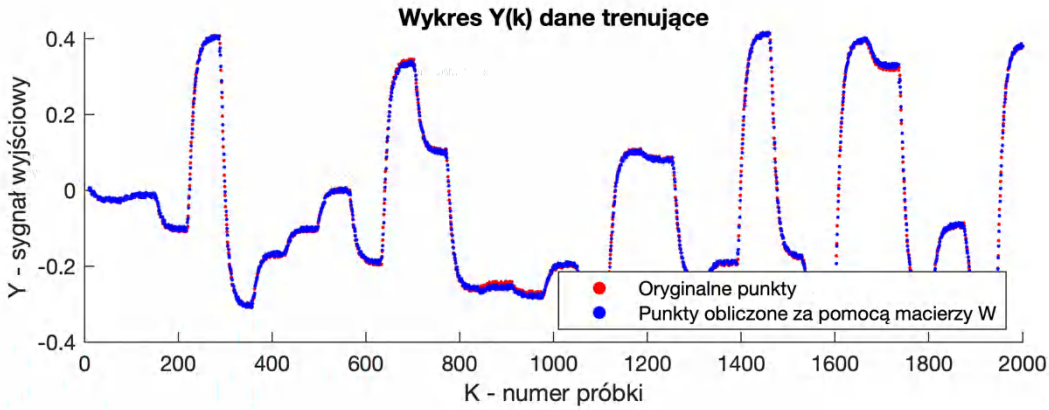
$$Y_{mod}(1) = b_1 u(st-1) + a_1 y(k-1)$$

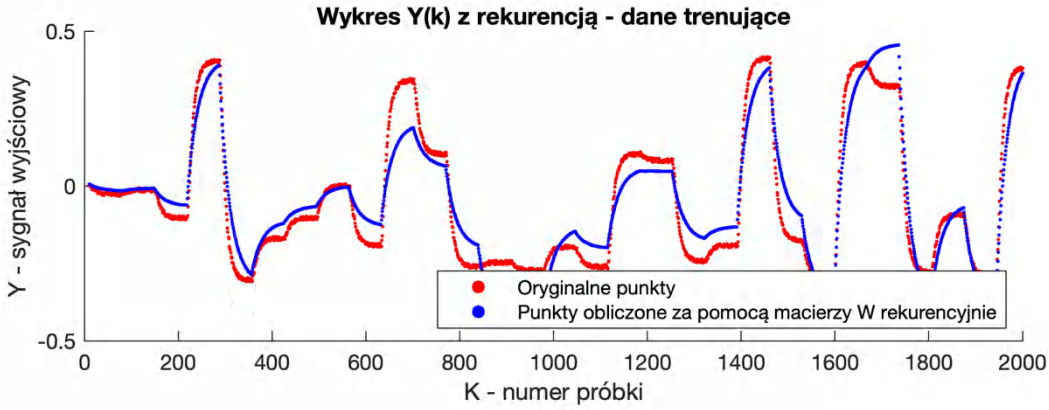
Gdzie st opisuje od której próbki zaczynamy wyznaczanie Y_{mod} , aby zachować jednakową ilość danych wyjściowych niezależnie od rzędu modelu, dzięki czemu pozbędziemy się potencjalnych zawyżonych wartości błędu z powodu większej ilości próbek dla modeli niższych rzędów. Dla kolejnych wartości Y_{mod} równanie ma postać:

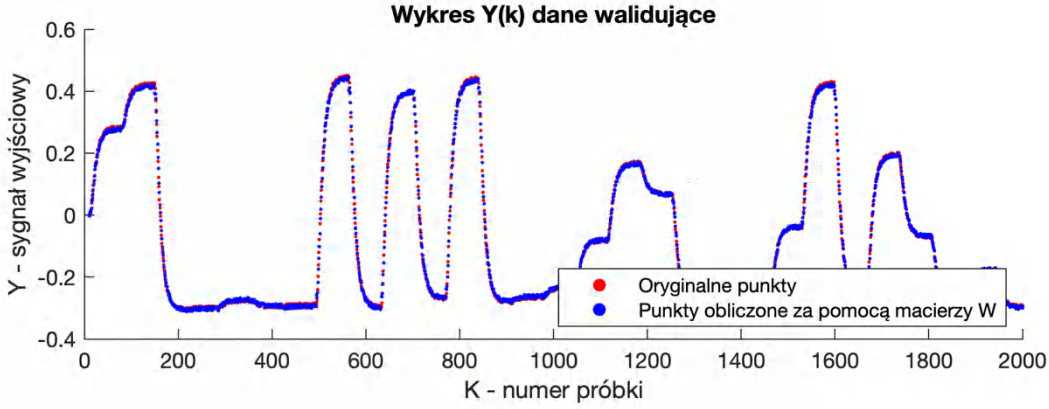
$$Y_{mod}(i) = b_1 u(st+i-2) + Y_{mod}(i-1)$$

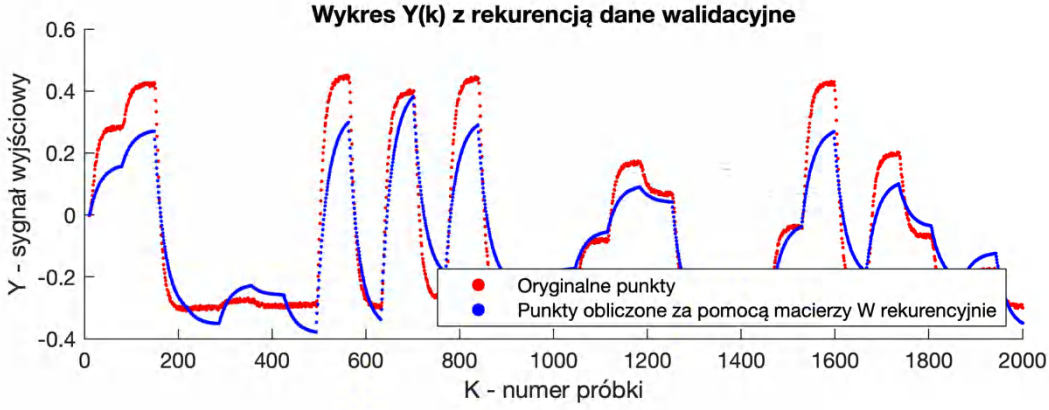
Dla $i = 2, \dots, n$

Wykresy przedstawiające dane dla danych walidacyjnych i trenujących dla modeli z rekurencją przedstawiam na poniższych grafikach:


Wykres Y(k) dane trenujące
Y - sygnał wyjściowy
K - numer próbki
Oryginalne punkty
Punkty obliczone za pomocą macierzy W


Wykres Y(k) z rekurencją - dane trenujące
Y - sygnał wyjściowy
K - numer próbki
Oryginalne punkty
Punkty obliczone za pomocą macierzy W rekurencyjnie


Wykres Y(k) dane walidujące
Y - sygnał wyjściowy
K - numer próbki
Oryginalne punkty
Punkty obliczone za pomocą macierzy W


Wykres Y(k) z rekurencją dane walidacyjne
Y - sygnał wyjściowy
K - numer próbki
Oryginalne punkty
Punkty obliczone za pomocą macierzy W rekurencyjnie

Wartości błędów zostaną przedstawione po wizualizacji danych dla modeli różnych rzędów w tabeli, gdzie zostaną porównane.

Model drugiego rzędu:

$$y(k) = b_1u(k-1) + b_2u(k-2) + a_1y(k-1) + a_2y(k-2)$$

Macierz M:

$$\mathrm{M} = \begin{bmatrix} u(st-1) & u(st-2) & y(st-1) & y(st-2) \\ \vdots & \vdots & \vdots & \vdots \\ u(st-1) & u(st-2) & y(st-1) & y(st-2) \end{bmatrix}$$

Macierz W:

$$\mathrm{W} = \begin{bmatrix} b_1 \\ b_2 \\ a_1 \\ a_2 \end{bmatrix}$$

Macierz Y:

$$Y = \begin{bmatrix} y(st) \\ \vdots \\ y(n) \end{bmatrix}$$

Po wykonaniu operacji matematycznych otrzymuje wektor W:

$$W = \begin{bmatrix} -0.000206266708357 \\ 0.017443436414471 \\ 1.208787430880915 \\ -0.247387555952806 \end{bmatrix}$$

Przy użyciu parametrów macierzy W wyznaczam Y_{mod} reprezentującą wartości obliczane rekurencyjnie. Na początku muszę zainicjalizować $Y_{mod}(1)$:

$$Y_{mod}(1) = b_1u(st-1) + b_2u(st-2) + a_1y(st-1) + a_2y(st-2)$$

Następnie inicjalizuję $Y_{mod}(2)$:

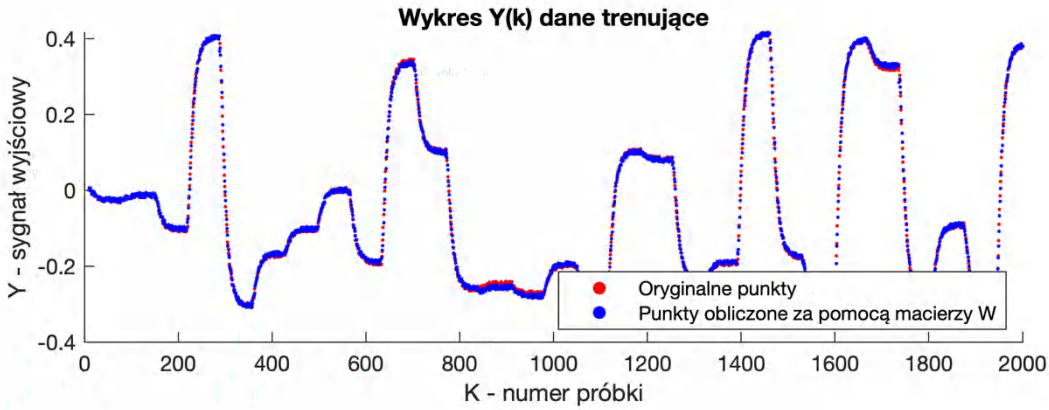
$$Y_{mod}(2) = b_1u(st) + b_2u(st-1) + a_1Y_{mod}(1) + a_2y(st-1)$$

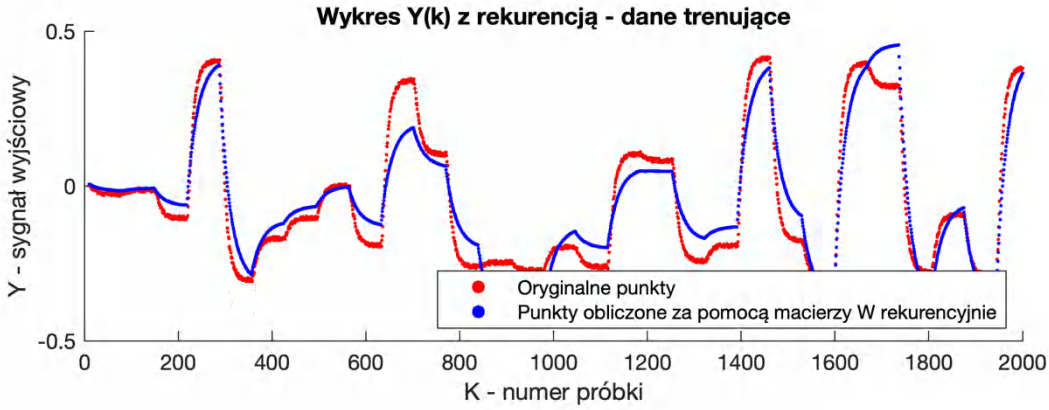
Dla kolejnych wartości równanie ma postać:

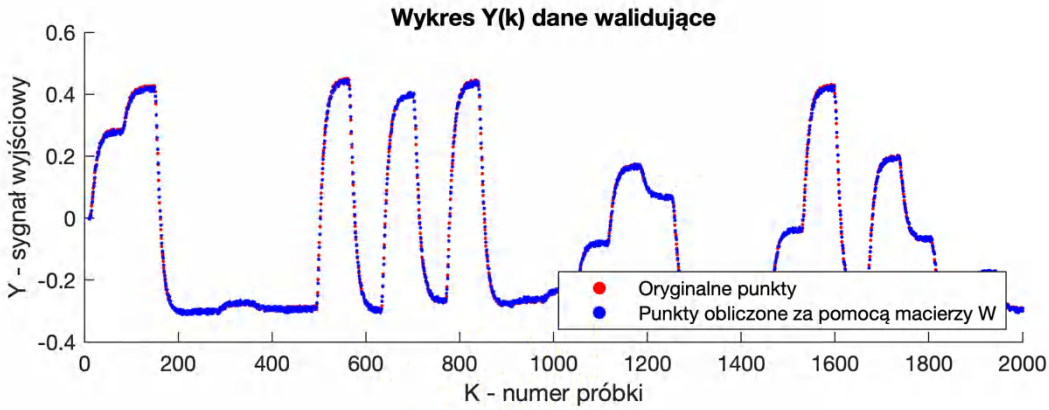
$$Y_{mod}(i) = b_1u(st+i-2) + \ +b_2u(st+i-3) + a_1Y_{mod}(i-1) + a_2Y_{mod}(i-2)$$

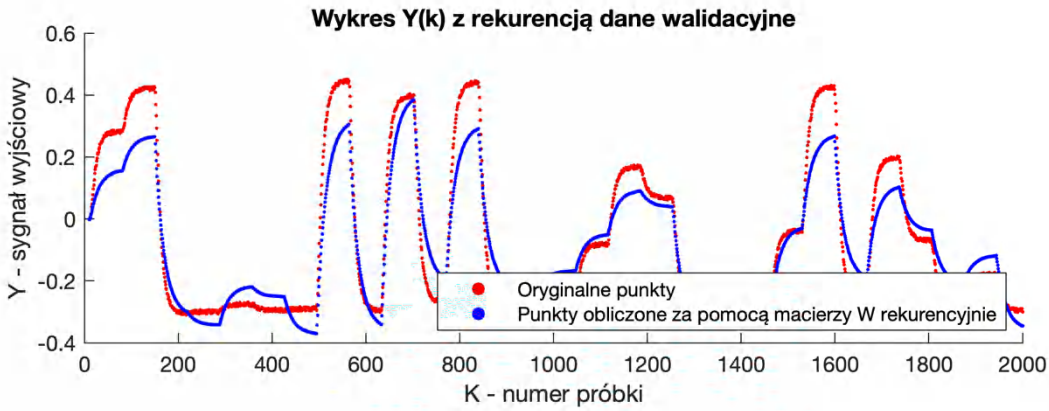
Dla $i = 3,..n$

Wykresy przedstawiające dane dla danych walidacyjnych i trenujących dla modeli z rekurencją przedstawiam na poniższych grafikach:


Wykres Y(k) dane trenujące
Y - sygnał wyjściowy
K - numer próbki
Oryginalne punkty
Punkty obliczone za pomocą macierzy W


Wykres Y(k) z rekurencją - dane trenujące
Y - sygnał wyjściowy
K - numer próbki
Oryginalne punkty
Punkty obliczone za pomocą macierzy W rekurencyjnie


Wykres Y(k) dane walidujące
Y - sygnał wyjściowy
K - numer próbki
Oryginalne punkty
Punkty obliczone za pomocą macierzy W


Wykres Y(k) z rekurencją dane walidacyjne
Y - sygnał wyjściowy
K - numer próbki
Oryginalne punkty
Punkty obliczone za pomocą macierzy W rekurencyjnie

Model trzeciego rzędu:

$$y(k) = b_1u(k-1) + b_2u(k-2) + b_3u(k-3) + a_1y(k-1) + a_2(y-2) + a_3y(k-3)$$

Macierz Y :

$$Y = \begin{bmatrix} y(st) \\ \vdots \\ y(n) \end{bmatrix}$$

Macierz M:

$$\mathrm{M} = \begin{bmatrix} u(st-1) & u(st-2) & u(st-3) & y(st-1) & y(st-2) & y(st-3) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u(n-1) & u(n-2) & u(n-3) & y(n-1) & y(n-2) & y(n-3) \end{bmatrix}$$

Macierz W:

$$\mathrm{W} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Wyliczone współczynniki macierzy W wynoszą:

$$W = \begin{bmatrix} -0.000816487698360 \\ 0.012137531530023 \\ 0.001916941993335 \\ 1.074287262214345 \\ 0.276215263063382 \\ -0.381258639265456 \end{bmatrix}$$

Przy użyciu parametrów macierzy W wyznaczam Y_{mod} reprezentującą wartości obliczane rekurencyjnie. Na początku muszę zainicjalizować :

$$Y_{mod}(1) = b_1u(st-1) + b_2u(st-2) + b_3u(st-3) + a_1y(st-1) + a_2y(st-2) + a_3y(st-3)$$

Następnie inicjalizuję $Y_{mod}(2)$:

$$Y_{mod}(2) = b_1u(st) + b_2u(st-1) + b_3u(st-2) + a_1Y_{mod}(1) + a_2y(st-1) + a_3y(st-2)$$

Następnie $Y_{mod}(3)$:

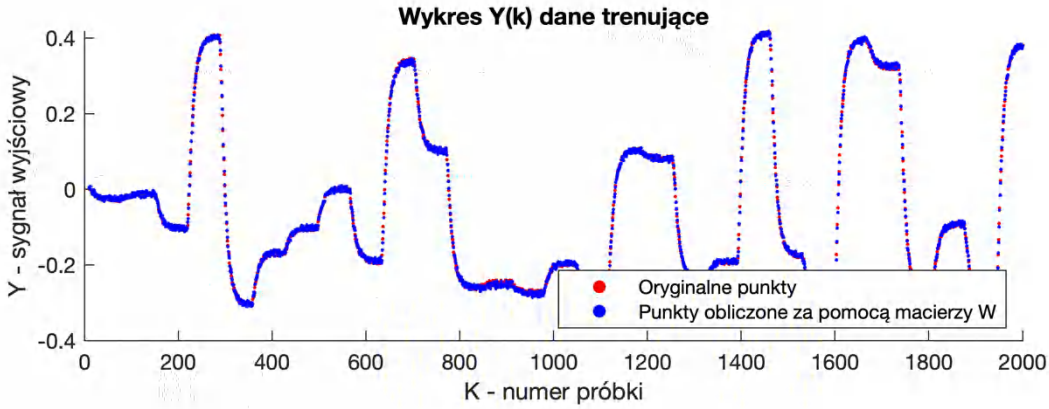
$$Y_{mod}(2) = b_1u(st+1) + b_2u(st) + b_3u(st-1) + a_1Y_{mod}(2) + a_2Y_{mod}(1) + a_3y(st-1)$$

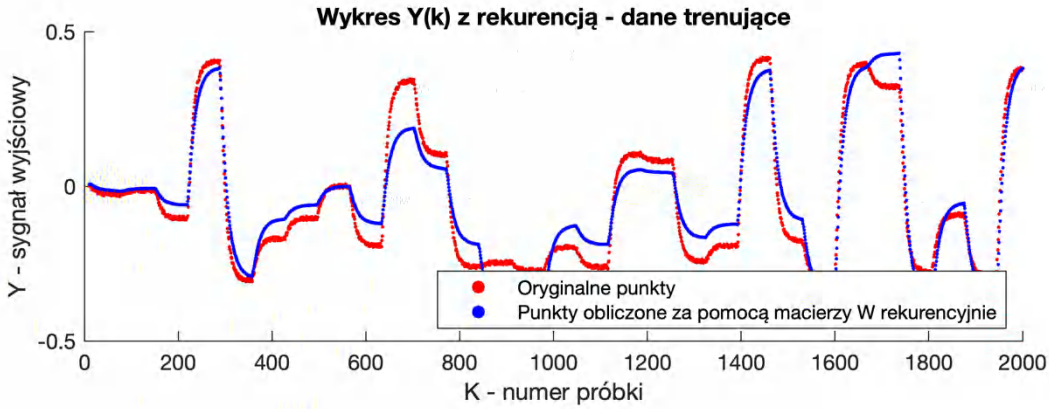
Dla kolejnych Y_{mod} równanie ma postać:

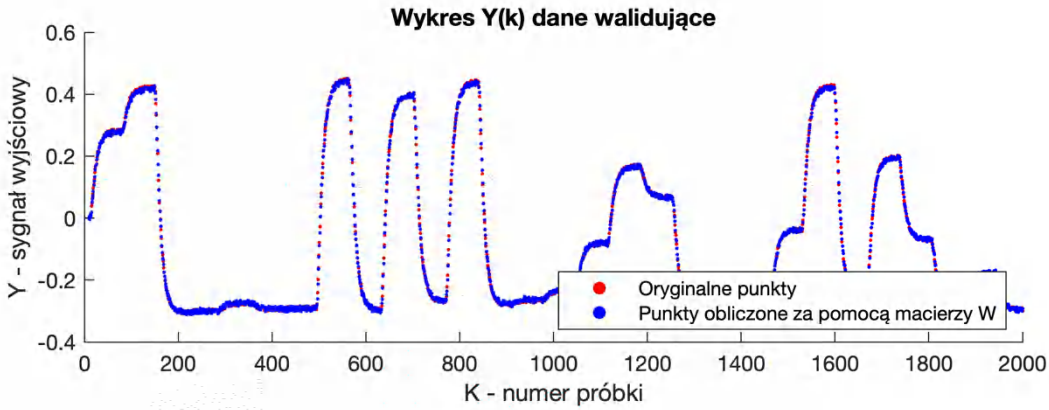
$$Y_{mod}(i) = b_1u(st+i-2) + b_2u(st+i-3) + b_3u(st+i-4) + a_1Y_{mod}(i-1) + a_2Y_{mod}(i-2) + a_3Y_{mod}(i-3)$$

Dla $i = 4, \dots, n$

Poniżej przedstawiam wizualizacje danych dla modeli rzędu 3:







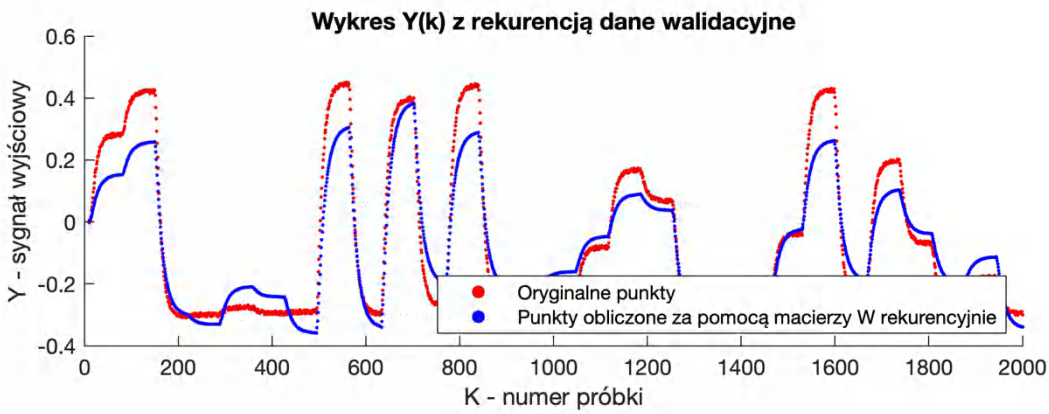


Tabela z błędami:

Rząd	Błąd dla danych trenujących		Błąd dla danych walidacyjnych	
	Bez rekurencji	Z rekurencją	Bez rekurencji	Z rekurencją
1	0.113714619556984	12.316005766341020	0.137234424936184	16.407179434436394
2	0.097477309262838	10.836410046524700	0.110591248308987	14.649360563818410
3	0.080012200309852	9.486346449070700	0.086051352314397	13.145263468262010

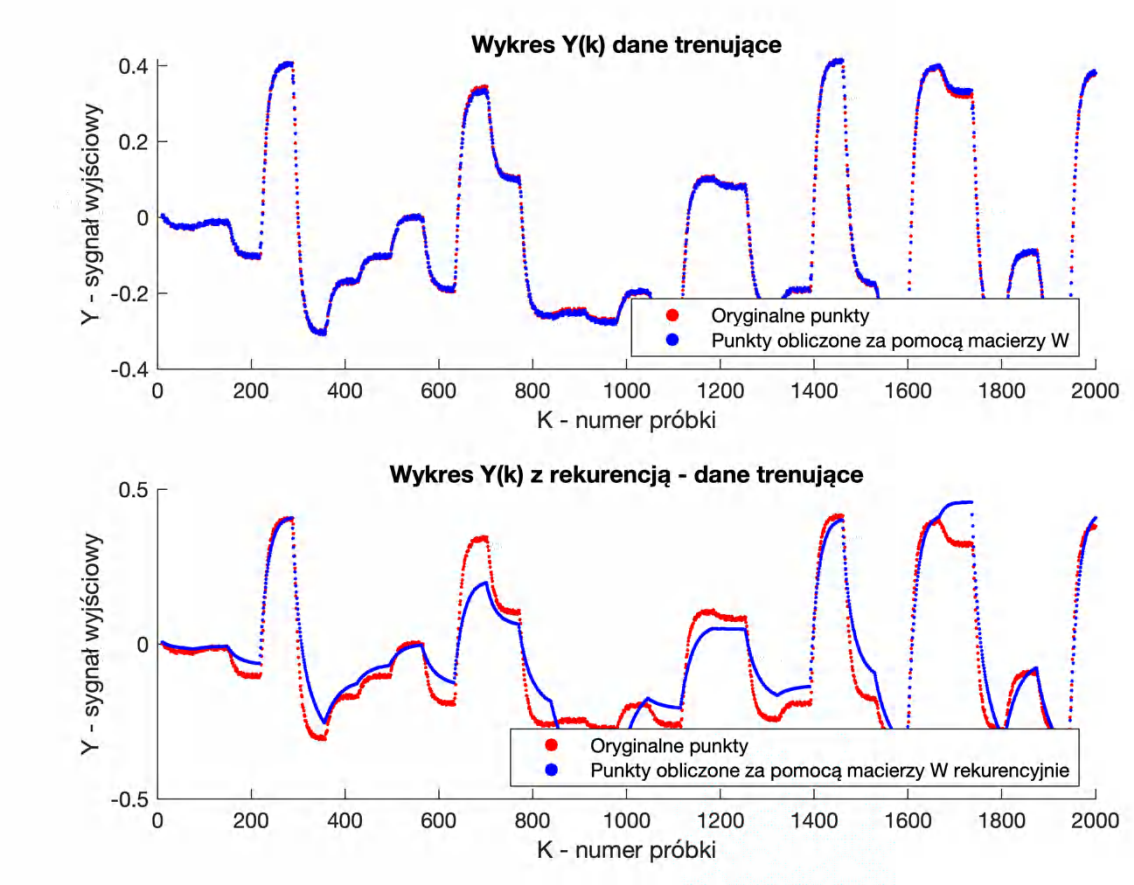
Poddając analizie powyższą tabelę możemy z łatwością stwierdzić, że zarówno dla modelu z rekurencją jak i modelu bez niej optymalnym rozwiązaniem będzie przyjęcie modelu trzeciego rzędu. Dzięki niemu odnotowujemy najmniejszy błąd zarówno dla rekurencji i jej braku co pozwala na jednoznaczne stwierdzenie o największej jakości modelu w stosunku do porównywanych w powyższej tabeli. Pomimo tego, że wybór modelu jest jednoznaczny warto zaznaczyć, że przez charakter sekwencyjny modelu ważniejszą wartością błędu jest błąd obliczany dla modelu z rekurencją, ponieważ on lepiej oddaje charakterystykę modelu.

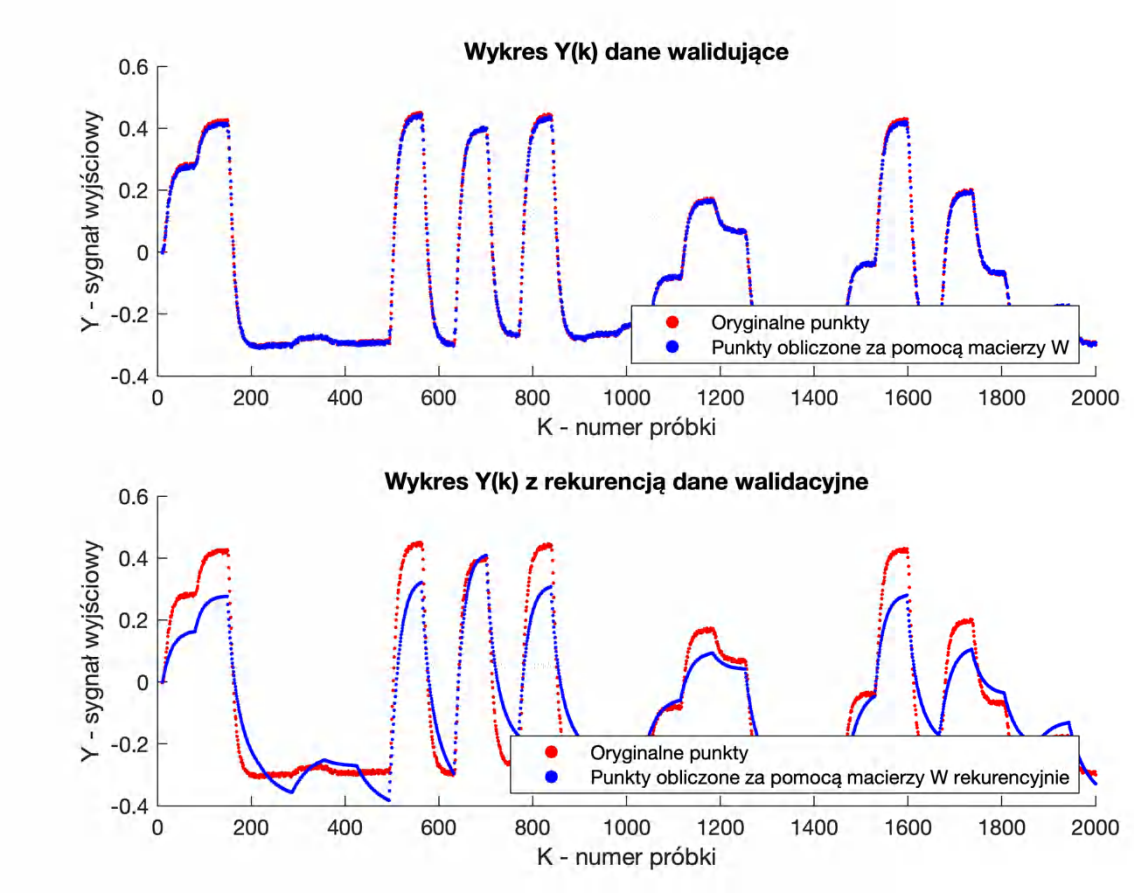
Rozważanie różnych modeli o różnym rzędzie i strukturze nieliniowości.

Pierwszym z modeli, którego będziemy analizować będzie model o rzędzie 1 i stopniu 2. Równanie opisujące ten model ma postać:

$$y(k) = w_1 u(k-1) + w_2 u^2(k-1) + w_3 y(k-1) + w_4 y^2(k-1)$$

Poniższe wykresy posłużą do wizualizacji charakterystyki podanego wyżej modelu.

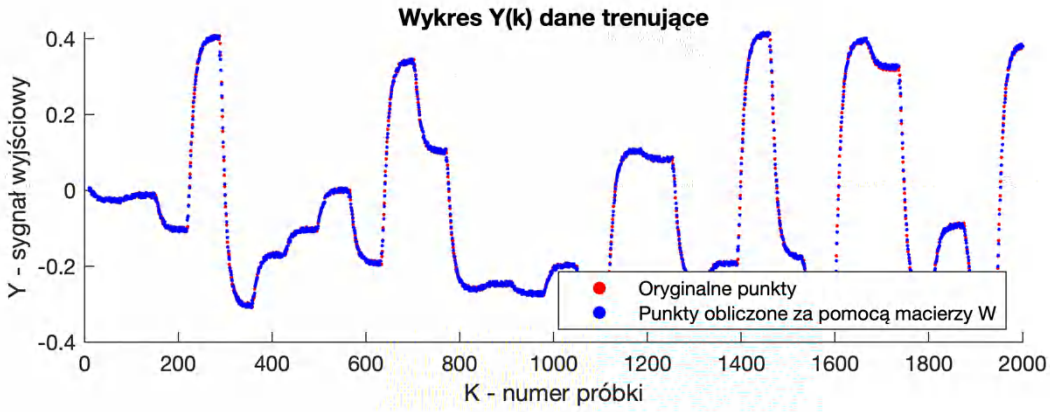


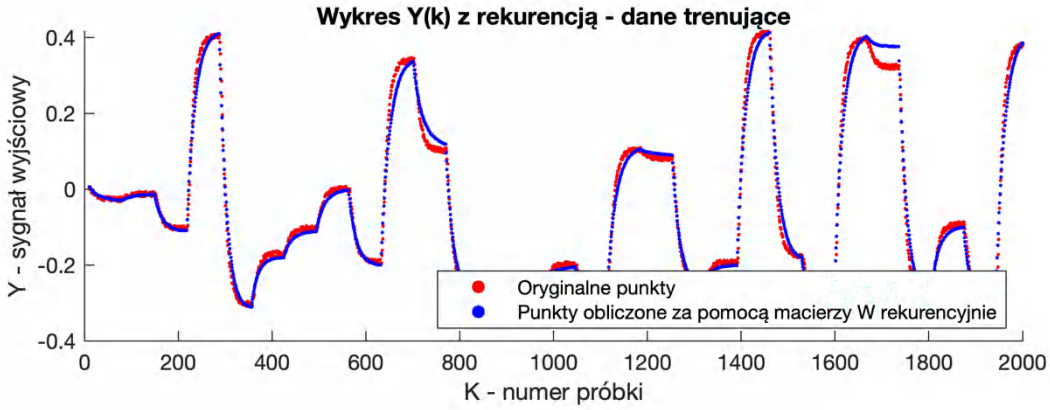


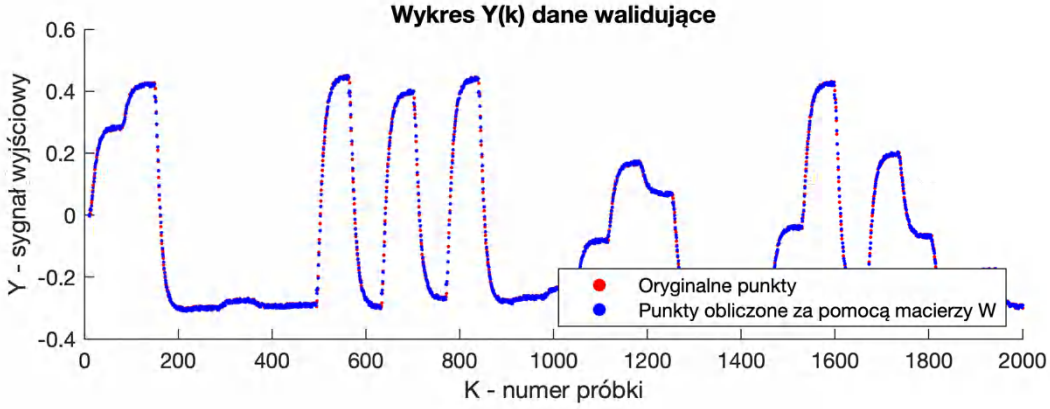
$$E_{trenujące} = 0.107696465888376$$
$$E_{trenującerec} = 11.333678889090553$$
$$E_{walidacyjne} = 0.144498073206522$$
$$E_{walidacyjnerec} = 14.911170738482150$$

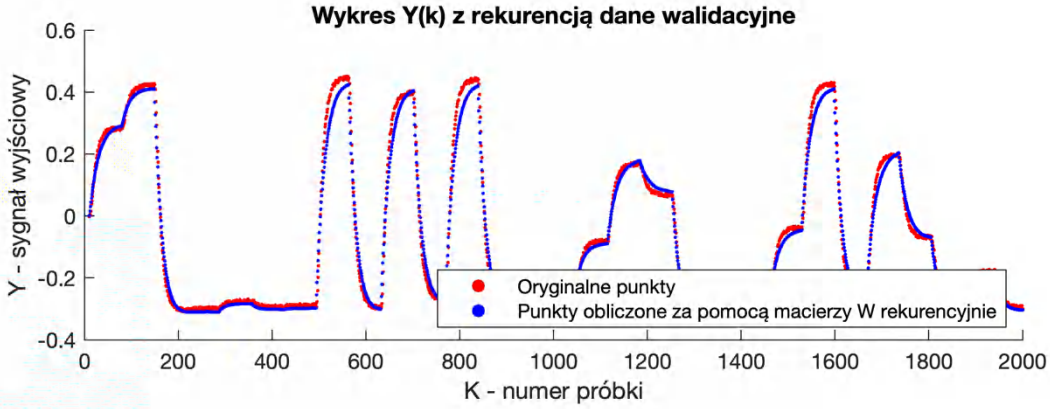
Jak możemy zauważyć odnotowujemy zmniejszenie się błędów w stosunku do modelu 1 rzędu i 1 stopnia, więc przystępujemy do badania modelu 1 rzędu i 3 stopnia:

$$y(k) = w_1u(k-1) + w_2u^2(k-1) + w_3u^3(k-1) + w_4y(k-1) + w_5y^2(k-1) + w_6y^3(k-1)$$


Wykres Y(k) dane trenujące
Y - sygnał wyjściowy
K - numer próbki
Oryginalne punkty
Punkty obliczone za pomocą macierzy W


Wykres Y(k) z rekurencją - dane trenujące
Y - sygnał wyjściowy
K - numer próbki
Oryginalne punkty
Punkty obliczone za pomocą macierzy W rekurencyjnie

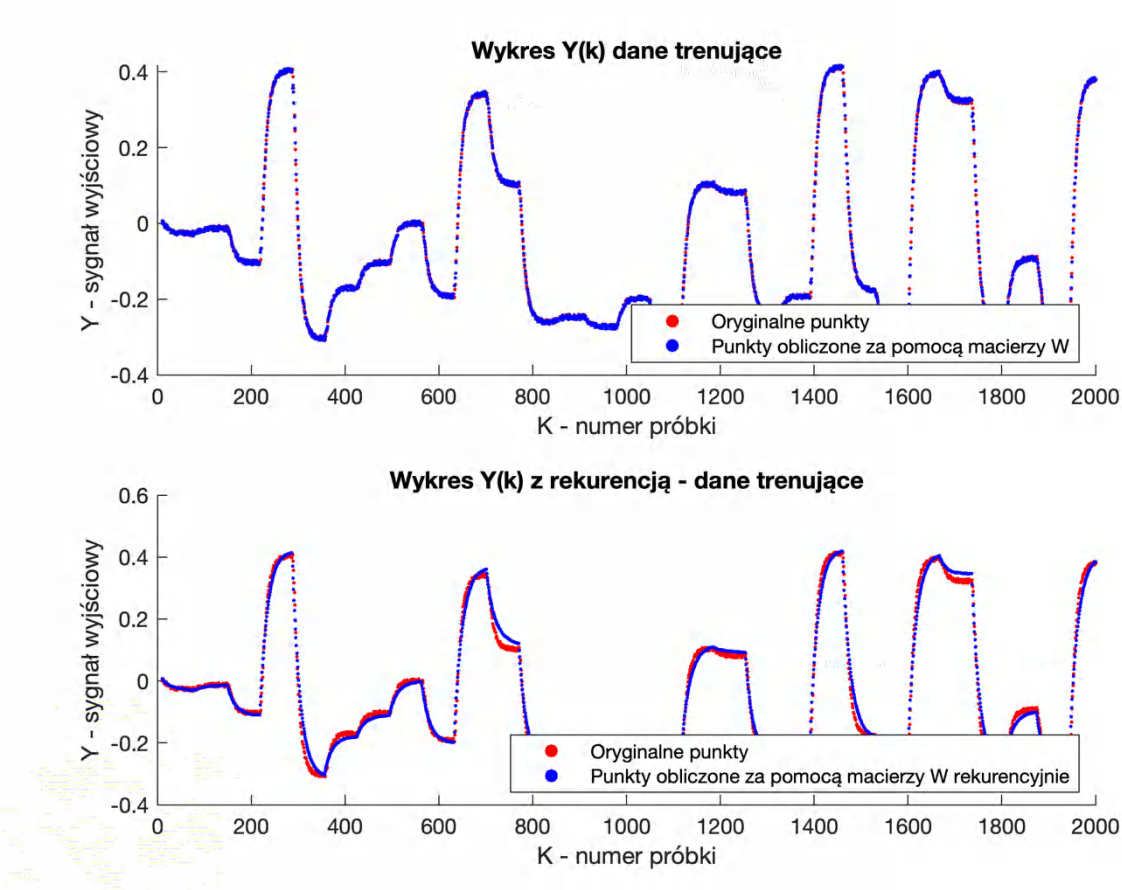

Wykres Y(k) dane walidujące
Y - sygnał wyjściowy
K - numer próbki
Oryginalne punkty
Punkty obliczone za pomocą macierzy W

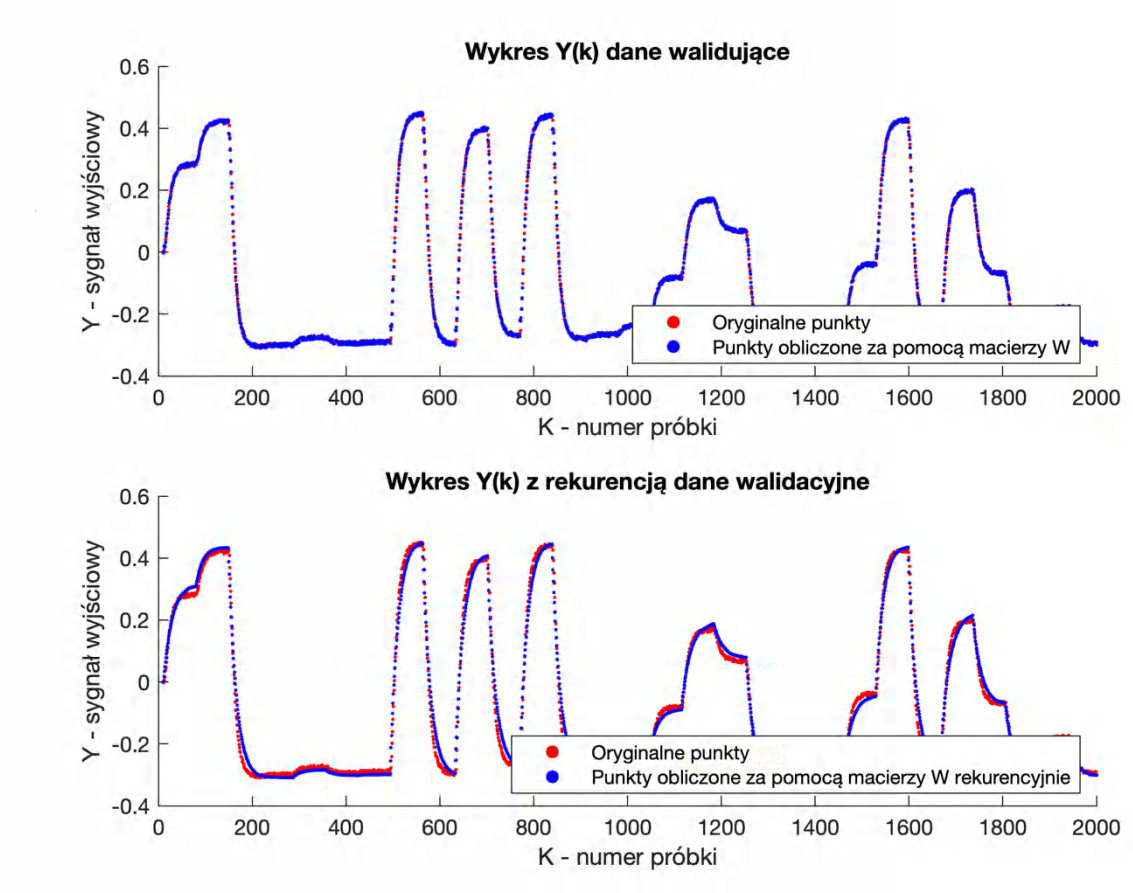

Wykres Y(k) z rekurencją dane walidacyjne
Y - sygnał wyjściowy
K - numer próbki
Oryginalne punkty
Punkty obliczone za pomocą macierzy W rekurencyjnie

$$E_{trenujące} = 0.082649380205501$$
$$E_{trenującerec} = 1.142916660280829$$
$$E_{walidacyjne} = 0.093809653922147$$
$$E_{walidacyjnerec} = 1.412204426550415$$

Odnotowujemy bardzo dużą poprawę dokładności modelu co możemy zauważyć poprzez drastyczne zmniejszenie błędu. W kolejnym kroku sprawdzam model 1 rzędu i 4 stopnia.

$$y(k) = w_1u(k-1) + w_2u^2(k-1) + w_3u^3(k-1) + w_4u^4(k-1) + w_5y(k-1) + w_6y^2(k-1) + w_7y^3(k-1) + w_8y^4(k-1)$$

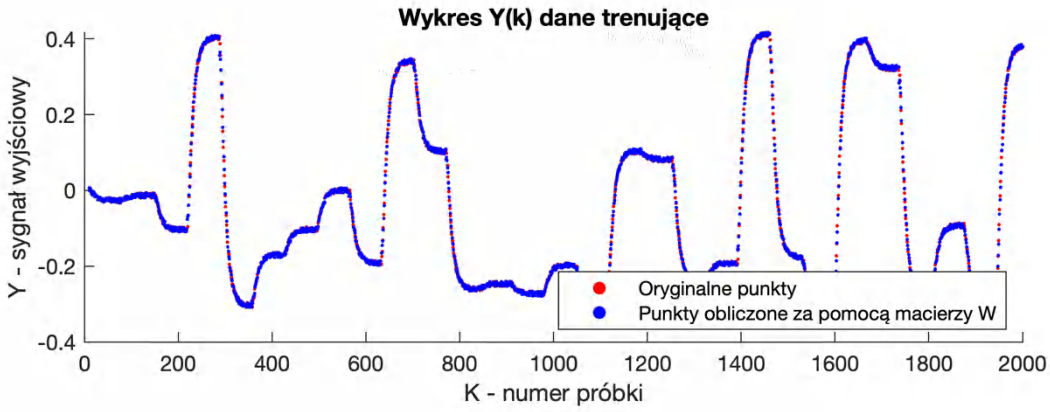


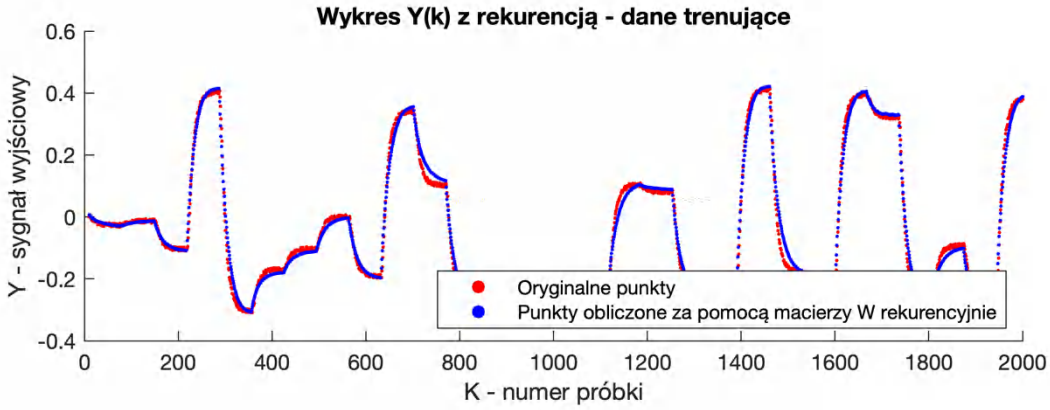


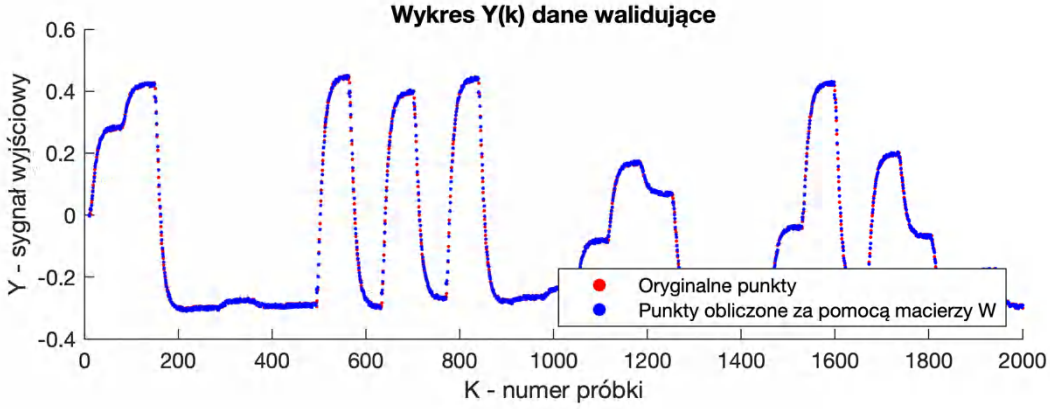
$$E_{trenujące} = 0.081656033037185$$
$$E_{trenującerec} = 0.919912861063571$$
$$E_{walidacyjne} = 0.093540475660551$$
$$E_{walidacyjnerec} = 1.177429099695852$$

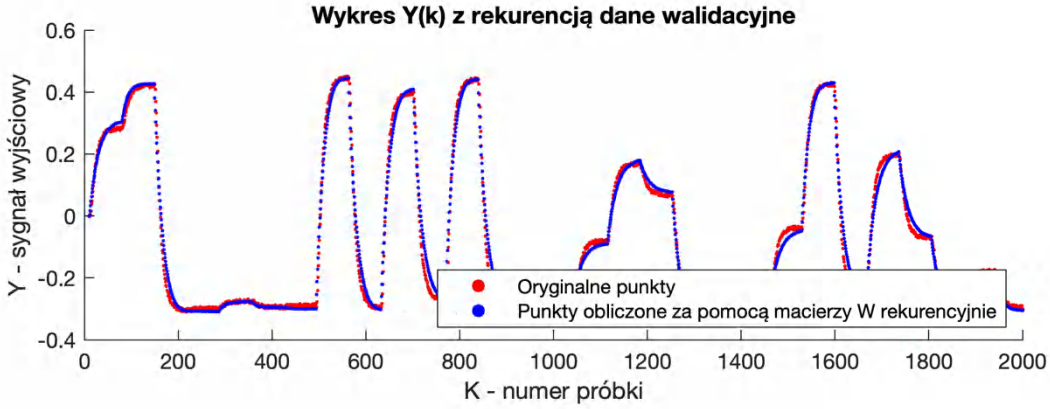
Po raz kolejny odnotowujemy zmniejszenie się błędów więc naturalnym krokiem będzie zbadanie zachowania modelu dla stopnia 5 i rzędu 1.

$$y(k) = w_1u(k-1) + w_2u^2(k-1) + w_3u^3(k-1) + w_4u^4(k-1) + w_5u^5(k-1) + w_6y(k-1) + w_7y^2(k-1) + w_8y^3(k-1) + w_9y^4(k-1) + w_{10}y^5(k-1)$$







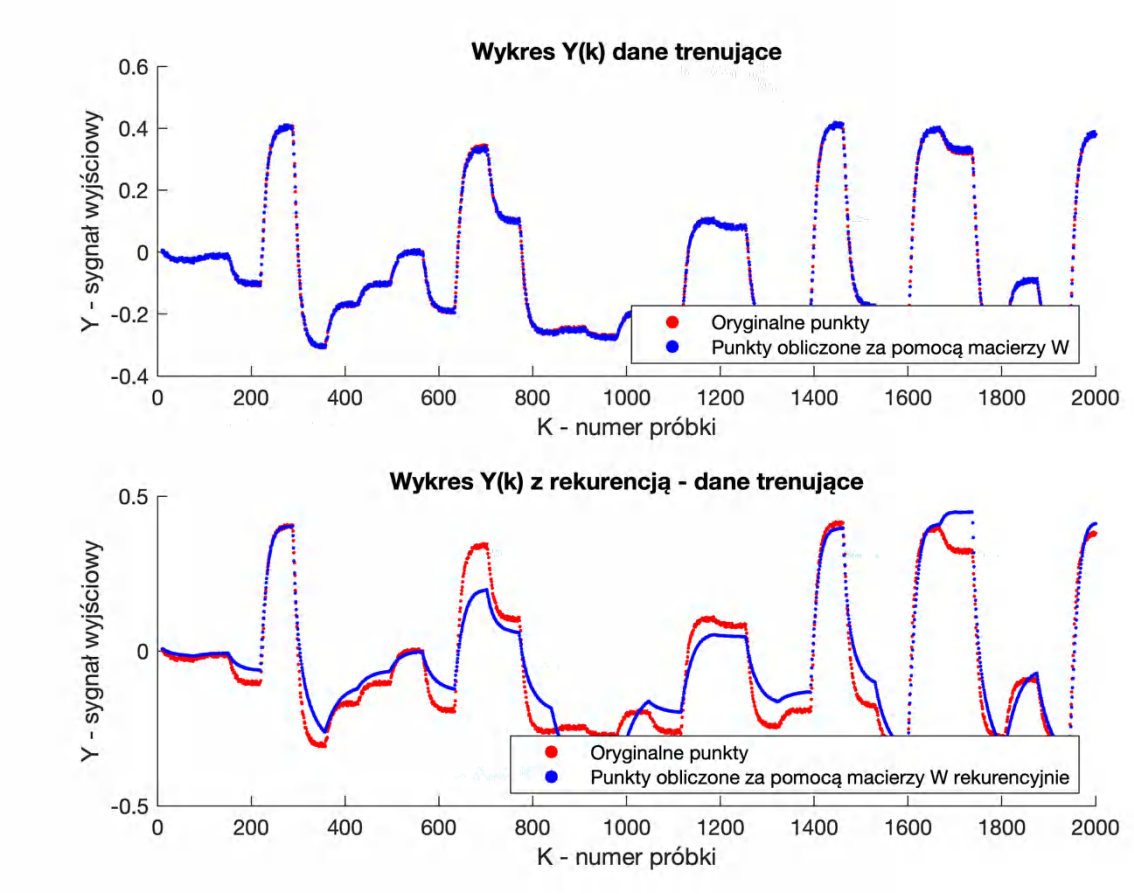


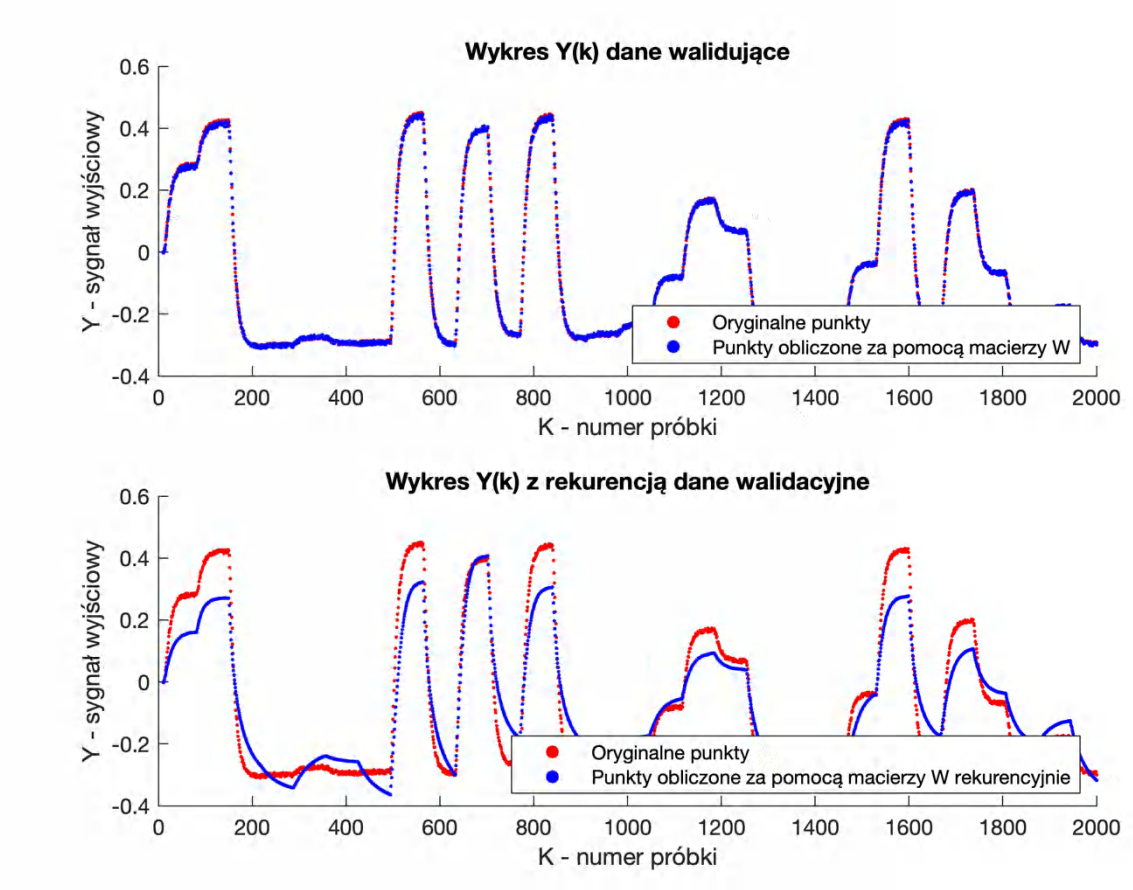
$$E_{trenujące} = 0.081152916710212$$

$$E_{trenującerec} = 0.863471085680167$$
$$E_{walidacyjne} = 0.093576414643954$$
$$E_{walidacyjnerec} = 1.117475505951403$$

Ponownie błąd się zmniejszył. Zastanówmy się co się stanie dla modeli rzędu 2 i zacznijmy analizę od modelu opisanego równaniem :

$$y(k) = w_1u(k-1) + w_2u^2(k-1) + w_3u(k-2) + w_3u^2(k-2) + w_5y(k-1) + w_6y^2(k-1) + w_7y(k-2) + w_8y^2(k-2)$$

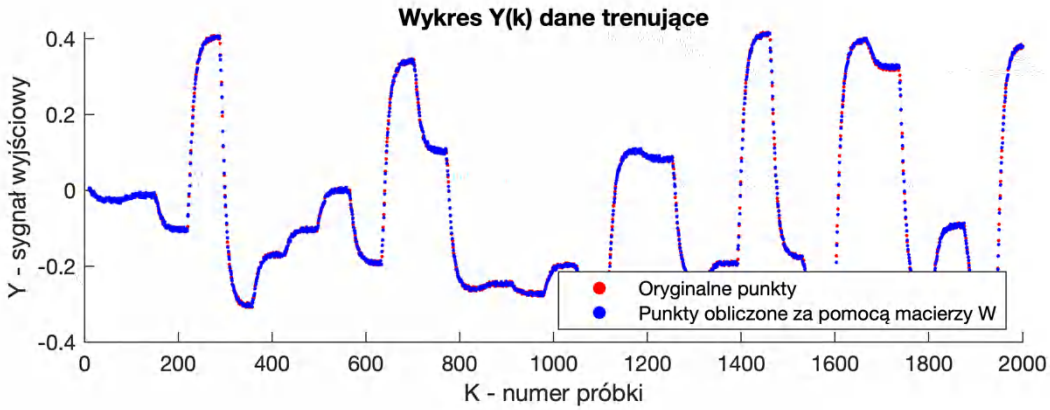


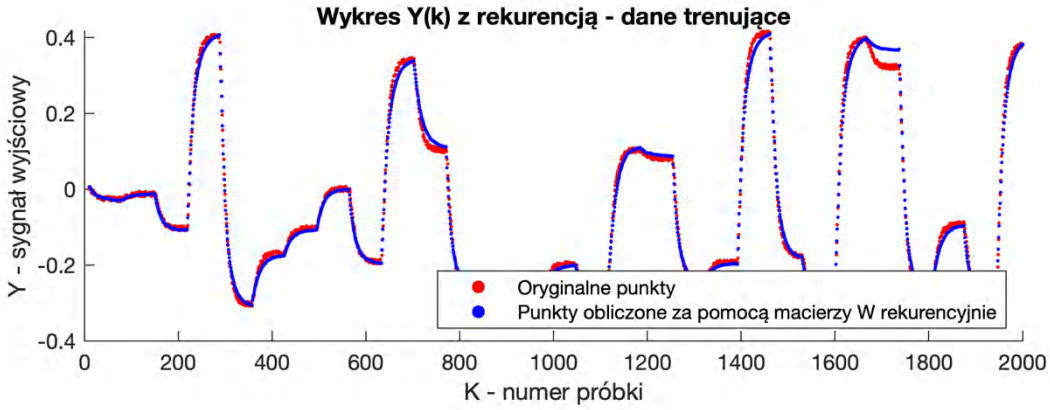


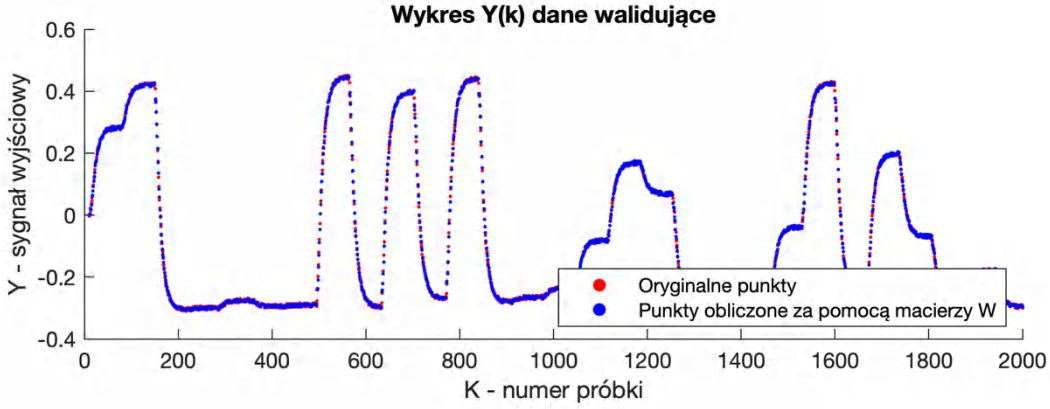
$$E_{trenujące} = 0.091479671228363$$
$$E_{trenującerec} = 9.333888787021714$$
$$E_{walidacyjne} = 0.111870964782965$$
$$E_{walidacyjnerec} = 13.097686937269827$$

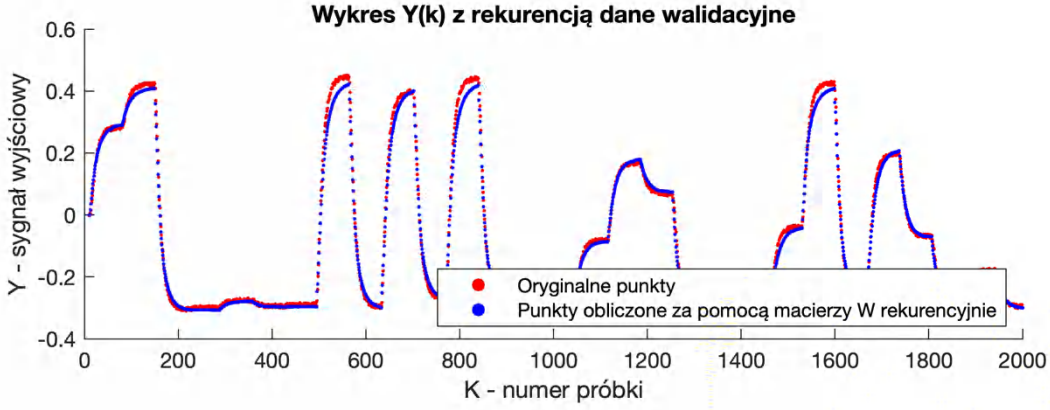
Kolejnym modelem, który będziemy analizować jest model o stopniu 3 i rzędzie drugim opisanym równaniem.

$$y(k) = w_1u(k-1) + w_2u^2(k-1) + w_3u^3(k-1) + w_4u(k-2) + w_5u^2(k-2) + w_6u^3(k-2) + w_7y(k-1) + w_8y^2(k-1) + w_9y^3(k-1) + w_{10}y(k-2) + w_{11}y^2(k-2) + w_{12}y^3(k-2)$$


Wykres Y(k) dane trenujące
Y - sygnał wyjściowy
K - numer próbki
Oryginalne punkty
Punkty obliczone za pomocą macierzy W


Wykres Y(k) z rekurencją - dane trenujące
Y - sygnał wyjściowy
K - numer próbki
Oryginalne punkty
Punkty obliczone za pomocą macierzy W rekurencyjnie


Wykres Y(k) dane walidujące
Y - sygnał wyjściowy
K - numer próbki
Oryginalne punkty
Punkty obliczone za pomocą macierzy W

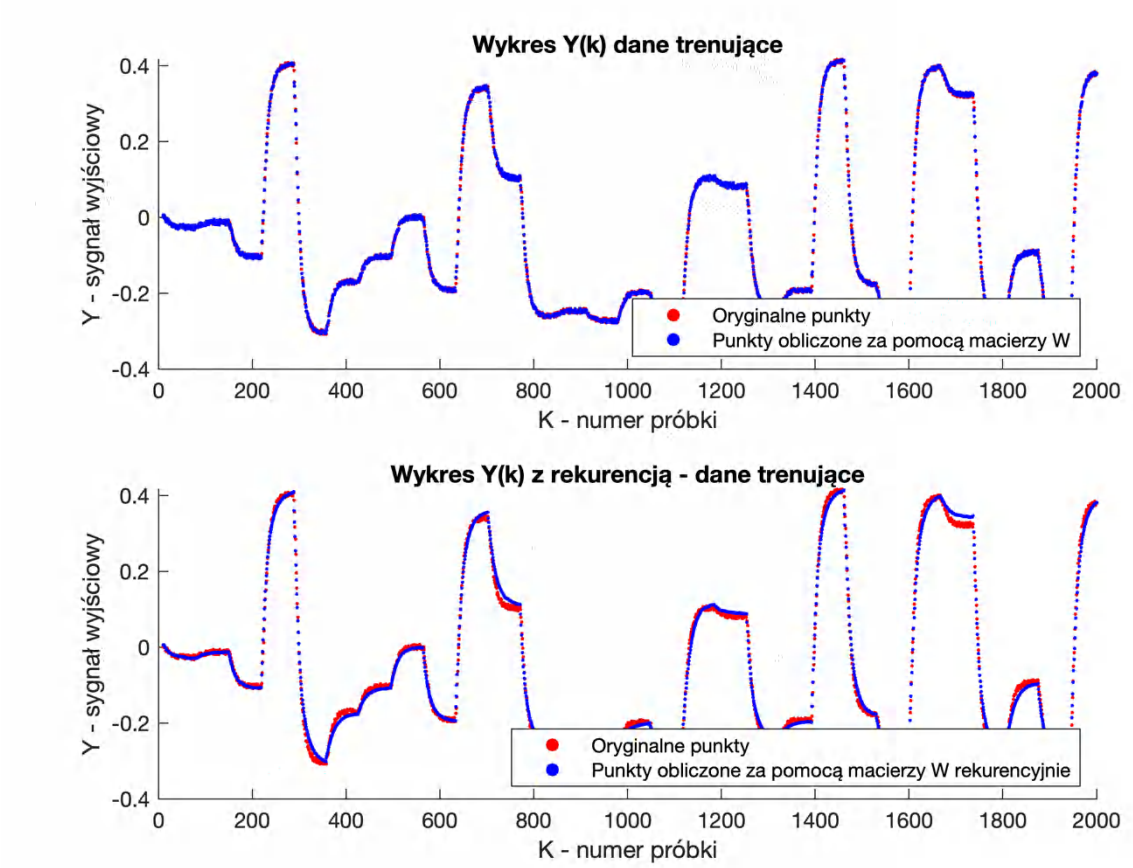

Wykres Y(k) z rekurencją dane walidacyjne
Y - sygnał wyjściowy
K - numer próbki
Oryginalne punkty
Punkty obliczone za pomocą macierzy W rekurencyjnie

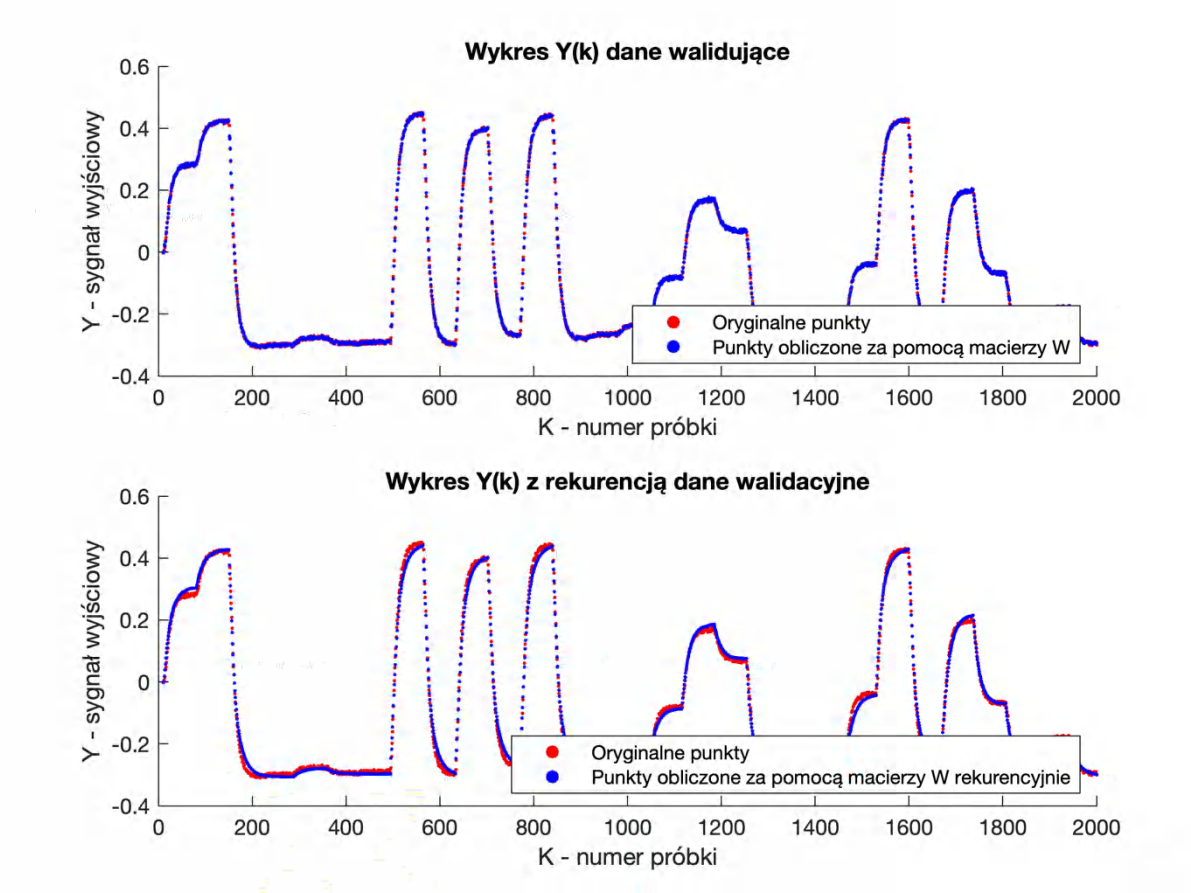
$$E_{trenujące} = 0.061184876127228$$
$$E_{trenującerec} = 0.500767125572999$$
$$E_{walidacyjne} = 0.065257695886117$$
$$E_{walidacyjnerec} = 0.681993732191418$$

Odnotowujemy znaczny spadek błędów. Analizując zachowanie powyższego modelu możemy stwierdzić, że działa on w wystarczająco satysfakcjonujący sposób w momencie, gdy jego stopień wynosi min. 3

Aby uniknąć zasypywania sprawozdania długimi równaniami pozwolę sobie pominąć wyprowadzanie równań a skupić się na opisaniu modelu za pomocą jego stopnia i rzędu

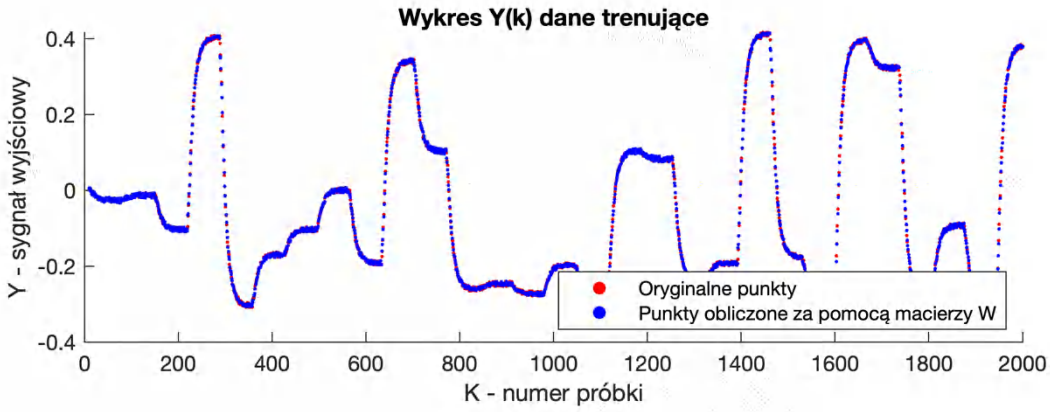
Rząd 3 Stopień 4:

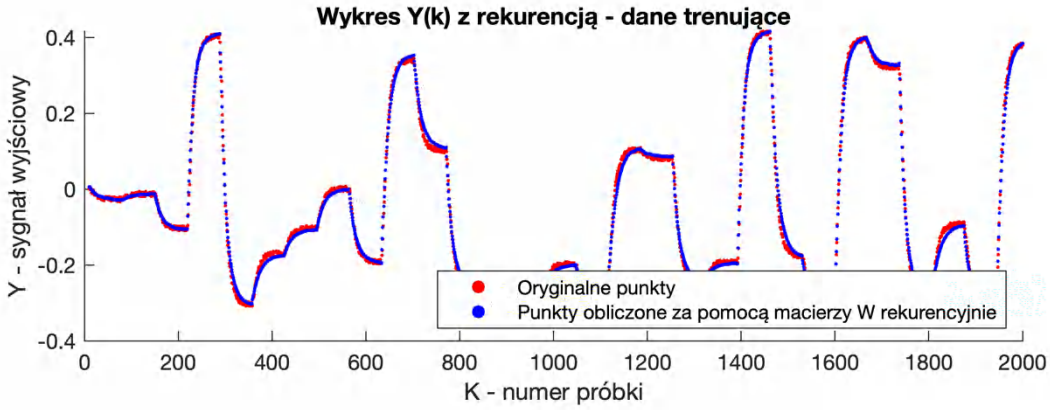


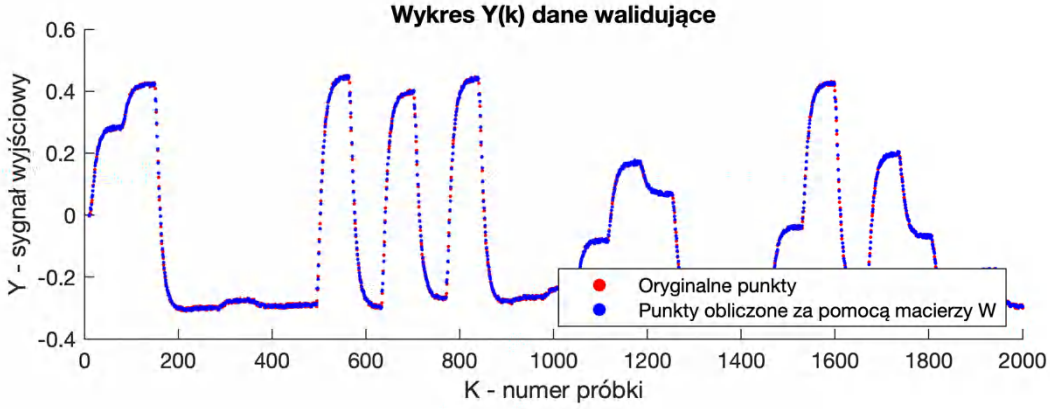


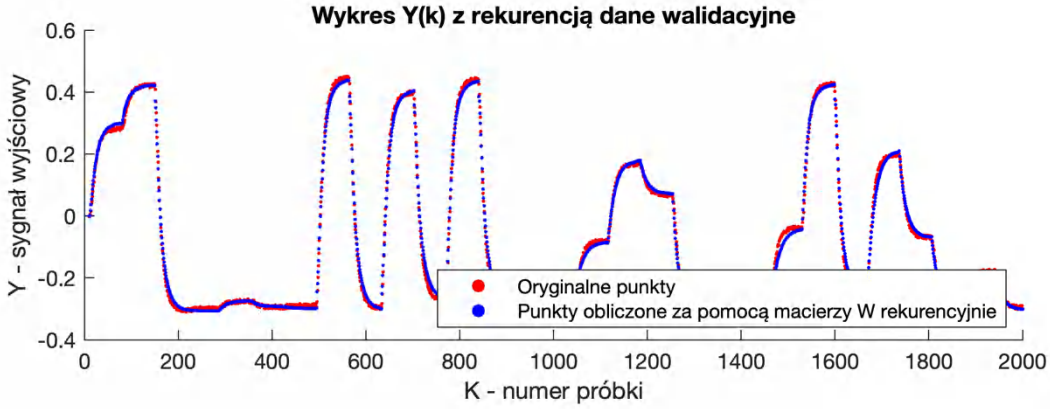
$$E_{trenujące} = 0.060319811807658$$
$$E_{trenującerec} = 0.369006465231484$$
$$E_{walidacyjne} = 0.064807258917069$$
$$E_{walidacyjnerec} = 0.454985219301132$$

Rząd 2 Stopień 5:





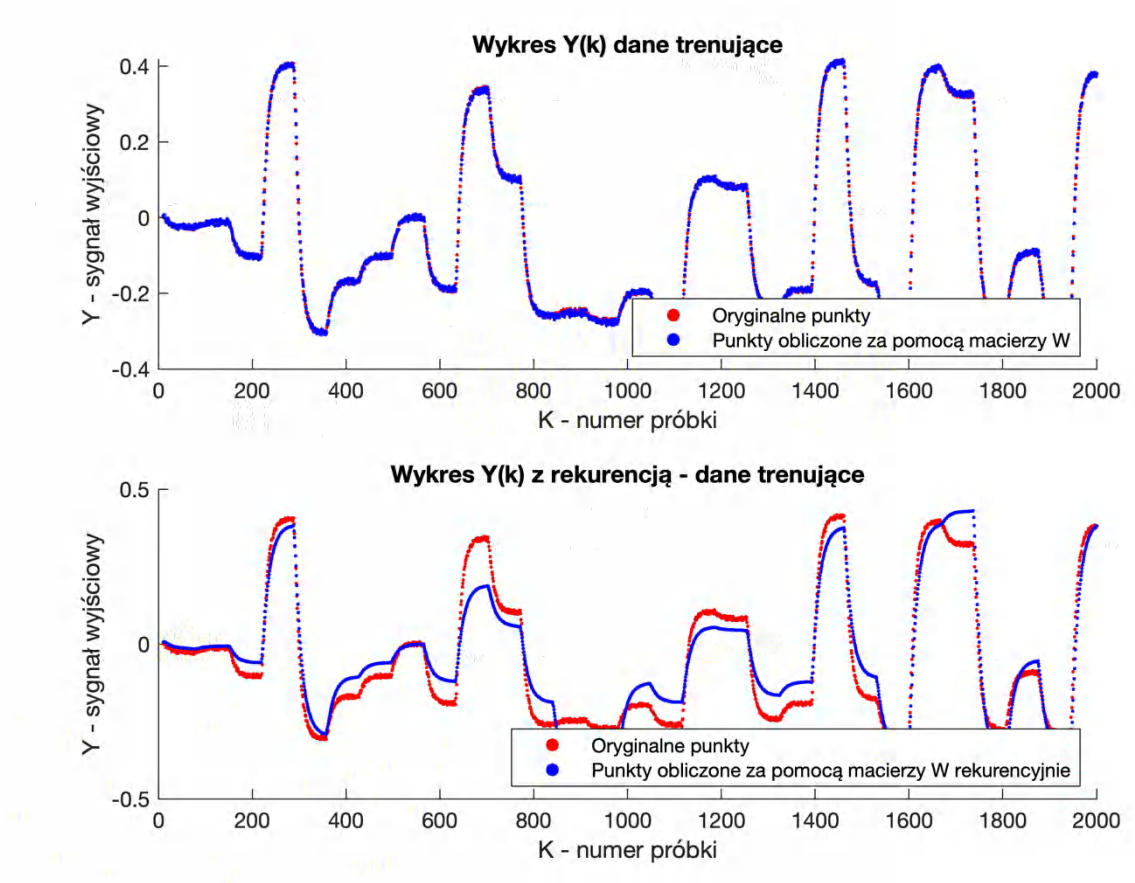


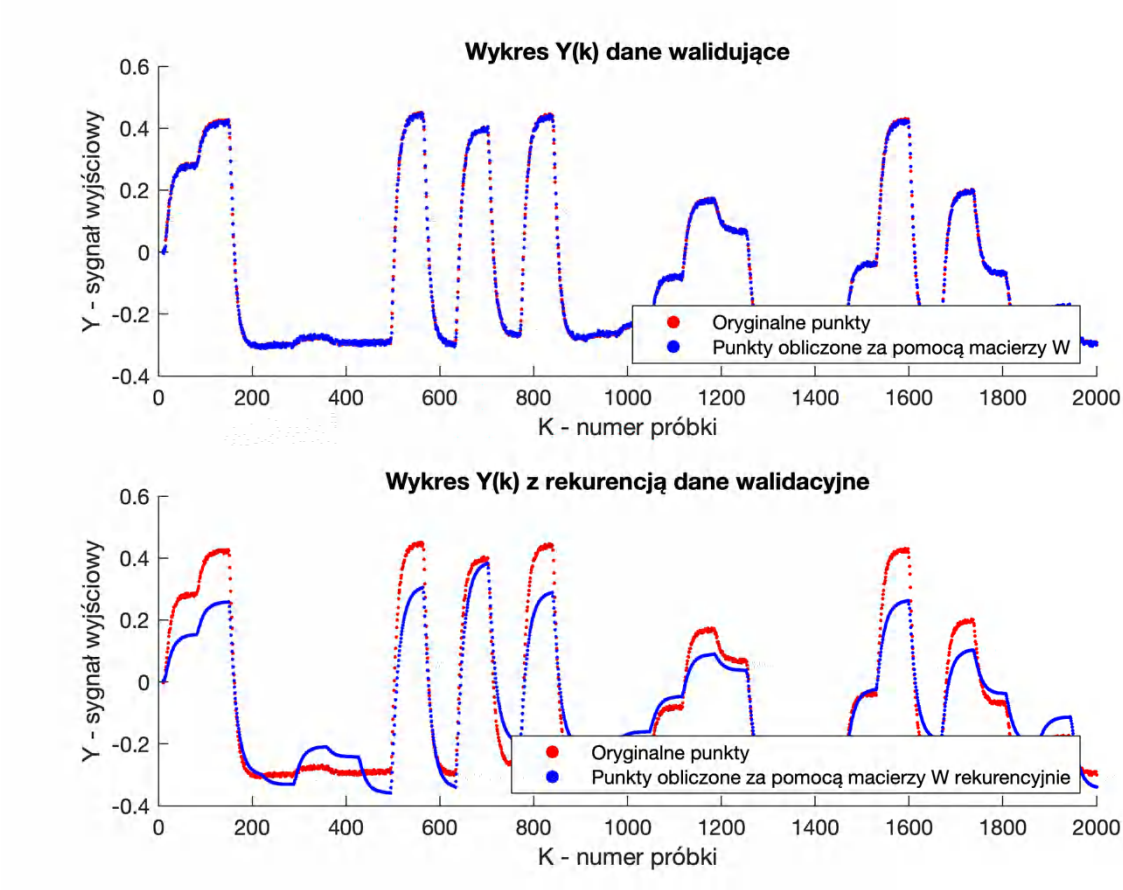


$$E_{trenujące} = 0.059325141053611$$

$$E_{trenującerec} = 0.316346349302739$$
$$E_{walidacyjne} = 0.064477160962602$$
$$E_{walidacyjnerec} = 0.391628315328667$$

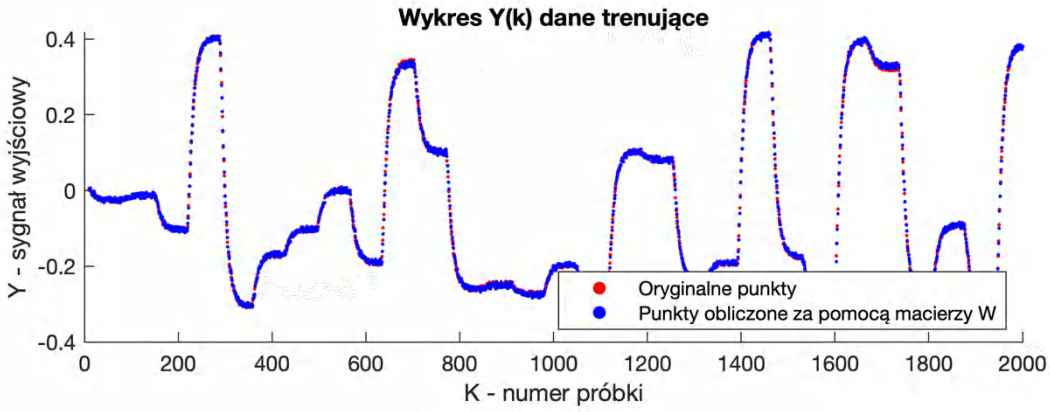
Rząd 3 stopień 1:

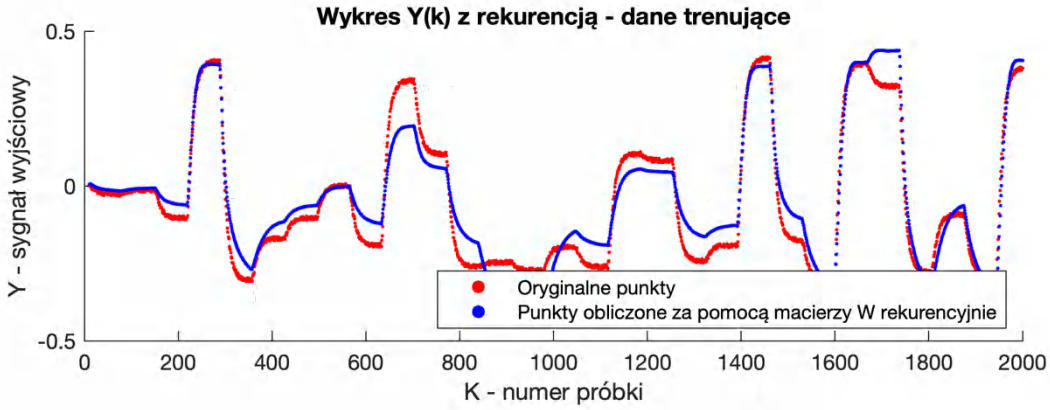


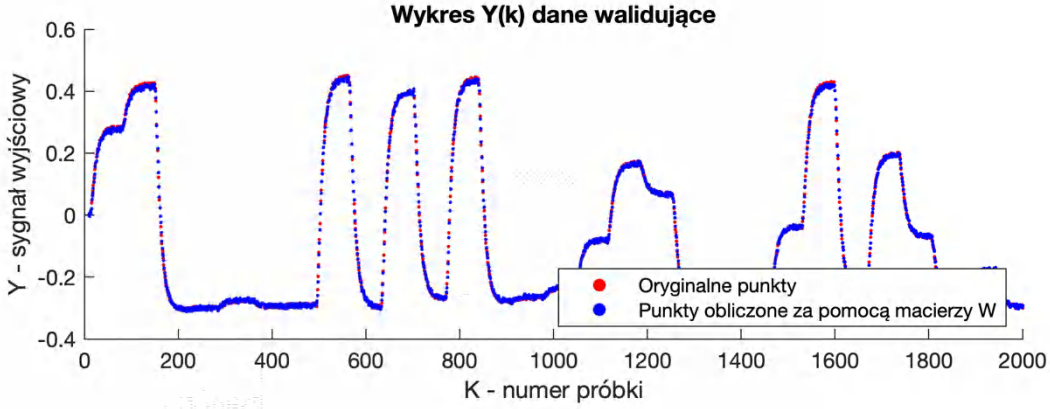


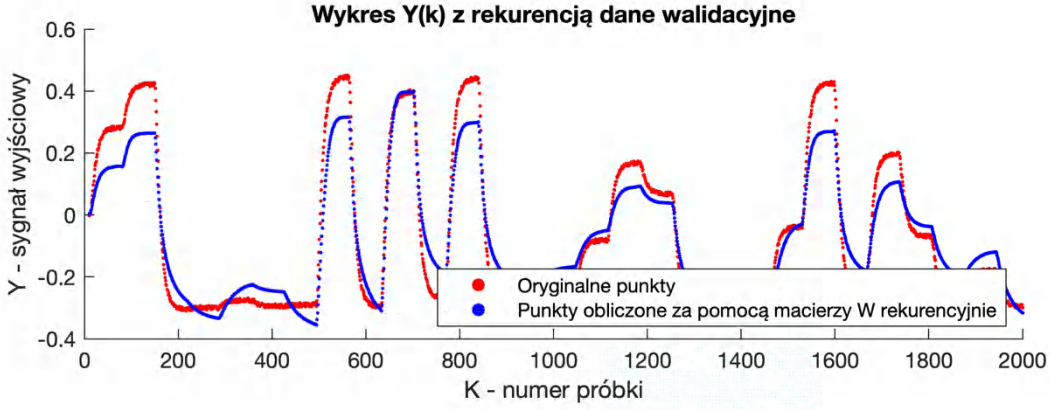
$$E_{trenujące} = 0.080012200309852$$
$$E_{trenującerec} = 9.486346449070700$$
$$E_{walidacyjne} = 0.086051352314397$$
$$E_{walidacyjnerec} = 13.145263468262010$$

Rząd 3 stopień 2:





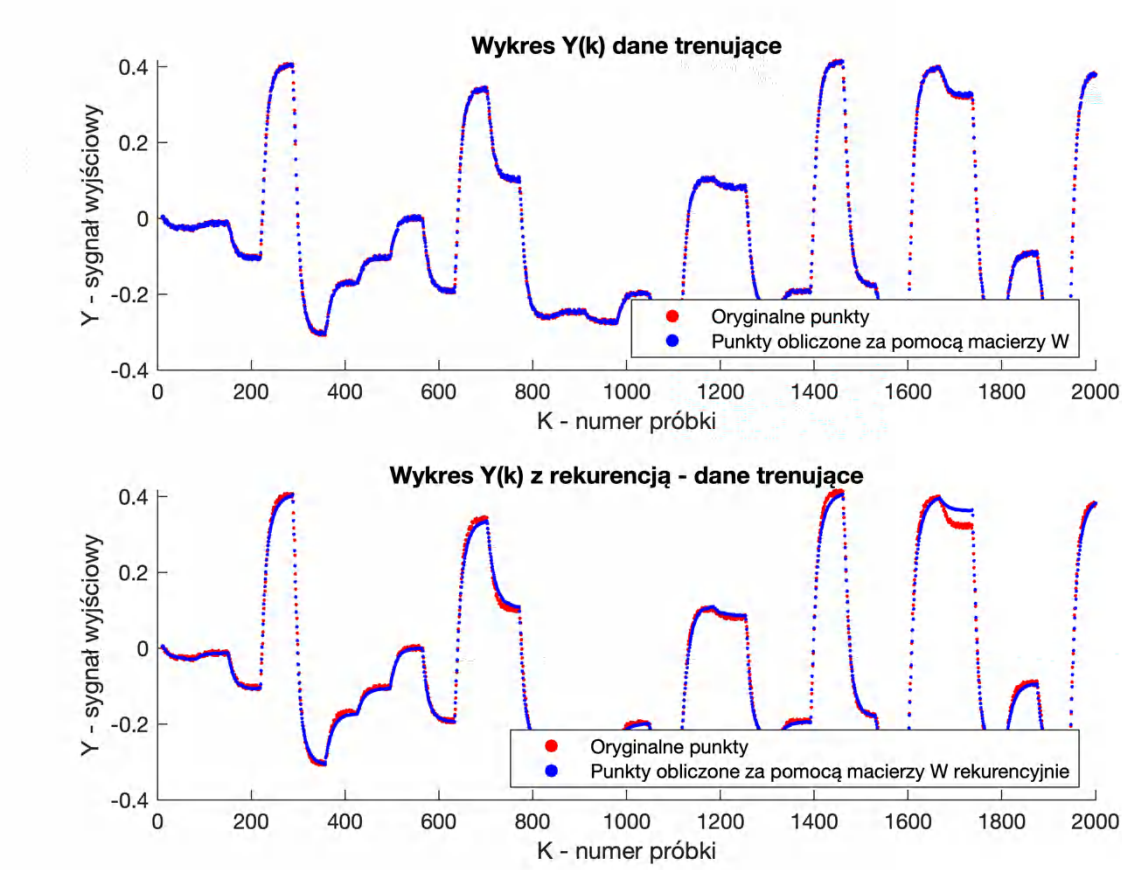


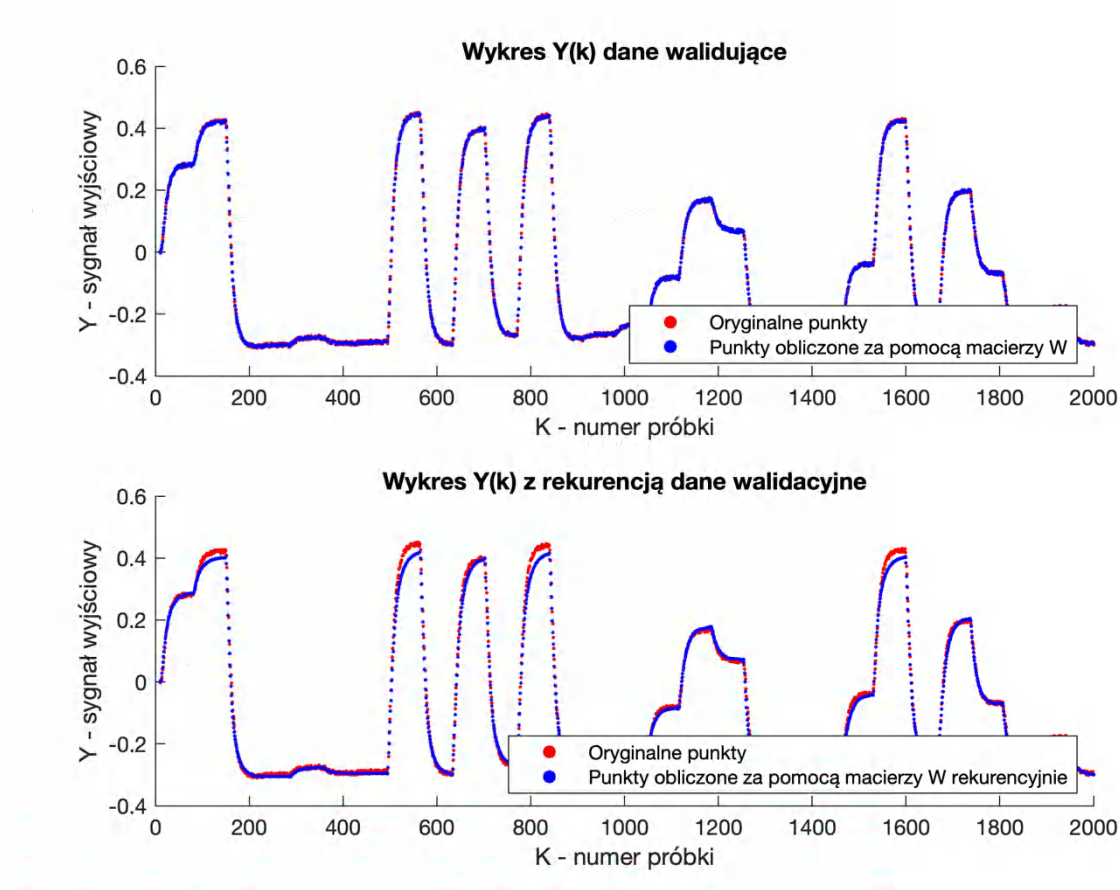


$$E_{trenujące} = 0.076774211847237$$

$$E_{trenującerec} = 8.228829453201229$$
$$E_{walidacyjne} = 0.088315869874434$$
$$E_{walidacyjnerec} = 11.692940152311902$$

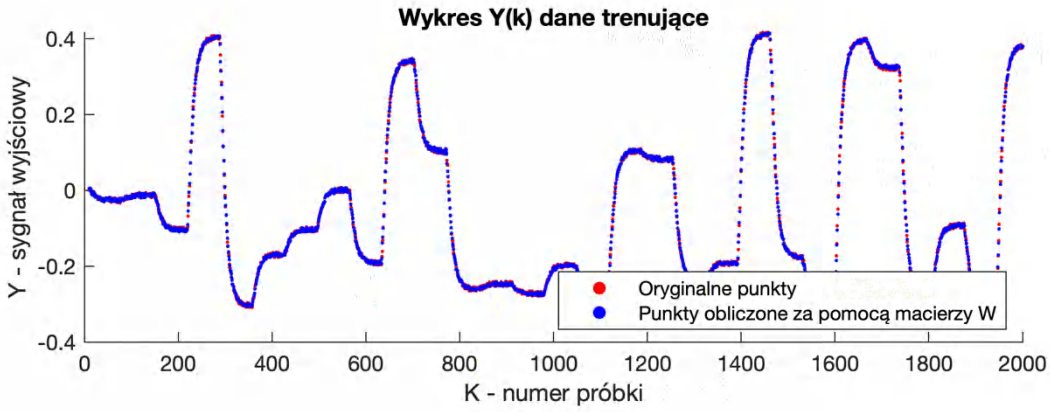
Rząd 3 stopień 3:

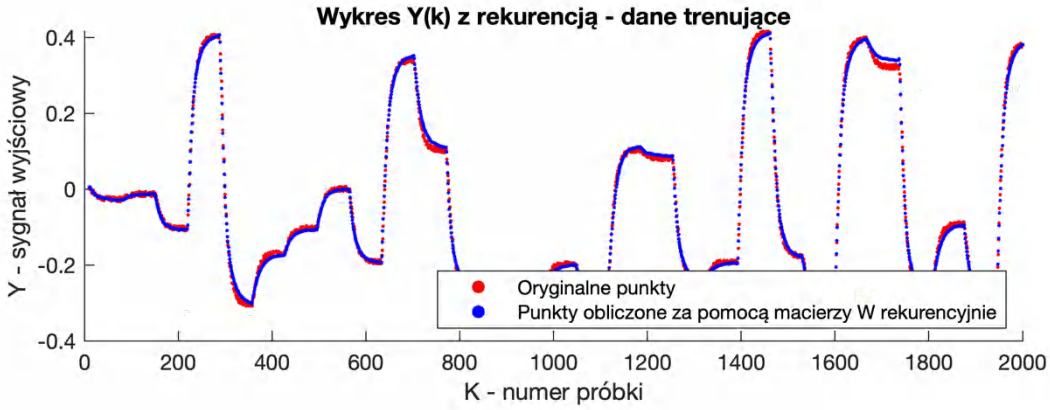


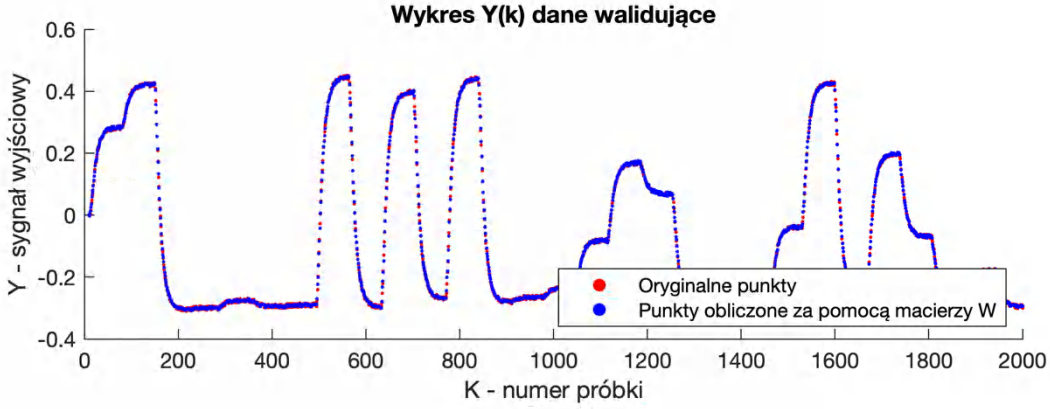


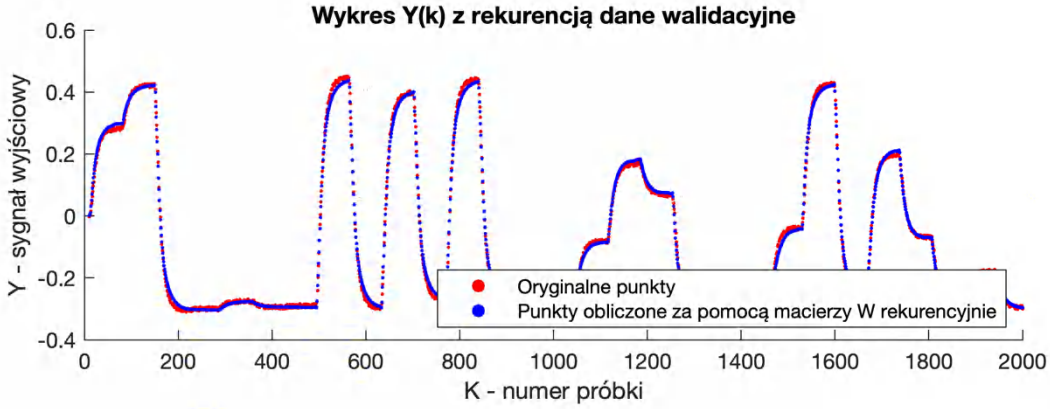
$$E_{trenujące} = 0.050462668839551$$
$$E_{trenującerec} = 0.322904235617871$$
$$E_{walidacyjne} = 0.052607699162180$$
$$E_{walidacyjnerec} = 0.513131459568140$$

Rząd 3 stopień 4:





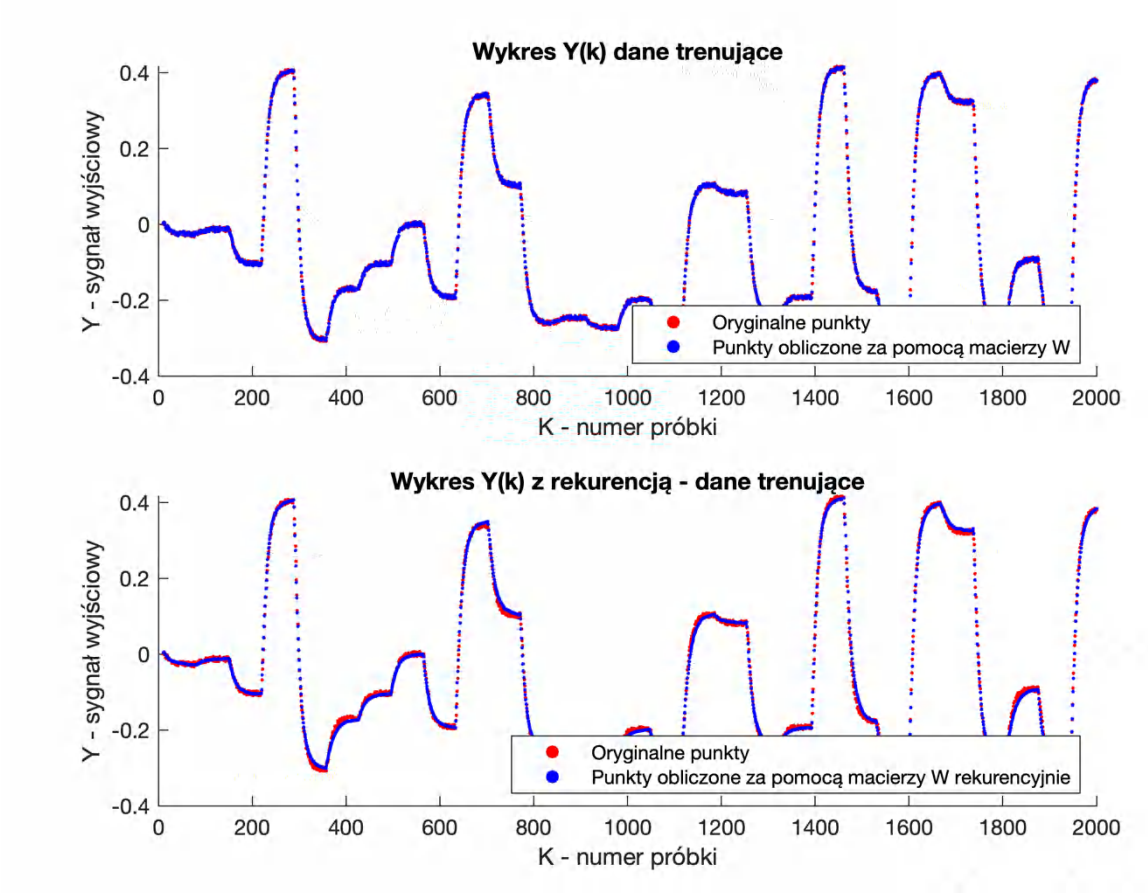


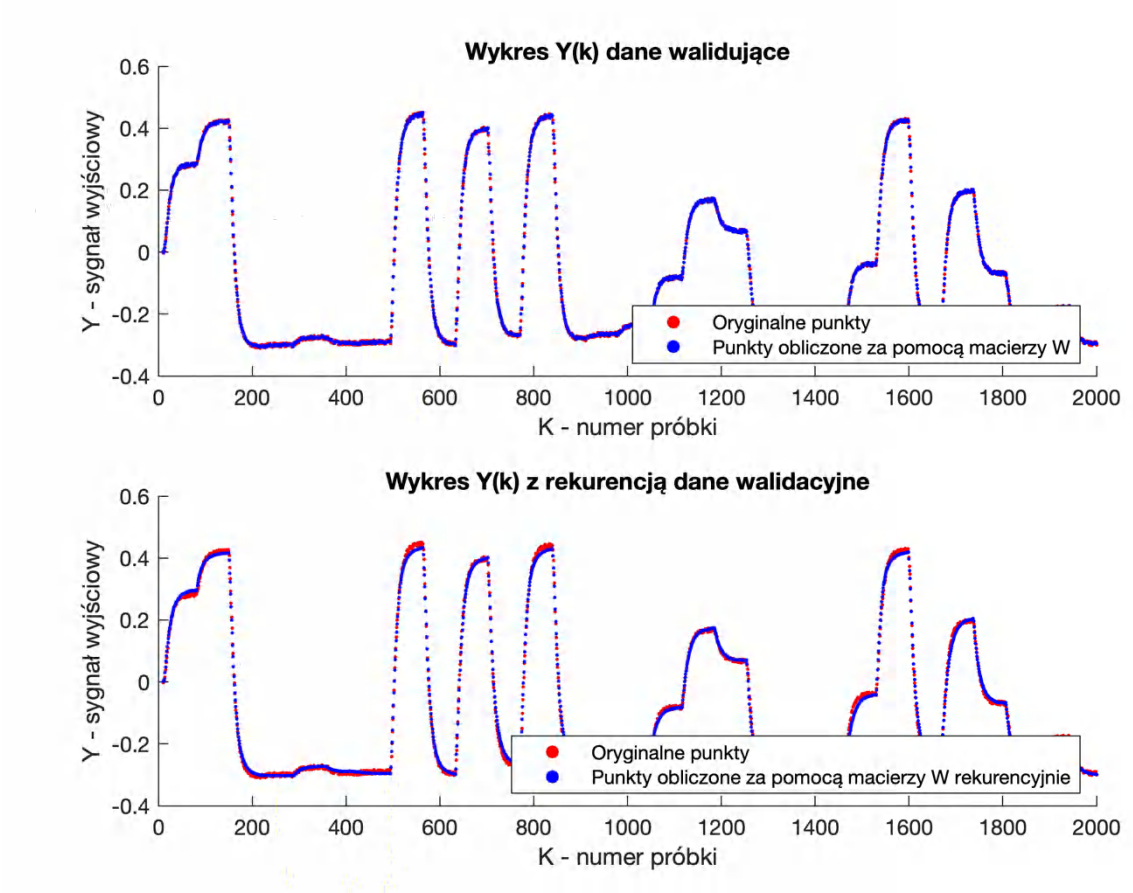


$$E_{trenujące} = 0.049395780319612$$

$$E_{trenującerec} = 0.207299429502114$$
$$E_{walidacyjne} = 0.051262615604847$$
$$E_{walidacyjnerec} = 0.254553460838299$$

Rząd 3 stopień 5:





$$E_{trenujące} = 0.047728825579350$$
$$E_{trenującerec} = 0.167892334952594$$
$$E_{walidacyjne} = 0.050643665191150$$
$$E_{walidacyjnerec} = 0.215990211380756$$

Analizując powyższe modele tworzę tabel, w której wpisuje błędy dla najdokładniejszego modelu z każdego rzędu.

Stopień	Rząd	Błąd dla danych trenujących		Błąd dla danych walidacyjnych	
		Bez rekurencji	Z rekurencją	Bez rekurencji	Z rekurencją
5	1	0.081152916710212	0.863471085680167	0.093576414643954	1.117475505951403
5	2	0.059325141053611	0.316346349302739	0.064477160962602	0.391628315328667
5	3	0.047728825579350	0.167892334952594	0.050643665191150	0.215990211380756

Dla powyższych modeli jednoznacznie możemy zauważyć, że dla stopnia 5 i rzędu 3 otrzymujemy najlepszy model z tych, które poddaliśmy analizie. Zarówno dla modelu z rekurencją i bez niej odnotowujemy najmniejsze błędy co świadczy o wysokiej precyzji modelu. Tryb z rekurencją pozwala lepiej pozwala na ocenę jakości ponieważ dane mają charakter sekwencyjny i są bezpośrednie zależne od wcześniej wyznaczonych wartości wyjść. Taki rozkład powoduje, że zdecydowanie lepszym kryterium doboru jest kryterium analizy błędu rekurencyjnego i przeanalizowanie jego zachowania dla poszczególnych stopniu i rzędów równań.

Użycie funkcji *fsolve* do wizualizacji statycznego modelu nieliniowego

Aby móc użyć skutecznie funkcji *fsolve* należy najpierw dokonać przekształcenia równania na statyczne. Z wcześniejszych modeli wybieram model:

$$y(k) = w_1u(k-1) + w_2u^2(k-1) + w_3u^3(k-1) + w_4u^4(k-1) + w_5u^5(k-1) + w_6u(k-2) + w_7u^2(k-2) + w_8u^3(k-2) + w_9u^4(k-2) + w_{10}u^5(k-2) + w_{11}u(k-3) + w_{12}u^2(k-3) + w_{13}u^3(k-3) + w_{14}u^4(k-3) + w_{15}u^5(k-3) + w_{16}y(k-1) + w_{17}y^2(k-1) + w_{18}y^3(k-1) + w_{19}y^4(k-1) + w_{20}y^5(k-1) + w_{21}y(k-2) + w_{22}y^2(k-2) + w_{23}y^3(k-2) + w_{24}y^4(k-2) + w_{25}y^5(k-2) + w_{26}y(k-3) + w_{27}y^2(k-3) + w_{28}y^3(k-3) + w_{29}y^4(k-3) + w_{30}y^5(k-3)$$

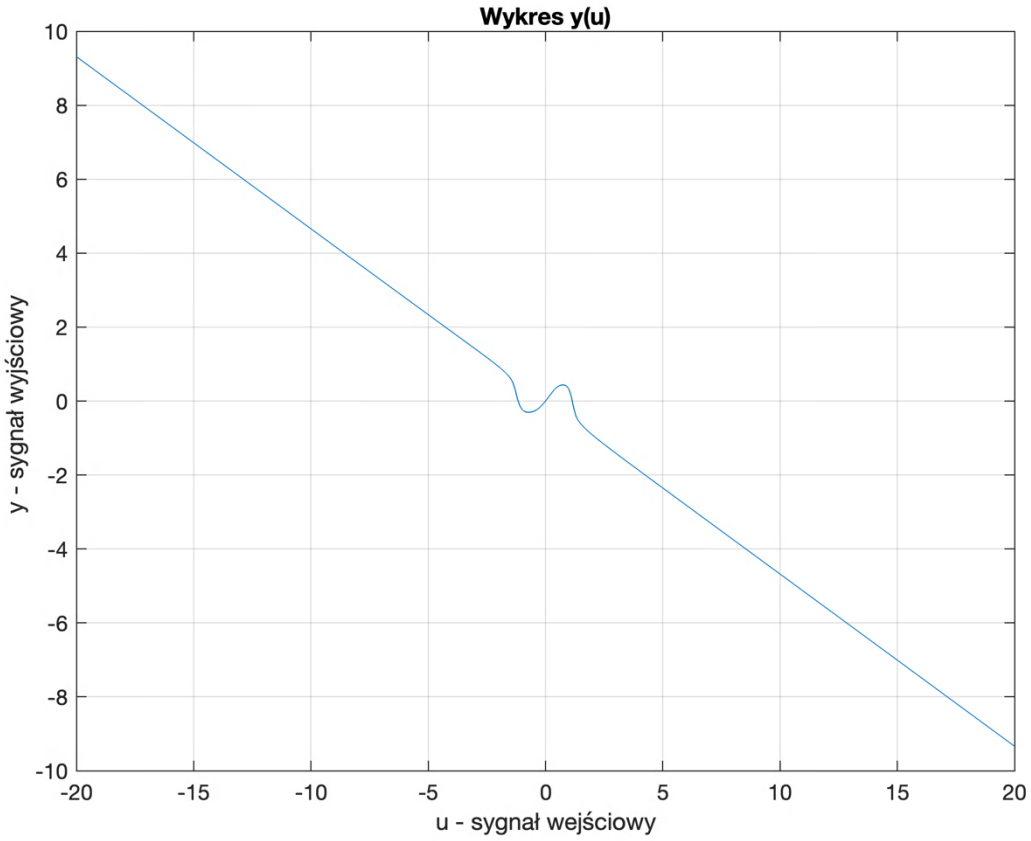
Następnie przekształcam model do postaci statycznego modelu nieliniowego:

$$y = w_1u + w_2u^2 + w_3u^3 + w_4u^4 + w_5u^5 + w_6u + w_7u^2 + w_8u^3 + w_9u^4 + w_{10}u^5 + w_{11}u + w_{12}u^2 + w_{13}u^3 + w_{14}u^4 + w_{15}u^5 + w_{16}y + w_{17}y^2 + w_{18}y^3 + w_{19}y^4 + w_{20}y^5 + w_{21}y + w_{22}y^2 + w_{23}y^3 + w_{24}y^4 + w_{25}y^5 + w_{26}y + w_{27}y^2 + w_{28}y^3 + w_{29}y^4 + w_{30}y^5$$

Po wyłączeniu przed nawias elementów powtarzających się:

$$y = u(w_1 + w_6 + w_{11}) + u^2(w_2 + w_7 + w_{12}) + u^3(w_3 + w_8 + w_{13}) + u^4(w_4 + w_9 + w_{14}) + u^5(w_5 + w_{10} + w_{15}) + y(w_{16} + w_{21} + w_{26}) + y^2(w_{17} + w_{22} + w_{27}) + y^3(w_{18} + w_{23} + w_{28}) + y^4(w_{19} + w_{24} + w_{29}) + y^5(w_{20} + w_{25} + w_{30})$$

Tak sformułowane równanie zostanie rozwiązane przeze mnie przy użyciu *fsolve*. Przed wykonaniem podzielę zbiór zawierający u na dwie częśći, jedna od 0 do prawej granicy zakresu, druga od 0 do lewej granicy zakresu $np. U = -20{:}\,0.1{:}\,20 - lewa\ granica\ zakresu{:}\,krok{:}\,prawa\ granica\ zakresu.$ Ma to na celu zapewnienie, że jedyny znajomy mi punkt, czyli $(0,0)$ bo w równaniu nie ma wyrazów wolnych posłuży mi jako wartość początkowa w funkcji $fsolve$. Charakterystyka dla tego równania wygląda w poniższy sposób:



Poprawność działania mojego solvera sprawdziłem w popularnym programie GeoGebra i oba wykresy przedstawiają to samo, więc solver działa poprawnie.