

Spis treści

1. Wstęp
 - 1.1. Cel i zakres pracy
 - 1.2. Spis głównych oznaczeń
2. Sztuczne Sieci Neuronowe
 - 2.1. Ogólne wiadomości o SSN
 - 2.2. Jednokierunkowe i rekurencyjne warstwowe SSN i metody ich uczenia
3. Filtr Kalmana jako metoda analizy dyskretnych procesów stochastycznych
 - 3.1. Liniowy Filtr Kalmana
 - 3.2. Rozszerzony Filtr Kalmana
4. Filtr Kalmana jako metoda uczenia SSN
 - 4.1. Aktualny stan zagadnienia - uczenie SSN z wykorzystaniem FK
 - 4.2. Dostosowanie ogólnego modelu analizy dyskretnych procesów stochastycznych za pomocą FK do procesu uczenia SSN
 - 4.2.1. Model nieliniowy, algorytm Extended Kalman Filter (EKF)
 - 4.2.2. Model rozprzężony nieliniowy, algorytm Decoupled Extended Kalman Filter (DEKF)
 - 4.2.3. Model liniowy rozprzężony wg. neuronów SSN, algorytm Node Decoupled Linear Kalman Filter (NDLKF)
 - 4.3. Uczenie SSN jednokierunkowych i rekurencyjnych jedno oraz dwuwarstwowych za pomocą Filtra Kalmana
5. Zastosowanie SSN uczonych za pomocą FK
 - 5.1. Symulacja Spektrow Odpowiedzi od wymuszeń parasejsmicznych
 - 5.1.1. Symulacja Akceleracyjnych Spektrow Odpowiedzi od wstrząsów górniczych
 - 5.1.2. Interakcja Przemieszczeniowych Spektrow Odpowiedzi grunt – fundament budynku
 - 5.2. Predykcja pętli histerezy od obciążeń niskocyklicznych
 - 5.2.1. Predykcja pętli histerezy dla kompozytu hierarchicznego złożonego z wiązek nadprzewodników

- 5.2.2. Symulacja i predykcja pętli histerezy w próbkach stalowych
- 5.2.3. Modelowanie i predykcja zależności naprężenie – odkształcenie dla cyklicznie obciążanych próbek betonowych

6. Projektowanie SSN uczonych za pomocą FK

- 6.1. Dobór architektury SSN w oparciu o algorytm obcinania (pruning)
- 6.2. Uczenie na podstawie analizy wartości macierzy kowariancji przybliżonych błędów filtracji
- 6.3. Dobór architektury SSN na podstawie algorytmu obcinania – wybór parametrów sterujących na podstawie testów numerycznych
- 6.4. Uczenie SSN na podstawie analizy wartości macierzy kowariancji przybliżonych błędów filtracji – wybór parametrów sterujących na podstawie testów numerycznych

7. Pakiet programów do uczenia oraz projektowania SSN

8. Zakończenie

- 8.1. Rezultaty pracy
- 8.2. Wykazane tezy oraz rezultaty w zakresie mechaniki
- 9.3. Kierunki dalszych prac
- 8.3. Wnioski ogólne w zakresie metodologii stosowania SSN uczonych za pomocą FK, do rozwiązywania rozpatrywanych problemów mechaniki
- 8.4. Wnioski szczegółowe
- 8.5. Elementy oryginalne rozprawy
- 8.6. Kierunki dalszych prac.

9. Literatura

10. Dodatki

- 10.1. Wyprowadzenie równań liniowego FK
- 10.2. Wyprowadzenie równań rozszerzonego FK
- 10.3. Definicje akceleracyjnego (przyspieszeniowego) i przemieszczeniowego spektrum odpowiedzi oraz wykorzystanie spektrum odpowiedzi do projektowania budynków i oceny szkodliwości drgań parasejsmicznych

Podziękowania

Autorka składa podziękowania Panu Profesorowi Zenonowi Waszczyszynowi za cierpliwość i pomoc udzieloną podczas pisania pracy. Dziękuje również wszystkim, którzy wspierali ją cennymi uwagami i życzliwością.

Badania, wykorzystane w niniejszej pracy, zostały wykonane min. w ramach Subsydium FNP dla Uczonych nr 13/2001 pt. "Zastosowanie sztucznych sieci neuronowych do analizy zagadnień inżynierii lądowej i wodnej" oraz B.P. 1507/T07/2005/28.

1. Wstęp

1.1. Cele i zakres pracy

a. Cele pracy

Cel główny pracy stanowi analiza wybranych problemów mechaniki konstrukcji i materiałów, trudnych do rozwiązania metodami tradycyjnymi (tzw. twardymi metodami), lub metodami miękkimi, Sztucznych Sieci Neuronowych (SSN). Wybrane problemy dotyczą predykcji spektrów odpowiedzi w dynamice konstrukcji oraz pętli histerezy w mechanice materiałów.

W niniejszej pracy zastosowano do rozwiązania ww. problemów - SSN, uczone za pomocą filtrów Kalmana (FK), jako metodzie uczenia matematycznie ugruntowanej, opartej na teorii dyskretnych procesów stochastycznych.

Cele szczegółowe pracy stanowiło:

1. Wykazanie numerycznej efektywności metody uczenia SSN, opartej o filtry Kalmana, do aproksymacji teoretycznych oraz doświadczalnych danych mechaniki konstrukcji i materiałów.
2. Dobór metodologii rozwiązywania ww. problemów za pomocą SSN.
3. Opracowanie własnego oprogramowania, pozwalającego na uczenie oraz projektowanie jednokierunkowych i rekurencyjnych SSN za pomocą FK.
4. Analiza wybranych zagadnień mechaniki konstrukcji i materiałów.

b. Zakres pracy

I. Problemy z zakresu mechaniki, rozważane w pracy dotyczą:

1. W dziedzinie mechaniki konstrukcji:

- symulacja Akceleracyjnych (Przyśpieszeniowych) Spektrów Odpowiedzi, ASO, od wstrząsów górniczych, w zależności od energii i odległości epicentralnej rozpatrywanego wstrząsu,
- predykcja Przemieszczeniowych Spektrów Odpowiedzi (PSO) wybranego piętra budynku, w zależności od PSO gruntu przed budynkiem.

2. W dziedzinie mechaniki materiałów, rozpatrywane zagadnienia dotyczą, w zakresie symulacji i predykcji pętli histerezy od obciążeń niskocyklicznych, dotyczą:

- nadprzewodnika kriogenicznego, jako elementu cewki typu Solenoid, wchodzącej w skład systemu magnesów nadprzewodzących Reaktora Termonuklearnego ITER (International Thermonuclear Experimental Reaktor - ITER),
- stali nierdzewnej
- różnych próbek betonowych, poddanych obciążeniu cyklicznemu typu:
 1. ściskanie
 2. rozciąganie-rozciąganie
 3. rozciąganie-ściskanie.

Wszystkie rozpatrywane problemy są oparte na danych doświadczalnych, pochodzących z badań terenowych (pomiarów przy monitorowanych budynkach) oraz badań laboratoryjnych (na odpowiednich próbkach).

II. Problemy efektywności numerycznej proponowanych SSN.

W zakresie testowania efektywności SSN, uczonych za pomocą algorytmu Kalmana, do rozwiązywania ww. problemów mechaniki, przedstawiono porównania rozwiązań tych zagadnień, z następującymi metodami:

- tradycyjne metody mechaniki, w tym:
 - a. metody empiryczne,
 - b. modelowanie metodami teorii sprężystości i plastyczności,
 - c. modelowanie metodami mechaniki zniszczenia,
- stosowane dotąd metody neuronowe, oparte o klasyczne algorytmy uczenia SSN, wraz ze stosowanymi - odpowiednimi dla ww. metod wektorami informacji wejściowej do SSN,
- metody oparte o klasyczne algorytmy uczenia SSN oraz taką samą architekturę i dobór wektorów wejścia, jak rozważane w metodologii FK.

W zakresie rozwoju metody uczenia SSN, wykorzystującej filtrowanie Kalmana, w pracy zaprezentowano następujące autorskie rozwiązania, wraz z testami doboru parametrów sterujących oraz efektywności numerycznej:

- projektowanie SSN na podstawie zmodyfikowanego algorytmu zmniejszania liczby parametrów SSN (pruning),
- uczenie SSN, oparte na zróżnicowaniu wartości przyrostów poszczególnych wag SSN, w trakcie trwania procesu uczenia,
- uczenie SSN, oparte na analizie wartości macierzy przybliżonych błędów filtracji w modelach teoretycznych,

- dwustopniowe uczenie SSN, wykorzystujące analizę rozrzutu wartości błędu uczenia dla poszczególnych losowań wag początkowych sieci.

Rezultaty obliczeń przy pomocy SSN, uczonych w oparciu o filtr Kalmana, zostały otrzymane na podstawie własnego oprogramowania, zaimplementowanego w systemie do obliczeń naukowo-inżynierskich MATLAB. Omówienie pakietu wykonanych programów zawiera:

- motywacje wyboru środowiska obliczeniowego,
- opis cech środowiska MATLAB, wykorzystanych przy projektowaniu systemu obliczeń neuronowych, opartych o algorytm filtracji Kalmana, jako metodę uczenia.
- opis wykonanego oprogramowania, z wyszczególnieniem:
 - a. charakteryzacji architektury rozważanych SSN,
 - b. zastosowanych modeli teoretycznych i algorytmów uczenia,
 - c. zaproponowanych autorskich rozwinięć i modyfikacji algorytmu podstawowego.

W zakresie algorytmów w pracy przedstawiono proces uczenia SSN, na podstawie: Rozprzężonego Liniowego Filtra Kalman (Node Decoupled Linear Kalman Filter, NDLKF) oraz Rozprzężonego Nieliniowego (rozszerzonego) Filtra Kalman (Node Decoupled Extended Kalman Filter, NDEKF) dla sieci warstwowych jednokierunkowych, oraz rekurencyjnych. Rozpatrywano modele teoretyczne bazujące na wyjściu z SSN, lub pożądanym potencjałom neuronów sieci, jako wielkości obserwowalnych w modelach. Równanie procesu zmiany wag SSN przyjęto w postaci liniowej.

W aspekcie teoretycznym pracy przedstawiono:

- ogólne wiadomości o SSN, z uwzględnieniem:
 - powiązań SSN z rzeczywistymi systemami nerwowymi istot żywych,
 - głównych cech przetwarzania danych, przy pomocy SSN,
 - omówienia budowy warstwowych SSN jednokierunkowych i rekurencyjnych oraz metod ich uczenia, z którymi porównywana jest metoda uczenia SSN, omawiania w niniejszej pracy,
 - przedstawienie idei SSN, jako systemu dynamicznego oraz charakteryzacji procesu uczenia SSN, na gruncie statystyki matematycznej,
- filtr Kalmana, jako metodę analizy dyskretnych procesów stochastycznych, z uwzględnieniem:
 - definicji procesu filtracji,
 - modeli liniowych oraz nieliniowych (rozszerzonych),
 - wyprowadzenie równań, realizujących algorytm Kalmana filtracji dyskretnego procesu stochastycznego, w modelu liniowym oraz nieliniowym,

- metodę uczenia SSN, opartą o Filtr Kalmana, w zakresie:
 - algorytmu liniowego, Linear Kalman Filter (LKF),
 - algorytmu nieliniowego, Extended Kalman Filter (EKF),
 - modeli rozprzężonych (algorytmów typu Decoupled),
 - doboru parametrów w modelach teoretycznych dla procesu uczenia SSN.

1.2. Spis głównych oznaczeń

Symbole (zmienne):

- \mathbf{x} wektor wejścia do SSN, $\mathbf{x}=\{x_1, \dots, x_N\}$
- \mathbf{y} wektor wyjścia z SSN, $\mathbf{y}=\{y_1, \dots, y_M\}$
- F funkcja aktywacji neuronów
- R^W przestrzeń wag uogólnionych SSN (tzn. wag połączeń i biasów)
- \mathbf{w} wektor wag SSN, $\mathbf{w}=\{w_1, \dots, w_W\}$
- \mathbf{w}^1 wektor wag połączeń pierwszej warstwy SSN
- \mathbf{w}^2 wektor wag drugiej połączeń warstwy SSN
- $b=-\theta$ bias (ujemna wartość progowa)
- x^p p -ty wzorzec uczący dla SSN
- \hat{y}^p odpowiedź SSN na p -ty wzorzec uczący,
- y^p p -ty znany wzorzec wyjściowy dla SSN, odpowiadający p -temu wzorcowi uczącemu
- $L = \{(x, y)^p\}_{p=1}^L$ zbiór uczący
- $T = \{(x, y)^p\}_{p=1}^T$ zbiór testujący
- h funkcja reprezentująca strukturę SSN
- \mathbf{E} wartość oczekiwana zmiennej losowej X
- $D^2(X)$ wariancja zmiennej losowej X
- $P(X)$ rozkład zmiennej losowej X
- E funkcja błędu SSN
- ∇ gradient funkcji

- H Hessian funkcji
- Q, R charakterystyki białego szumu w modelach filtru Kalmana
- \mathbf{K} macierz wzmocnienia Kalmana
- \mathbf{P} macierz przybliżonych błędów filtracji
- k – krok czasu dyskretnego

Użyte skróty (akronimy):

- MSE błąd średniokwadratowy SSN
- FLNN Feed-forward Layered Neural Network
- RLNN Reccurent Layered Neural Network

2. Sztuczne Sieci Neuronowe

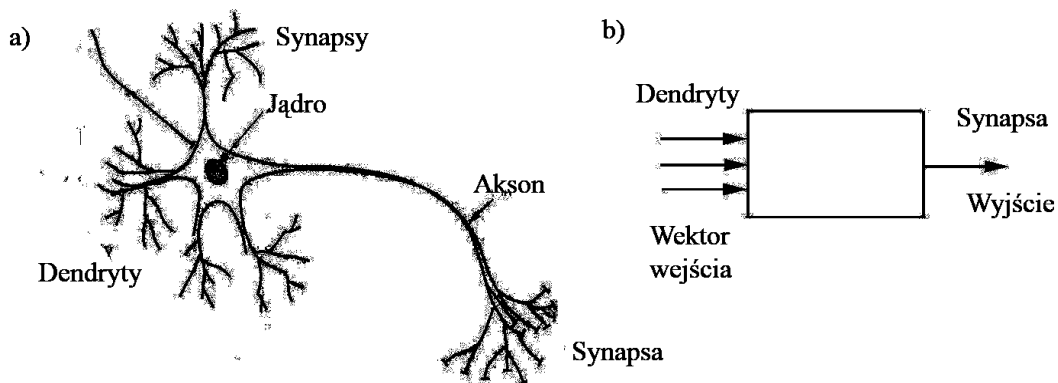
2.1. Ogólne wiadomości o SSN

Sztuczne sieci neuronowe (SSN) to algorytmy, napisane w jednym z języków programowania, których działanie inspirowane jest obserwacją natury i sposobu przesyłania informacji w systemie nerwowym istot żywych. Ich powstanie odzwierciedla próby stworzenia programów o podobnym do biologicznego odpowiednika efektywności przetwarzania i gromadzenia informacji.

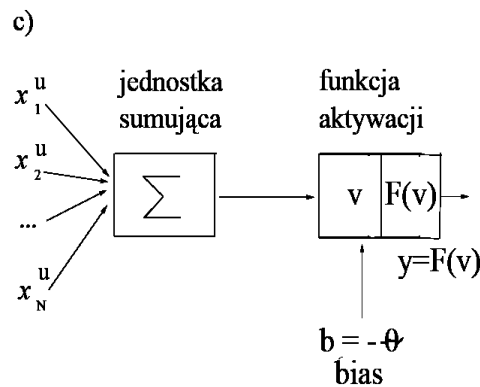
Rozwiązywanie problemów i przechowywanie danych za pomocą SSN różni się znacznie od rozwiązywania problemów konwencjonalnymi metodami obliczeniowymi (KMO). Do głównych różnic tych podejść zaliczyć można:

- faza przygotowawcza: SSN – wybór architektury, metody uczenia, wartości początkowych wag, przygotowanie wzorców uczących, KMO – opracowanie programu komputerowego;
- przetwarzanie danych: SSN – uczenie poprzez zmianę wartości wag sieci, KMO – wykorzystanie programu;
- przechowywanie informacji: SSN – wartości wag sieci, KMO – pamięć komputera;
- cel: znalezienie wielkości określonych przebiegiem programu – wykorzystanie zdolności generalizacyjnych SSN;
- wrażliwość na błędy: SSN – niewielka, KMO duża, por.[1].

2.2.1. Inspiracja biologiczna przy tworzeniu SSN – neuron biologiczny a neuron sztuczny



Rys.2.1.a) Schemat biologicznego neuronu, b) Sztuczny neuron



Rys.2.1.c Schemat sztucznego neuronu

Podstawowym elementem systemu nerwowego żywej istoty, jak i jego sztucznych odpowiedników, jest jednostka zwana neuronem, por. rys. 2.1.a, b. Zaprogramowane na wzór komórek nerwowych elementy, połączone ze sobą, tworzą układ, który nazywać dalej będziemy Sztuczna Siecią Neuronową. Podobnie jak komórki rzeczywiste, elementy SSN mają odpowiedniki synaps, czyli wielu połączeń dochodzących do komórki oraz aksonu – będącego jedynym połączeniem, którym sygnał opuszcza neuron. Za zróżnicowanie pobudzenia, które wywołane jest tym samym sygnałem docierającym do neuronu za pośrednictwem różnych synaps, odpowiada w systemie komputerowym przyjęcie różnych wag dla poszczególnych połączeń wewnątrz sieci. Przyjęcie wag SSN jako liczb rzeczywistych stanowi odpowiednik synaps pobudzających (liczby dodatnie) i hamujących przewodzenie sygnału (liczby ujemne). Rzeczywista komórka nerwowa zostaje pobudzona, jeśli jej równowaga elektryczna, wyrażona sumą pobudzeń i hamowań, przekracza pewien poziom. Wtedy impuls nerwowy jest przekazywany przez akson na zewnątrz. Podobnie odbywa się to w SSN: sygnały dochodzące do neuronu są sumowane z wagami przypisanymi określonym połączeniom, jako potencjał neuronu. Rolę progu aktywacji spełnia czynnik stały tzw. bias, natomiast za poziom sygnału wychodzącego z neuronu odpowiada jego funkcja aktywacji [2].

2.1.2. Główne cechy przetwarzania danych za pomocą SSN

Opierając się na zestawieniu, przeprowadzonym w [3], można wyliczyć następujące cechy przetwarzania przez SSN:

- adaptacja do danych: SSN mają zdolność do adaptacji wartości swoich wag w zależności od zmieniającego się środowiska (zbioru uczącego przykładów),
- zdolności generalizacyjne SSN: prawidłowo nauczona sieć ma zdolność do rozszerzania zbioru przykładów/wzorców, na których potrafi działać efektywnie, poza te ze zbioru uczącego,

- niejawny charakter modelowania: możliwość odtworzenia środowiska bez konieczności tworzenia jego modelu matematycznego,
- odwzorowanie wejście-wyjście: podczas uczenia nadzorowanego SSN przetwarza dane podawane jej na wejście w dane wyjściowe, gromadząc informację o wzajemnej relacji między nimi za pomocą wartości wag,
- niska wrażliwość na zakłócenia, dotycząca zarówno imperfekcji w zbiorze przetwarzanych danych, jak i np. uszkodzeń w strukturze sieci,
- nieliniowość: każdy neuron SSN może przetwarzać sygnał w sposób nieliniowy,
- rozproszony charakter przetwarzania i przechowywania informacji,
- możliwość autoasocjacji danych: SSN ma zdolność rekonstrukcji prawidłowego stanu podanego na wejściu "uszkodzonego" środowiska,
- możliwość klasyfikacji danych: przypisywanie danych do odpowiednich klas elementów zbioru wejściowego,
- relatywnie krótki czas obliczeń,
- możliwość wykorzystania raz nauczonej sieci jako "czarnej skrzynki", działającej na podanych jej przykładach, bez konieczności posiadania wiedzy o jej strukturze wewnętrznej,
- cechy "inteligentnego" przetwarzania informacji, takie jak zastosowanie zdobytej wiedzy, celem rozwiązania problemów nie tożsamyh z już znanymi, czy zdobywanie wiedzy na podstawie doświadczania środowiska zewnętrznego, tj. bodźców płynących z poza SSN.

2.2. Jednokierunkowe i rekurencyjne warstwowe SSN oraz podstawowe metody ich uczenia

2.2.1. Budowa neuronu SSN

Sztuczny neuron zbudowany jest z następujących elementów, por.rys. 2.1c:

- zbiór połączeń dochodzących do neuronu (z każdym z nich skojarzona jest odpowiednia waga SSN: w_{ij}) oraz jednego połączenia wyjściowego
- jednostka sumująca sygnały dochodzące do neuronu, ważone wagami odpowiadającymi określonym połączeniom, daje część potencjału neuronu u
- jednostka progowa – najczęściej dodaje się w tym celu jedno połączenie o sygnale stale równym 1 (tzw. bias b) o wadze w_0 celem otrzymania potencjału $v = u + b$

- funkcja aktywacji F , przekształcająca potencjał w wyjście neuronu, y .

Niech $x = \{x_1, \dots, x_N\}$ będzie wektorem wejścia neuronu, $b = -\theta = w_0$ jest biasem (czynnikiem stałym). Potencjał neuronu wyrażony jest związkiem:

$$v = u + b = \sum_{j=1}^N w_j x_j + b = \sum_{j=0}^N w_j x_j \quad (2.1)$$

natomiast sygnał wychodzący z neuronu przyjmuje wartość:

$$y = F(v). \quad (2.2)$$

2.2.2. Podstawowe typy neuronów w SSN warstwowych

Wyróżnia się najczęściej neurony sigmoidalne, dla których F jest funkcją ciągłą

a) sigmoidą unipolarną

$$F(\phi) = \frac{1}{1 + e^{-a\phi}}, \quad (2.3)$$

b) sigmoidą bipolarną

$$F(\phi) = \tanh(\phi/2) = \frac{1 - e^{-a\phi}}{1 + e^{-a\phi}}, \quad (2.4)$$

oraz neurony liniowe, których funkcja aktywacji jest

c) liniowa

$$F(\phi) = a\phi. \quad (2.5)$$

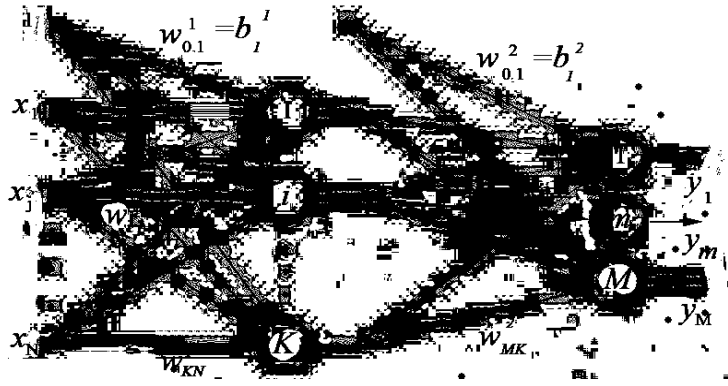
W powyższych wzorach $a > 1 \in R$ przyjmuje się najczęściej równe 1, co w przypadku neuronów liniowych daje funkcję identyczności.

2.2.3. Budowa warstwowych SSN.

Spośród wielu rodzajów SSN, wyróżnimy tutaj sieci warstwowe, a w ich obrębie: sieci z transmisją przebiegającą jedynie od wejścia do wyjścia (SSN jednokierunkowe), oraz sieci rekurencyjne, dla których następuje przekazywanie sygnału także w kierunku od wyjścia, do wejścia SSN.

2.2.3.1. Sieć wielowarstwowa, jednokierunkowa.

SSN wielowarstwowa, oprócz neuronów wejściowych i warstwy wyjściowej, ma warstwy ukryte. Neurony warstw ukrytych mają połączenia jedynie z neuronami wejściowymi i warstwy kolejnej (w przypadku pierwszej warstwy ukrytej), lub neuronami warstw poprzednich i następnych (w przypadku dalszych warstw ukrytych), por. rys. 2.3. Dla przejrzystości przyjmijmy, że jest jedna warstwa ukryta.



Rys.2.3. Schemat SSN jedнокierunkowej, dwuwarstwowej jedna warstwa ukryta

Zwykle przyjmuje się taką samą funkcję aktywacji neuronów w obrębie warstwy.

Niech N – oznacza liczbę neuronów wejściowych, M – liczbę neuronów wyjściowych. Przyjmując liczbę neuronów w warstwie ukrytej równą K , w odpowiedzi na wektor wejściowy $\mathbf{x} = \{x_1, \dots, x_N\}$, dla sieci o powyższej strukturze, otrzymujemy wartość sygnału wychodzącego z m -tego ($m = 1, \dots, M$) wyjścia SSN w postaci:

$$y_m = F^2([1, x_1^u, \dots, x_K^u][b_m^2, w_{m,1}^2, \dots, w_{m,K}^2]^T) = F^2\left(\sum_{i=0}^K x_i^u w_{m,i}^2\right) \quad (2.6)$$

przy czym

$$x_i^u = F^1([1, x_1, \dots, x_N][b_i^1, w_{i,1}^1, \dots, w_{i,N}^1]^T) = F^1\left(\sum_{j=0}^N x_j w_{i,j}^1\right) \quad (2.7)$$

czyli ostatecznie:

$$y_m = F^2\left(\sum_{i=0}^K F^1\left(\sum_{j=0}^N x_j w_{i,j}^1\right) w_{m,i}^2\right) \quad (2.8)$$

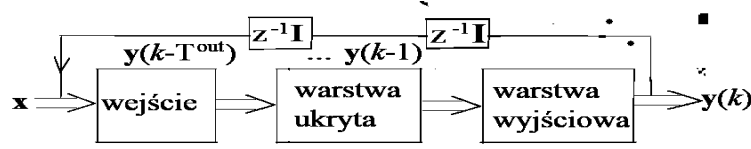
gdzie F^2, F^1 – oznaczają funkcje aktywacji neuronów warstwy 2 oraz 1, $w_{i,m}^2$ – oznacza wagę dla połączenia i -tego ($i = 1, 2, \dots, K$) neuronu warstwy ukrytej z m -tym neuronem warstwy wyjściowej ($m = 1, 2, \dots, M$), $w_{i,j}^1$ – oznacza wagę dla połączenia j -tego ($j = 1, 2, \dots, N$) neuronu wejściowego z k -tym neuronem warstwy ukrytej. Wektor $[x_1^u, \dots, x_K^u]$ złożony jest z sygnałów wychodzących z K -neuronów warstwy ukrytej. Wielkości $w_{m,0}^1 = b_m^1 = -\theta_m^1$ oraz $w_{i,0}^2 = b_i^2 = -\theta_i^2$ są wagami sygnałów od biasów warstwy ukrytej oraz wyjściowej.

2.2.3.2. Sieci warstwowe rekurencyjne.

Omówione poniżej sieci opierają się na przedstawionych wcześniej strukturach, powstając przez dodanie sprzężeń zwrotnych z opóźnieniem czasowym.

Sieć z rekurencją w obrębie wyjście-wejście

Sieć ta w literaturze oznacza jest RMLP (Reccurent Multulayer Perceptron) [3], lub Recurrent Back Propagation Network [4]. Powstaje ona z sieci jednokierunkowej, przez dodanie sprzężenia zwrotnego z opóźnieniem czasowym jednego lub więcej kroków, między warstwą wyjściową, a wejściem sieci. W ten sposób wektor wejściowy sieci zostaje rozszerzony o sygnał płynący nie "z zewnątrz" środowiska, lecz z samej SSN. Struktura takiej sieci przedstawiona jest na rys. 2.4.



Rys.2.4. Schemat sieci z rekurencją w obrębie wyjście-wejście

Przyjmijmy indeksację kroku czasowego $k = 1, 2, \dots$. Zachowując oznaczenia dotyczące sieci jednokierunkowych, dla wektora wejściowego związanego z sygnałem spoza SSN: $[x_1, \dots, x_N]$ oraz $T^{out} \in \{1, 2, \dots\}$ liczby jednostek opóźniających dla sygnału wyjściowego otrzymujemy zbiorczy wektor wejścia do SSN postaci:

$$\mathbf{x} = [x_1, \dots, x_N, y(k-1), \dots, y(k-T^{out})] \quad (2.9)$$

Wartość sygnału wychodzącego wyraża się wtedy jako:

$$y = F^2\left(\sum_{i=0}^K x_i^u w_i^2\right) \quad (2.10)$$

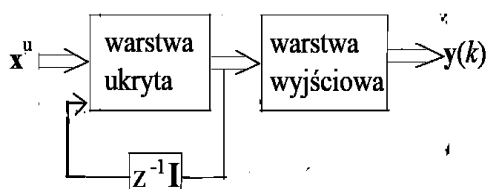
gdzie sygnał wychodzący z k -tego neuronu warstwy ukrytej ma postać:

$$x_i^u = F^1\left([1, x_1, \dots, x_N, y(k-1), \dots, y(k-T^{out})] * [b_i, w_{i,1}^1, \dots, w_{i,N}^1 + T^{out}]^T\right) = F^1\left(\sum_{j=0}^N +T^{out} x_j w_{i,j}^1\right) \quad (2.11)$$

W niniejszej pracy, obie sieci o powyższych strukturach, zostały oznaczone od sposobu przekazywania sygnału wewnątrz sieci (z pominięciem wejścia), które w obu przypadkach jest typu jednokierunkowego. Przyjęto oznaczenie FLNN (Feed-forward Layered Neural Network). To czy dana sieć zawiera element rekurencyjny, łączący wyjście z wejściem, jest jednoznacznie określone poprzez podanie elementów odpowiednich składowych wektora wejścia.

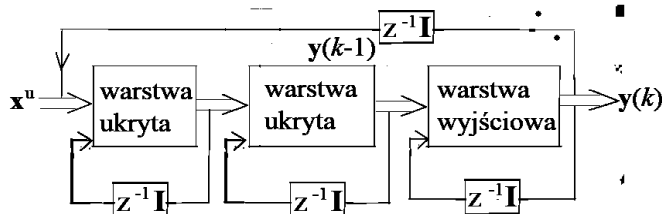
Sieć z rekurencją w obrębie poszczególnych neuronów

Dla sieci tego typu, rekurencja występuje w obrębie neuronów warstw ukrytych: sygnał wyjściowy z danego neuronu jest z opóźnieniem czasowym jednego kroku rekurencyjnie zwracany do neuronów tej samej warstwy. Warstwę wyjściową tworzą neurony nie posiadające sprzężeń zwrotnych. Ponownie dla ustalenia uwagi przyjęto sieć o jednej warstwie ukrytej i jednym neuronie wyjściowym. Sieć taka określana jest jako rekurencyjna, typu Elmana, [2]. Schemat przedstawiono na rys.2.5.



Rys.2.5. Schemat SSN z rekurencją w obrębie poszczególnych neuronów, typu Elmana

Niech $\mathbf{v} = \{x_1^u(k-1), \dots, x_K^u(k-1)\}$ oznacza wektor sygnałów wyjściowych ze wszystkich neuronów warstwy ukrytej, opóźniony o jeden krok czasowy w stosunku do kroku k .



Rys.2.6. Schemat sieci RLNN z wejściem autoregresyjnym

Wektor wejściowy dla i -tego neuronu warstwy ukrytej może mieć w kroku czasowym k postać:

$$[x_1, \dots, x_{N+K}] = [x_1, x_2, \dots, x_N, x_1^u(k-1), \dots, x_K^u(k-1)] \quad (2.12)$$

gdzie N oznacza liczbę neuronów wejściowych, K – jest liczbą neuronów w warstwie ukrytej. $x_i^u(k-1)$ oznacza wartość sygnału wyjściowego z i -tego neuronu warstwy ukrytej w kroku czasowym $k-1$, dla $i = 1, 2, \dots, K$.

Dla sieci o takiej strukturze otrzymujemy wartość sygnału wychodzącego postaci:

$$y = F^2\left(\sum_{i=0}^K x_i^u w_i^2\right) \quad (2.13)$$

gdzie sygnał wychodzący z i -tego neuronu warstwy ukrytej ma postać:

$$x_i^u = F^1([x_1, \dots, x_N, x_1^u(k-1), \dots, x_K^u(k-1)]^T * [b_i, w_{i,1}^1, \dots, w_{i,N+K}^1]) \quad (2.14)$$

W niniejszej pracy, zastosowano strukturę SSN Elmana w połączeniu ze sprzężeniem zwrotnym, także w obrębie warstwy wyjściowej, por. rys. 2.6. Sieci o takiej architekturze zostały oznaczone RLNN (Recurrent Layered Neural Network), by podkreślić rekurencyjne połączenia w obrębie neuronów. Sieć RLNN może mieć również połączenie rekurencyjne warstwy wyjściowej, z wejściem. Najczęściej stosowano jeden krok czasowy opóźnienia, czyli wyjście w sposób autoregresyjny było przekazywane do wejścia SSN.

2.2.4. Proces uczenia i testowania SSN

Jedną z najbardziej istotnych cech SSN jest ich możliwość uczenia i poprawy efektywności działania, jako efektu przyswajania wiedzy ze środowiska. Poprawa efektywności sieci jest procesem odbywającym się w czasie. Przyjmuje się następującą definicję procesu uczenia SSN, [3]:

Proces uczenia SSN polega na adaptacji parametrów sieci (wag neuronów i biasów) w procesie stymulacji przez środowisko, w którym sieć jest osadzona. Proces uczenia jest zdeterminowany przez wyznaczony sposób zmiany parametrów sieci.

W niniejszej pracy została wykorzystana metoda uczenia nadzorowanego, zakładająca istnienie "nauczyciela" sieci, który posiada wiedzę o środowisku zewnętrznym. Jest ono reprezentowane przez pewien podzbiór par elementów przykładów (wzorców) o strukturze:

- pierwszym elementem pary jest wektor wejściowy (wektor pobudzenia) dla sieci $\mathbf{x}^p \in \Omega_{input} \subset R^N$
- drugim elementem pary jest znany wzorec wyjściowy $\mathbf{y}^p \in \Omega_{target} \subset R^l$ (wektor odpowiedzi na pobudzenie, *ang. target vector*), będący zgodną ze środowiskiem zewnętrznym sieci, prawidłową odpowiedzią na powyższy wektor wejściowy \mathbf{x}^p , gdzie p przebiega liczbę dostępnych wzorców P .

Powyższy zbiór wzorców zostaje podzielony na zbiór wzorców uczących

$$L = \{(\mathbf{x}, \mathbf{y})^p\}_{p=1}^L \quad (2.15)$$

i rozłączny z nim zbiór wzorców testujących

$$T = \{(\mathbf{x}, \mathbf{y})^p\}_{p=1}^T \quad (2.16)$$

dla SSN. Na podstawie elementów zbioru uczącego, następuje korekta wag sieci. Wagi są dobierane w taki sposób, aby na podstawie błędu dla danego wzorca, definiowanego jako pewna miara różnicy między odpowiedzią sieci na dany wzorec uczący, a odpowiadającym mu znanym wyjściem, podanym przez nauczyciela, zminimalizować zadaną funkcję błędu sieci.

Najczęściej wykorzystuje się przyrostową formę zmiany wag SSN. Jeśli $\mathbf{w}_k \in R^W$ oznacza wartość wektora wag sieci w chwili k , wtedy wartość wektora \mathbf{w} w czasie $k + 1$ jest przyjmowana w postaci:

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \Delta \mathbf{w}_k \quad (2.17)$$

gdzie $\Delta \mathbf{w}_k$ zależy od przyjętego sposobu uczenia SSN oraz przyjętej miary błędu sieci przy przetwarzaniu bierzącego lub większej ilości wzorców uczących. (Schemat, otrzymywania wartości \mathbf{w}_{k+1} na podstawie \mathbf{w}_k nazywamy algorytmem uczenia SSN.)

Błąd uzyskany dla elementów zbioru uczącego nazywamy błędem uczenia SSN.

Sieć nauczona jest testowana następnie na zbiorze testującym T . Mierzony jest poziom przyjętego błędu odtworzenia przez sieć elementów tego zbioru. Błąd uzyskany dla elementów zbioru testującego nazywamy błędem testowania SSN.

Zdolność poprawnej (zgodnej z rzeczywistym środowiskiem zewnętrznym) reakcji sieci na wzorce nie należące do zbioru uczącego nazywamy zdolnością generalizacji SSN.

Z procesem uczenia SSN związane są dwa możliwe do wystąpienia, niekorzystne zjawiska:

- niedoszacowanie, '*underfitting*' – nie osiągnięcie przez SSN wystarczającej efektywności uczenia (spowodowanej np. jej zbyt małymi rozmiarami w stosunku do złożoności środowiska, lub pewnych niedostatków w ramach samego algorytmu uczącego), wyrażająca się dużymi błędami uczenia i testowania
- ponadprzystosowanie, '*overfitting*' – zjawisko prowadzące do bardzo dobrej adaptacji do zbioru uczącego, z równoczesną, bardzo słabą generalizacją na zbiór testujący.

2.2.5. SSN jako dyskretny układ dynamiczny

Poniżej sformalizowany zostanie model środowiska, w którym osadzona jest SSN, proces podawania sieci informacji oraz przedstawiona koncepcja SSN jako dyskretnego modelu dynamicznego, [5].

Niech $T \subset \mathbb{R}_+ \cup 0$ będzie zbiorem dyskretnym, rozumianym jako chwile czasowe, $[k_0, k_f] \subset T$, gdzie k_0 – czas początkowy, k_f – czas końcowy (finalny), $k_0 < k_f$, niech Ω_Z będzie pewnym zbiorem zdarzeń.

Definicja.2.1. Funkcję $u : [k_0, k_f] \rightarrow \Omega_Z$ nazywamy funkcją historii zdarzeń generowanych ze zbioru T w Ω_Z .

Definicja.2.2. Każdy podzbiór Z zbioru $\{u$ takich, że u jest funkcją historii zdarzeń generowanych ze zbioru T w $\Omega_Z\}$ nazywamy środowiskiem zdarzeń ze zbioru T w Ω_Z .

Definicja.2.3. Niech $\Omega \subset \mathbb{R}$, niech Z będzie środowiskiem zdarzeń, ze zbioru T w Ω_Z . Dyskretnym systemem dynamicznym nazywamy funkcję $\Psi : \Omega \times T \times T \times Z \rightarrow \Omega$ taką że:

1. stan końcowy w czasie t_f :

$$x_f = \Psi(x_0, t_0, t_f, \eta) \quad (2.18)$$

jest determinowany przez pewien stan początkowy $x_0 = \Psi(x_0, t_0, t_0, \eta)$, w którym system się znajduje w czasie t_0 i pewną funkcję historii zdarzeń $\eta : [t_0, t_f] \rightarrow \Omega_Z) \in Z$, przy czym $\forall t_0 \in T \forall x_0 \in \Omega \forall \eta \in Z : x_0 = \Psi(x_0, t_0, t_0, \eta)$

2. $\forall x_a \in \Omega \forall t_a, t_b, t_c \in T$ takich że $t_a < t_b < t_c$ zachodzi równość

$$\Psi(\Psi(x_a, t_a, t_b, \eta_a), t_b, t_c, \eta_b) = \Psi(x_a, t_a, t_c, \eta_c) \quad (2.19)$$

gdzie

$$(\eta_a : [t_a, t_b] \rightarrow \Omega_Z) \in Z \quad (2.20)$$

$$(\eta_b : [t_b, t_c] \rightarrow \Omega_Z) \in Z$$

$$(\eta_c : [t_a, t_c] \rightarrow \Omega_Z) \in Z$$

$\eta_c = \eta_a$ na $[t_a, t_b)$ oraz $\eta_c = \eta_b$ na $[t_b, t_c)$.

Zbiór Ω nazywamy zbiorem stanów systemu dynamicznego, zbiór Ω_Z zbiorem zdarzeń podawania wzorców.

Dla SSN zbiór stanów systemu to $\Omega = \Omega_{hidden} \times \Omega_{output} \times \mathbb{R}^W$, gdzie: Ω_{hidden} jest zbiorem stanów pamięci ukrytej (niepustym dla sieci rekurencyjnych), Ω_{output} jest zbiorem sygnałów wyjściowych z SSN, \mathbb{R}^W jest zbiorem wag SSN. Środowisko zdarzeń przyjmuje postać: $Z = Z_{input} \times Z_{target}$, gdzie: Z_{input} jest środowiskiem zdarzeń ze zbioru T w zbiór Ω_{input} - wzorców podawanych sieci na wejście, Z_{target} jest środowiskiem zdarzeń ze zbioru T w zbiór Ω_{target} - znanych wzorców uczących dla sieci. Funkcje historii zdarzeń wyznaczają kolejność prezentacji elementów zbioru wzorców

podawanych na wejście oraz wprowadzania wiedzy o wzorcach będących znanymi odpowiedziami SSN na wzorce wejściowe.

Warunek 1. definicji 2.3. mówi, że historia stanów SSN (czyli wartości jej wag, sygnałów wyjściowych z poszczególnych neuronów oraz dla sieci niejednokierunkowych zbioru stanów pamięci ukrytej) wyznaczona jest jednoznacznie przez stan początkowy oraz funkcję historii podawania sieci wzorców. Warunek 2. oznacza, iż bieżący stan SSN jednoznacznie wyznacza jej stan przyszły.

2.2.6. Uczenie nadzorowane układu dynamicznego

Interpretując SSN jako układ dynamiczny, proces uczenia polega na wyznaczeniu stanów SSN w określonym czasie k , na podstawie funkcji historii zdarzeń generowanych przez wzorce \mathbf{x}^p ze zbioru Ω_{input} elementów, podawanych sieci na wejście oraz odpowiadające im wzorce \mathbf{y}^p ze zbioru Ω_{target} będące podawanymi przez nauczyciela wzorcami znanymi, pochodzącymi ze zbioru uczącego L [5].

Celem uczenia jest taki dobór wartości wag SSN, a więc elementów zbioru R^W , aby były one najlepsze z punktu widzenia określonej miary błędu (efektywności uczenia) $E : R^W \rightarrow R_+ \cup 0$, której wartości zależą od pewnego zbioru elementów zbioru uczącego.

Przykładem funkcji błędu może być najczęściej stosowany błąd średniokwadratowy, o postaci:

$$\begin{aligned} E(\mathbf{w}) &= MSE(\{(\mathbf{x}, \mathbf{y})^p\}_{p=1}^L, \mathbf{w}) := \frac{1}{L} \sum_{p=1}^L (\mathbf{y}^p - \hat{\mathbf{y}}^p)^2 \\ &= \frac{1}{L} \sum_{p=1}^L (\mathbf{y}^p - \mathbf{h}(\mathbf{x}^p, \mathbf{w}))^2 \end{aligned} \quad (2.21)$$

gdzie $\mathbf{y}^p \in \Omega_{target} \subset R^M$ jest znanym wzorcem, będącym prawidłową odpowiedzią SSN na wzorec wejściowy $\mathbf{x}^p \in \Omega_{input} \subset R^N$, $\hat{\mathbf{y}}^p \in \Omega_{output} \subset R^M$ jest odpowiedzią SSN na wzorec wejściowy \mathbf{x}^p , zależną od wektora wag sieci \mathbf{w} . L oznacza liczbę wzorców w zbiorze uczącym.

2.2.7. Charakteryzacja procesu uczenia SSN na gruncie statystyki matematycznej.

Poniżej przedstawiona zostanie idea formalnego uzasadnienia, iż uczenie SSN w oparciu o algorytm minimalizujący funkcję błędu uczenia na wybranej próbce "środowiska" powoduje, przy odpowiedniej liczności próbki uczącej, poprawne modelowanie niejawne przez SSN całego "środowiska".

Niech elementy zbioru uczącego podawane sieci na wejście: \mathbf{x}^p , będą próbką z rozkładu zmiennej losowej \mathbf{X} , przyjmującej wartości w zbiorze R^N (reprezentującej środowisko,

w którym osadzona jest SSN), przy czym rozkład prawdopodobieństwa zmiennej \mathbf{X} jest nieznaną, [3]. Niech zmienna losowa \mathbf{Y} , przyjmująca wartości w zbiorze R^M , będzie zależna od zmiennej losowej \mathbf{X} oraz niech $\mathbf{Y} = [\mathbf{y}^1, \mathbf{y}^2, \dots, \mathbf{y}^P]$ będzie próbką z rozkładu zmiennej losowej Y , przy czym:

$$\mathbf{Y} = \mathbf{g}(\mathbf{X}) + \varepsilon \quad (2.22)$$

gdzie \mathbf{g} jest pewną nieznaną funkcją. ε jest zmienną losową o wartości oczekiwanej warunkowej

$$\mathbf{E}(\varepsilon|\mathbf{X}) = 0 \quad (2.23)$$

reprezentującą na przykład błąd urządzenia pomiarowego odczytującego wartości \mathbf{y} lub zaszumienie wzorców. Model ten zostaje przybliżony poprzez model neuronowy postaci:

$$\mathbf{Y} = \mathbf{h}(\mathbf{X}) + \varepsilon \quad (2.24)$$

gdzie \mathbf{h} reprezentuje strukturę SSN.

Niech wektor wyjściowy z sieci, w odpowiedzi na wzorzec wejściowy x^p , ma postać: $\hat{y}^p = h(x^p, \mathbf{w})$, gdzie \mathbf{w} jest wektorem wag SSN.

Miarą dopasowania SSN do modelu probabilistycznego jest zmienna losowa postaci:

$$E(\mathbf{w}) = \|\mathbf{Y} - \mathbf{h}(\mathbf{X}, \mathbf{w})\|^2 \quad (2.25)$$

gdzie $\|\cdot\|$ oznacza normę euklidesową w R^W . Wartość oczekiwana tak zdefiniowanej zmiennej losowej wynosi:

$$\mathbf{E}(E(\mathbf{w})) = \int \|\mathbf{Y} - \mathbf{h}(\mathbf{X}, \mathbf{w})\|^2 dP(\mathbf{X}, \mathbf{Y}) \quad (2.26)$$

gdzie \int rozumiana jest jako całka Riemmana - Stieltjesa [6], $P(\mathbf{X}, \mathbf{Y})$ jest rozkładem łącznym zmiennej losowej (\mathbf{X}, \mathbf{Y}) . By wyznaczyć SSN, dla której dopasowanie do modelu pobabilistycznego jest najlepsze, należy optymalizować funkcję $\mathbf{E}(E(\mathbf{w}))$ względem zmiennej \mathbf{w} . Wyznaczenie powyższej wartości oczekiwanej jest jednak trudne, ze względu na fakt iż: $P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{Y}|\mathbf{X})P(\mathbf{X})$ gdzie $P(\mathbf{Y}|\mathbf{X})$ jest warunkowym rozkładem zmiennej losowej \mathbf{Y} , [7], przy danym \mathbf{X} , $P(\mathbf{X})$ - oznacza rozkład zmiennej losowej \mathbf{X} , który nie jest znany.

Według [8] przy odpowiednio dużej liczności L zbioru wzorców uczących, problem wyznaczenia \mathbf{w} wektora minimalizującego funkcję $\mathbf{E}(E(\mathbf{w}))$:

$$\mathbf{w} \rightarrow \int \|\mathbf{Y} - \mathbf{h}(\mathbf{X}, \mathbf{w})\|^2 dP(\mathbf{X}, \mathbf{Y}) \quad (2.27)$$

można zamienić na problem wyznaczenia wektora \mathbf{w} minimalizującego funkcję:

$$\mathbf{w} \rightarrow \frac{1}{L} \sum_{p=1}^L (\mathbf{h}(\mathbf{x}^p, \mathbf{w}) - \mathbf{y}^p)^2 = \frac{1}{P} \sum_{p=1}^L (\hat{\mathbf{y}}^p - \mathbf{y}^p)^2 = MSE(\mathbf{w}) \quad (2.28)$$

w przestrzeni R^W wag SSN.

Fakt ten wynika między innymi z tego, iż $MSE(\mathbf{w})$ jest estymatorem z próby dla wartości oczekiwanej zmiennej $E(\mathbf{w})$. Funkcja $MSE(\mathbf{w})$ nie zależy od nieznanego rozkładu prawdopodobieństwa, a jedynie znanych danych i struktury SSN, jej minimalizację można więc przeprowadzić. Znaleziony w ten sposób wektor wag sieci daje SSN, rozwiązującą problem poszukiwania związku pomiędzy \mathbf{Y} oraz \mathbf{X} .

2.2.8. Dylemat bias-wariancja

Podczas uczenia SSN jednym z problemów do rozwiązania jest tzn. dylemat bias-wariancja [3]. Wartości wag SSN, nauczonej w oparciu o wybrany zbiór wzorców uczących L , zależne są od elementów tego zbioru. Więc $\mathbf{h}(\mathbf{X}, \mathbf{w}) = \mathbf{h}(\mathbf{X}, \mathbf{w}, L)$ gdzie \mathbf{h} jest funkcją reprezentującą SSN o wektorze wag \mathbf{w} .

Niech $(\mathbf{g} - \mathbf{h}(\mathbf{X}, \mathbf{w}, L))^2$, por. (2.25), oznacza miarę efektywności dopasowania SSN do danego modelu środowiska, opisanego równaniem (2.22). Można wykazać, iż por. [3]:

$$\mathbf{E}(\mathbf{g} - \mathbf{h}(\mathbf{X}, \mathbf{w}, L))^2 = (\mathbf{E}(\mathbf{g} - \mathbf{h}(\mathbf{X}, \mathbf{w}, L)))^2 + \mathbf{E}((\mathbf{h}(\mathbf{X}, \mathbf{w}, L) - \mathbf{E}(\mathbf{h}(\mathbf{X}, \mathbf{w}, L))))^2). \quad (2.29)$$

Pierwszy składnik powyższej sumy, reprezentuje tzw. *bias* przybliżenia funkcji \mathbf{g} poprzez SSN (w kwadracie). Oddaje on dokładność modelowania środowiska zewnętrznego, za pomocą SSN. Drugi składnik reprezentuje wariancję funkcji \mathbf{h} i zależy zarówno od architektury przyjętej SSN (liczby warstw, neuronów, struktury połączeń), która wyraża się w postaci funkcji \mathbf{h} oraz od konkretnych wartości wag \mathbf{w} , znalezionych w procesie uczenia. Ich wartości zależne są także od elementów zbioru uczącego L .

Jeżeli funkcja aproksymująca dane środowisko przy pomocy SSN, czyli $\mathbf{h}(\mathbf{X}, \mathbf{w}, L)$ jest różna od funkcji \mathbf{g} , wtedy jest ona obciążonym estymatorem funkcji \mathbf{g} (*ang. biased*) i $\mathbf{E}\mathbf{h}(\mathbf{X}, \mathbf{w}, L) \neq \mathbf{g}(\mathbf{X})$. Wtedy pierwszy składnik powyższej sumy jest różny od zera. Jeśli SSN stanowi nieobciążony estymator dla funkcji \mathbf{g} (*ang. unbiased*), pierwszy składnik sumy jest równy zero. W obu tych przypadkach drugi składnik sumy może być większy od zera, gdyż zależy tylko od struktury SSN i zbioru wzorców uczących.

Jeśli SSN jest uczona na określonym zbiorze przykładów, gdy osiągnęte są małe wartości *biasu*, równocześnie wariancja funkcji \mathbf{h} przyjmuje dużą wartość. Efekt nie wystąpiłby tylko jeśli możliwe byłoby uczenie SSN na nieskończenie liczny zbiór uczącym. Przez konstrukcję SSN o odpowiedniej architekturze, lub przeprowadzaniu odpowiednio dokładnego procesu uczenia, możliwe jest sterowanie wielkością poszczególnych składników w sumie (2.29).

Podczas procesu uczenia i testowania SSN konieczne jest osiągnięcie kompromisu, pomiędzy złożonością architektury SSN, a długością trwania uczenia. SSN, która

jest niedostatecznie złożona w stosunku do modelowanego środowiska może dać duży *bias*, SSN o zbyt dużej liczbie parametrów może prowadzić do ponadprzystosowania, dużej wartości wariancji i słabej zdolności generalizacji. SSN uczona zbyt małą liczbą epok może nie stanowić reprezentatywnego modelu zbioru uczącego. Mały błąd uzyskany dla zbioru wzorców uczących, może jednak prowadzić do dużych wartości błędu testowania. Wybór konkretnej architektury SSN oraz długości trwania uczenia, wyważenia biasu i wariancji w obrębie konkretnych zbiorów danych nierzadko podyktowany jest użytecznością nauczonej SSN i wymaganą dokładnością np. przy rozwiązywaniu konkretnych problemów inżynierskich.

2.2.9. Algorytmy uczenia SSN

Poniżej zaprezentowane zostaną najczęściej stosowane "standartowe" metody uczenia SSN, [4], por. tab. 2.1. Z nimi będzie zestawiona efektywność uczenia SSN, za pomocą omawianej w pracy metody uczenia z wykorzystaniem filtra Kalmana.

Przedstawione algorytmy należą do klasy metod wstecznej propagacji błędów (*ang. back propagation*), gdyż zakładają dwa przebiegi:

- przebieg wprost, gdy podczas przesyłu sygnału od wejścia do wyjścia SSN, znajdująca jest odpowiedź SSN na dany wzorzec uczący
- przebieg wstecz, gdy następuje uaktualnienie wartości wag sieci.

	Algorytmy gradientowe	Relaksacja
<i>metody pierwszego rzędu</i>	Wstecznej propagacji błędów off-line on-line	perturbacja wag
<i>metody adaptacyjne pierwszego rzędu</i>	globalne adaptacyjne lokalne adaptacyjne Rprop	
<i>metody drugiego rzędu</i>	QRprop metody Newtona metody pseudo-Newtona Levenberga-Marquardta	symetryczna relaksacja (pseudoinwersja) asymetryczna relaksacja

Rys.2.7. Wybrane rodzaje algorytmów uczenia, za [4]

1. Podstawowy algorytm wstecznej propogacji błędów

Algorytm opiera się na metodzie największego spadku wyznaczania minimum funkcji błędu sieci w przestrzeni wag R^W . Niech $\mathbf{w}_k = [w_1, \dots, w_W] \in R^W$ oznacza wartość wektora wag SSN w k -tym kroku uczenia. Przez E oznaczmy funkcję błędu SSN na zadanym zbiorze wzorców uczących (głównie $E = MSE$). Adaptacja wag odbywa się zgodnie ze wzorem:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta(k)\nabla E \quad (2.30)$$

gdzie

$$\nabla E = \left[\frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_W} \right]^T \Big|_{\mathbf{w}_k} \quad (2.31)$$

jest gradientem funkcji błędu, obliczonym w punkcie \mathbf{w}_k , $\eta(k)$ jest współczynnikiem uczenia.

Ta metoda uczenia, nazywana jest algorytmem uczenia największego spadku (*ang. gradient descent*). Metoda wykorzystuje rozwinięcie funkcji błędu w szereg potęgowy w sąsiedztwie punktu \mathbf{w}_k tylko z dokładnością do członu zawierającego jej pierwsze pochodne cząstkowe (tzw. algorytm pierwszego rzędu).

Zbieżność algorytmu zależy od kształtu powierzchni błędu w przestrzeni wag SSN oraz od wartości $\eta(k)$. Główną wadą metody jest fakt, iż dla niektórych wartości tego parametru proces uczenia może oscylować, być rozbieżny lub wolno zbierzy, zależnie od kształtu powierzchni błędu [3]. Algorytm nie gwarantuje znalezienia takich wag dla SSN, dla których osiągnane jest minimum globalne zadanej miary błędu.

2. Algorytm Levenberga - Marquardta

Jest to algorytm, w którym stosuje się rozwinięcie funkcji błędu SSN w sąsiedztwie \mathbf{w}_k z dokładnością do członu zawierającego jej drugie pochodne cząstkowe, [2]. Adaptacja wag odbywa się zgodnie ze wzorem:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta(k)[H(\mathbf{w}_k)]^{-1}\nabla E \quad (2.32)$$

gdzie

$$H(\mathbf{w}_k) = \left[\frac{\partial^2 E}{\partial w_i \partial w_j} \right]_{i,j=1,\dots,W} \Big|_{\mathbf{w}_k} \quad (2.33)$$

jest Hessianem, czyli macierzą drugich pochodnych cząstkowych funkcji E w punkcie \mathbf{w}_k . Jest on wyznaczany w przybliżeniu, w oparciu o Jacobian (metoda tzw. pseudo drugiego rzędu), w postaci:

$$\mathbf{H} \approx \mathbf{J}^T \mathbf{J} + \alpha(k)\mathbf{I} \quad (2.34)$$

gdzie

$$\mathbf{J} = \left[\frac{\partial \zeta_p}{\partial w_i} \right]_{i=1,\dots,W}^{p=1,\dots,P} \Big|_{\mathbf{w}_k} \quad (2.35)$$

$$\zeta_p = \widehat{\mathbf{y}}^p - \mathbf{y}^p = \mathbf{h}(\mathbf{w}_k, \mathbf{x}^p) - \mathbf{y}^p \quad (2.36)$$

dla $p = 1, 2, \dots, L$ przebiegającego elementy zbioru uczącego, przy czym parametr $\alpha(k)$ wyznacza się w oparciu o bieżący przebieg uczenia:

$$\alpha(k) = \alpha(k-1)/\beta \quad (2.37)$$

gdzie $MSE(k) \leq MSE(k-1)$

$$\alpha(k) = \alpha(k-1)\beta \quad (2.38)$$

gdzie $MSE(k) > MSE(k-1)$ gdzie $\beta > 1$. Jest on zmniejszany, po każdej redukcji wielkości błędu SSN, zwiększany, gdy jego dotychczasowa wartość podniosłaby wartość funkcji błędu sieci, [10].

W obu powyższych metodach współczynnik uczenia $\eta(k)$ wyznacza się często empirycznie dla konkretnego problemu uczenia, [4].

3. Algorytm Resilient-Propagation (RProp)

Jest to algorytm, oparty o lokalną (dla każdej z wag SSN osobno) metodę wyznaczania współczynników uczenia $\eta_i(k)$ oraz biorący pod uwagę jedynie znak składowej gradientu funkcji błędu [4].

Adaptacja i -tej wagi SSN, w_i , dla $i = 1, 2, \dots, W$, odbywa się zgodnie ze wzorem, por. [11]:

$$w_{ik+1} = w_{ik} - \eta_i(k) \operatorname{sgn}(S_i(k)) \quad (2.39)$$

gdzie $S_i(k) = \frac{\partial E}{\partial w_i} | w_i$,

$$\eta_i(k) = \min\{1.2\eta_i(k-1), \eta_{max}\} \quad (2.40)$$

jeżeli $S_i(k)S_i(k-1) > 0$

$$\eta_i(k) = \max\{0.5\eta_i(k-1), \eta_{min}\} \quad (2.41)$$

jeżeli $S_i(k)S_i(k-1) < 0$

$$\eta_i(k) = \eta_i(k) \quad (2.42)$$

w pozostałych przypadkach.

Wielkość $\eta_i(k-1)$ oznacza wartość współczynnika uczenia dla i -tej składowej wektora wag sieci w $k-1$ -wszym kroku uczenia, natomiast $\eta_{max} = 50$, $\eta_{min} = 10^{-6}$, funkcja sgn oznacza znak argumentu.

3. Filtr Kalmana jako metoda analizy dyskretnych procesów stochastycznych

Filtr Kalmana jest narzędziem do analizy ogólnych modeli przestrzeni stanów [7,12]. W modelach tych część parametrów przyjmowana jest jako losowa i nieobserwowalna, część natomiast, jako obserwowalna oraz dodatkowo zależna od pewnych nielosowych parametrów, innych wielkości losowych oraz zmiennych nieobserwowalnych.

3.1. Liniowy Filtr Kalmana

Rozważmy liniowy, dyskretny układ dynamiczny dany przez:

1. Równanie procesu \mathbf{x} :

$$\mathbf{x}_{k+1} = \mathbf{F}_{k+1,k}\mathbf{x}_k + \omega_k \quad (3.1)$$

2. Równanie obserwacyjne ze względu na \mathbf{y} :

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \nu_k \quad (3.2)$$

gdzie

$k = 1, 2, 3, \dots$ – dyskretny pseudoczas \mathbf{x}_k , \mathbf{y}_k – zmienne losowe n oraz p wymiarowe odpowiednio,

$\mathbf{F}_{k+1,k}$, \mathbf{H}_k – macierze o wymiarach $n \times n$ oraz $p \times n$, $p \leq n$, o wyrazach rzeczywistych,

ω_k, ν_k – n oraz p -wymiarowe, niezależne zmienne losowe, o rozkładzie gaussowskim i charakterystykach:

$$\mathbf{E}(\omega_k) = 0, \mathbf{E}(\nu_l) = 0 \quad (3.3)$$

$$\mathbf{E}(\omega_k \omega_l^T) = \mathbf{Q}_k \delta_{kl} \quad (3.4)$$

$$\mathbf{E}(\nu_k \nu_l^T) = \mathbf{R}_k \delta_{kl} \quad (3.5)$$

dla $l = 1, 2, \dots$, $\delta_{kl} = 1$ jeżeli $k = l$, $\delta_{kl} = 0$ jeżeli $k \neq l$,
 ω_k, ν_l są nieskorelowane, por. [13].

Niech \mathbf{x}_0 będzie zmienną losową o rozkładzie normalnym, o wartości oczekiwanej $\mathbf{E}(\mathbf{x}_0) = \bar{\mathbf{x}}_0$ i macierzy kowariancji $\mathbf{E}(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T = \Sigma_0$.

W przedstawionym modelu (3.1)-(3.2), zadanie filtracji dyskretnego procesu stochastycznego \mathbf{x} , polega na wyznaczeniu estymatora $\hat{\mathbf{x}}_k$ dla \mathbf{x}_k , minimalizującego błąd średniokwadratowy estymacji o postaci:

$$\mathbf{E}(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T \quad (3.6)$$

gdy dany jest zbiór obserwacji procesu \mathbf{y} : $\mathbf{Y}(k) = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k\}$.

Algorytm filtracji realizowany jest przez komplet równań, zaproponowanych przez R.E. Kalmana, [14], nazywanych filtrem Kalmana. Zakłada on wyznaczenie kolejno :

– estymatora wektora stanu a priori:

$$\hat{\mathbf{x}}_k^- = \mathbf{F}_{k,k-1}(\hat{\mathbf{x}}_{k-1}) \quad (3.7)$$

– macierzy kowariancji błędów estymacji a priori:

$$\mathbf{P}_k^- = \mathbf{F}_{k,k-1}\mathbf{P}_{k-1}\mathbf{F}_{k,k-1}^T + \mathbf{Q}_{k-1} \quad (3.8)$$

– macierzy wzmocnienia Kalmana:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k]^{-1} \quad (3.9)$$

– estymatora wektora stanu a posteriori, po uwzględnieniu obserwacji \mathbf{y}_k :

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k [\mathbf{y}_k - h_k(\hat{\mathbf{x}}_k^-)] \quad (3.10)$$

– macierzy kowariancji błędów estymacji a aposteriori:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (3.11)$$

By zainicjować obliczenia przyjmuje się:

$$\hat{\mathbf{x}}_0 = \mathbf{E}(\mathbf{x}_0) = \bar{\mathbf{x}}_0 \quad (3.12)$$

czyli wartość estymatora zmiennej losowej \mathbf{x}_0 równą jej wartości oczekiwanej oraz

$$\mathbf{P}_0 = \Sigma_0 \quad (3.13)$$

czyli błąd jej estymacji równy jej wariancji.

Formalne wyprowadzenie równań (3.7)-(3.12) znajduje się w Dodatku 1 niniejszej pracy.

3.2. Rozszerzony Filtr Kalmana

Rozszerzony filtr Kalmana pozwala na analizę następującego nieliniowego dyskretnego układu dynamicznego:

$$\mathbf{x}_{k+1} = \mathbf{f}_{k+1}(\mathbf{x}_k) + \omega_k \quad (3.14)$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \nu_k \quad (3.15)$$

gdzie:

$\mathbf{f}_{k+1} : R^n \rightarrow R^n$, $\mathbf{h}_k : R^n \rightarrow R^p$ – funkcje o argumentach i wartościach wektorowych, nieliniowe, nielosowe, mające ciągłe pierwsze pochodne.

Proces filtracji w modelu (3.14)-(3.15) realizowany jest przez wyznaczenie kolejno wielkości:

– estymator wektora stanu 'a priori':

$$\widehat{\mathbf{x}}_k^- = f_k(\widehat{\mathbf{x}}_{k-1}) \quad (3.16)$$

– przybliżenie macierzy kowariancji błędów estymacji a priori:

$$\mathbf{P}_k^- = \mathbf{F}_{k,k-1} \mathbf{P}_{k-1} \mathbf{F}_{k,k-1}^T + \mathbf{Q}_{k-1} \quad (3.17)$$

– macierz wzmocnienia Kalmana:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k]^{-1} \quad (3.18)$$

– estymator wektora stanu a posteriori, po uwzględnieniu obserwacji \mathbf{y}_k :

$$\widehat{\mathbf{x}}_k = \widehat{\mathbf{x}}_k^- + \mathbf{K}_k [\mathbf{y}_k - h_k(\widehat{\mathbf{x}}_k^-)] \quad (3.19)$$

– przybliżenie macierzy kowariancji błędów estymacji 'a posteriori':

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (3.20)$$

gdzie dla $f = \{f^1, \dots, f^n\}$, $\mathbf{x} = \{x^1, \dots, x^n\}$:

$$\mathbf{F}_{k+1,k} = \left[\frac{\partial f^i(\mathbf{x})}{\partial x^j} \right]_{\substack{i=1,\dots,n \\ j=1,\dots,n}} \Big|_{\mathbf{x}=\widehat{\mathbf{x}}_k} \quad (3.21)$$

oraz dla $h = \{h^1, \dots, h^p\}$

$$\mathbf{H}_k = \left[\frac{\partial h^i(\mathbf{x})}{\partial x^j} \right]_{\substack{i=1,\dots,p \\ j=1,\dots,n}} \Big|_{\mathbf{x}=\widehat{\mathbf{x}}_k} \quad (3.22)$$

są macierzami linearyzacji równania (3.14) oraz (3.15), z pominięciem reszty wzoru Taylora. Formalne wyprowadzenie równań (3.16)-(3.20) znajduje się w Dodatku 2 niniejszej pracy.

4. Filtr Kalmana jako metoda uczenia SSN

4.1. Aktualny stan zagadnienia - uczenie SSN z wykorzystaniem FK

1. Porównanie metody uczenia SSN opartej o filtr Kalmana, ze standartowymi metodami gradientowymi

W standardowych metodach nadzorowanego uczenia SSN, opartych o algorytmy gradientowe, por. rozdział 2. kolejne wartości wag SSN wyznaczone są w oparciu o numeryczne wyznaczanie minimum zadanej funkcji błędu SSN. Kontrola szybkości zmian wag sieci w przestrzeni parametrów odbywa się poprzez wprowadzanie różnego rodzaju współczynników korekcji, których wartości mogą zależeć od szybkości zmian samej funkcji błędu i być wspólne dla wszystkich wag sieci, lub dobrane dla każdej z nich osobno. W powyższych metodach uczenia nie zawsze znajdowane jest globalne minimum funkcji błędu. Nierzadko wyznaczane jest minimum lokalne, por. [3].

W metodzie uczenia SSN opartej na filtrach Kalmana podejście jest inne, pomimo faktu, iż wagi w kolejnym etapie ich adaptacji również są dobierane na podstawie różnic między wzorcami pożądanymi na wyjściu, a obliczonymi przez SSN. W metodzie tej zmiany wagi sieci kontrolowane są zgodnie przyjętym modelem. Rolę parametru regularyzacyjnego pełnią charakterystyki białego szumu gaussowskiego, będącego składnikiem modelu. Postać modelu zależy od architektury przyjętej sieci neuronowej. Adaptacja wag, rozumianych jako zmienne losowe, odbywa się zgodnie z algorytmem filtracji w tym modelu, w oparciu o kryterium najmniejszego błędu średniokwadratowego estymacji. Wyznaczenie wartości wag w kolejnym etapie uczenia następuje na podstawie wartości błędu, dla pojedynczego wzorca uczącego. Zmiana wagi po jego prezentacji jest zgodna z przyjętym modelem matematycznym, proporcjonalna do różnicy pomiędzy wartością pożądaną na wyjściu z sieci, a podana przez SSN w odpowiedzi na wzorec uczący. Estymacja zależy od błędów dla wzorców poprzednich poprzez rekurencyjną konstrukcję algorytmu.

Do głównych cech tych dwóch podejść należy zaliczyć :

- Metody gradientowe:
 - podejście deterministyczne,
 - realizacja algorytmu minimalizacji,
 - cel uczenia: dobór wag, minimalizujących funkcję błędu sieci dla wzorców ze zbioru uczącego,
 - efektywność działania algorytmu mierzona wartością błędu uczenia sieci dla wyznaczonych przez algorytm wag sieci.
- Metody oparte o filtr Kalmana:
 - podejście stochastyczne,

- realizacja algorytmu filtracji,
- cel uczenia: dobór optymalnego estymatora wag sieci w założonym modelu,
- efektywność działania algorytmu estymacji jest określona przez ślad macierzy kowariancji błędów estymacji,
- efektywność uczenia sieci neuronowej mierzona sumą przeskalowanych zgodnie z parametrami regularyzacyjnymi: błędu średniokwadratowego odtworzenia wzorców oraz błędu estymacji.

2. Rozwój metody

Rozwiązanie problemu filtracji przedstawione przez Kalmana w roku 1960 [14], w odniesieniu do modeli przestrzeni stanu, pierwszy raz wykorzystano jako algorytm do uczenia SSN w latach 80tych. Singhal i Wu, por.[13], zaproponowali algorytm w wersji globalnej (*ang. Global Extended Kalman Filter*, GEKF) do uczenia sieci wielowarstwowych. Algorytm, po rozszerzeniu rozwiązania poprzez linearyzację również dla równań nieliniowych, zakładał konstrukcję modelu dla wektora składającego się ze wszystkich wag sieci neuronowej. Ze względu na duże nakłady obliczeniowe tak skonstruowanej procedury, Puskorius i Feldkamp, zaproponowali wersję zdekomponowaną (*ang. Decoupled Extended Kalman Filter*, DEKF) algorytmu, por. [13]. Wprowadzona zmiana polegała na zastąpieniu jednego (budowanego dla całej sieci) modelu – wieloma modelami osobnymi. Poszczególne modele obejmować mogły pojedyncze neurony lub pojedyncze warstwy SSN. Dobór kolejnych wartości wag jest realizowany w tym przypadku dla każdego z neuronów, niezależnie od neuronów pozostałych lub dla każdej z warstw sieci niezależnie. Takie postępowanie pozwala na redukcję wymiarów macierzy, na których wykonywane są obliczenia i w rezultacie skrócenie czasu trwania obliczeń. Modele te zakładały liniowość równania procesu i nieliniowe równanie obserwacyjne. Zmienną obserwowalną stanowiło wyjście z sieci.

W pracach Lou i Pereza, 1995, [15] zaproponowano algorytm oparty o model czysto liniowy, w którym rolę zmiennej obserwowalnej pełniło nie wyjście z sieci neuronowej, ale potencjał każdego z neuronów. Zmienna ta, dla neuronów innych niż w warstwie wyjściowej, była wyznaczana na podstawie wyjścia podanego przez sieć poprzez algorytm wstecznej propagacji.

Uwzględnienie w modelu teoretycznym sygnału rekurencyjnie "zawracanego" w obrębie każdego neuronu czy warstwy, lub też połączeń neuronów w obrębie tej samej warstwy ukrytej, ewentualnie sprzężenia wyjścia z wejściem, umożliwiło uczenie metodą filtru Kalmana również sieci o strukturach niejednokierunkowych, [16].

Pomimo, iż możliwe jest wprowadzenie w modelu teoretycznym nieliniowego równania procesu, jednak, w zastosowaniach algorytmu Kalmana do uczenia SSN, ze względu na prostotę implementacji i wystarczającą efektywność numeryczną, wykorzystuje się liniową postać równania procesu zmiany wag sieci, por. [13].

3. Przegląd dotychczasowych zastosowań SSN uczonych metodą filtru Kalmana w zakresie modelowania i analizy danych eksperymentalnych

Pierwsze prace dotyczące zastosowania algorytmu uczenia SSN, opartego na filtrze Kalmana, do modelowania rzeczywistych danych pomiarowych dotyczyły problemów obróbki sygnałów z dziedziny przemysłu samochodowego, por. [13], modelowania pracy poszczególnych elementów pojazdu oraz estymacji poziomu parametrów określonych procesów zachodzących podczas pracy silnika.

W zakresie wspomagania sterowania urządzeniami, metodę uczenia SSN, w oparciu o filtr Kalmana stosowano do korekcji błędów podczas operacji chirurgicznych, [17], oraz kontroli pracy kolektorów słonecznej elektrowni Platforma Solar de Almeria w Hiszpani, [18].

SSN uczone z wykorzystaniem FK stosowano w zadaniach modelowania szeregów czasowych: mieszanego autoregresji i średniej ruchomej Box-Jenkins ilości dwutlenku węgla, pozostającego po spalaniu określonej ilości metanu [19] oraz natężenia światła lasera, [13].

SSN, uczone z wykorzystaniem FK, stosowano jako narzędzia pozwalające na wyznaczanie zależności między wyznaczonymi wielkościami (np. prostymi do wyznaczenia w sposób eksperymentalny, a wielkościami z nimi skorelowanymi, trudnymi do określenia w sposób doświadczalny). Można tutaj wymienić następujące zagadnienia: predykcja zużycia energii w budynku [20], estymacja zawartości ołowiu oraz cynku w gruncie, na podstawie naturalnego promieniowania gamma próbki, gęstości próbki wyznaczonej za pomocą prześwietlania promieniowaniem gamma, oraz wyników prób w polu magnetycznym [21], estymacja parametrów gruntu: pierwiastka średniej wysokości terenu i stałej dielektrycznej na podstawie obrazów satelitarnych danego terenu [22], czy wyznaczanie zależności pomiędzy metanem, a podtlenkiem azotu w atmosferze [23].

Algorytm EKF wykorzystany został także do modelowania pracy urządzenia hydraulicznego sterującego ramieniem robota [24] oraz do estymacji mocy wytwarzanej przez turbinę wiatrową [25] i modelowania systemu nagrzewania powietrza [26] w budynku. Filtry Kalmana zostały wykorzystane także w zagadnieniu z dynamiki budowli, polegającym na identyfikacji wybranych parametrów konstrukcji, [27].

Za pomocą SSN, uczonych w oparciu o filtr Kalmana, rozwiązano pomyślnie również szereg problemów klasyfikacji, [28], [29], [30] oraz problemów obróbki sygnałów różnego typu [31], [32], w tym sygnału mowy [33], [34].

Metodę te niejednokrotnie uznawano za narzędzie obiecujące [35-39].

4.2. Dostosowanie ogólnego modelu analizy dyskretnych procesów stochastycznych za pomocą FK do procesu uczenia SSN

Wykorzystanie filtru Kalmana, jako metody uczenia nadzorowanego SSN, zakłada dostosowanie ogólnego modelu, omówionego w Rozdz. 3. do potrzeb analizy dyskretnego procesu stochastycznego, jakim jest w tej metodzie proces obliczania wartości parametrów (wag neuronów i biasów) sieci. Po założeniu pewnego modelu zmian procesu nieobserwowalnego parametrów sieci (może być liniowy, lub nieliniowy), wyborze, która z wielkości stanowić będzie proces obserwowalny (może to być np. wyjście z SSN, lub potencjał neuronów), buduje się dla SSN równanie obserwacyjne w modelu teoretycznym zależne od, por.[13]:

- zewnętrznego sygnału sterującego, który stanowią elementy zbioru uczącego podawanego SSN na wejście oraz odpowiadające im znane wartości wyjścia: $(\mathbf{x}_k, \mathbf{y}_k) \in L$
- wektora parametrów sieci (wag i biasów) $\mathbf{w} \in \mathbf{R}^W$
- założonej architektury sieci (liczby neuronów wejściowych N , liczby neuronów w warstwie wyjściowej M , liczby warstw ukrytych i neuronów ukrytych H , struktury połączeń między neuronami)
- dla SSN zawierających połączenia rekurencyjne od struktury połączeń, którymi sygnał zawracany jest w obrębie SSN, reprezentowanych przez wektor \mathbf{v} .

4.2.1 Model nieliniowy, algorytm Extended Kalman Filter (EKF)

Najczęściej stosowany jest model, w którym równanie procesu zmian wag SSN jest liniowe, zmienną obserwowalną jest wyjście z sieci, natomiast równanie obserwacyjne odzwierciedla budowę SSN o nieliniowych funkcjach aktywacji neuronów – nieliniową zależność pomiędzy wektorem wejścia, a wyjściem SSN.

Niech $W1$ będzie liczbą nierekurencyjnych parametrów SSN, czyli jej wag i biasów związanych z sygnałem przechodzącym od wejścia do wyjścia sieci, $W2$ – liczbą parametrów SSN, związanych z sygnałem "zwrotnym" rekurencyjnie tj. przechodzącym w kierunku od wyjścia do wejścia sieci ($W=W1+W2$). Wtedy dla SSN o takiej budowie, model ten ma następującą postać::

$$\{\mathbf{w}_{k+1}, \mathbf{v}_{k+1}\} = \{\mathbf{w}_k, \mathbf{v}_k\} + \omega_k \quad (4.1)$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{w}_k, \mathbf{x}_k, \mathbf{v}_k) + \nu_k \quad (4.2)$$

gdzie

$k = 1, 2, 3, \dots$ – dyskretny pseudoczas

\mathbf{w}_k – zmienna losowa $W1$ -wymiarowa, wektor wag nierekurencyjnych, obliczonych w k -tym kroku uczenia,

\mathbf{v}_k – zmienna losowa $W2$ -wymiarowa, będąca wektorem 'rekurencyjnych' wag SSN obliczonych w k -tym kroku uczenia,

\mathbf{y}_k – zmienna losowa M -wymiarowa, oznaczająca wektor wyjścia z sieci,

\mathbf{x}_k – zmienna losowa N -wymiarowa, oznaczająca wektor wejścia SSN

$\mathbf{h}_k : R^{W1} \times R^N \times R^{W2} \rightarrow R^M$ – funkcja o argumentach i wartościach wektorowych, nieliniowa, nielosowa, mająca ciągłe pierwsze pochodne

ω_k, ν_k – $W1 + W2$ oraz M -wymiarowe odpowiednio, niezależne zmienne losowe, o rozkładzie gaussowskim i charakterystykach:

$$\mathbf{E}(\omega_k) = 0, \mathbf{E}(\nu_l) = 0 \quad (4.3)$$

$$\mathbf{E}(\omega_k \omega_l^T) = \mathbf{Q}_k \delta_{kl} \quad (4.3)$$

$$\mathbf{E}(\nu_k \nu_l^T) = \mathbf{R}_k \delta_{kl} \quad (4.5)$$

przy czym zakładamy, że ω_k, ν_l są nieskorelowane.

Stan początkowy procesu,

$$\{\mathbf{w}_0, \mathbf{v}_0\} \quad (4.6)$$

jest zmienną losową o rozkładzie normalnym, o wartości oczekiwanej równej

$$\{\bar{\mathbf{w}}_0, \bar{\mathbf{v}}_0\} \quad (4.7)$$

i ustalonej macierzy kowariancji.

Elementami wektora \mathbf{v} , mogą być sygnały wyjściowe z neuronów warstwy wyjściowej, rekurencyjnie zwracane podawane na wejście SSN, (dla sieci FLNN o wejściu autoregresyjnym, por. rys. 2.4) lub sygnał zwracany w obrębie neuronów tej samej warstwy (dla sieci RLNN por. rys. 2.5, 2.6).

Zmiana wag SSN przy prezentacji k -go wzorca uczącego, odbywa się zgodnie z równaniami EKF: (3.16)-(3.22), por. 3., które dla uproszczonego modelu (4.1)-(4.2) sprowadzają się do:

– wyznaczenia macierzy wzmocnienia Kalmana:

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k]^{-1} \quad (4.8)$$

– wyznaczenia estymatora wektora stanu sieci a posteriori, po uwzględnieniu obserwacji \mathbf{y}_k :

$$\{\widehat{\mathbf{w}}_{k+1}, \widehat{\mathbf{v}}_{k+1}\} = \{\widehat{\mathbf{w}}_k, \widehat{\mathbf{v}}_k\} + \mathbf{K}_k \xi_k \quad (4.9)$$

– wyznaczenia przybliżonej macierzy kowariancji błędów estymacji a posteriori:

$$\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{G}_k \mathbf{H}_k) \mathbf{P}_k + \mathbf{Q}_k \quad (4.10)$$

Wielkość

$$\xi_k = \mathbf{y}_k - \widehat{\mathbf{y}}_k \quad (4.11)$$

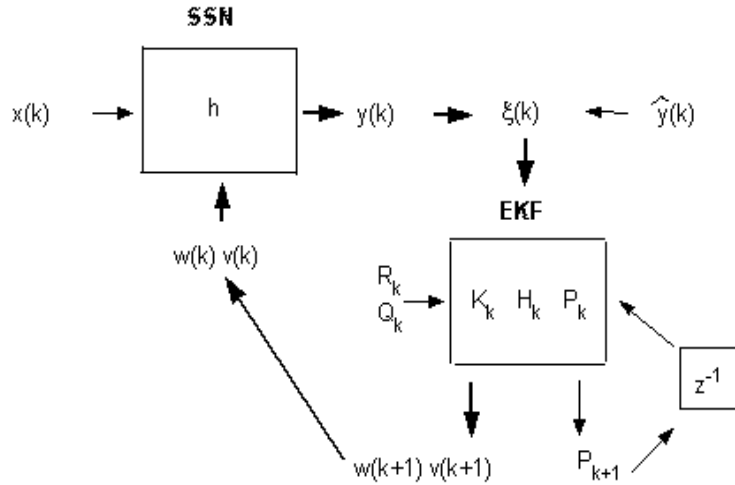
gdzie $\widehat{\mathbf{y}}_k = \mathbf{h}_k(\widehat{\mathbf{w}}_k, \mathbf{x}_k, \widehat{\mathbf{v}}_k)$ to błąd SSN przy prezentacji k -go wzorca uczącego, $h_k(\widehat{\mathbf{w}}_k, \mathbf{x}_k, \widehat{\mathbf{v}}_k)$ stanowi wyjście obliczone przez SSN, jako odpowiedź na wektor wejściowy \mathbf{x}_k , zależne od stanu sieci z poprzedniego kroku uczenia.

Macierz linearyzacji równania obserwacyjnego ma postać:

$$H_k = \left[\frac{\partial h^i(\mathbf{z})}{\partial \mathbf{z}^j} \right]_{i=1, \dots, M} \Big|_{j=1, \dots, W_1+W_2} \Big|_{\mathbf{z}=\{\widehat{\mathbf{w}}_k, \widehat{\mathbf{v}}_k\}} \quad (4.12)$$

gdzie pochodne cząstkowe liczone są względem kolejnych elementów wektora $\mathbf{z} = \{\mathbf{w}, \mathbf{v}\}$.

Schemat działania algorytmu przedstawiony jest na diagramie na rysunku 4.1.



Rys. 4.1. Schemat działania algorytmu uczenia SSN w oparciu o FK

Model, w którym w równaniu (4.1) występuje wektor wszystkich parametrów SSN nazywany jest także Globalnym Rozszerzonym Filtrem Kalmana (*ang. Global Extended Kalman Filter, GEKF*), w odróżnieniu od modeli rozprzężonych, które omówione

będą w dalszej części pracy.

4.2.2 Model rozprzężony nieliniowy, algorytm Decoupled Extended Kalman Filter (DEKF)

Model ten został wprowadzony, by zmniejszyć nakłady obliczeniowe, konieczne do wyznaczania kolejnych wielkości w modelu pełnym. Dla algorytmu GEKF wymiary macierzy kowariancji przybliżonych błędów filtracji \mathbf{P} oraz macierzy wzmocnienia Kalmana \mathbf{K} równe są liczbie parametrów SSN. Zaproponowano więc, rozprzężenie wektora wag sieci na rozłączne grupy, a następnie budowę modelu teoretycznego oraz obliczenia nowych wartości parametrów sieci, dla każdej z grup osobno, [13]. Rozprzężenie to może dotyczyć poszczególnych warstw SSN, mówimy wtedy o algorytmie Layer Decoupled EKF (LDEKF), lub poszczególnych neuronów SSN: Node Decoupled EKF (NDEKF).

Jeśli sieć ma $W1+W2$ parametrów, to macierze konieczne do wyznaczenia wg. GEKF kolejnych ich wartości mają wymiary $W1+W2 \times W1+W2$. Zazwazmy przykładową SSN o 1 warstwie ukrytej, posiadającej odpowiednio: n_1, n_2, n_3 parametrów w obrębie wejścia oraz warstwy ukrytej i wyjściowej ($n_1 + n_2 + n_3 = W1 + W2$). Dla algorytmu LDEKF, w każdym kroku algorytmu, do wyznaczenia wszystkich parametrów SSN konieczne są obliczenia na 3 macierzach o wymiarach: $n_1 \times n_1, n_2 \times n_2, n_3 \times n_3$ odpowiednio. Algorytm rozprzężony wymaga więc mniejszych nakładów obliczeniowych.

Schematycznie zostało to przedstawione dla algorytmu LDEKF dla opisanego rozprzężenia wektora parametrów sieci w tabeli 4.1, gdzie w polach oznaczonych 'X' znajdują się odpowiednie wielkości obliczane w procesie uaktualnienia parametrów SSN. Dla modelu pełnego konieczne są obliczenia na całej macierzy, przedstawionej schematycznie, jako tabela 4.1.

	1	2			...				W
1	X	X	n_1						
2	X	X							
	n_1		X	X	X	X			
			X	X	X	X	n_2		
...			X	X	X	X			
			X	X	X	X		n_3	
				n_2			X	X	X
						n_3	X	X	X
W							X	X	X

Tabela 4.1. Redukcja liczby obliczanych wielkości w LDEKF vs GEKF

W niniejszej pracy stosowane jest rozprzężenie wektora parametrów sieci wg. poszczególnych neuronów sieci. Poniżej zostanie przedstawione sformułowanie równań

modelu oraz kolejne etapy obliczeń.

4.2.2.1 Algorytm Node Decoupled Extended Kalman Filter (NDEKF)

Niech $i = 1, 2, \dots, K \cdot M$ oznacza liczbę neuronów w sieci, (por. Rozdz. 2, rys. 2.3) oraz niech $\mathbf{w}^i \in R^{W_i}$ ($W_1 + W_2 + \dots + W_{K \cdot M} = W$) oznacza wektor wszystkich parametrów (wag, biasów, połączeń rekurencyjnych) i -tego neuronu sieci. Zamiast modelu (4.1)-(4.2), budowanych jest wtedy dla SSN $K \cdot M$ modeli, różniących się równaniem procesu.

Dla i -tego neuronu model przyjmuje postać:

$$\mathbf{w}_{k+1}^i = \mathbf{w}_k^i + \omega_k^i \quad (4.13)$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{w}_k, \mathbf{x}_k) + \nu_k \quad (4.14)$$

gdzie

$$k = 1, 2, \dots, i = 1, \dots, K$$

ω_k^i – W_i -wymiarowe, niezależne zmienne losowe, o rozkładzie gaussowskim i charakterystykach:

$$\mathbf{E}(\omega_k^i) = 0 \quad (4.15)$$

$$\mathbf{E}(\omega_k^i \omega_l^{iT}) = \mathbf{Q}_k^i \delta_{kl} \quad (4.16)$$

przy czym ω_k^i, ν_l są nieskorelowane.

Zmiana wektora \mathbf{w}^i przy prezentacji k -go wzorca uczącego odbywa się poprzez wyznaczenie kolejno, [13]:

– macierzy wzmocnienia Kalmana, w skład której wchodzi macierze linearyzacji równania (4.13), dla każdego modelu osobno, względem odpowiednich wektorów parametrów, związanych z i -tym neuronem SSN dla wszystkich $K \cdot M$ modeli:

$$\mathbf{K}_k^i = \mathbf{P}_k \mathbf{H}_k^T \left[\sum_{j=1}^{K \cdot M} \mathbf{H}_k^j \mathbf{P}_k^j \mathbf{H}_k^{jT} + \mathbf{R}_k \right]^{-1} \quad (4.17)$$

– estymatora wektora stanu parametrów i -tego neuronu, po uwzględnieniu obserwacji \mathbf{y}_k :

$$\widehat{\mathbf{w}}_{k+1}^i = \widehat{\mathbf{w}}_k^i + \mathbf{K}_k^i \xi_k \quad (4.18)$$

– przybliżonej macierzy kowariancji błędów estymacji w i -tym modelu:

$$\mathbf{P}_{k+1}^i = (\mathbf{I} - \mathbf{G}_k^i \mathbf{H}_k) \mathbf{P}_k^i + \mathbf{Q}_k^i \quad (4.19)$$

Macierz linearyzacji równania obserwacyjnego względem j -tego neuronu sieci ma postać:

$$\mathbf{H}_k^j = \left[\frac{\partial \mathbf{h}^l(\mathbf{z})}{\partial \mathbf{z}^{ll}} \right]_{l=1, \dots, M} \Big|_{ll=1, \dots, W_j} \Big|_{\mathbf{z} = \widehat{\mathbf{w}}_k^j} \quad (4.20)$$

gdzie pochodne cząstkowe liczone są względem elementów wektora $\mathbf{z} = \mathbf{w}^j$. W_j oznacza wymiar wektora \mathbf{w}^j . Linearyzacja odbywa się więc względem parametrów każdego z modeli osobno w ten sposób, że dla j -tego modelu funkcja \mathbf{h} rozpatrywana jest jako funkcja wektora \mathbf{w}^j .

4.2.3. Model liniowy rozprężony wg. neuronów SSN, algorytm Node Decoupled Linear Kalman Filter (NDLKF)

Rozprężenie algorytmu względem poszczególnych neuronów pozwala na budowę modelu teoretycznego liniowego, także w sytuacji, gdy funkcje aktywacji neuronów w sieci są nieliniowe. W modelu takim, oba równania: procesu zmian wag SSN oraz obserwacyjne są liniowe, por. [15]. Zmienną obserwowalną jest pożądaný potencjał i -tego neuronu.

Zachowując poprzednie oznaczenia, dla i -tego neuronu model przyjmuje postać:

$$\mathbf{w}_{k+1}^i = \mathbf{w}_k^i + \omega_k^i \quad (4.21)$$

$$\mathbf{d}_k^i = \mathbf{X}_k^{iT} \mathbf{w}_k^i + \nu_k \quad (4.22)$$

gdzie

\mathbf{d}_k^i – pożądaný potencjał i -tego neuronu,

\mathbf{X}_k^i – wektor sygnałów docierających do i -tego neuronu poprzez połączenia z neuronami warstwy wcześniejszej (lub wejścia), będących reakcjami SSN na podanie na wejście pierwszego elementu k go wzorca uczącego.

Dla $k, l = 1, 2, \dots, i = 1, \dots, KM$ wielkości ω_k^i oraz ν_k , jak w modelu (4.13)-(4.14).

Niech, zgodnie z wprowadzonymi wcześniej oznaczeniami, $(\mathbf{x}_k, \mathbf{y}_k)$ będzie k -tym wzorcem uczącym, gdzie \mathbf{y}_k stanowi znany k -ty wektor wyjścia SSN. Wektor \mathbf{X}_k wyznacza się na podstawie propagacji sygnału 'wprzód', po podaniu sieci na wejście wielkości \mathbf{x}_k . Stanowią go sygnały wychodzące z neuronów, mających połączenia wagowe z tym neuronem, dla którego budowany jest model. Pożądane potencjały neuronów $\widehat{\mathbf{d}}_k^i$ wyznacza się na podstawie drugiego elementu pary uczącej, czyli znanej odpowiedzi SSN na wzorec wejściowy.

Dla neuronów warstwy wyjściowej, pożądaný potencjał neuronu definiuje się jako:

$$\widehat{\mathbf{d}}_k^i = F_i^{-1}(\widehat{y}_k) \quad (4.23)$$

gdzie F_i oznacza funkcję aktywacji i -tego neuronu. Jest to więc znana odpowiedź na wzorec wejściowy, przekształcona przez odwrotną funkcję aktywacji neuronu. Dla neuronów warstw poprzednich, wyznacza się tę wielkość, korygując rzeczywisty potencjał tego neuronu, w oparciu o jedną z metod numerycznej minimalizacji funkcji błędu SSN np. metodę gradientową największego spadku, otrzymując za [15]:

$$\widehat{\mathbf{d}}_k^i = \mathbf{d}_k^i - \mu \frac{\partial E}{\partial \mathbf{d}_k^i} \quad (4.24)$$

gdzie E jest przyjętą funkcją błędu sieci (np. $E = MSE$), μ parametrem uczenia, wyznaczanym jak w standardowych metodach gradientowych uczenia SSN, por. Rozdz. 2.

4.3. Uczenie SSN jednokierunkowych i rekurencyjnych jedno oraz dwuwarstwowych za pomocą Filtra Kalmana

Przeprowadzenie procesu uczenia wymaga dodatkowo: wyznaczenia macierzy linearyzacji w modelu DEKF oraz przyjęcia parametrów szumu w modelu teoretycznym.

4.3.1. Linearyzacja równania obserwacyjnego.

W modelu teoretycznym, w którym występuje nieliniowa funkcja \mathbf{h} , reprezentująca strukturę sieci, istnieje konieczność wyznaczenia jej pochodnych względem odpowiednich parametrów SSN. Wielkości te można określić poprzez znalezienie postaci funkcji \mathbf{h} (określenie postaci związku pomiędzy wektorem wejścia, a wejściem do sieci) i wyznaczenie pochodnych względem odpowiednich wielkości analitycznie [2], lub poprzez różniczkowanie numeryczne metodą *Back Propagation Through Time* [4].

4.3.2. Parametry białego szumu w modelu teoretycznym.

Wartości macierzy kowariancji białego szumu w modelu teoretycznym \mathbf{R} oraz \mathbf{Q} , ustalane są w sposób następujący: dla k -go wzorca uczącego przy jego prezentacji w czasie s -tej epoki uczenia przyjmuje się na przykład:

$$\mathbf{R}_k(s) = \alpha \exp(-s/\beta) \mathbf{I} \quad (4.25)$$

$$\mathbf{Q}_k(s) = \alpha' \exp(-s/\beta') \mathbf{I} \quad (4.26)$$

gdzie

s – numer bierzącej epoki uczenia sieci

\mathbf{I} – macierz identycznościowa, której wymiar wynika z postaci modelu,

$\alpha, \alpha', \beta, \beta'$ – stałe, por. [13].

Szum ma więc stałe parametry w obrębie tej samej epoki, malejące eksponentalnie wraz z przejściem do kolejnego cyklu prezentacji zbioru uczącego. Parametry te pełnią rolę regularyzacyjną dla algorytmu filtrowania. Malejące wartości kowariancji

białego szumu w równaniu procesu powodują, iż w miarę postępu uczenia, zmiany wag nie są tak duże, jak w fazie początkowej. Przyjęcie macierzy kowariancji szumu w równaniu obserwacyjnym równej szacowanemu błędowi pomiarowemu, w przypadku wprowadzania rzeczywistych wielkości, wyznaczonych eksperymentalnie z pewnym znanym lub szacowanym błędem (np. błędem urządzenia pomiarowego), pozwala na uwzględnienie tej wiedzy w modelu teoretycznym.

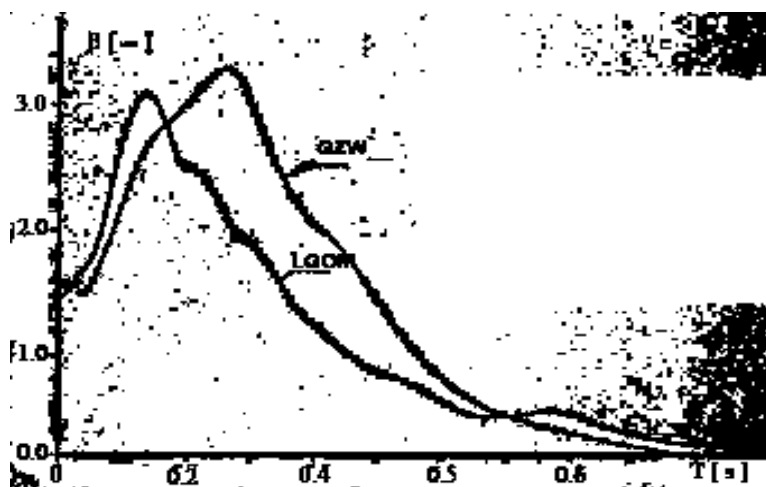
5. Zastosowanie SSN uczonych za pomocą FK

5.1. Symulacja Spektrow Odpowiedzi od wymuszeń parasejsmicznych

Drgania parasejsmiczne są obciążeniami budowli, powstałymi w czasie kinematycznego ruchu jej podstawy, powodowanymi działalnością człowieka. Ich źródłami mogą być zarówno: tąpnięcia w kopalniach i wyrobiskach, wybuchy przemysłowe, podziemne eksplozje nuklearne, jak i ruch pojazdów drogowych, kolejowych, czy wbijanie pali w sąsiedztwie budowli.

W niniejszej pracy rozpatrywane są wstrząsy parasejsmiczne, powstające na terenie Polski, w wyniku podziemnego wydobywania węgla na terenach Górnośląskiego Zagłębia Węglowego (GZW) oraz w wyniku odstrzałów w kamieniołomach rudy miedzi, na terenach Legnicko-Głogowskiego Okręgu Miedziowego (LGOM).

W obydwu tych regionach występuje gęsta zabudowa, a najsilniejsze wstrząsy powodują uszkodzenia budynków, gdyż nie zostały one w fazie projektowania przystosowane do obciążeń tego typu [40]. Jedną z metod oceny szkodliwości wstrząsów parasejsmicznych, przydatną także przy projektowaniu budynków w rejonach poddanych wymuszeniom tego typu, jest metoda Spektrum Odpowiedzi [41,55]. Najczęściej stosuje się modelowe Spektra Odpowiedzi, wyznaczone na podstawie wielu wstrząsów metodą uśredniania Spektrow Odpowiedzi od wstrząsów pojedynczych, [42,43]. Przeprowadzone w ww. regionach pomiary wskazują na istnienie znacznej różnicy w kształtach Spektrow Odpowiedzi (SO), zarówno między tymi dwoma regionami kraju, por. rys. 5.1, jak i w stosunku do regionów aktywnych parasejsmicznie w krajach sąsiednich.



Rys.5.1. SO dla GZW i LGOM, za [40]

Szczegółowa definicja akceleracyjnego (przyspieszeniowego) i przemieszczeniowego spektrum odpowiedzi, oznaczanych dalej skrótami ASO i PSO oraz wykorzystanie

SO do projektowania budynków i oceny szkodliwości drgań parasejsmicznych przedstawione zostały w Dodatku 3.

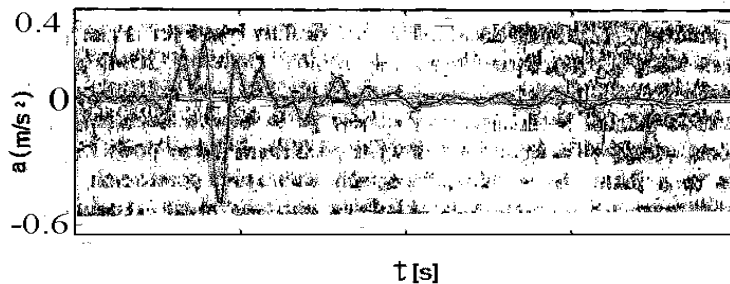
Ponieważ nie jest możliwe monitorowanie każdego budynku w zagrożonych sejsmicznie regionach, dlatego SSN są stosowane, w celu predykcji (generowania) Spektrow Odpowiedzi dla poszczególnych budynków [44, 46, 47].

5.1.1. Symulacja ASO od wstrząsów górniczych

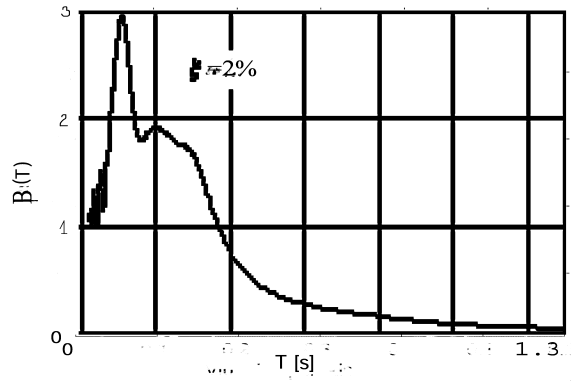
W rozdziale niniejszym rozpatrywany jest problem zależności Akceleracyjnych (Przyśpieszeniowych) Spektrow Odpowiedzi (ASO) od energii wstrząsu E i odległości r_e epicentrum wstrząsu górniczego od miejsca rejestracji przebiegu drgań gruntu przy monitorowanych budynkach. Pomiary prowadzono na terenie Górnośląskiego Zagłębia Węglowego (GZW). Praca dowiązuje głównie do [45], gdzie również analizowano zbiory omawianych w niniejszym rozdziale danych, a w celu symulacji SO wykorzystano standartowe gradientowe metody uczenia SSN.

1. Dane pomiarowe

Oparto się na danych z okręgu GZW, przyjmując do analizy dane z [61], dotyczące 145 wstrząsów o energii $E \in [2 \cdot 10^4, 4 \cdot 10^6] J$ i odległości epicentralnej $r_e \in [0, 1200] m$, zarejestrowanych w stacjach sejsmicznych w Katowicach oraz Rybniku, na przestrzeni kilku lat. Mierzono składowe poziome przyśpieszenia gruntu przy wybranych budynkach. Na ich podstawie wyznaczano Względne Przyśpieszeniowe Spektra Odpowiedzi, przyjmując ułamek tłumienia $\xi = 2\%$, jako tłumienie występujące w większości typowych budynków regionu GZW. Na rys.5.2.a pokazano jeden z zarejestrowanych akcelerogramów, na rys.5.2.b odpowiadające mu ASO. Maksymalna zarejestrowana wartość przyśpieszenia gruntu w przypadku drgań poziomych wyniosła $a_{max} \approx 0.5 m/s^2$.



Rys.5.2.a Akcelerogram zarejestrowany dn.25.02.1996 dla $E = 2 \cdot 10^6 J, r_e = 740m$, za [45]



Rys.5.1.b. Bezwymiarowe ASO, za [45]

Za pracą [45] przyjęto spektra w postaci

$$\beta_k = S_a/a_{max} = \beta(T_k) \quad (5.1)$$

dla $k = 1, 2, \dots, 198$, przyjmując $T_k \in [0.02, 1.33]s$, co odpowiada częstotliwościom $f_k \in [0.75, 50.0]Hz$ i krokowi $\Delta f_k = 0.25Hz$, gdzie $T_k = 2\pi/\omega_k = 1/f_k$. Dla pojedynczego spektrum odpowiedzi znana była energia E oraz odległość epicentralna r_e , charakteryzująca wstrząs na podstawie którego zarejestrowano akcelerogram odpowiadający temu Spektrum.

2. Wzorce uczące i testujące dla SSN

Powyższe dane pomiarowe dają zbiór $P = 145 \cdot 198 = 28710$ wzorców, gdzie przez wzorec rozumiana jest pojedyncza wartość β_k . W celu porównania jakości odwzorowania spektrów, za pomocą SSN uczonych algorytmem FK, z metodami wcześniej stosowanymi, przyjęto podział spektrów na uczące i testujące, jak w [45] tzn. 113 losowo wybranych z dostępnych ASO do uczenia sieci, pozostałe 32 ASO do testowania. Otrzymano w ten sposób: $L = 113 \cdot 198 = 22374$ (ok.78%) wzorców do uczenia SSN oraz $T = 32 \cdot 198 = 6336$ (ok.22%) wzorców do testowania. Na potrzeby algorytmu obliczeniowego, dane wejściowe dla SSN przeskalowano do wartości zawierających się w przedziale $[0,1]$. Wielkości o t.j. $\beta_k \in [0.158, 4.288]$ zostały podzielone przez największą wartość w ich ciągu $max_{k=1, \dots, 198} \beta_k = 4.288$, dane o dużych wartościach (E, r_e), uprzednio przekształcono przez funkcję logarytmu naturalnego. Dla różnych wartości ww. paramametrów, spektra różniły się znacznie min. wartościami maksymalnymi i zakresem okresów drgań, dla których wartości spektrów są bliskie maksymalnym.

3. Architektury rozważanych SSN

W celu modelowania zależności wartości spektrum β_k dla poszczególnych okresów drgań T_k , od odległości epicentralnej oraz energii wstrząsu, zaproponowano dla SSN następujący wektor wejścia \mathbf{x} :

$$\mathbf{x}(k) = \{\beta_{k-1}, E, r_e\}, \quad (5.2)$$

na wyjściu z sieci oczekując wartości

$$\hat{y}(k) = \beta_k \quad (5.3)$$

Zaproponowano dwa typy SSN:

- Sieć warstwową, typu FLNN o architekturze 3-15-1: 3 wejścia, 15 neuronów w warstwie ukrytej, jeden neuron w warstwie wyjściowej. Zastosowano rekurencyjne połączenie wyjścia SSN z wejściem. Zawracany sygnał stanowiło opóźnione o jeden krok czasowy wyjście z SSN, por. rys. 2.4., Dla neuronów warstwy ukrytej przyjęto sigmoidalne funkcje aktywacji (2.4), dla wyjścia liniową funkcję aktywacji (2.5) (por. rozdz. 2., dla $N=3$, $K=15$, $M=1$, $T^{in} = 1$).
- Sieć RLNN o architekturze jak powyżej, ale z dodatkową rekurencją typu Elmana tj. w obrębie neuronów (por. rys. 2.5., dla $N=3$, $K=15$, $M=1$). Rekurencja występuje w obrębie neuronów warstw ukrytych i neuronu wyjściowego. Sygnał wyjściowy z danego neuronu (stanowiący jego potencjał, przekształcony przez funkcję aktywacji) jest z opóźnieniem czasowym jednego kroku rekurencyjnie zawracany do neuronów tej samej warstwy. Rekurencyjne połączenie typu Elmana zostało dodatkowo, po wstępnych testach, zredukowane tylko do połączenia każdego z neuronów z samym sobą (dotyczy neuronów warstwy ukrytej oraz wyjściowego), por. rys.2.6.

Sieci uczono z urzyciem rozprężonego według poszczególnych neuronów algorytmu NDEKF, wykorzystującego Rozszerzony Filtr Kalmana dla wyjścia z sieci, jako wielkości obserwowalnej (por. rozdz. 4). W modelu teoretycznym (4.13)-(4.14) wektor procesu $\{w^i, v^i\}$ przyjęto postaci:

- dla FLNN: $w^i \in R^4$, oraz $v^i \in R$ dla neuronów $i = 1, \dots, 15$ warstwy ukrytej, $w^{16} \in R^{16}$ oraz brak wektora sprzężenia zwrotnego dla wyjścia
- dla RLNN: $w^i \in R^4$, oraz $v^i \in R$ dla neuronów $i = 1, \dots, 15$ warstwy ukrytej, $w^{16} \in R^{16}$ oraz $v^{16} \in R$ dla wyjścia.

Funkcja h , odzwierciedlająca struktury przyjętych SSN, jest w postaci (2.9)-(2.14).

Dla sieci FLNN przyjęto dla wszystkich $k = 1, 2, \dots, 198$ parametry szumów:

$$R_k = 7exp((s - 1)/50) \quad (5.4)$$

dla równań obserwacyjnych oraz:

$$\mathbf{Q}_k = 0.001exp((s - 1)/50)\mathbf{I} \quad (5.5)$$

dla równań procesu, gdzie s – oznacza numer epoki uczenia. Dla sieci RLNN przyjęto odpowiednio

$$R_k = 10exp((s - 1)/30) \quad (5.6)$$

$$\mathbf{Q}_k = 0.01 \exp((s-1)/10) \mathbf{I} \quad (5.6)$$

gdzie \mathbf{I} jest macierzą jednostkową, o wymiarze zależnym od liczby połączeń dochodzących do neuronu, dla którego konstruowany jest model teoretyczny, por. rozdz. 4., czyli 16x16 lub 4x4, w zależności od numeru neuronu.

Uczenie rozpoczęto od początkowych wartości parametrów SSN, wylosowanych z przedziału $[-0.5, 0.5]$, zgodnie z rozkładem jednostajnym. Proces uczenia kontrolowano poprzez obserwację spadku poziomu błędów średniokwadratowych uczenia $MSEU$ oraz testowania $MSET$, (2.21). Zastosowano kryterium stopu algorytmu uczącego, oparte o określoną liczbę epok uczenia SSN lub określony poziom błędu uczenia $MSEU$.

4. Wyniki uczenia i testowania sieci FLNN i RLNN za pomocą algorytmu Kalmana

Wstępne uczenie przeprowadzono zakładając kryterium wczesnego stopu algorytmu $S = 100$ epok dla sieci RLNN oraz 500 epok dla sieci FLNN.

W tabeli 5.1. przedstawiono wyniki uczenia i testowania obu SSN w zakresie błędów $MSEU$ oraz $MSET$. Wyszczególniono także dla zbioru testującego, r_T , współczynnik korelacji liniowej, między wartościami spektrum obliczonymi na podstawie pomiarów, $y(p) = \beta_p$, a podanymi przez SSN, $\hat{y}(p) = \hat{\beta}_p$, por. rozdz. 4. Współczynnik ten dany jest wzorem:

$$r_T = \frac{\sum_p (\beta_p - \bar{\beta})(\hat{\beta}_p - \bar{\hat{\beta}})}{\sqrt{\sum_p (\beta_p - \bar{\beta})^2 \sum_p (\hat{\beta}_p - \bar{\hat{\beta}})^2}} \quad (5.7)$$

gdzie

$$\bar{\beta} = (1/6336) \sum_p \beta_i \quad (5.9)$$

$$\bar{\hat{\beta}} = (1/6336) \sum_p \hat{\beta}_p \quad (5.10)$$

wyraża związek liniowy między tymi wielkościami (sumowanie dla $p = 1, 2, \dots, 6336$).

SSN/algorytm	liczba epok uczenia	$MSEU \cdot 10^3$	$MSET \cdot 10^3$	r_T
RLNN/NDEKF	24	1.01	0.88	0.9861
	27	0.87	0.76	0.9865
	100	0.41	0.37	0.9874
FLNN/NDEKF	188	1.00	1.17	0.9877
	300	0.94	1.10	0.9877
	500	0.87	1.02	0.9877
Rprop	10000	3.1	4.1	0.781

Tab.5.1. Błędy średniokwadratowe uczenia i testowania, metoda FK i Rprop dla wektora wejścia do SSN w postaci (5.12)

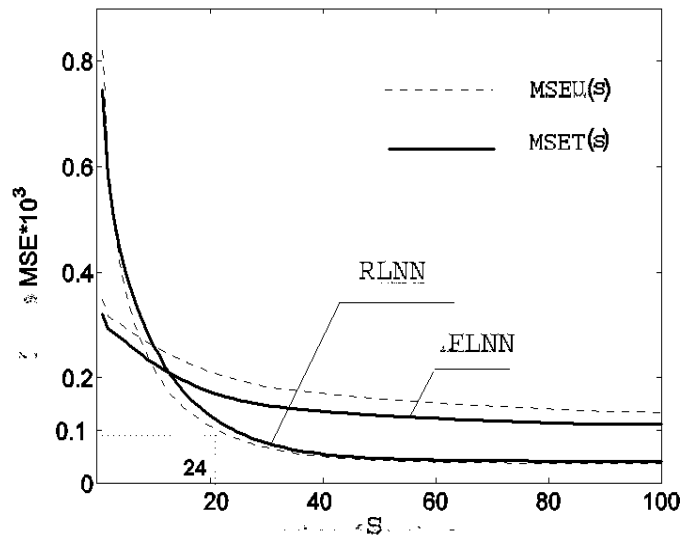
By otrzymać wartość SO dla wybranego okresu drgań T_k dla SSN o wejściu $\{\beta_{k-1}, E, r_e\}$, należy wykonać określoną liczbę k przebiegów obliczeń SSN, wg. schematu:

$$T_n \Rightarrow SSN(\mathbf{x}(n)) \quad (5.11)$$

gdzie $n = 1, 2, \dots, k$, $SSN(\mathbf{x}(n))$ oznacza obliczenie wyjścia z sieci, dla wektora wejścia $\mathbf{x}(n)$.

W zakresie uzyskanych błędów uczenia i testowania sieć RLNN okazała się o wiele efektywniejsza – oba typy błędów są wyraźnie niższe. Sieć RLNN uczy się szybciej, podczas 100 epok uczenia sieci rekurencyjnych RLNN uzyskano mniejsze błędy uczenia i testowania, niż podczas 300 epok uczenia sieci nie rekurencyjnej FLNN, por. rys.5.4. Dla sieci rekurencyjnej RLNN błąd $1 \cdot 10^{-4}$ został osiągnięty po 24 epokach uczenia. Zastosowanie sieci rekurencyjnej RLNN pozwoliło znacznie przyspieszyć algorytm uczenia, po $s = 27$ epokach uczenia błąd MSE osiągnął poziom taki, jak otrzymany po $s = 300$ epokach uczenia sieci jednokierunkowej FLNN.

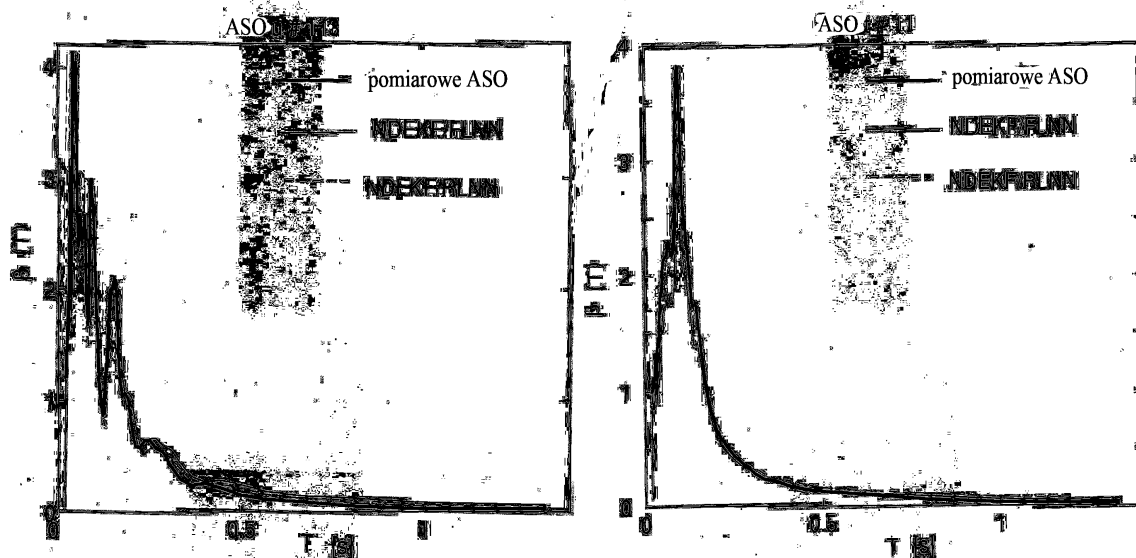
Model teoretyczny dla sieci RLNN różni się od zastosowanego w sieci FLNN tylko jedną dodatkową wagą dla każdego neuronu, nie powoduje więc znacznego zwiększenia wymiarów macierzy koniecznych do przeprowadzenia procesu uczenia sieci. Rozbudowanie modelu, przez wprowadzenie rekurencji w obrębie każdego z neuronów, pozwoliło jednak na znacznie lepsze odtworzenie wzorców uczących oraz znaczące obniżenie poziomu błędu testowania.



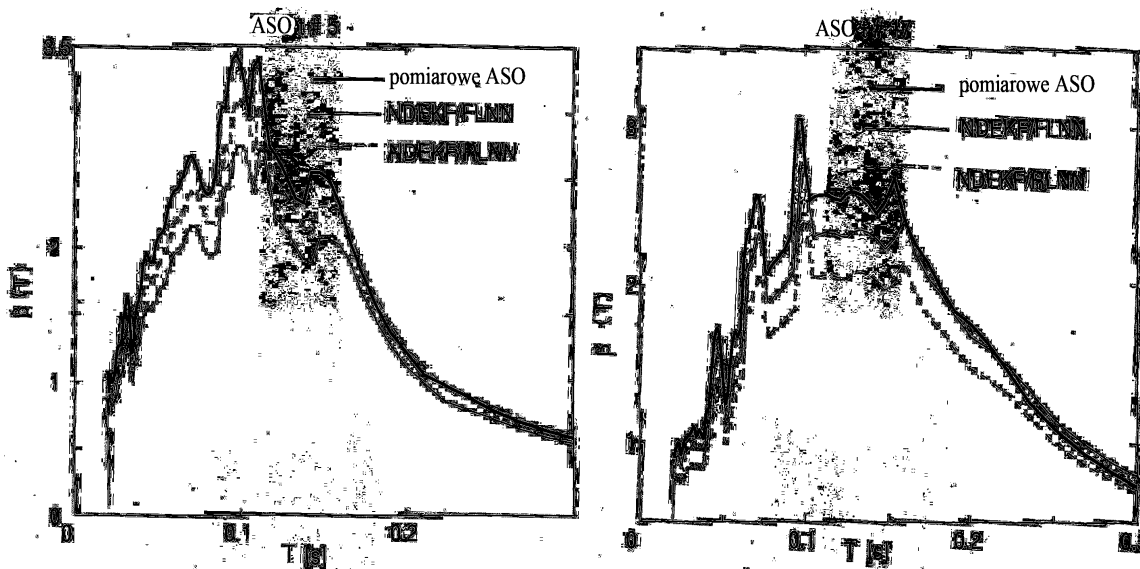
Rys.5.4. Uzyskane poziomy błędów MSEU, MSET dla sieci RLNN, FLNN, w zależności od nr epoki uczenia s

Na rys. 5.5. zestawiono wybrane spektrum uczące (ASO u #113) oraz testowe (ASO t #11), symulowane przez oba typy SSN, porównane z odpowiednimi spektrami obliczonymi na podstawie pomiarów, oznaczonymi jako 'pomiarowe ASO'. Na rys. 5.6.

pokazano części ASO dla przedziału $T_k \in [0.02, 0.308]s$, co odpowiada częstotliwościom $f_k \in [3.25, 50]Hz$. Taki zakres obejmuje charakterystyki mieszkalnych budynków prefabrykowanych o średniej wysokości [65].



Rys.5.5. Porównanie odtworzenia SO uczącego nr 113 i testowego nr 11 przez SSN z odpowiednimi spektrami obliczonymi na podstawie pomiarów



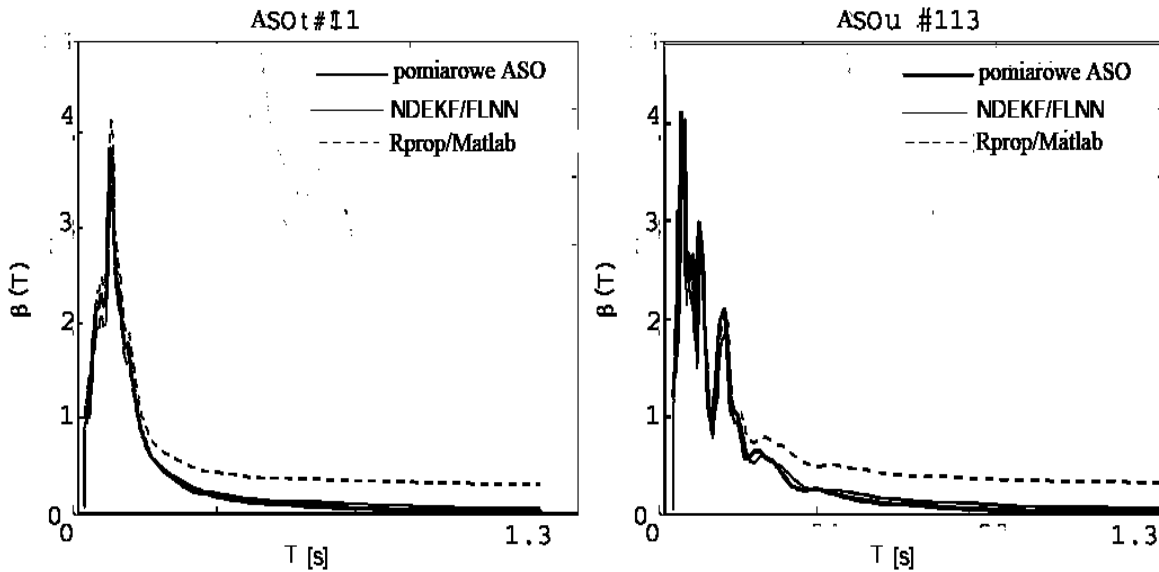
Rys.5.6. Wybrane SO testowe dla okresów drgań odpowiadających budynkom o średnich wysokościach

Widocznym jest, iż spektra symulowane przez SSN (ASO t # 11, 5, 17) dają dobrą aproksymację spektrów, otrzymanych na podstawie pomiarów. Stwierdzono, iż

symulacją neuronową Spektrow daje przybliżenie wykresów eksperymentalnych 'od dołu' w przypadku obliczeń wykonanych zarówno przy pomocy FLNN jak i RLNN.

5. Porównanie efektywności SSN uczonej metodą FK, ze standartową metodą uczenia Rprop

Obliczenia wykonano również za pomocą sieci o takich samych strukturach i funkcjach aktywacji neuronów, jak omawiane powyżej, do uczenia jednak zastosowano symulator MATLAB Neural Network Toolbox [11] oraz metodę uczenia Rprop, jako zalecaną w [4] dla dużej liczby wzorców uczących, por. rozdz. 2.



Rys.5.7. Wybrane SO, SSN uczone metodą Rprop/MATLAB i FK

SSN/algorytm	liczba epok uczenia	$MSEU \cdot 10^3$	$MSET \cdot 10^3$	r_T
RLNN/Rprop	50	0.84	1.067	0.9854
	200	0.54	0.74	0.9864
	1000	0.42	0.46	0.9870
	10000	0.40	0.54	0.9877
FLNN/Rprop	200	0.94	2.4	0.9329
	1000	0.57	1.20	0.9683
	10000	0.42	0.56	0.9845

Tab.5.2. Błędy średniokwadratowe uczenia i testowania, metoda Rprop

W tab.5.2 pokazano błędy aproksymacji sieci uczonej tą metodą do 10 tys. epok włącznie. Na rysunku 5.7. przedstawiono odtworzenie tych samych spektrów (nr 113

uczące oraz nr 11 testowe), które zostały pokazane na rys.5.5. Widoczna jest różnica w efektywności działania SSN nauczonych metodą FK (DEKF) oraz metodą Rprop i MATLAB Neural Network Toolbox (Rprop/Matlab) w zakresie wyższych okresów drgań.

Z porównania tab. 5.1 oraz 5.2. wynika, iż metoda FK jest zbieżna szybciej, niż metoda uczenia Rprop (mniej epok uczenia potrzeba do osiągnięcia podobnych rezultatów). Czas trwania epoki uczenia w przypadku SSN uczonej z wykorzystaniem FK jest jednak ok. 38% dłuższy. Głównym celem uczenia SSN jest znalezienie SSN, o jak najlepszych własnościach generalizacyjnych. Z porównania uzyskanych błędów testowania, $MSET$, wynika, iż metoda oparta o FK jest w tym zakresie efektywniejsza. Dłuższy czas uczenia sieci nie wpływa na czas otrzymywania wyników symulacji w trakcie testowania sieci.

5. Porównanie otrzymanych wyników z dotychczas stosowanymi metodami predykcji ASO na terenie GZW

5.1. Porównanie z metodami wykorzystującymi SSN.

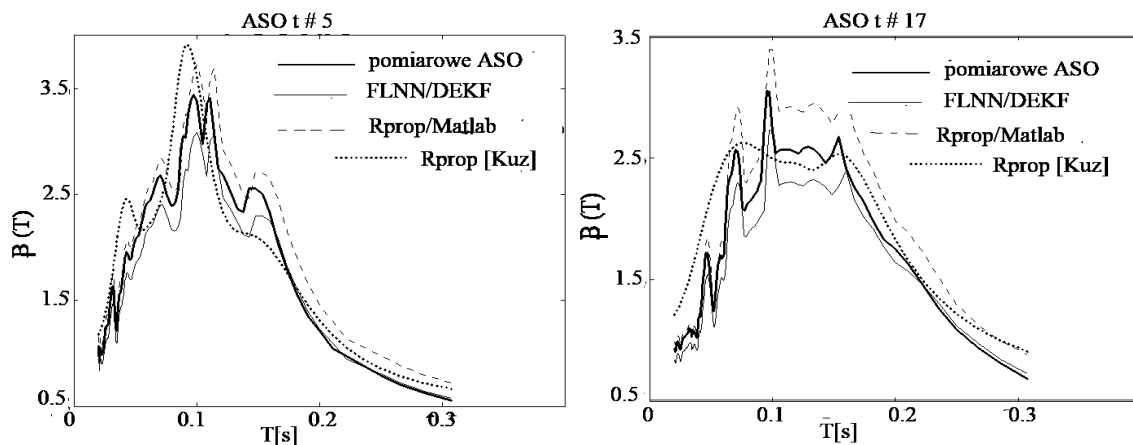
Dotychczasowe prace w zakresie symulacji kształtu ASO w regionie GZW dotyczyły wykorzystania klasycznych SSN neuronowych wstecznej propagacji błędów (por. rozdz. 2). zaimplementowanych w symulatorze neuronowym - SNNS [48], w połączeniu z metodą uczenia Rprop. Konstrukcję wektora wejścia oparto o składową T_k , zamiast składowej autoregresyjnej β_k , otrzymując wektor wejściowy do SSN postaci:

$$\mathbf{x} = \{E, r_e, T_k\} \quad (5.12)$$

Sieci uczone i testowane w oparciu o te same zbiory wzorców do uczenia i testowania. W pracy [49] za najdokładniejszą uznano sieć o architekturze 3-6-12-5-1. Uzyskane poziomy błędów uczenia i testowania przedstawiono w tab. 5.1.

Zastosowanie SSN, uczonych przy pomocy FK, o konstrukcji wektora wejścia (5.2) pozwala więc znacznie podnieść dokładność symulacji spektrów.

Na rys. 5.8. przedstawiono spektra testujące ASO nr 5 i 17, rozpatrywane w [49], oznaczając je jako Rprop[Kuz]. Z porównania wykresów na rys. 5.8. i 5.5. wynika, że predykcja Rprop[Kuz] i dużą siecią 3-6-12-5-1 bardziej odbiega od ASO, obliczonych na podstawie pomiarów, niż dla aproksymacji uzyskanych znacznie mniejszymi sieciami FLNN 3-15-1, RLNN 3-15-1, uczonymi z wykorzystaniem filtru Kalmana.



Rys.5.8. Odtworzenie spektrum testowego metodami Rprop/Matlab, Rprop[Kuz] za [49], FLNN/NDEKF

Z porównania wykresów wynika, że aproksymacja FLNN/Rprop, RLNN/Rprop daje oszacowania ASO pomiarowego od góry (otrzymywane były w przypadku 5324 wzorców, na 6336 wzorców zbioru testującego). Aproksymacja FLNN/NDEKF, a także RLNN/NDEKF daje oszacowanie od dołu dla znacznych części wykresów (zwłaszcza dla niskich wartości okresów T [sek]). Zastosowanie SSN o autoregresyjnej budowie wektora wejścia dla różnych metod uczenia pozwala otrzymać przedziałowe oszacowanie wartości spektrum z dużą dokładnością. Zastosowanie wektora wejścia T_k , zamiast składowej autoregresyjnej, wydaje się nie dawać takich możliwości.

5.2. Porównanie z metodami empirycznymi, stosowanymi na terenie GZW.

Maksymalne wartości przyspieszeń, na podstawie energii wstrząsu oraz odległości epicentralnych, szacuje się w oparciu o wzory empiryczne, modyfikowane w miarę rejestrowania kolejnych wstrząsów na terenie GZW, jednak wg [40], otrzymywane wyniki symulacji często odbiegają od rzeczywistych danych pomiarowych, co podnosi atrakcyjność metody opartej o analizę z wykorzystaniem SSN.

5.1.2. Interakcja Przemieszczeniowych Spektrów Odpowiedzi grunt – fundament budynku

Zjawisko współpracy konstrukcji z podłożem, powstałe na skutek oddziaływania konstrukcja – podłoże podczas wstrząsów sejsmicznych i parasejsmicznych, określa się mianem interakcji dynamicznej [41]. Fale sejsmiczne są źródłem naprężeń kontaktowych pomiędzy podłożem, a fundamentem budynku. Wywołują one deformację podłoża pod obiektem oraz szereg efektów w samej konstrukcji. Interakcja dynamiczna obejmuje szereg zjawisk dotyczących wzajemnego sprzężenia zachowania się gruntu oraz budynku. Drgający grunt oddziałuje na fundament budynku, a siły powstające w gruncie w wyniku wstrząsu powodują jego deformacje. Mogą one mieć

wpływ na odpowiedź budynku. Ruch zarejestrowany w pobliżu konstrukcji może być różny od tego, jaki byłby na tym samym obszarze w przypadku braku konstrukcji, ruch konstrukcji może także mieć dodatkowo wpływ na deformację podłoża. Badanie zjawiska interakcji dynamicznej jest trudne, ze względu na złożone cechy gruntu, geometrię fundamentów budynku i charakter drgań parasejsmicznych [41]. Jedną z metod analizy interakcji dynamicznej jest analiza spektrów odpowiedzi od drgań zarejestrowanych na gruncie obok budynku, zestawionych ze spektrami odpowiedzi od drgań zarejestrowanych za pomocą urządzeń ustawionych na fundamentach budynku, np. w okienkach piwnic, lub na posadzce piwnic [40].

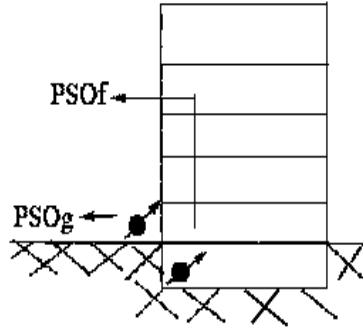
Niniejszy rozdział dotyczy metody wyznaczania Przemieszczeniowych Spektrów Odpowiedzi (PSO) dla fundamentu budynku, za pomocą SSN uczonych w oparciu o PSO dla gruntu. Oparto się na danych pochodzących z pomiarów na terenach LGOM, dotyczących budynków średniowysokich, których drgania są szczególnie trudne do konwencjonalnych analiz dynamicznych [44].

Metoda modelowania spektrów oparta o SSN może być atrakcyjnym narzędziem w stosunku do wyznaczenia spektrów na podstawie pomiarów w budynkach, wykonywanych za pomocą kosztownych urządzeń pomiarowych czy konieczności stałego monitorowania budynków, w oczekiwaniu na występujące wstrząsy. Także pewne cechy SSN jako systemu modelowania (por rozdz. 2) mogą stanowić o wyjątkowej ich przydatności w analizie wielkości często obarczonych błędami pomiarowymi.

1. Dane pomiarowe

Dane w formie zdyskretyzowanych PSO, zaczerpnięto z [50]. Stanowiły one zbiór PSO, obliczonych na podstawie pomiarów wartości przemieszczeń wykonanych dla 5-cio piętrowych ścianowych budynków prefabrykowanych na terenie Legnicko-Głogowskiego Okręgu Miedziowego.

Analizowano składowe poziome drgań powierzchniowych, jako najistotniejsze. Mierzono przemieszczenia gruntu w odległości kilku metrów od budynku (w celu zminimalizowania wpływu drgań budynku na drgania gruntu w miejscu przeprowadzenia pomiaru) oraz przemieszczenia budynków, rejestrowane na fundamentach lub okienkach ścian nośnych piwnic. PSO obliczano jak w przypadku analizy ASO, przyjmując współczynnik tłumienia równy $\xi = 2\%$. Para spektrów odnosi się do tego samego budynku, przy czym PSO_g odnosi się do pomiaru na zewnątrz budynku na poziomie gruntu, PSO_f odnosi się do drgań fundamentu analizowanego budynku, por. rys. 5.9.



Rys.5.9. Ilustracja sposobu wykonywania pomiarów

Łącznie posługiwano się 12-ma parami (PSOg, PSOf). Spektra od drgań zarejestrowanych na terenie LGOM, różnią się znacznie od spektrów obliczonych na podstawie drgań zarejestrowanych na terenie GZW [40], $PSOf, PSOg \in [0.1, 133.83]\mu m$.

Maksymalna zarejestrowana wartość przyspieszenia gruntu w przypadku drgań poziomych wyniosła $a_{max} \approx 1m/s^2$ [44].

W odróżnieniu od problemu poprzedniego, nie dysponowano energią wstrząsu i odległością epicentralną poszczególnych pojedynczych wstrząsów.

2. Wzorce uczące i testujące dla SSN

Pojedyncze Przemieszczeniowe Spektrum Odpowiedzi składa się z 198 punktów postaci:

$$Sd(k) = Sd(T_k) \quad (5.13)$$

dla $k = 1, 2, \dots, 198$, gdzie Sd jest odpowiednią wartością zdyskryzowanego PSO. 50% losowo wybranych par spektrów przeznaczono do uczenia SSN, pozostałe 50% przeznaczając do testowania SSN nauczonej. Otrzymano w ten sposób: $L = 6 \cdot 198 = 1188$ wzorców do uczenia SSN oraz $T = 6 \cdot 198 = 1188$ wzorców do testowania. Na potrzeby algorytmu obliczeniowego, dane wejściowe dla SSN przeskalowano do wartości zawierających się w przedziale $[0,1]$. Spektra od pomiarów na gruncie przy budynkach różnią się znacznie od spektrów od pomiarów w piwnicach, [44].

3. Przyjęte SSN, algorytm uczenia, wyniki neuronowej symulacji PSO

Zastosowano SSN o architekturze FLNN 2-5-1 z autoregresyjnym sprzężeniem wyjścia SSN – z wejściem. Wektor wejściowy SSN przyjęto w postaci:

$$\mathbf{x} = \{Sdg(T_{k-1}), Sdf(T_{k-1})\} \quad (5.14)$$

Indeksem 'g' oznaczono zdyskryzowane wartości SO gruntu, PSOg, natomiast indeksem 'f' - odpowiednie wartości SO dla fundamentu budynku. Na wyjściu z sieci oczekiwano wartości:

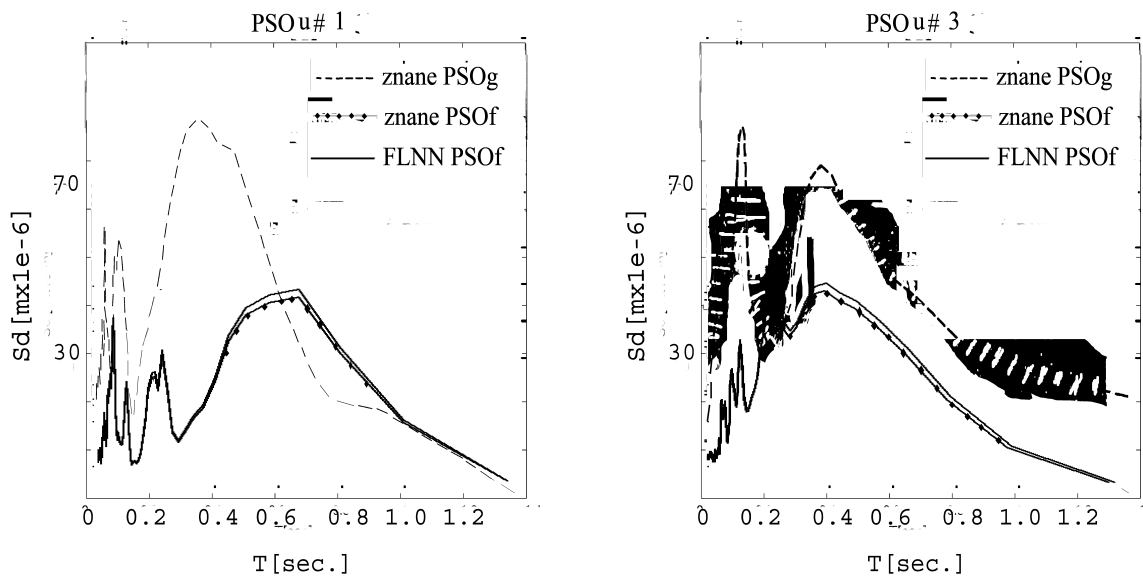
$$Sdf(T_k) \quad (5.15)$$

SSN uczono z wykorzystaniem rozprężonego według poszczególnych neuronów algorytmu NDEKF, wykorzystującego Rozszerzony Filtr Kalmana dla wyjścia z sieci, jako wielkości obserwowalnej (por. rozdz. 4). W modelu teoretycznym (4.13)-(4.14) wektor procesu $\{w^i, v^i\}$ przyjęto postaci: $w^i \in R^2$, oraz $v^i \in R$ dla neuronów $i = 1, \dots, 15$ warstwy ukrytej, $w^i \in R^6$ oraz brak wektora sprzężenia zwrotnego dla wyjścia.

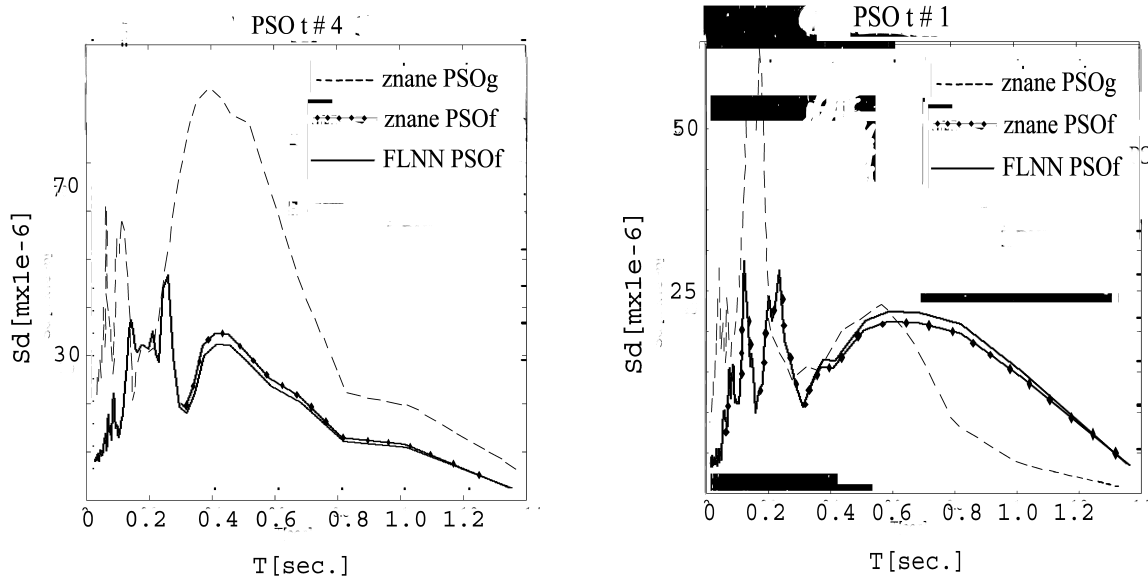
W tab. 5.4. zestawiono błędy uczenia $MSEU$ oraz testowania $MSET$ i wartości współczynnika korelacji liniowej r_T dla zbioru testującego, dla wielkości Spektrow przeskalowanych do przedziału $[0.1, 0.9]$, dla liczby epok uczenia S . Na rysunku 5.11 przedstawiono odtworzenie dwóch wybranych SO uczących. Na rys. 5.12 pokazano wybrane SO testowe, symulowane przez SSN.

SSN/algorytm	S	$MSEU \cdot 10^6$	$MSET \cdot 10^6$	r_T	$st\varepsilon$
FLNN/NDEKF 2-5-1	300	3.90	6.50	0.9582	0.0318
6-5-1/Rprop	7000	137	125	0.876	0.036

Tab 5.4. Błędy uczenia oraz testowania dla FLNN 2-6-1, uczonej NDEKF i Rprop dla wektora wejścia postaci (5.15)



5.11. Symulacja PSO, wybrane Spektra uczące



5.12. Predykcja PSO, wybrane Spektra testowe

Widoczna jest bardzo dobra zgodność SO symulowanych przez SSN ze spektrami pomiarowymi.

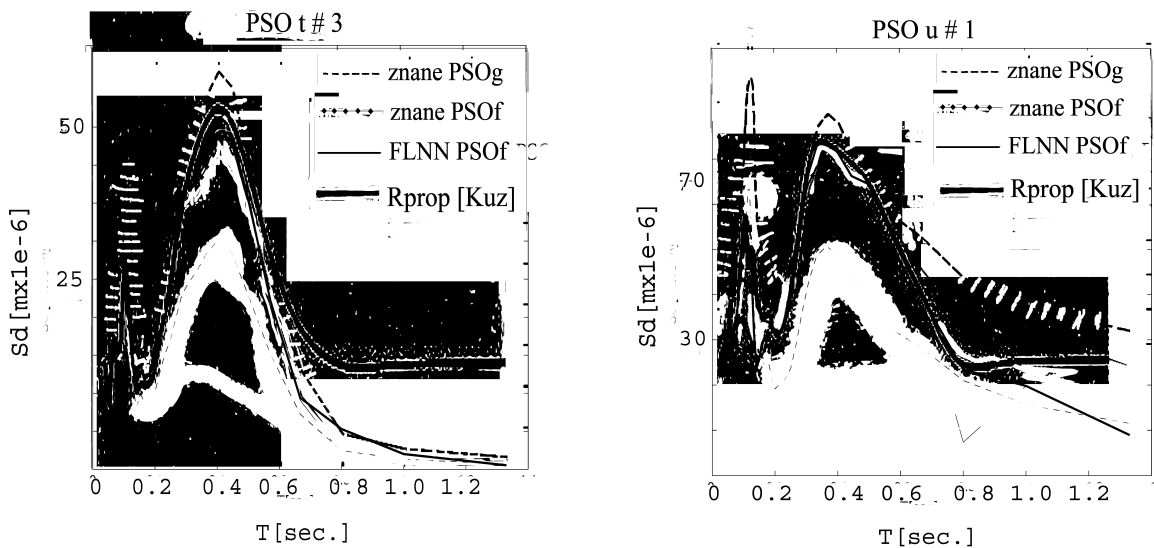
5. Porównanie otrzymanych wyników z dotychczas stosowanymi metodami analizy PSO na terenie LGOM

W pracach [50, 51] analizowano te same grupy spektrów za pomocą SSN uczonych tradycyjnym algorytmem Resilient-propagation z wykorzystaniem symulatora SSN - SNNS [48] oraz sieci neuro-rozmyte - uczone z użyciem systemu ANFIS [52, 53], zaimplementowanym w symulatorze Fuzzy Logic Toolbox MATLAB [54]. W obu przypadkach wykorzystano ideę okien czasowych (ramek) do konstrukcji wektora wejścia, przyjmując go w postaci:

$$\mathbf{x} = \{Sdg(T_{k-2}), Sdg(T_{k-1}), Sdg(T_k), Sdg(T_{k+1}), Sdg(T_{k+2}), T_k\} \quad (5.15)$$

Za najefektywniejszą uznano sieć o architekturze 6-5-1. Wyniki uczenia i testowania, w zakresie uzyskanych błędów, dla liczby epok uczenia S przedstawiono w tab. 5.4. Dla sieci ANFIS otrzymane wyniki są praktycznie identyczne.

Różnice w dokładności predykcji spektrów dla drgań w piwnicy dla sieci uczonych metodą FK (krzywa oznaczona DEKF) oraz Rprop w pracy [51] (krzywa oznaczona Rprop[Kuz]) przedstawiono na rys 5.13. Dopasowanie do krzywych pomiarowych jest dokładniejsze w przypadku SSN uczonych z wykorzystaniem FK. Widoczne to jest szczególnie w zakresie wyższych okresów drgań.



5.13. Porównanie symulacji spektrów za pomocą SSN i metod FK oraz Rprop, wybrane Spektrum uczące

5. Wnioski końcowe

- Zastosowanie filtru Kalmana w postaci algorytmu NDEKF otwiera nowe możliwości poprawienia jakości i dokładności predykcji akceleracyjnych (przyspieszeniowych) i przemieszczeniowych spektrów odpowiedzi od wymuszeń parasejsmicznych.
- Wykazano, że posłużenie się autokorelacyjnym wejściem, które odpowiada rzędnej spektrum z jedną chwilą opóźnienia, istotnie polepsza wyniki aproksymacji neuronowej, zarówno przy stosowaniu uczenia sieci neuronowych za pomocą algorytmu DEKF, jak też metodą Rprop.
- Prosta SSN, nauczona za pomocą Filtru Kalmana, umożliwia łatwe obliczanie PSO dla interakcji budynek-podłoże.
- Algorytm NDEKF, w połączeniu z siecią rekurencyjną w obrębie każdego z neuronów wydaje się być dużo bardziej efektywny, w zadaniu predykcji ASO w oparciu o energię wstrząsu i odległość epicentralną, niż algorytm NDEKF zastosowany do uczenia sieci neuronowej jednokierunkowej w obrębie sygnału przekazywanego przez poszczególne neurony.
- Nie wykazano takiej poprawy wyników dla zadania modelowania wybranych spektrów dla problemu interakcji.

- Zastosowanie filtra Kalmana do uczenia SNN warstwowej, wraz z zastosowaniem autoregresyjnego wejścia, pozwala znacznie zwiększyć dokładność symulacji spektrów odpowiedzi analizowanych budynków, w porównaniu z dotychczas stosowanymi metodami analizy tych zagadnień z wykorzystaniem SSN. Wydaje się być także skuteczniejsza, w porównaniu z standardowymi, nie opartymi o SSN, metodami analizy SO.

5.2. Predykcja pętli histerezy od obciążeń niskocyklicznych

5.2.1. Predykcja pętli histerezy dla kompozytu hierarchicznego złożonego z wiązek nadprzewodników

Zajęto się problemem modelowania pętli histerezy w układzie: siła – przemieszczenie dla przewodu wielożyłowego, złożonego z wiązek nadprzewodników, poddanego działaniu cyklicznie zmiennej siły.

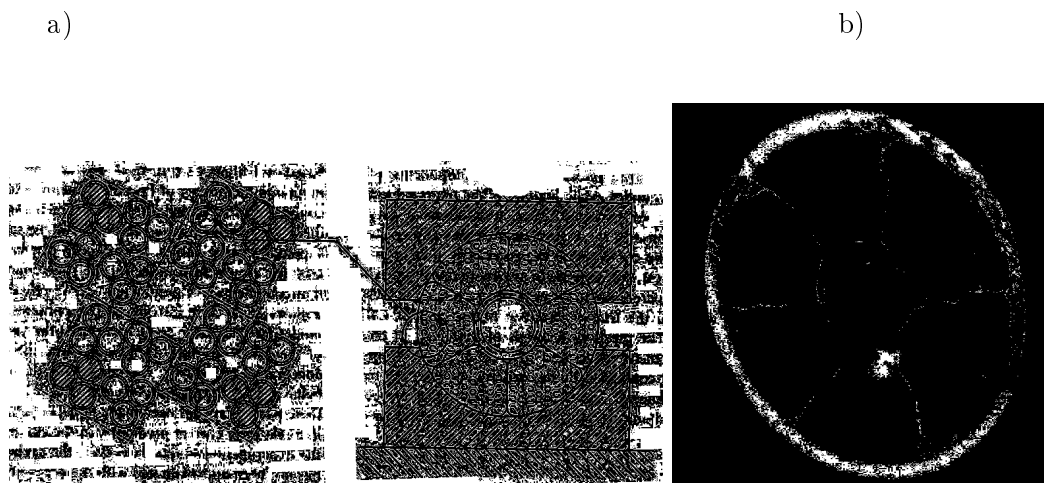
Badany przewód, o bardzo złożonej budowie i właściwościach mechanicznych, został zaprojektowany jako składnik cewki Central Solenoid CS1 (cewka indukcyjna utworzona przez przewód nawinięty na rdzeń w kształcie walca), wykorzystanej jako składnik systemu magnesów nadprzewodzących w powstającym w Cadarache (Aix-en-Provence, Francja) Międzynarodowym Eksperymentalnym Reaktorze Termonuklearnym (International Thermonuclear Experimental Reactor, ITER [55]). Podczas testów na obciążenie, przeprowadzonych na przewodzie w skali naturalnej, badano jego odpowiedź na siłę działającą w płaszczyźnie poprzecznej. Dane doświadczalne uzyskane w ten sposób wykorzystane zostały do zaprojektowania i nauczania sieci neuronowych, odtwarzających zachowanie się tego materiału pod wpływem przyłożonej siły oraz dokonujących predykcji wybranych fragmentów eksperymentu, por. [56].

1. Rozważany materiał, motywacja badań, przeprowadzony eksperyment

Przewód ma konstrukcję wielożyłową: w stalowej, prostokątnej osłonie (na poziomie 'makro') znajduje się 6 głównych żył kabli nadprzewodzących oraz system chłodzący, w formie centralnej rury chłodzącej. Główne żyły kabla posiadają własne osłony. Każda żyła przewodu zawiera ok. 1000 włókien nadprzewodników niobowo-cynowych, Nb₃Sn, o średnicy ok. 0.8mm każdy, zatopionych w przewodzie miedzianym. Włókna skupione są w heksagonalne klastry, które z kolei po ok. 80 sztuk wchodzi w skład pojedynczego drutu. Druty splatane są w triplety, a triplety w czwórki. Materiał ma więc budowę hierarchiczną wg. pięcio-stopniowej skali, por. rys.5.12.a,b. Zachowanie mechaniczne tego materiału, ze względu na nieznaną właściwości mechanicznych pojedynczych drutów oraz na złożony charakter oddziaływania poszczególnych elementów, potraktowany jest jako materiał jednorodny w makro-skali [56].

Motywację do podjęcia badań, stanowiły powstające pod wpływem siły Lorentza efekty mechaniczne, zaobserwowane podczas pracy przewodu. Powstające pole elektromagnetyczne powoduje poprzeczne obciążenie przewodu oraz szereg zjawisk mechanicznych. Dodatkowo stwierdzono, iż procesy te zależą od kąta skręcenia włókien w wiązkach, siły nacisku oraz historii nacisku.

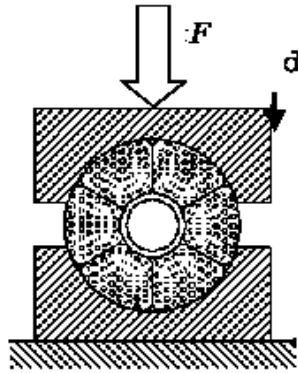
Efekty te są przyczyną powstawania m.in. zmian we właściwościach magnetycznych przewodu, które wpływają istotnie na jakość jego eksploatacji [57].



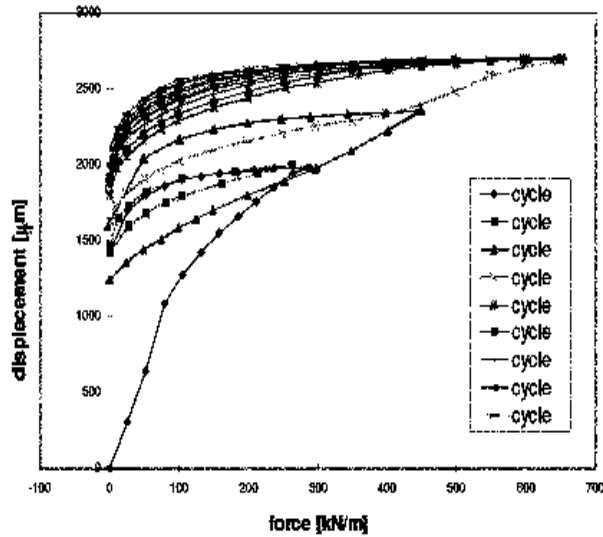
Rys.5.12.a) Schemat przekroju poprzecznego przewodu, widoczna chierarchiczna struktura przewodu, b) fragment przewodu

Magnesy nadprzewodzące, których elementem konstrukcyjnym jest rozważany przewód, są usytuowane w ekranowanym termicznie termostacie niskotemperaturowym, zapewniającym odpowiednią do uzyskania efektu nadprzewodnictwa temperaturę [55]. Eksperyment wykonano więc w temperaturze właściwej dla pracy przewodu, na prasie kriogenicznej (Twente Cryogenic Press [57]) w środowisku ciekłego helu. Próbkę przewodu w skali naturalnej obciążano cyklicznie zmienną siłą poprzeczną F [kN/m] w temperaturze 4.2K ($-268.96^{\circ}C$). Rozważano płaski stan przemieszczenia d [μm], w płaszczyźnie prostopadłej do osi przewodu. Znak przemieszczenia wywołanego siłą ściskającą przyjęto jako dodatni. Wykonano 38 cykli obciążania, a następnie odciażania, z początkowo wzrastającą siłą maksymalną F , następnie ustaloną na poziomie ok. 650 kN/m. Przez pojedynczą pętlę obciążenia rozumiemy obciążenie materiału od stanu $F = 0$, z siłą rosnącą do wartości maksymalnej w obrębie cyklu, następnie odciażanie poprzez zmniejszanie wartości działającej siły, aż do otrzymania wartości $F = 0$. Mierzono przemieszczenie górnej części stalowej osłony, względem dolnej części. Czujniki zamocowano w 6 punktach osłony. Przemieszczenie przyjęto w postaci zagregowanej: jako wartość średnią odczytów. Maksymalna przyłożona siła wynosiła: 655.4 kN/m, por.rys. 5.14.

Pierwsze 3 cykle różnią się znacznie od pozostałych. Zaobserwowano duże nieodwracalne osiadanie przewodu w początkowej fazie eksperymentu. Także pierwsze 11 cykli różni się od pozostałych 27. Różnice pomiędzy cyklami 1-11 są znaczące, natomiast pomiędzy cyklami 27-38 są dużo mniejsze. Kolejne wartości siły oraz odpowiadające im przemieszczenia przedstawiono na rys.5.14.



Rys.5.13. Schemat eksperymentu



Rys.5.14. Wyniki doświadczenia, za [57]

2. Wzorce uczące, testujące, rozważane typu wejścia do SSN oraz architektury sieci

Do analizy za pomocą SSN wybrano 9 reprezentatywnych pętli eksperymentalnych o numerach: 1,..6,11,21 oraz 38. Poszczególne pętle różnią się liczbą pomiarów, oraz siłą maksymalną. Otrzymano 245 par postaci

$$(F(k), d(k)) \quad (5.14)$$

gdzie

$$k = 1, 2, \dots, 245$$

$F(k) \in [-2.1, 6.554e + 002][kN/m]$ – przyłożona siła (dla obciążania $F(k) < F(k + 1)$, dla odciążania $F(k) > F(k + 1)$)

$d(k) \in [0, 2711][\mu m]$ – przemieszczenie wywołane siłą F ,

k – indekse pseudoczasu, numerujący kolejne, następujące po sobie wartości przyłożonej siły i przemieszczenia.

Jako preprocesing danych dla SSN, unormowano wartości sił i przemieszczeń do przedziału $[0,1]$, przez podzielenie odpowiednio wszystkich wartości przez największą wartość w ciągu sił i przemieszczeń.

Rozważano trzy typy podziału dostępnych wzorców na zbiory uczący i testujący, przyjmując:

- a) Testowanie na wylosowanych 25% lub 50% punktów $(F(k), d(k))$ na pętłach histerezy, dających punkty zbiorów testującego i uczącego rozmieszczone nieregularnie. W przypadku wyboru 25% wzorców do testowania otrzymano $T=61$ wzorców do testowania i $L=244-61=183$ wzorce do uczenia SSN.
- b) Siedem pierwszych pętli, zawierających $L=180$ wzorców, dla $k = 1, 2, \dots, 180$, przeznaczono do uczenia i $T=244-180=64$ wzorców, dla $k = 180, \dots, 244$ do testowania (testowanie na dwóch ostatnich pętłach histerezy, zbiór ten stanowi ok. 26% dostępnych wzorców),
- c) Dwie pierwsze pętłe histerezy, zawierające $T=35$ wzorców, dla $k = 1, 2, \dots, 35$ przeznaczono do testowania (ok. 13% dostępnych wzorców), pozostałe $L=244-35=209$ wzorców do uczenia SSN.

Mimo, iż zbiór uczący w punkcie c) stanowi większy procent dostępnego zbioru wzorców, niż w pozostałych przypadkach, jednak pętłe testowe w tym przypadku bardzo różnią się od wszystkich pozostałych pętli, gdyż dotyczą zaobserwowanego we wstępnej fazie eksperymentu nieodwracalnego osiadania przewodu. W punkcie b) zaproponowanego podziału, modelowane jest zachowanie materiału w dalszej części eksperymentu, na podstawie wprowadzonej do SSN wiedzy o jego początkowej fazie. Punkt c) to próba modelowania zachowania się materiału we wstępnej fazie (podczas zaobserwowanego mniej regularnego zachowania się materiału), gdy zbiór uczący dla SSN stanowią pętłe, dotyczące reakcji materiału na przykładaną siłę w dalszej części eksperymentu.

Zaproponowano szereg wektorów wejścia dla SSN, mającej modelować przeprowadzony eksperyment, dla $k = 1, 2, \dots, 244$:

$$\mathbf{x} = [F(k), d(k), F(k + 1)] \quad (5.15)$$

$$\mathbf{x} = [F(k), d(k), F(k + 1), 1 - k/244] \quad (5.16)$$

$$\mathbf{x} = [F(k), F(k + 1), 1 - k/244] \quad (5.17)$$

$$\mathbf{x} = [d(k), F(k + 1), 1 - k/244, l(k)] \quad (5.18)$$

gdzie składowe wektorów wejścia stanowią:

$k, k + 1$ ($k = 1, 2, \dots, 244$) – bierzący i kolejny parametr pseudoczasu,

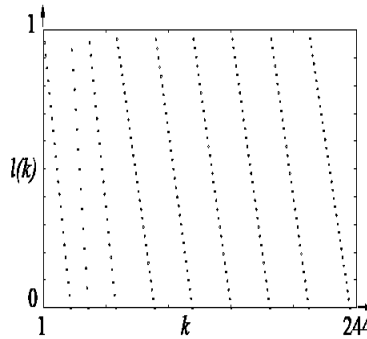
$F(k), F(k + 1)$ – wartości przyłożonej siły,

$1 - k/244$ malejący ciąg liczb dotyczący zaawansowania eksperymentu.

Dla k właściwego każdej z Pętli histerezy osobno, parametr $l(k)$ jest skojarzony z liczbą pomiarów w danej pętli, niezależnie od pętli pozostałych. Przyjmuje on kolejno, w równych odstępach unormowane wartości od jeden do:

$$1 - 1/N(1), 1 - 2/N(2), \dots, 1 - N(l)/N(l) \quad (5.19)$$

gdzie $N(l)$ oznacza liczbę pomiarów przeprowadzonych w l -tym cyklu. Wartość parametru l jest zgodna z kolejnością cykli, por. rys. 5.15.



Rys.5.15. Wartości parametru $l(k)$

Na wyjściu przyjęto wartość przemieszczenia wywołanego siłą $F(k + 1)$, czyli wielkość $d(k + 1)$.

Rozważano wiele grup SSN, różniących się ilością warstw, liczbą neuronów w poszczególnych warstwach oraz kierunkiem przepływu sygnału wewnątrz sieci:

- Jednowarstwowe i dwuwarstwowe, jednokierunkowe i o sprzężeniu autoregresyjnym wyjścia z wejściem, typu FLNN. Tam, gdzie wynika to z konstrukcji wektora wejścia SSN, zawracany na wejście SSN sygnał stanowiło opóźnione o jeden krok czasowy wyjście z SSN, por. rys. 2.4., Dla neuronów warstwy ukrytej przyjęto sigmoidalne funkcje aktywacji, zob. (2.4), dla wyjścia liniową funkcję aktywacji zob. (2.5) (por. Rozdz. 2.)
- Jednowarstwowe, rekurencyjne, typu RLNN (por. Rozdz. 2). Rekurencja występuje w obrębie neuronów warstw ukrytych i neuronu wyjściowego: sygnał wyjściowy z danego neuronu (stanowiący jego potencjał, przekształcony przez

funkcje aktywacji) jest z opóźnieniem czasowym jednego kroku rekurencyjnie zawracany do tego neuronu, por. rys.2.6. Funkcje aktywacji neuronów przyjęto jak dla FLNN.

- W obrębie powyższych sieci o różnej liczbie parametrów.

Sieci uczono z wykorzystaniem rozprężonego według poszczególnych neuronów algorytmu NDEKF, wykorzystującego Rozszerzony Filtr Kalmana dla wyjścia z sieci, jako wielkości obserwowalnej (por. rozdz. 4.). W modelu teoretycznym (4.13)-(4.14) wektor procesu przyjęto postaci $\{\mathbf{w}^i, v^i\}$, dla neuronów w których występował sygnał zawracany rekurencyjnie oraz $\{\mathbf{w}^i\}$ dla neuronów, do których sygnał dochodził jedynie zgodnie z kierunkiem wejście-wyjście SSN. Funkcja h , odzwierciedlająca struktury przyjętych SSN, jest postaci (2.9)-(2.14).

Dla wszystkich SSN przyjęto parametry szumów:

$$R_k = \exp((s - 1)/50) \quad (5.20)$$

dla równań obserwacyjnych oraz:

$$\mathbf{Q}_k = \exp((s - 1)/50)\mathbf{I} \quad (5.21)$$

dla równań procesu, gdzie

s – numer bieżącej epoki uczenia sieci

\mathbf{I} – macierz identycznościowa, której wymiar zależy od ilości połączeń dochodzących do neuronu, dla którego konstruowany jest model teoretyczny, por. Rozdz. 4.

Uczenie rozpoczęto od początkowych wartości parametrów SSN, wylosowanych z przedziału $[-0.5, 0.5]$, zgodnie z rozkładem jednostajnym. Proces uczenia kontrolowano poprzez obserwację spadku poziomu błędów średniokwadratowych uczenia $MSEU$ oraz testowania $MSET$, por. Rozdz.2, (2.21). Zastosowano kryterium zatrzymania algorytmu uczącego, oparte o określoną ilość epok uczenia SSN. Proces uczenia każdej SSN powtarzano stukrotnie, dla różnych zestawów wartości parametrów początkowych sieci.

3. Wyniki neuronowego modelowania zachowania materiału pod wpływem przyłożonej siły dla różnych wektorów wejściowych

Wyniki neuronowego modelowania przedstawiono w tabelach 5.5, 5.6, 5.7. Po wstępnych testach, przyjęto kryterium stopu algorytmu uczenia na $S=500$ epok. Wartości błędów uczenia i testowania zostały otrzymane po 500 epokach uczenia SSN.

Dodatkowa składowa wektora wejściowego $1 - k/244$, związana z numerem bieżąco przetwarzanego wzorca, pozwoliła podnieść dokładność zarówno odtworzenia zbioru

uczącego, jak i predykcji neuronowej w obrębie wzorców testowych. Przykładowo, dla sieci FLNN 4-4-5-1, otrzymano znacznie niższe błędy $MSEU$ oraz $MSET$, niż dla sieci FLNN 3-4-5-1, mającej taką samą liczbę neuronów w warstwach ukrytych, lecz nie posiadającej dodatkowej składowej w wektorze wejścia.

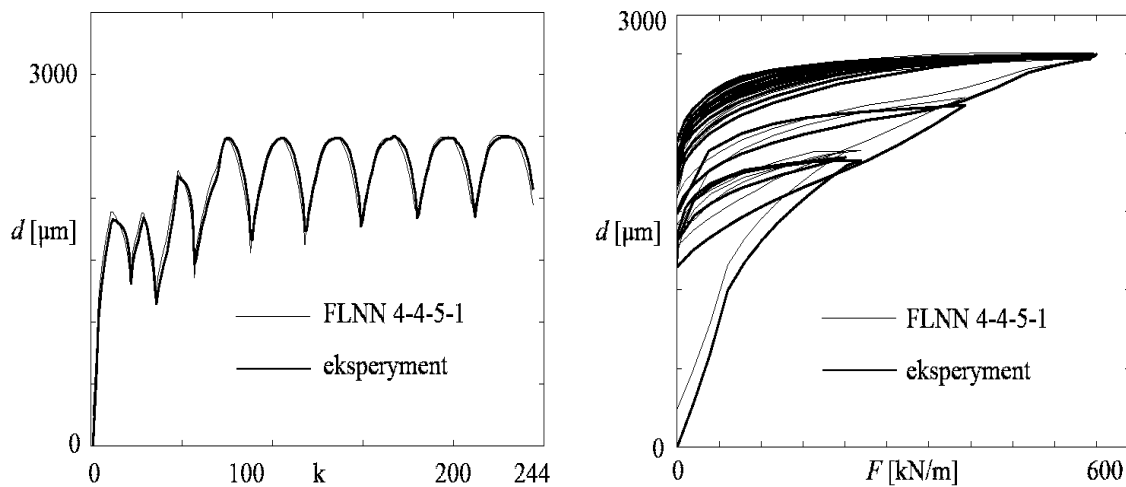
Najlepsze rezultaty otrzymano w przypadku zastosowania jednocześnie obydu znaczników $1 - k/244$ oraz $l(k)$.

3.1. Przypadek a) wyboru zbioru uczącego

Na rys. 5.16 przedstawiono wyniki uczenia i testowania, w przypadku testowania na 25% wybranych losowo z całej historii obciążenia punktów. Zaprezentowane rezultaty dotyczą SSN najbardziej efektywnej numerycznie. Zachowanie się materiału pod wpływem przyłożonej siły zostało oddane poprawnie.

Nr	SSN	Wejście	$MSEU * 10^3$	$MSET * 10^3$
1 (25%)	FLNN 4-4-5-1	$[d(k), F(k+1), 1 - k/244, l(k)]$	0.27	0.37
1 (50%)	FLNN 4-4-5-1	$[d(k), F(k+1), 1 - k/244, l(k)]$	0.36	0.30

Tabela 5.5. Zestawienie wyników obliczeń – testowanie na losowo wybranych punktach histerezy



Rys.5.16 Test nr 1. Pętle histerezy eksperymentalne vs symulowane przez SSN, testowanie na 25 % losowo wybranych wzorców

3.2. Przypadek b) wyboru zbioru uczącego

Porównywano wyniki symulacji, otrzymane za pomocą jedno i dwuwarstwowych sieci typu FLNN i RLNN, por. tab. 5.6. Najlepsze rezultaty otrzymano w przypadku

zastosowania wektora wejścia postaci:

$$\mathbf{x} = [1 - k/244, d(k), l(k), F(k + 1)] \quad (5.22)$$

por. test nr 8. Przypadek 8, sieć typu FLNN 4-4-5-1, o licznie parametrów $W = 57$ wydaje się być najbardziej efektywna. SSN, nauczona na danych dotyczących zachowania się przewodu w początkowej fazie eksperymentu, poprawnie odtwarza odpowiedź materiału na przyłożoną siłę w dalszej części doświadczenia, gdy jedyną informacją podawaną "z zewnątrz" są wartości przykładanej siły.

Zastosowanie sieci bez autoregresyjnego wejścia, por. test nr 9, daje bardzo duże błędy uczenia i testowania.

Osiągnięte poziomy błędów średniokadratowych MSE uczenia, są wyższe, niż w przypadku a), gdyż pętle o mniejszej regularności wchodziły w całości w skład zbioru uczącego. Błędy testowania w przypadkach a) i b) otrzymano podobnego rzędu.

Nr	W	SSN	Wejście	$MSEU * 10^3$	$MSET * 10^3$
2	92	RLNN 3-15-1	$[F(k), d(k), F(k + 1)]$	0.86	0.35
3	76	FLNN 3-15-1	$[F(k), d(k), F(k + 1)]$	0.88	0.46
4	146	FLNN 3-15-5-1	$[F(k), d(k), F(k + 1)]$	0.90	0.32
5	49	FLNN 3-6-3-1	$[F(k), d(k), F(k + 1)]$	0.94	0.44
6	57	FLNN 4-4-5-1	$[F(k), d(k), F(k + 1), 1 - k/244]$	0.78	0.35
7	63	RLNN 4-4-5-1	$[F(k), d(k), F(k + 1), 1 - k/244]$	0.88	0.37
8	57	FLNN 4-4-5-1	$[d(k), F(k + 1), 1 - k/244, l(k)]$	0.39	0.35
9	57	FLNN 4-4-5-1	$[F(k), F(k + 1), 1 - k/244, l(k)]$	46	1.8

Tabela 5.6. Zestawienie wyników obliczeń - testowanie na dwóch ostatnich pętlach histerezy

3.2. Przypadek c) wyboru zbioru uczącego

Do testowania SSN przeznaczono dwie pierwsze pętle histerezy. SSN uczone były więc w oparciu o pętle bardziej regularnych i testowane na pętlach odbiegających znacznie od tych, ze zbioru uczącego. Z tego powodu otrzymano wyższe błędy testowania i niższe błędy uczenia niż w przypadkach a) oraz b).

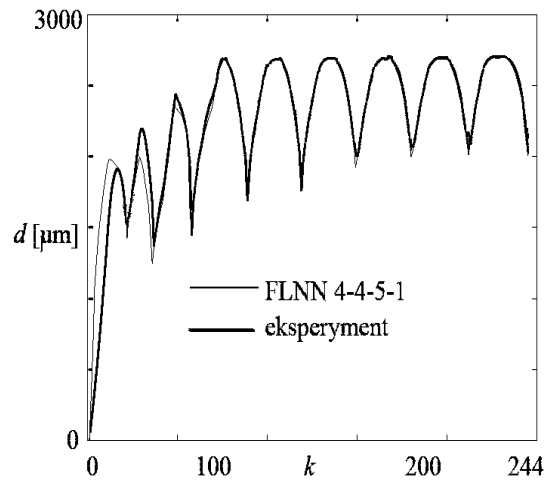
Nieregularne w stosunku do dalszej fazy eksperymentu zachowanie się materiału pod wpływem przyłożonej siły dla pierwszych cykli obciążenia zostało oddane poprawnie, por. 5.17. Podobnie jak w przypadkach a) oraz b) najlepsze rezultaty otrzymano w przypadku zastosowania wektora wejścia postaci:

$$\mathbf{x} = [1 - k/244, d(k), l(k), F(k + 1)] \quad (5.23)$$

por. test nr 16.

Nr	SSN	Wejście	$MSEU * 10^3$	$MSET * 10^3$
10	RLNN 3-15-1	$[F(k), d(k), F(k+1)]$	0.53	2.03
11	FLNN 3-15-1	$[F(k), d(k), F(k+1)]$	0.671	2.09
12	FLNN 3-15-5-1	$[F(k), d(k), F(k+1)]$	0.674	2.03
13	FLNN 3-6-3-1	$[F(k), d(k), F(k+1)]$	0.69	2.1
14	FLNN 4-4-5-1	$[F(k), d(k), F(k+1), 1 - k/244]$	0.78	1.90
15	RLNN 4-4-5-1	$[F(k), d(k), F(k+1), 1 - k/244]$	0.67	1.92
16	FLNN 4-4-5-1	$[d(k), F(k+1), 1 - k/244, l(k)]$	0.26	1.0

Tabela 5.7. Zestawienie wyników obliczeń - testowanie na dwóch ostatnich pętach histerezy



Rys.5.17. Test 16. Pętle histerezy eksperymentalna vs symulowane przez SSN, testowanie na dwóch pierwszych pętach histerezy

3.3. Wpływ architektury SSN na dokładność symulacji neuronowej

Wykonane symulacje pozwalają na stwierdzenie, że dokładność aproksymacji neuronowej nie jest silnie związana z ilością neuronów w warstwach ukrytych sieci, por. tab. 5.6, 5.7.

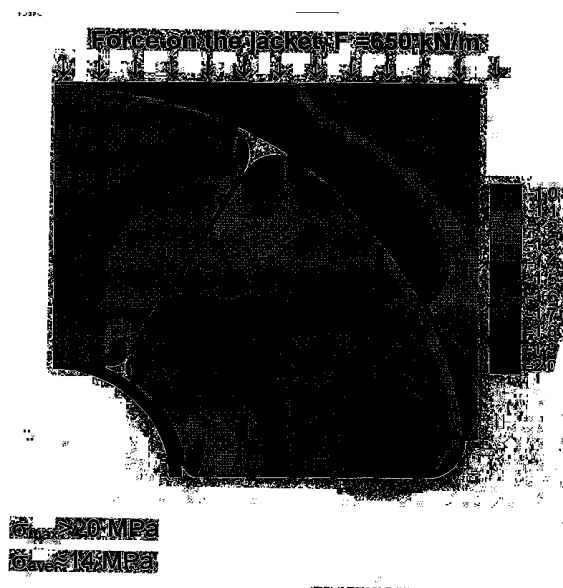
Dodanie rekurencyjnego połączenia w obrębie neuronów warstw ukrytych i wyjściowej sieci nie podniosło znacząco dokładności symulacji neuronowej w przypadku sieci o niewielkiej liczbie parametrów, por. testy nr 6,7 i 14,15 oraz pozwoliło obniżyć uzyskany błąd testowania dla sieci dużej, por. testy 2,3 i 10,11. Zastosowanie SSN o znacznie większej liczbie parametrów nie pozwoliło obniżyć błędu uczenia w stosunku do sieci mniejszych, o tym samym wektorze wejściowym, por. testy 4,5 i 12,13.

Zastosowanie SSN o niewielkiej liczbie parametrów, w przypadku wszystkich przyjętych zbiorów uczących i testowych, pozwoliło na otrzymanie symulacji neuronowej

oddającej charakter zachodzącego zjawiska fizycznego. Pozwoliło także skrócić czas uczenia sieci.

4. Porównanie neuronowego modelowania eksperymentu z otrzymanym za pomocą modeli klasycznych.

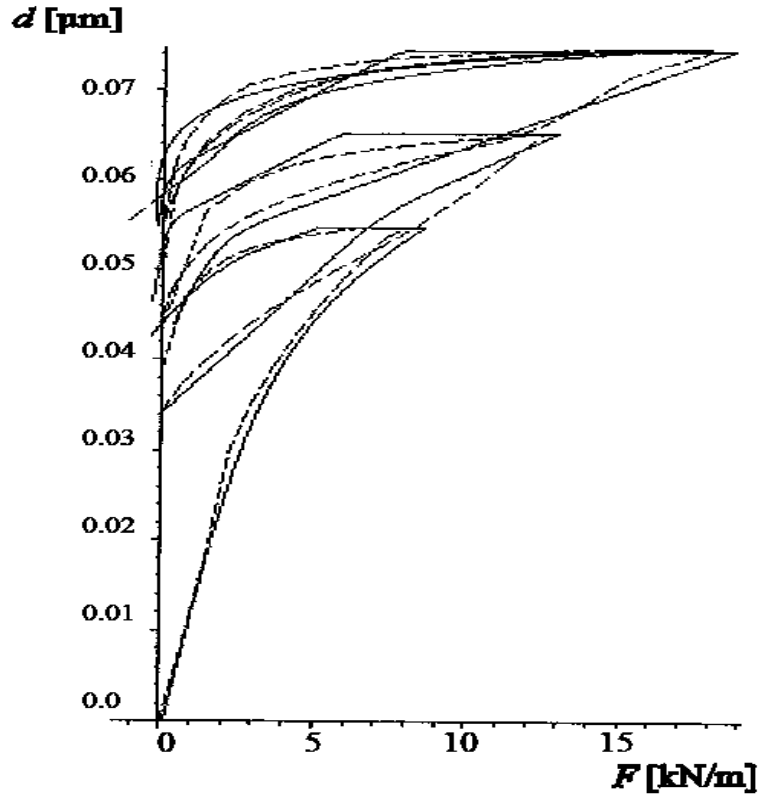
W [57] materiał modelowano wstępnie Metodą Elementów Skończonych, bez uwzględnienia tarcia pomiędzy elementami, jako materiał sprężysty. Otrzymano bardzo zróżnicowane wartości naprężeń wewnątrz przewodu, wynikające z jego segmentowej konstrukcji. Rozkład naprężeń przedstawiony na rys.5.18.



Rys.5.18. Rozkład naprężeń, FEM, model wstępny-uproszczony, za [57]

Rozważany materiał został zamodelowany w [58] jako sprężysto-plastyczny, uogólniony materiał Prandtla. Dla każdej z pętli obciążenia model ten zależy od wielu parametrów, których liczba zwiększa się dla każdej kolejnej fazy eksperymentu. Ich wartości różnią się, zarówno w zakresie obciążanie-odciążanie, jak i przy przejściu do kolejnych pętli histerezy. Z tego powodu problem znalezienia modelu, opisującego zachowanie się tego materiału uznano za trudny i pracochłonny. Za najdokładniejszy uznano w [59] model, w którym ww. wielkości, dla każdej z pętli osobno oraz po rozdzieleniu na fazę obciążania i odciążania – identyfikowano na podstawie wyników doświadczeń za pomocą pomocy SSN.

Rezultaty identyfikacji modelu przedstawiono na rys.5.20. Otrzymano model poprawnie opisujący zachowanie się przewodu pod wpływem przyłożonej siły, jednak z nieuzasadnionymi eksperymentalnie zmianami wartości pochodnej przemieszczenia w połowie cyklu odciążania. Zjawisko to nie występuje dla niejawnego modelu, otrzymanego za pomocą Sieci Neuronowych, por. rys. 5.16.



Rys.5.20. Krzywe teoretyczne – linia ciągła, wykres doświadczalny – linia przerywana, za [59]

5. Porównanie neuronowego modelowania eksperymentu za pomocą SSN uczonych FK z otrzymanym za pomocą uczenia SSN metodami klasycznymi.

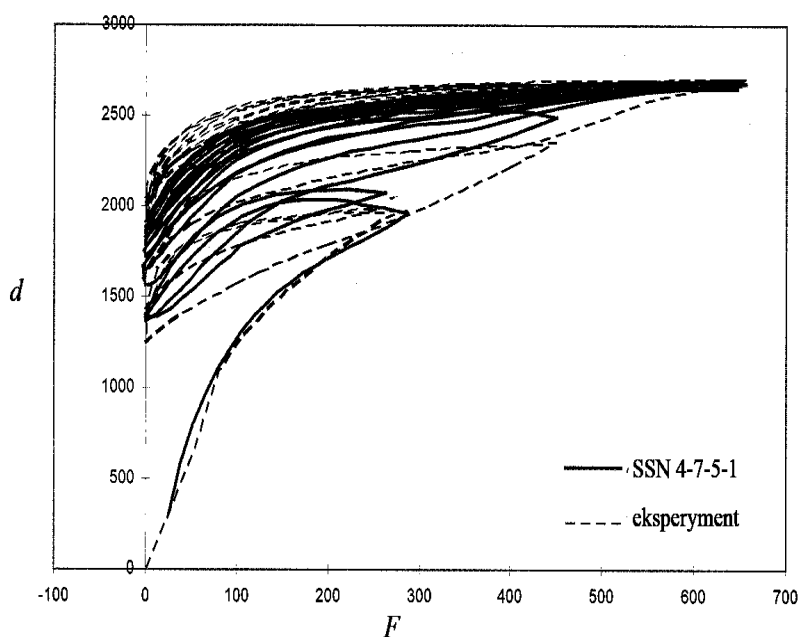
W pracy [56] testowano szereg sieci o niskiej liczbie parametrów: 3-6-3-1, 3-5-4-1, 3-4-5-1, 3-6-5-1, 3-4-6-1. Przyjęto wektor wejścia SSN w postaci (5.15).

Sieci uczone i testowano w oparciu o te same zbiory danych, jednak jako algorytm uczący zastosowano klasyczny algorytm wstecznej propagacji błędów, por. Rozdz. 2. Stwierdzono znaczną różnicę otrzymywanych wyników w obrębie sieci o wymienionych architekturach. Wartości błędów średniokwadratowych uczenia i testowania okazały się wrażliwe na dobór ilości neuronów w poszczególnych warstwach SSN. Przykładowo błąd $RMSE = \sqrt{MSE}$ testowania niższy niż $25 \mu m$ otrzymano dla SSN różniących się o jeden neuron warstwy ukrytej dla 40% wzorców lub 60% wzorców, zależnie od przejętej architektury sieci. Wybrano sieć 3-6-5-1 jako najefektywniejszą. Dokładność odtworzenia danych eksperymentalnych poprawiono, stosując dodatkową

składową wektora wejścia w postaci:

$$e(k) = \sum_{i=1}^k 0.5(F(i+1) - F(i))(d(i+1) - d(i)) \quad (5.24)$$

Również uzyskano duży rozrzut wyników, w zależności od architektury SSN. Rezultaty uczenia i testowania dla sieci najefektywniejszej numerycznie (4-7-5-1) przedstawiono na rys. 5.21. W niniejszej pracy przeprowadzono analogiczną symulację dla sieci dwuwarstwowych, o niewielkiej liczbie parametrów, dla najefektywniejszego znalezionego wektora wejściowego (5.19), uczonych algorytmem Kalmana. Otrzymane rezultaty potwierdziły, iż zastosowanie tej metody uczenia, wraz z przyjętym wektorem wejściowym (5.19), nie jest wrażliwe na dobór architektury SSN: różnice w ramach sieci o różnych architekturach, zarówno dla błędów średniokwadratowych uczenia, jak i testowania można było uznać za pomijalne.



Rys.5.20. Wyniki symulacji neuronowej, za [56]

5. Wnioski końcowe

Zaprezentowane wyniki upoważniają do wyciągnięcia następujących wniosków:

- Zastosowanie filtrów Kalmana, jako metody uczenia SSN pozwoliło na predykcję pętli histerezy rozważanego materiału z dużą dokładnością, za pomocą SSN o niewielkiej licznie parametrów (FLNN 4-4-5-1).

- Metoda uczenia SSN, oparta o filtry Kalmana wydaje się być dużo mniej wrażliwa na wybór architektury SSN, w stosunku do SSN uczonych za pomocą standartowego algorytmu wstecznej propagacji błędów, por. 5
- Zwiększenie rozmiarów SSN oraz wprowadzenie rekurencji w obrębie poszczególnych neuronów SSN nie podnosi efektywności symulacji, konieczne jest jednak wprowadzanie rekurencji w obrębie wyjście-wejście SSN, o charakterze autoregresyjnym.
- Wprowadzenie dodatkowej składowej wektora wejścia $1 - k/244$, zależnej od numeru bieżącego wzorca dla SSN oraz znacznika numeru wzorca w obrębie danej pętli $l(k)$ pozwala uzyskać niższe błędy zarówno uczenia, jak i testowania, w porównaniu z SSN o tej samej architekturze.
- Konstrukcja niejawnego, neuronowego modelu tego materiału, pozwala ominąć etap pracochłonnej konstrukcji modelu matematycznego, w szczególności fazę doboru parametrów, koniecznych do prawidłowego odtworzenia zjawisk fizycznych, dających rozważaną krzywą doświadczalną.
- Wykorzystanie SSN, jako narzędzia obliczeniowego, pozwala modelować eksperyment i przewidywać końcową fazę eksperymentu, poprzedzającą zniszczenie materiały, z zadowalającą dokładnością.

5.2. Predykcja pętli histerezy od obciążeń niskocyklicznych

5.2.2. Symulacja i predykcja pętli histerezy w próbkach stalowych

Niniejszy rozdział dotyczy neuronowej symulacji związku między odkształceniem, a naprężeniem dla cyklicznie obciążanej próbki stali nierdzewnej, [60].

Zadanie podzielono na dwa etapy. Problem pierwszy dotyczy wykorzystania zdolności ekstrapolacyjnych SSN, w celu predykcji zachowania materiału w fazie eksperymentu poprzedzającej bezpośrednio zniszczenie materiału, na podstawie danych dotyczących wcześniejszej jego fazy. W drugim, własności interpolujące SSN wykorzystane zostały w celu wyznaczenia wartości naprężeń na podstawie wartości odkształceń, także dla naprężeń nie wchodzących w skład zbioru uczącego, w losowo wybranym punkcie historii obciążenia. Rozdział zakończy porównanie rezultatów modelowania przeprowadzonego eksperymentu za pomocą SSN, z modelowaniem opartym o klasyczne podejście mechaniki zniszczenia ośrodka ciągłego.

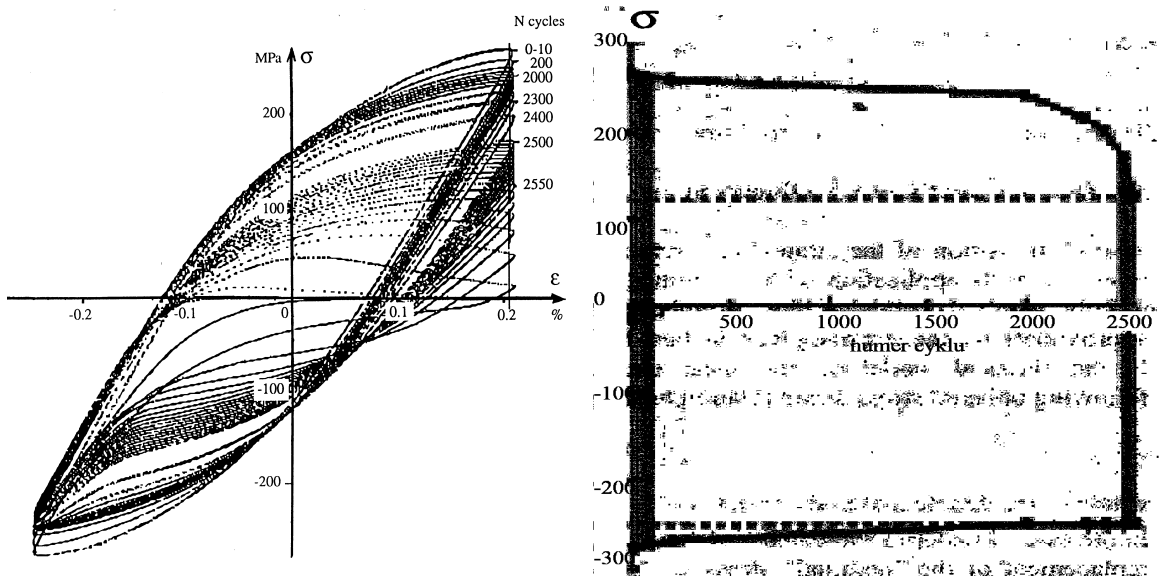
Przeprowadzona analiza eksperymentu wskazuje na jakościowo różny, trzystopniowy, przebieg procesu zniszczenia, w tym jednostronny proces uszkodzeń, [60]. Próby modelowania eksperymentu klasycznymi metodami mechaniki wskazują na występowanie efektów zamykania i otwierania się rys oraz efektu lokalizacji plastycznych odkształceń, [61].

1. Dane eksperymentalne, wybór danych do symulacji neuronowej.

Rozpatrywane dane doświadczalne, są rezultatem jednoosiowej próby niskocyklowego zmęczenia próbki stali nierdzewnej AISI 316L, w teście ściskanie–rozciąganie, ze stałą amplitudą odkształcenia [60]. Z przebiegu eksperymentu uzyskano 2556 cykli, na płaszczyźnie odkształcenie ε – naprężenie σ , przedstawionych na rys. 5.22.

Fazy eksperymentu można podzielić na trzy etapy:

- dla cykli 0–2000: zniszczenie typu izotropowego – maksymalne wartości naprężeń w poszczególnych cyklach maleją liniowo bardzo wolno w stosunku do numeru cyklu obciążenia, por. rys. 5.22, wartości najmniejsze w obrębie cykli zwiększają się w sposób liniowy również bardzo wolno, amplituda naprężenia się zmniejsza,
- dla cykli 2000–2400 zachodzi jednostronny proces uszkodzeń: amplituda naprężenia maleje w sposób znacznie szybszy, przy czym minimalne wartości naprężeń pozostają praktycznie niezmienione, wartości maksymalne maleją wraz ze wzrostem numeru pętli doświadczenia liniowo, zgodnie z krzywymi o zwiększającym się nachyleniu w stosunku do osi odkształceń,
- dla cykli poprzedzających zniszczenie (2400–2556) uaktywnia się lokalizacja plastycznych odkształceń.



Rys.5.22. Dane doświadczalne na płaszczyźnie $\sigma-\varepsilon$ oraz wartości maksymalne i minimalne w obrębie cykli na płaszczyźnie $\sigma-k$, za [60],[61]

Do analizy neuronowej wybrano 12 reprezentatywnych pętli, dzielących eksperyment na równą ilość 12 części, ze stałym ujemnym przyrostem maksymalnej wartości naprężenia w wybranych cyklach, numerując je kolejnymi liczbami naturalnymi. Pętle przeznaczone do analizy neuronowej o numerach 1 i 2 pochodziły z zakresu 0–2000 pętli eksperymentalnych. Z zakresu 2000–2400 wybrano 3 pętli, pozostałe z zakresu powyżej 2400, dla którego zmiany wartości naprężeń były największe. Pętli o numerach 10–12 odpowiadają trzem doświadczalnym ostatnim pętlom eksperymentalnym, poprzedzającym zniszczenie próbki. Każdą z pętli przyjęto w formie 45 danych odczytanych z osi rzędnych $\sigma(k)$ dla wartości odciętych

$$\varepsilon(k) = 0.2 - (k - 1) * \Delta\varepsilon \quad (5.24)$$

dla $k = 1, 2, \dots, 25$,

$$\varepsilon(k) = -0.26 + (k - 25) * \Delta\varepsilon \quad (5.25)$$

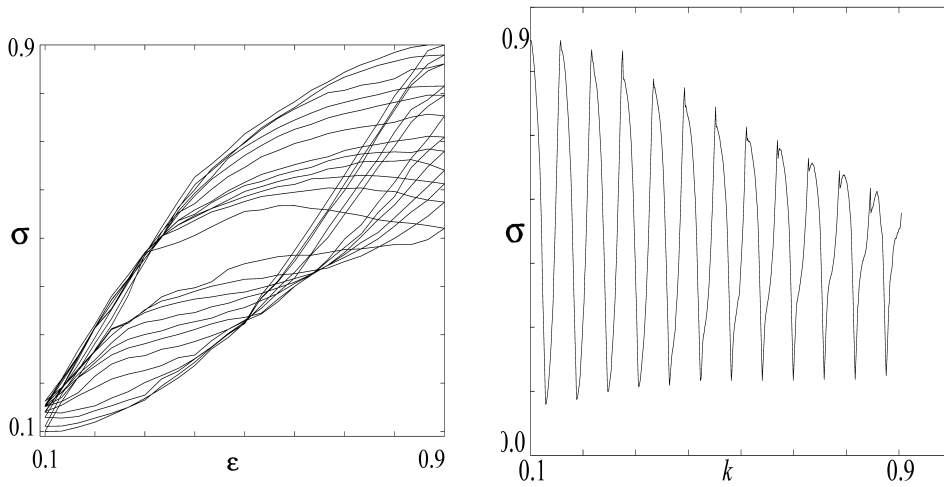
dla $k = 26, \dots, 48$ z długością kroku

$$\Delta\varepsilon(k) = 0.002\% \quad (5.26)$$

To daje razem $P=12*49=588$ punktów do uczenia i testowania SSN. Nie dysponowano danymi eksperymentalnymi w postaci numerycznej, przyjęto wartości zdygitalizowane za pomocą programu do obróbki grafiki cyfrowej Photoshop 6.0 [62], co spowodowało znaczne 'zaszumienie' danych pomiarowych, związane z określoną rozdzielczością ekranu monitora komputerowego.

Dane, dla celów symulacji neuronowej przeskalowano do przedziału $[0.1, 0.9]$, poprzez funkcję afiniczną. Dane przeskalowane, w układzie odkształcenie – naprężenie, oraz

te same dane, gdy na osi poziomej występuje numer kolejnego punktu 'pomiarowego' k , dla $k = 1, 2, \dots, 587$ przedstawiono na rys. 5.23.



Rys.5.23 Dane do analizy neuronowej

2. Rozpatrywane podejścia do modelowania krzywej histerezy

Modelowano krzywą histerezy w oparciu o różną informację wejściową dla SSN. Rozróżniono elementy wektora wejścia do SSN, zawierające zarówno wielkości z dziedziny mechaniki, jak też liczbowe znaczniki numerujące. Dodatkowo, tylko dla niektórych konfiguracji danych wejściowych, etap zmniejszania się wartości odkształceń jest rozróżniony od etapu wzrostu wartości odkształceń przez podanie z zewnątrz SSN rosnących, lub malejących, kolejno wartości wektora wejścia.

Elementy wektora wejścia w kroku k , $k = 1, 2, \dots, 587$, stanowiły :

1. Poprzednia, podana przez SSN wartość naprężenia: $\sigma(k - 1)$.
2. Unormowany znacznik postępu eksperymentu, związany z nr. punktu na histerezie, ze wzmocnieniem wartości w miarę postępu eksperymentu: $k/587$ lub malejąco wraz z numerem punktu eksperymentalnego: $k - 1/587$.
3. Unormowany znacznik postępu eksperymentu w obrębie pętli: $\text{mod}(k, 49)/49$ lub $(\text{mod}(k, 49) + 1)/49$.
4. Wartość odkształcenia $\epsilon(k)$.

Na wyjściu sieci oczekiwano wartości naprężenia $\sigma(k)$, dla $k = 2, 3, \dots, 588$.

W tab. 5.7. podano współczynniki korelacji z opóźnieniem zero oraz z opóźnieniem jednego kroku czasowego pomiędzy sygnałem $\{\sigma(k)$ dla $k = 1, 2, \dots, 588\}$, a odpowiednim ciągiem informacji wejściowej do SSN oraz zakres opóźnień, dla których

otrzymano współczynniki korelacji powyżej 0.8.

sygnał	sygnał	opóźnienie 0	opóźnienie -1	powyżej 0.8
$\sigma(k)$	$\sigma(k)$	1	0.994	-1,...,-13
$\sigma(k)$	$k/587,$	0.768	0.767	-
$\sigma(k)$	$\text{mod}(k, 49/49)$	0.709	0.694	-
$\sigma(k)$	$\varepsilon(k)$	0.96744	0.95378	-1,...,-8

Tabela 5.7. Korelacja między elementami wejścia, a wyjściem sieci

Widoczna jest silna zależność bieżącej wartości naprężenia od historii naprężenia oraz bardzo wysoka korelacja pomiędzy wybranym sygnałem znaczników eksperymentu, a wartością naprężenia.

3. Podział na zbiory uczący i testujący, architektury SSN, algorytmy uczenia

Rozważano dwa typu podziału zbioru dostępnych wzorców na podzbiory do uczenia i testowania SSN:

a) celem uczenia jest predykcja zachowania się materiału przed zniszczeniem. Zbiór testujący typu ekstrapolacyjnego, sprawdzający zdolność nauczonej SSN do prawidłowego wyznaczenia zachowania się materiału dla pętli eksperymentalnych poprzedzających zniszczenie materiału. Na podstawie podawanej w procesie uczenia informacji dotyczącej wcześniejszego przebiegu doświadczenia. Do testowania przeznaczono pętle o numerach 10–12. To daje $L = 49 \cdot 9 = 441$ wzorców do uczenia SSN oraz $T = 49 \cdot 3 = 147$ wzorców do testowania.

b) celem uczenia jest otrzymanie niejawnego modelu związku odkształcenie–naprężenie: zbiór testujący typu interpolacyjnego, sprawdzający zdolność nauczonej SSN do prawidłowego wyznaczania wartości naprężeń w dowolnej fazie eksperymentu na podstawie zbioru wzorców uczących, obejmujących wyznaczone punkty na krzywej histerezy. Do testowania przeznaczono 25% losowo wybranych punktów na histerezie. To daje $L = 0.75 \cdot 588 = 441$ wzorców do uczenia SSN oraz $T = 0.25 \cdot 588 = 147$ wzorców do testowania.

Zaproponowano sieć warstwową, typu FLNN o architekturze 3-6-3-1. Zastosowano rekurencyjne połączenie wyjścia SSN z wejściem. Zawracany sygnał stanowiło opóźnione o jeden krok czasowy wyjście z SSN, por. rys. 2.4. Dla neuronów warstwy ukrytej przyjęto sigmoidalne funkcje aktywacji (2.4), dla wyjścia liniową funkcję aktywacji (2.5) (por. rozdz. 2., dla $N=3$, $K=6$, $M=1$, $T^{in} = 1$).

Sieci uczono do 2000 epok z użyciem rozprzężonego według poszczególnych neuronów algorytmu NDEKF, wykorzystującego Rozszerzony Filtr Kalmana dla wyjścia z sieci, jako wielkości obserwowalnej (por. rozdz. 4). W modelu teoretycznym (4.13)-(4.14) wektor procesu $\{w^i, v^i\}$ przyjęto w postaci:

- dla SSN bez sprzężenia zwrotnego: $w^i \in R^4$, oraz brak v^i dla neuronów $i = 1, \dots, 6$ pierwszej warstwy ukrytej, $w^i \in R^7$ oraz brak v^i dla neuronów $i = 7, \dots, 9$ drugiej warstwy ukrytej, $w^{10} \in R^4$ oraz brak wektora sprzężenia zwrotnego dla wyjścia
- dla SSN ze sprzężeniem zwrotnym: $w^i \in R^3$ oraz brak $v^i \in R$ dla neuronów $i = 1, \dots, 6$ pierwszej warstwy ukrytej, pozostałe jak dla sieci bez sprzężenia zwrotnego.

Funkcja h , odzwierciedlająca struktury przyjętych SSN, jest w postaci (2.9)-(2.14).

Dla sieci wszystkich typów, przyjęto dla $k = 1, 2, \dots, 198$ parametry szumów:

$$R_k = \exp((s - 1)/50) \quad (5.27)$$

dla równań obserwacyjnych oraz:

$$\mathbf{Q}_k = \exp((s - 1)/50)\mathbf{I} \quad (5.28)$$

dla równań procesu, gdzie s – oznacza numer epoki uczenia, \mathbf{I} jest macierzą jednostkową, o wymiarze zależnym od liczby połączeń dochodzących do neuronu, dla którego konstruowany jest model teoretyczny, por. rozdz. 4., czyli 4x4 lub 7x7, w zależności od numeru neuronu.

Uczenie rozpoczęto od początkowych wartości parametrów SSN, wylosowanych z przedziału $[-0.5, 0.5]$, zgodnie z rozkładem jednostajnym. Proces uczenia kontrolowano poprzez obserwację spadku poziomu błędów średniokwadratowych uczenia $MSEU$ oraz testowania $MSET$, (2.21). Zastosowano kryterium stopu algorytmu uczącego, oparte o określoną liczbę epok uczenia SSN, lub określony poziom błędu uczenia $MSEU$.

4. Wyniki neuronowej symulacji zachowania się materiału w końcowej fazie eksperymentu - przypadek a) podziału zbioru wzorców na uczący i testowy

W tabeli 5.8. wyszczególniono wyniki uczenia oraz testowania SSN dla wybranych wektorów wejścia. Podano wartości błędów średniokwadratowych uczenia $MSEU$ oraz testowania $MSET$. Sygnał podawany na niektóre składowe wektorów wejściowych został dodatkowo wzmocniony, poprzez zastosowanie mnożników. Dla modelowania w oparciu o wartości odkształcenia rozpatrywano trzy typu sygnału:

- $\varepsilon_{unorm}^{[0.1,0.9]} \in [0.1, 0.9]$, czyli wartości unormowane do przedziałów o identycznych wartościach, co modelowane wartości naprężeń,
- $\varepsilon_{org} \in [-0.2; 0.2]$ oryginalne ekperymentalne wartości z zachowaniem znaku odkształcenia,

- $\varepsilon_{unorm}^{[0,0.2]} = |\varepsilon_{org}|$, nie zawierające informacji o znaku odkształcenia, lecz z zachowaniem oryginalnych wartości eksperymentalnych.

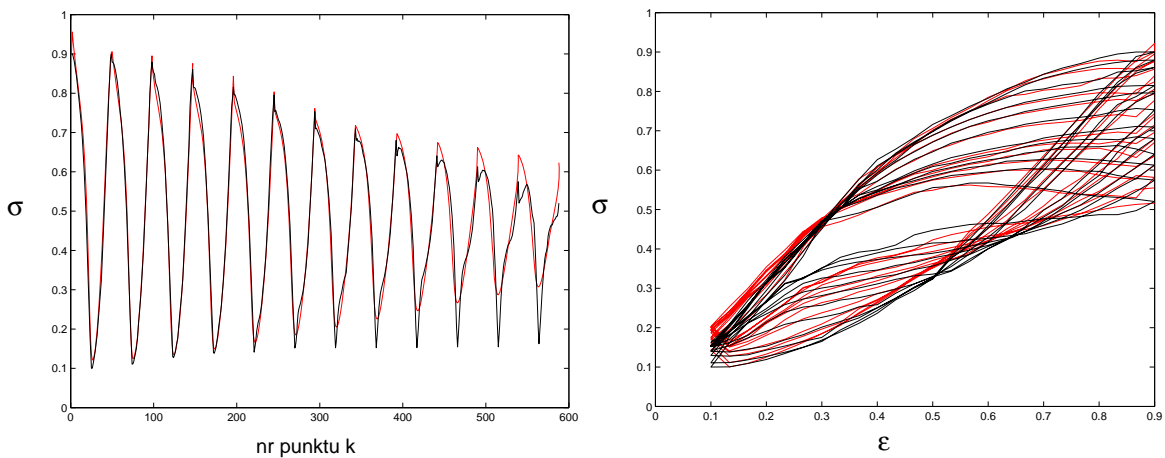
Nr	Wejście SSN	$MSEU \cdot 10^3$	$MSET \cdot 10^3$
1.	$[\sigma(k-1), 1.5(k/587), 1.5 \text{mod}(k, 49)/49]$	0.521	2.28
2.	$[\sigma(k-1), 2(k/587), 2 \text{mod}(k, 49)/49]$	0.620	2.76
3.	$[\sigma(k-1), 3(k/587), \text{mod}(k, 49)/49]$	0.414	2.21
4.	$[\sigma(k-1), 3(k/587), 3 \text{mod}(k, 49)/49]$	0.439	2.21
5.	$[\sigma(k-1), k/587, \text{mod}(k, 49)/49]$	0.862	2.25
6.	$[2(k-1)/587, 2(k/587), 2 \text{mod}(k, 49)/49]$	2.40	1.06
7.	$[2(k/587), 2 \text{mod}(k, 49)/49, 2(\text{mod}(k, 49) + 1)/49]$	1.09	1.27
8.	$[2(k/587), 2 \text{mod}(k, 49)/49, 2(1 - \text{mod}(k, 49) + 1)/49]$	3.80	2.54
9.	$[2(k/587), 2 \text{mod}(k, 49)/49, 2(\varepsilon_{unorm}^{[0.1;0.9]}(\text{mod}(k, 49)))]$	12.176	7.6
10.	$[2(k/587), 2 \text{mod}(k, 49)/49, \varepsilon_{unorm}^{[0.1;0.9]}(\text{mod}(k, 49))]$	10.7	2.90
11.	$[2(k/587), 2 \text{mod}(k, 49)/49, \varepsilon_{org}(\text{mod}(k, 49))]$	4.676	3.68
12.	$[2(k/587), 2 \text{mod}(k, 49)/49, \varepsilon_{unorm}^{[0;0.2]}(\text{mod}(k, 49))]$	9.143	1.70
13.	$[2(1 - k/587), 2 \text{mod}(k, 49)/49, \varepsilon_{unorm}^{[0;0.2]}(\text{mod}(k, 49))]$	18.22	8.5
14.	$[\sigma(k-1), k/587, \varepsilon_{unorm}^{[0;0.2]}(\text{mod}(k, 49))]$	1.31	1.51

Tabela 5.8. Zestawienie wyników dla testowania na 3 ostatnich pętach histerezy $\sigma - \varepsilon$

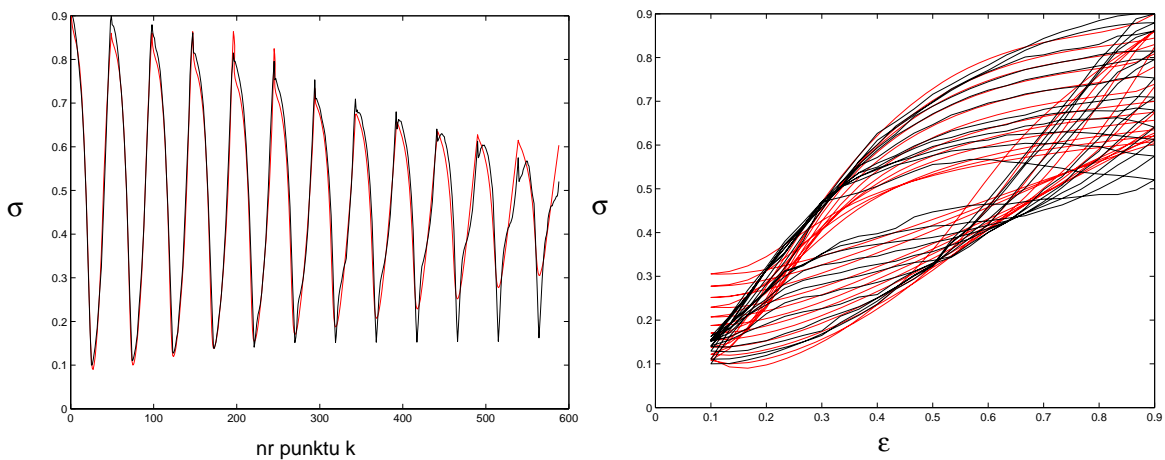
Testy 1-5 i 14 dotyczą SSN uczonych i testowanych z wykorzystaniem autoregresyjnego sprzężenia zwrotnego w obrębie wyjście – wejście sieci. Dobór wektora wejściowego dla tej grupy sieci nie mających składowej związanej z wielkością mechaniczną (1-5) miał mały wpływ na uzyskany poziom błędu $MSET$ testowania SSN. Dla sieci nie mających sprzężenia zwrotnego uzyskano duży rozrzut osiągniętych wyników, zarówno w zakresie uczenia, jak i testowania SSN, w zależności od wektora wejścia SSN. W każdej grupie wektorów wejściowych udało się otrzymać SSN o niższych poziomach błędów testowania, niż SSN z członem autoregresyjnym i wektorem wejścia nie zawierającym informacji o wielkości odkształcenia, jednak kosztem wyższego poziomu błędu uczenia, por. testy 6, 7, 12, 14. Duży wpływ na poziom uzyskiwanych błędów miała rosnąca lub malejąca wraz ze zwrotem osi k informacja w postaci znaczników eksperymentu, por. testy 7 i 8 oraz 12 i 13.

Jako najbardziej efektywne wskazać można sieci uzyskane w testach 6 i 14.

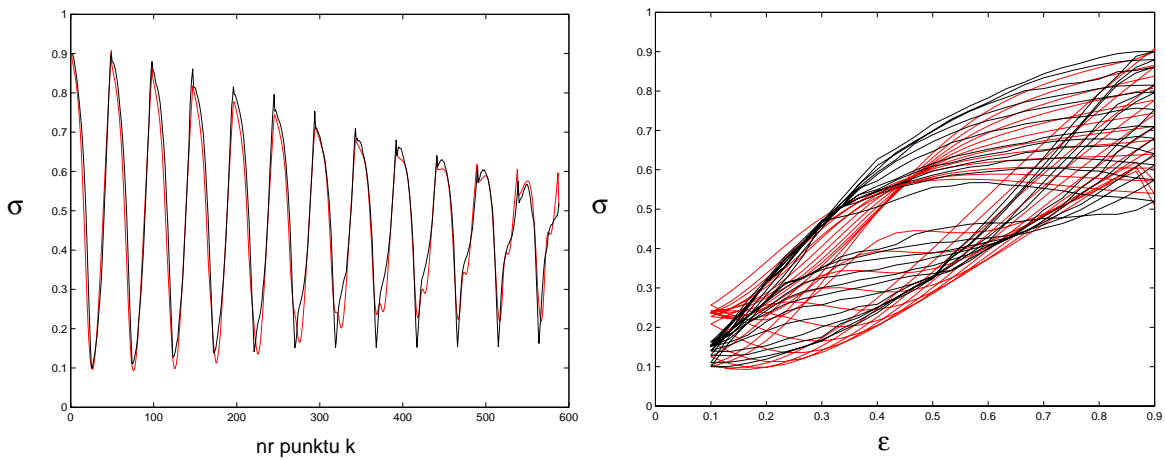
Na rys. 5.24 przedstawiono wyniki poszczególnych testów, w dwóch układach współrzędnych: numer k punktu w zbiorze danych przeznaczonych do analizy vs wartość naprężenia $\sigma(k)$ oraz wartość odkształcenia $\varepsilon(k)$ vs naprężenie $\sigma(k)$. Zobrazowano konieczność prawidłowego doboru wektora wejścia dla sieci.



Rys.5.24.a. Test 3. Wejście: $[\sigma(k - 1), 3(k/587), \text{mod}(k, 49)/49]$

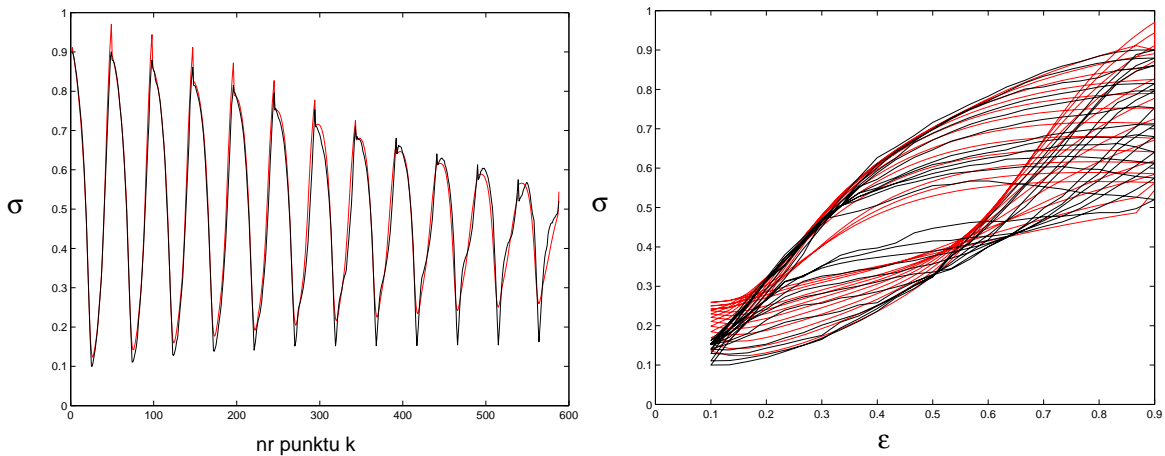


Rys.5.24.b. Test 5. Wejście: $[\sigma(k - 1), k/587, \text{mod}(k, 49)/49]$

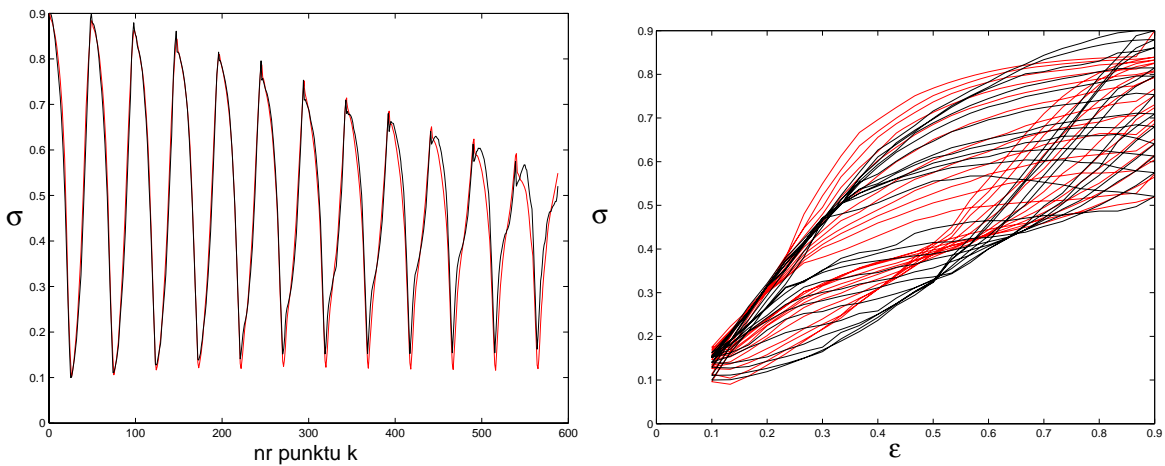


Rys.5.24.c. Test 6. Wejście: $[2(k - 1/587), 2(k/587), 2\text{mod}(k, 49)/49]$

Linia czerwona - symulacja SSN, linia czarna - wartości doświadczalne.



Rys.5.24.d. Test 7. Wejście: $[2(k/587), 2\text{mod}(k, 49)/49, 2(\text{mod}(k, 49) + 1)/49]$



Rys.5.24.e. Test 14. Wejście: $[\sigma(k - 1), k/587, \varepsilon_{unorm}^{[0;0.2]}(\text{mod}(k, 49))]$

Linia czerwona - symulacja SSN, linia czarna - wartości doświadczalne.

Najdokładniejsze wyniki uzyskano dla zadania predykcji wartości naprężenia jeden krok w przód z silnie wzmocnionym sygnałem dotyczącym numeracji wzorca, por. rys. 5.24.a.

Doświadczalną krzywą histerezy dla rozpatrywanego materiału udało się także zamodelować bez podawania SSN informacji o historii naprężenia, wyłącznie na podstawie przyjętych liczników postępu doświadczenia, opartych o parametr k , por. rys. 5.24.d, aczkolwiek otrzymano mniejszą dokładność w zakresie symulacji dla niskich wartości odkształcenia. Dla takich wektorów wejścia (nie zawierających bezpośrednio wartości odkształcenia), by wyznaczyć dla nauczonej SSN wartość naprężenia dla n -tego cyklu doświadczenia dla wartości odkształcenia $\varepsilon \in [-0.2, 0.2]$, należy wartość parametru k wyznaczyć ze związku

$$k = n \cdot 49 + m \quad (5.29)$$

gdzie:

$$\begin{cases} m=1-(\varepsilon_0 - 0.2)/\Delta\varepsilon \\ m=25+(\varepsilon_0 + 0.26)/\Delta\varepsilon \end{cases} \quad (5.30)$$

, gdzie równość pierwszą stosujemy na etapie malenia wartości odkształceń ($\varepsilon(k) < \varepsilon(k-1)$), a drugą w fazie ich wzrostu ($\varepsilon(k) > \varepsilon(k-1)$). $\Delta\varepsilon = 0.002$.

5. Wyniki neuronowej symulacji zachowania się materiału - przypadek b) podziału zbioru wzorców

W tab. 5.9. wyszczególniono wyniki uczenia oraz testowania SSN dla najefektywniejszych numerycznie wektorów wejścia. Podano wartości błędów średniokwadratowych uczenia $MSEU$ oraz testowania $MSET$.

Nr	Wejście SSN	$MSEU \cdot 10^3$	$MSET \cdot 10^3$
15.	$[\sigma(k-1), 3(k/587), \text{mod}(k, 49)/49]$	0.561	1.84
16.	$[2(k/587), 2\text{mod}(k, 49)/49, 2(\text{mod}(k, 49) + 1)/49]$	1.19	1.17
17.	$[\sigma(k-1), k/587, \varepsilon_{unorm}^{[0;0.2]}(\text{mod}(k, 49))]$	1.11	1.20

Tabela 5.9. Zestawienie wyników dla testowania na 25% losowo wybranych punktów histerezy $\sigma - \varepsilon$

Dla tego typu problemu uzyskano podobne wyniki w zakresie doboru wektorów wejścia, jak dla problemu a). Wprowadzenie autoregresyjnej składowej wektora wejścia SSN, bez informacji o wartościach odkształceń, pozwoliło podnieść dokładność uczenia, lecz błędy testowania otrzymano niższe w przypadku podawania SSN wartości odkształcenia z zachowanym znakiem (dodatnim bądź ujemnym), por. testy 15 i 17.

6. Porównanie wyników neuronowej symulacji z metodami modelowania za pomocą równań klasycznej mechaniki zniszczenia

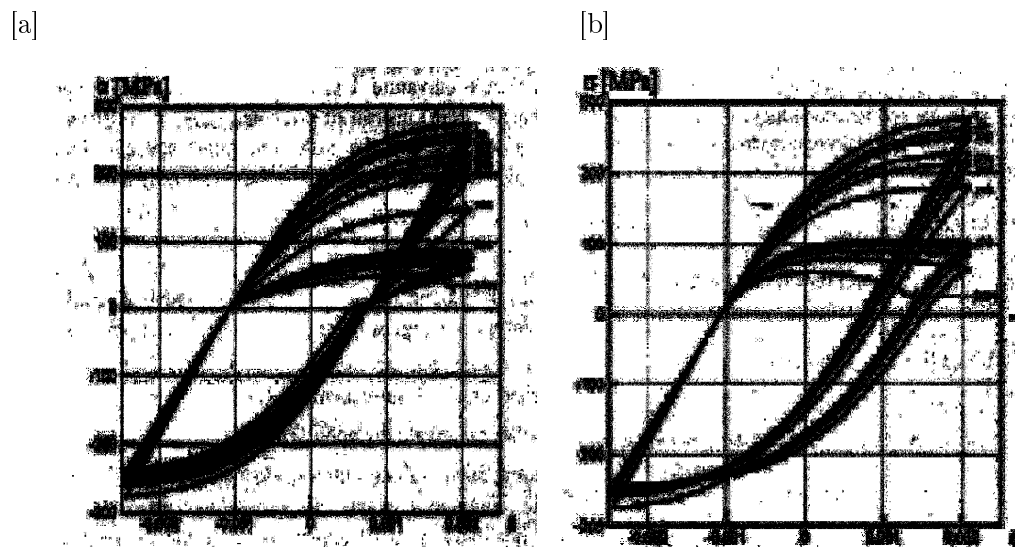
W pracy [61] zaproponowano model teoretyczny związku pomiędzy naprężeniem, a odkształceniem dla rozpatrywanego materiału.

Model oparto na jednowymiarowym, nieliniowym, sprężysto–plastycznym modelu Ylinaena [63], otrzymując związek między naprężeniem, a odkształceniem o postaci innej dla fazy ściskania, innej dla rozciągania materiału. Przyjęto dwie zmienne charakteryzujące uszkodzenie: zmienną odpowiadającą jednostronnemu uszkodzeniu oraz izotropowemu uszkodzeniu. Dla etapu lokalizacji plastycznych odkształceń wykorzystano przybliżony, trójwymiarowy, opis parametrów zniszczenia. Zaproponowano dwa modele: pierwszy odzwierciedlający dyskretnie otwieranie i zamykanie się rys, otrzymując zależność pomiędzy odkształceniem, a naprężeniem przedstawioną na rys.

5.25.[a], drugi, umożliwiający zamodelowanie efektu ciągłego otwierania i zamykania się rys, rys. 5.25.[b]. Parametry ww. modeli otrzymano z warunków zamknięcia pierwszej pętli histerezy oraz warunków minimalizacji różnic pomiędzy modelem teoretycznym, a pierwszą pętlą histerezy. Parametry materiałowe moduł plastyczności Younga oraz liczbę Poissona przyjęto w modelach za [60].

W zaproponowanym modelu [a] występuje nieuzasadniona eksperymentem nieciągłość w wartości pochodnej $\partial\sigma/\partial\varepsilon$. Zarówno model [a], jak i model [b] nie odtwarza poprawnie krzywej dla zakresu ujemnych wartości, zarówno naprężenia, jak i odkształcenia w fazie ściskania. Dla neuronowej symulacji eksperymentu dla testu 1., żaden z tych niekorzystnych efektów nie występuje. Dla testu 7., w którym nie uwzględniono wartości naprężenia z kroku poprzedniego, symulacja neuronowa tej fazy zachowania się materiału wydaje się być bliższa obrazowi doświadczalnemu. Dla wejścia do SSN występującego w teście 14. zmiany krzywizny krzywej histerezy oddane są również poprawniej.

Prawidłowe odtworzenie relacji naprężenie–odkształcenie dla zaproponowanego modelu teoretycznego materiału wymaga znajomości stałych materiałowych, poprawnego wyboru oraz kalibracji modelu teoretycznego, poprawnego modelowania parametrów zniszczenia. Dla metody opartej o SSN żaden z tych etapów nie jest konieczny. Poszukiwany związek jest modelowany tylko w oparciu o dane doświadczalne oraz proste znaczniki eksperymentu, które pełnią rolę różnicującą kolejne fazy doświadczenia. Związek między wielkościami mechanicznymi nie jest dany kompletem równań, ale w postaci niejawnej – informacji o wartościach wag nauczonej SSN.



Rys.5.25
Wyniki modelowania równaniami mechaniki zniszczenia, za [61]

5. Wnioski końcowe

Zaprezentowane wyniki upoważniają do wyciągnięcia następujących wniosków:

- Zastosowanie filtrów Kalmana, jako metody uczenia SSN pozwoliło na predykcję pętli histerezy rozważanego materiału, za pomocą SSN o niewielkiej liczbie parametrów (3-6-3-1);
- Wykorzystanie SSN, jako narzędzia obliczeniowego, pozwala modelować przeprowadzony eksperyment i przewidywać zachowanie się materiału poprzedzające jego zniszczenie;
- Wprowadzenie rekurencji w obrębie wyjście–wejście SSN, o charakterze autoregresyjnym, pozwala otrzymać obraz bliższy wykresom eksperymentalnym, niż w przypadku zastosowania sieci jednokierunkowych;
- Porównanie wyników zawartych w tab. 5.8 oraz wykresów na rysunkach 5.24 daje podstawy do stwierdzenia, iż uzyskany poziom błędu średniokwadratowego nie powinien być jedynym kryterium wyboru SSN, najefektywniej symulującej przebieg eksperymentu;
- Konstrukcja niejawnego, neuronowego modelu tego materiału, pozwala uzyskać obraz zachowania się tego materiału w różnych fazach eksperymentu;
- Problem doboru parametrów modelu matematycznego mechaniki zniszczenia zamieniony zostaje na problem poszukiwania odpowiedniego wektora wejściowego do SSN.

5.2. Predykcja pętli histerezy od obciążeń niskocyklicznych

5.2.3. Modelowanie zależności naprężenie – odkształcenie dla cyklicznie obciążanych próbek betonowych

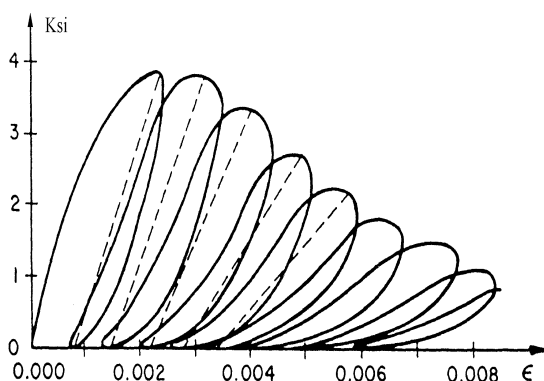
Zajęto się problemem neuronowego wyznaczania zależności między naprężeniem, a odkształceniem dla próbek betonowych, obciążanych według różnych programów obciążeń cyklicznych. Dane wykorzystane podczas symulacji pochodziły z doświadczeń wykonanych na rzeczywistym materiale. Obejmowały 3 grupy badań: [64,65,66], dotyczące doświadczeń po przekroczeniu obciążenia krytycznego w zakresie ściskania oraz rozciągania-ściskania.

1. Rozważane materiały, przeprowadzone eksperymenty – grupa 1 badań.

Dla tej grupy badań doświadczenia wykonano na 12 cylindrycznych próbkach o wymiarach 3x6 in., por. [66].

Obciążenie cykliczne ściskania realizowano wg. następującego schematu, realizując co najmniej 7 pełnych cykli obciążenia każdej próbki:

1. Monotoniczne zwiększanie obciążenia, do osiągnięcia wartości maksymalnej, odpowiadającej wytrzymałości betonu na ściskanie;
2. Odciążanie do uzyskania wartości ok. 0
3. Ponowne obciążanie próbki do uzyskania maksymalnej wartości naprężenia;
4. Próbkę odciążano i dociążano wg 2-3, por. rys. 5.30.



Rys. 5.30. Wyznaczona eksperymentalnie zależność naprężenie nominalne – wydłużenie, za [66], 1 Ksi = 6.895 MPa

2. Rozważane materiały, przeprowadzone eksperymenty – grupa 2 badań.

Grupa badań, oznaczona jako 2, została zaczerpnięta z [64]. Do eksperymentów użyto pięciu rodzajów betonu zwykłego, o projektowanej wytrzymałości gwarantowanej 45N/m^2 . Doświadczenia wykonywano na próbkach prostopadłościennych o wymiarach: $150 \times 60 \times 50$ mm. W celu zlokalizowania strefy zniszczenia, w połowie długości próbek zostały wykonane symetryczne nacięcia szerokości i głębokości 5×5 mm, redukujące efektywny przekrój poprzeczny próbek do 50×50 mm. Próbki rozciągano w warunkach kontroli przemieszczeń, z ustaloną prędkością przyrostu wydłużeń $0.08 \mu\text{m/s}$, por [64],[67]. Dla próbek wykonanych z poszczególnych rodzajów betonu wyznaczono średnią wytrzymałość na ściskanie i rozciąganie przy rozłupywaniu, które wahały się w granicach 2.38-2.91 MPa. Badania wykonywano wg. trzech różnych programów obciążeń statycznych:

- próba rozciągania monotonicznego
- w zakresie pokrytycznym próba rozciągania cyklicznego po przekroczeniu obciążenia maksymalnego, wykonywana w zakresie rozciąganie-rozciąganie
- w zakresie pokrytycznym próba rozciągania-ściskania cyklicznego, po przekroczeniu maksymalnego obciążenia rozciągającego.

Wykonano badania co najmniej trzech próbek każdego z pięciu rodzajów betonów, we wszystkich 3 typach obciążeń. W niniejszej pracy rozważane są wyłącznie obciążenia cykliczne.

Wartości sił w przedziałach 10-30 sek. przeliczono na wartości naprężeń nominalnych $[\sigma] = \text{N/mm}^2 = \text{MPa}$, wg zależności:

$$\sigma = P/F \quad (5.31)$$

gdzie

P – wartość przykładanej siły,

F – początkowe pole powierzchni próbki pomiędzy nacięciami.

Wydłużenie całkowite $[\delta_c] = \mu\text{m}$ stanowiły uśrednione odczyty z 8 czujników zamocowanych na próbce. Składało się na nie: wydłużenie sprężyste, wydłużenie nieodwracalne i szerokość rozwarcia rysy.

Obciążenie cykliczne rozciąganie-rozciąganie realizowano wg. następującego schematu, realizując co najmniej 7 pełnych cykli obciążenia każdej próbki:

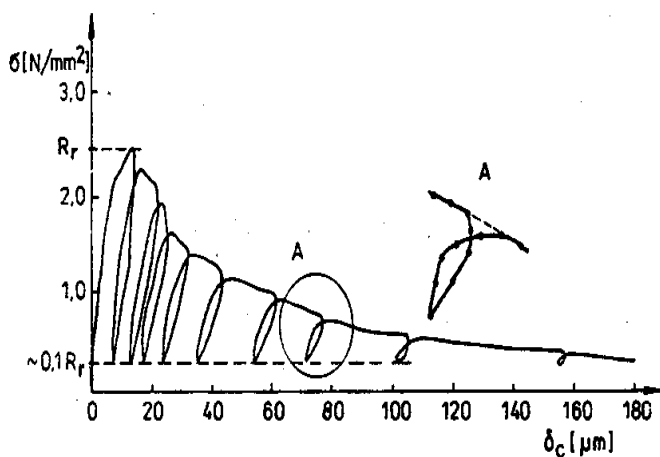
1. Monotoniczne zwiększanie obciążenia, do osiągnięcia wartości maksymalnej, odpowiadającej wytrzymałości betonu na rozciąganie R_r ;
2. W warunkach kontroli przyrostu wydłużenia próbki, dopuszczano do obniżenia wartości naprężenia nominalnego o ok. 0.16 MPa, do poziomu $\sigma = R_r - 0.16$;

3. Odciążanie do uzyskania wartości obciążenia ok. $0.1R_r$;
4. Dociążanie próbki do uzyskania maksymalnej wartości naprężenia
5. Dopuszczano swobodny spadek naprężeń o ok. 5%, tj. do poziomu 0.95 % maksymalnej wartości naprężenia uzyskanej w punkcie poprzednim;
6. Próbkę odciażano i dociażano wg 3-4-5, por. rys. 5.26.a.

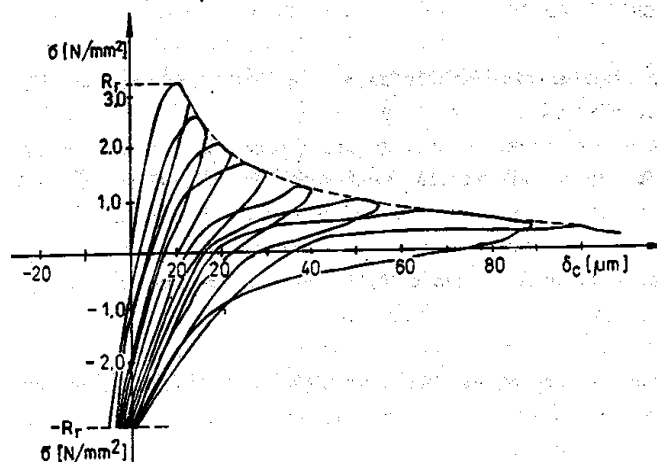
Zastosowane oznaczenie R_r przyjęto za pracę [64].

Obciążenie cykliczne rozciąganie-ściskanie realizowano wg. następującego schematu:

1. Monotoniczne zwiększanie obciążenia, do osiągnięcia wartości maksymalnej, odpowiadającej wytrzymałości betonu na rozciąganie R_r ;
2. W warunkach kontroli przyrostu wydłużenia próbki, dopuszczano do obniżenia wartości naprężenia nominalnego o ok. 0.16 MPa, do poziomu $\sigma = R_r - 0.16$;
3. Odciażanie do uzyskania wartości zerowego obciążenia, a następnie płynne ściśkanie, aż do wartości naprężenia nominalnego ok. $\sigma = -R_r$;
4. Odciażanie próbki, przez redukcję siły ściskającej do zera, po czym płynne obciążanie jej siłą rozciągającą, aż do uzyskania maksymalnej wartości naprężenia;
5. Dopuszczano swobodny spadek naprężeń o ok. 5%, tj. do poziomu 0.95 % maksymalnej wartości naprężenia uzyskanej w punkcie poprzednim;
6. Próbkę ściśkano i rozciągano wg 3-4-5, por. rys. 5.26.b.



Rys. 5.26.a Grupa 2 badań – wyznaczona eksperymentalnie zależność naprężenie nominalne–wydłużenie dla testu rozciąganie-rozciąganie, za [64]



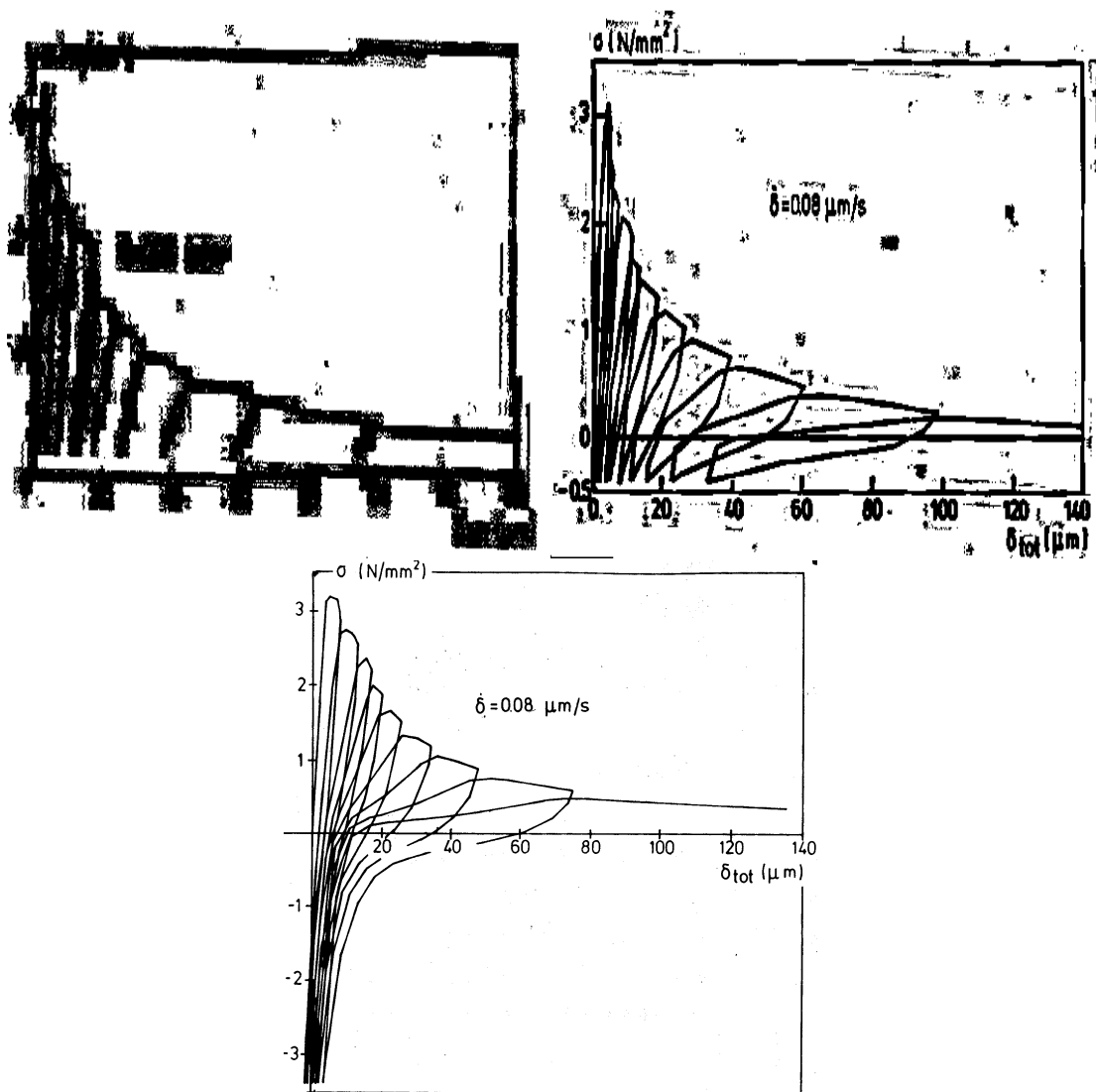
Rys. 5.26.b Grupa 2 badań – wyznaczona eksperymentalnie zależność naprężenie nominalne–wydłużenie dla testu rozciąganie-ściskanie, za [64]

3. Rozważane materiały, przeprowadzone eksperymenty – grupa 3 badań.

Badania wykonano na dwóch grupach próbek różnej wielkości, testowanych w warunkach kontroli wydłużenia całkowitego $[\delta_{tot}] = \mu\text{m}$, [65], por. rys. 5.27. Mniejsze próbki prostopadłościowe, o wymiarach 250x60x50 mm, nacięto w połowie długości symetrycznie, wykonując nacięcia szerokości i głębokości 5x5 mm, zmniejszając ich efektywny przekrój poprzeczny do 50x50 mm. Większe próbki prostopadłościowe, o wymiarach 22x250x50 mm, nacięto symetrycznie na szerokości i głębokości 5x20 mm. Do wykonania wszystkich próbek użyty został ten sam materiał.

3.1. Próbki mniejsze.

Próby rozciąganie-rozciąganie oraz rozciąganie-ściskanie wykonywano wg tego samego schematu, jak opisane w 2.1., przy czym dla testu rozciąganie-rozciąganie próbę wykonywano do uzyskania wartości obciążenia ok. $5\%R_r$, dla testu rozciąganie-ściskanie do uzyskania wartości naprężenia nominalnego ok. $\sigma = -5\%R_r$ lub $\sigma = -R_r$, por. rys. 5.27. Dla tego rodzaju próbek zaobserwowano rozkład naprężeń taki sam w całym przekroju próbki. Zniszczenie próbek (w sensie destrukcji próbki) następowało dla wartości wydłużenia $\delta_{tot} = 250\mu\text{m}$. Podobnie, jak dla grupy 1 badań, mierzono działającą siłę, wartości której, po podzieleniu przez początkowy przekrój poprzeczny próbki pomiędzy wykonanymi nacięciami, dały wartości naprężeń nominalnych $[\sigma] = \text{N/mm}^2$.



Rys. 5.27. Grupa 3 badań, próbki małe – test rozciąganie-rozciąganie oraz rozciąganie-ściskanie, za [65]

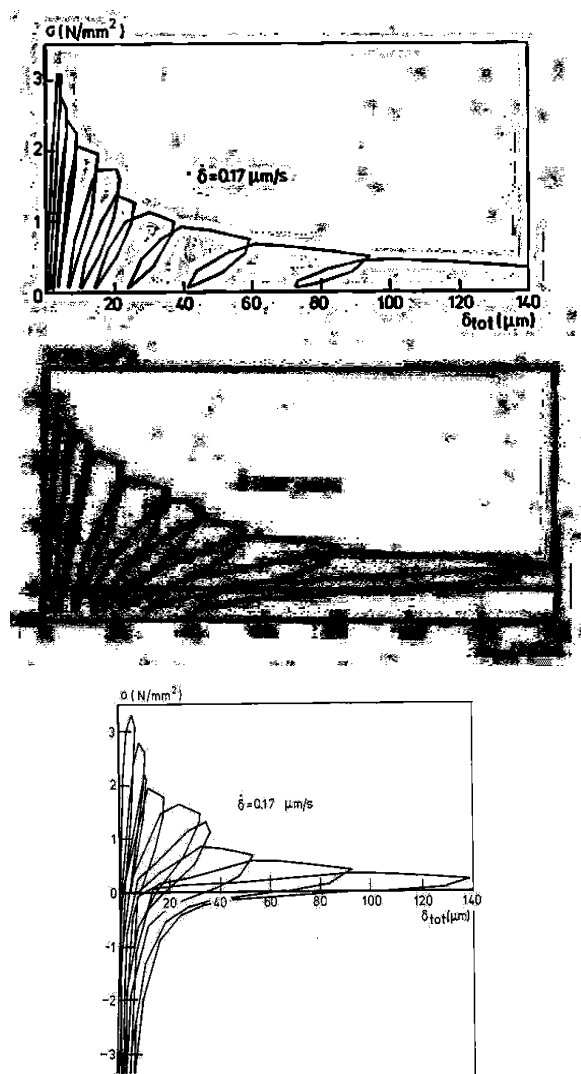
3.2. Próbki większe.

Celem rozważania próbek o tych rozmiarach było uzyskanie niejednorodnego rozkładu naprężeń w przekroju próbki. Rozważano dwa rodzaje testów:

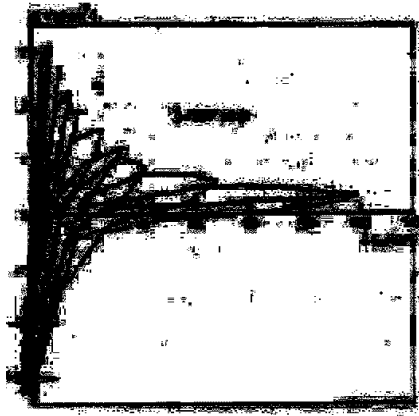
- Testy wykonywane do pełnego odciążenia. Wykonywane zgodnie z programem opisanym w punkcie 2.1, zarówno dla przypadku rozciąganie-rozciąganie, jak dla przypadku rozciąganie-ściskanie, tj. do uzyskania maksymalnej wartości odkształcenia na obwodni testu niecyklicznego. Dla testu rozciąganie-rozciąganie

odciążanie wykonywano do uzyskania wartości obciążenia ok. $5\%R_r$, dla testu rozciąganie-ściskanie odciążanie wykonywano do uzyskania wartości naprężenia nominalnego ok. $\sigma = -5\%R_r$ lub $\sigma = -R_r$, por. rys. 5.28.

- Testy wykonywane do niepełnego odciążenia. Dla testu rozciąganie-rozciąganie obciążenie i odciążanie wykonywano dla wartości naprężenia nominalnego $\sigma \in [0.25R_r, 0.85R_r]$, dla testu rozciąganie-ściskanie obciążenie i odciążanie wykonywano do uzyskania wartości naprężenia nominalnego $\sigma \in [-R_r, 0.85R_r]$, por. rys. 5.29.



Rys. 5.28. Grupa 2 badań, próbki duże, testy do pełnego odciążenia – wyznaczona eksperymentalnie zależność naprężenie nominalne–wydłużenie dla testów rozciąganie-rozciąganie oraz rozciąganie-ściskanie, za [65]



Rys. 5.29. Grupa 2 badań, próbki duże, testy do niepełnego odciążenia – wyznaczona eksperymentalnie zależność naprężenie nominalne–wydłużenie dla testów rozciąganie-rozciąganie oraz rozciąganie-ściskanie, za [65]

4. *Wzorce uczące i testujące dla sieci, architektura SSN, algorytmy uczenia.*

Dysponowano interpolowanymi liniowo w [64] oraz [65] krzywymi zależności $\sigma - \delta_c$ (grupa 2 badań) oraz w [66] $\sigma - \delta_{tot}$ (grupa 3) i $\sigma - \varepsilon$ (grupa 1). Wykonano ich digitalizację, przy użyciu programu do cyfrowej obróbki sygnałów graficznych Graph Digitizer Version 1.9, [68], ze stałą długością kroku, którą wyznaczała przyjęta jednostka długości digitalizowanej krzywej. Digitalizację danych z 1 grupy przeprowadzono pobierając punkty 'próbkowe' z krzywej naprężenie – odkształcenie ze stałą długością kroku naprężenia. W tab. 5.10. wyszczególniono licznosc P danych, którymi dysponowano po digitalizacji. Jako 'preprocessing' danych przeprowadzono, ze względu na przyjętą funkcję aktywacji neuronów, skalowanie za pomocą funkcji afinicznej do przedziału $[0.1, 0.9]$.

Grupa	Doświadczenie	Zakres	P
grupa 1	ściskanie	$[0, R_c]$	405
2	rozciąganie-rozciąganie	$[0.1R_r, R_r]$	1352
2	rozciąganie-ściskanie	$[-R_r, R_r]$	2954
3 próbki małe	rozciąganie-rozciąganie	$[0.05R_r, R_r]$	1088
3 próbki małe	rozciąganie-ściskanie	$[-0.05R_r, R_r]$	1463
3 próbki małe	rozciąganie-ściskanie	$[-R_r, R_r]$	2189
3 próbki duże	rozciąganie-rozciąganie	$[0.05R_r, R_r]$	1517
3 próbki duże	rozciąganie-ściskanie	$[-0.05R_r, R_r]$	1796
3 próbki duże	rozciąganie-ściskanie	$[-R_r, R_r]$	3086
3 próbki duże	rozciąganie-rozciąganie	$[0.25R_r, 0.85R_r]$	1355
3 próbki duże	rozciąganie-ściskanie	$[-R_r, 0.85R_r]$	1893

Tab. 5.10. Licznosci zbiorów danych do uczenia i testowania SSN, dla poszczególnych grup pomiarów

Dostępne dane rozdzielono na zbiór uczący i testowy. Zbiór testowy przyjęto w formie dwóch ostatnich (2 i 3 grupa pomiarów) lub 3 ostatnich (1 grupa pomiarów) pętli rozpatrywanych krzywych. Przyjęcie takiego podziału, w przypadku prawidłowo nauczonej SSN, pozwala na predykcję zachowania się materiału, w dalszej części eksperymentu, na podstawie danych dotyczących wcześniejszej fazy doświadczenia.

Rozpatrywano szereg wektorów wejścia do SSN, dzieląc je na różne grupy, zależnie od rodzaju wprowadzanej informacji. Kolejne ich składowe przyjęto w postaci:

- znacznika numerującego przebieg eksperymentu w czasie

$$k/P \quad (5.32)$$

dla $k = 1, 2, \dots, P$, gdzie P jest licznością dostępnego zbioru wzorców

- znacznika numerującego przebieg eksperymentu w czasie w obrębie każdej z pętli osobno. Parametr wejściowy *licznik* dobrano, numerując wzorce w obrębie każdej z pętli histerezy oddzielnie, normalizując wskaźnik do przedziału $[0,1]$:

$$\text{licznik} = \quad (5.33)$$

$$[1/N_1, 2/N_1, \dots, N_1/N_1, \dots, 1/N_i, 2/N_i, \dots, N_i/N_i, \dots, 1/N_9, 2/N_9, \dots, N_9/N_9]$$

gdzie N_i oznacza licznosc wzorców przypadających na i -tą pętlę histerezy.

- znacznika rozróżniającego fazę wzrostu wartości naprężeń (obciążenie materiału) od fazy zmniejszania wartości naprężeń (odciążenie materiału). Przyjęto parametry wejściowe licznik_1 oraz licznik_2 . W obrębie i -tej pętli histerezy:

$$\text{licznik}_{1,i} = \quad (5.33)$$

$$[1/M_i, 2/M_i, \dots, M_i/M_i, (M_i - 1)/M_i, (M_i - 2)/M_i, \dots, (M_i - N_i)/M_i]$$

gdzie M_i oznacza ilość kroków obciążenia materiału, N_i ilość kroków odciążenia materiału, przypadających na i -tą pętlę rozpartywanego eksperymentu.

Parametr $\text{licznik}_{2,i}$ skonstruowano poprzez przekształcenie parametru $\text{licznik}_{1,i}$ przez funkcję afiniczną, tak, by przyjmował on wartości w $[0.1, 0.9]$, czyli do przedziału identycznego, jak przeskalowane, podawane SSN, elementy zbioru uczącego.

- poprzednia wartość naprężenia, podaną przez SSN:

$$\sigma_{ssn}(k - 1) \quad (5.35)$$

- oraz dodatkowo dla grupy 2 badań, aproksymację potęgową obwiedni doświadczalnej testu monotonicznego, postaci:

$$\text{obw}(k) = \sigma(\delta) = 1 - (\delta/200)^{0.267} \quad (5.36)$$

która została uznana przez autorów badań, za najlepiej oddającą charakter eksperymentu, por. [64].

Poszczególne wektory wejścia, przedstawiono w tabelach, obrazujących uzyskane wyniki. Na wyjściu SSN oczekiwano wartości naprężenia nominalnego

$$\sigma(k) \tag{5.37}$$

Rozważano SSN dwuwarstwowe, jednokierunkowe i o sprzężeniu autoregresyjnym wyjścia z wejściem, typu FLNN. Za względu na słabą zależność wyników symulacji neuronowych od przyjętej architektury SSN, po wstępnych testach wybrano architekturę 3-6-3-1. Dla wektorów wejścia, mających składową postaci (5.37), zawracany na wejście SSN sygnał stanowiło opóźnione o jeden krok czasowy wyjście z SSN, por. rys. 2.4., Dla neuronów warstwy ukrytej przyjęto sigmoidalne funkcje aktywacji, zob. (2.4), dla wyjścia liniową funkcję aktywacji zob. (2.5) (por. Rozdz. 2.)

Sieci uczono z wykorzystaniem rozprężonego według poszczególnych neuronów algorytmu NDEKF, wykorzystującego Rozszerzony Filtr Kalmana dla wyjścia z sieci, jako wielkości obserwowalnej (por. rozdz. 4.). W modelu teoretycznym (4.13)-(4.14) wektor procesu przyjęto postaci $\{\mathbf{w}^i, v^i\}$, dla neuronów w których występował sygnał zawracany rekurencyjnie oraz $\{\mathbf{w}^i\}$ dla neuronów, do których sygnał dochodził jedynie zgodnie z kierunkiem wejście – wyjście SSN. Funkcja h , odzwierciedlająca struktury przyjętych SSN, ma postać (2.9)-(2.14).

Dla wszystkich SSN przyjęto parametry szumów w modleach teoretycznych postaci:

$$R_k = \exp((s - 1)/50) \tag{5.38}$$

dla równań obserwacyjnych oraz:

$$\mathbf{Q}_k = \exp((s - 1)/50)\mathbf{I} \tag{5.39}$$

dla równań procesu, gdzie

s – numer bieżącej epoki uczenia sieci

\mathbf{I} – macierz identycznościowa, której wymiar zależy od ilości połączeń dochodzących do neuronu, dla którego konstruowany jest model teoretyczny, por. Rozdz. 4.

Uczenie sieci rozpoczęto od początkowych wartości parametrów SSN, wylosowanych z przedziału $[-0.5, 0.5]$, zgodnie z rozkładem jednostajnym. Proces uczenia kontrolowano poprzez obserwację spadku poziomu błędów średniokwadratowych uczenia $MSEU$ oraz testowania $MSET$, por. Rozdz. 2, (2.21).

Ze względu na różny przebieg procesu uczenia dla poszczególnych grup badań oraz zróżnicowanych wektorów wejścia SSN, zastosowano kryterium zatrzymania algorytmu uczącego, oparte o kryterium stabilizacji wartości błędów uczenia i testowania sieci.

Wyniki symulacji zostaną przedstawione w kolejności rozwiązywania ww. zagadnień.

5. Wyniki modelowania neuronowego testu ściskanie – grupa 1 badań.

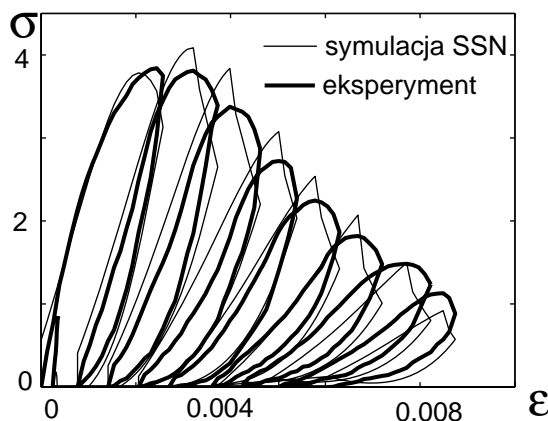
Do uczenia sieci przeznaczono 6 pierwszych pętli histerezy, na które składały się $L = 273$ punkty pomiarowe, zdolności generalizacyjne sieci testując na 3 ostatnich pętlach, zawierających $T = 132$ punkty pomiarowe.

Dla tej grupy badań, najefektywniejsze dobrane wektory wejścia przedstawiono w tab. 5.11, gdzie k – oznacza numer bieżącego wzorca uczącego lub testowego, związanego z numerem punktu na zdigitalizowanej krzywej histerezy, $k = 1, \dots, 405$. Na rys. 5.31. przedstawiono odwzorowanie zbioru uczącego oraz neuronową symulację dalszej części doświadczenia, uzyskaną za pomocą SSN z najefektywniejszym dobranym wektorem wejściowym. Wszystkie z zaproponowanych SSN dla tej grupy badań uczono przez 1000 epok.

Otrzymano wyniki, w zakresie błędów MSE silnie zależne od przyjętego wektora wejścia SSN. Jako najefektywniejszą, ze względu na najmniejszy uzyskany błąd $MSET$ wybrano SSN z testu nr 3.

Nr	Wejście SSN	$MSEU$	$MSET$
1	$[1 - k/405, 1 - licznik, licznik_1]$	0.0107	0.0039
2	$[1 - k/405, 1 - licznik, licznik_2]$	0.0035	0.0016
3	$[1 - licznik, licznik_2, k/405]$	0.0031	0.0008
4	$[1 - licznik, licznik_2, \sigma_{ssn}(k - 1)]$	0.0049	0.0062

Tab. 5.11. Wyniki symulacji dla grupy 1 badań



Rys. 5.31. Odtworzenie zbioru uczącego (pierwsze 6 pętli histerezy) oraz testującego (pozostałe pętle) dla testu nr 3, dla grupy 1 badań)

Widoczna jest prawidłowa predykcja charakteru doświadczenia, w zakresie obniżania się obwiedni testu oraz zmniejszania kąta nachylenia kolejnych pętli doświadczalnych w stosunku do osi odkształceń.

6. Wyniki modelowania neuronowego testu rozciąganie-rozciąganie – grupa 2 badań.

Dla tej grupy badań doświadczalne pętle histerezy mają odmienny kształt, niż te uzyskane dla materiału testowanego w [66]. Wartości maksymalne naprężeń maleją znacznie szybciej, pętle histerezy znacznie silniej się zwężają. Wybrano parametry *licznik₂* oraz *licznik* i wskaźnik postępu eksperymentu postaci $k/1352$ dla $k = 1, 2, \dots, 1352$, jako efektywniejsze. Konstruowano je analogicznie, jak dla grupy 1 badań, z uwzględnieniem liczności zbioru uczącego i testującego oraz liczby punktów doświadczalnych na poszczególnych zdigitalizowanych pętlach histerezy dla tej grupy badań. Do uczenia sieci przeznaczono 8 pierwszych pętli histerezy, na które składało się $L = 1168$ punktów pomiarowych, zdolności generalizacyjne sieci testując na 2 ostatnich pętlach, zawierających $T = 184$ punkty pomiarowe.

Poszczególne wektory wejściowe, wraz z wynikami uczenia i testowania w zakresie otrzymanych błędów średniokwadratowych uczenia *MSEU* i testowania *MSET*, dla SSN uczonych do S epok uczenia włącznie, przedstawione zostały w tab. 5.12. i na rys 5.32.

Zastosowanie wektorów wejścia, jak dla grupy 1 badań, dało mniej dokładne odwzorowanie przebiegu doświadczenia, szczególnie w zakresie odwzorowania obwiedni testu, por. tab. 5.11, i 5.12. Zaistniała więc konieczność zmiany wektora wejścia, na efektywniejszy numerycznie.

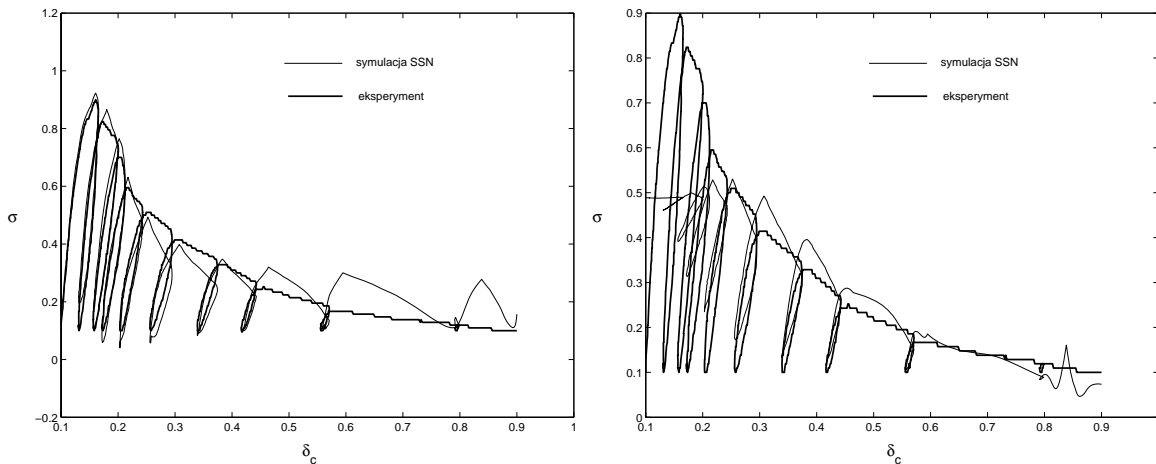
Ze względu na fakt, iż w pracy [64] zweryfikowana została pozytywnie dla rozpatrywanego materiału hipoteza równoważności obwiedni dla obciążenia monotonicznego oraz obu programów obciążeń cyklicznych, do konstrukcji wektora wejścia dla omawianych danych pochodzących z testów cyklicznych, wykorzystano za [64] aproksymację potęgową doświadczalnej obwiedni testu monotonicznego, postaci (5.36).

Zaproponowano szereg wektorów wejścia, opartych o informację o postaci obwiedni testu, w tym wprowadzenie informacji o wartości obwiedni dla poszczególnych punktów pomiarowych (por. tab. 5.12, testy 4,5) oraz przemnożenie sygnału wejściowego, rozpatrywanego dla grupy 1 badań, przez wartość obwiedni dla bieżącego punktu pomiarowego (por. tab. 5.12, test 3).

Nr	Wejście SSN	MSEU * 10 ²	MSET * 10 ²	S
1	$[k/1352, licznik, licznik_2]$	0.26	16.0	1300
2	$[k/1352, 1 - licznik, licznik_2]$	0.16	0.70	1000
3	$obw(k)[(1 - k/1352), (1 - licznik), licznik_2]$	0.27	0.20	500
4	$[k/1352, obw(k), licznik_2]$	0.35	1.23	500
5	$[1 - k/1352, obw(k), licznik_2]$	0.53	1.30	500
6	$[(k/1352)^2, (k/1352)^2, obw(k), (k/1352)^2 licznik_2]$	2.01	0.49	500
7	$(1 - k/1352)^2 [1, obw(k), licznik_2]$	0.08	0.11	425
8	$[1 - k/1352, (1 - k/1352)^2 obw(k), (1 - k/1352)^2 licznik_2]$	0.51	0.13	2000
9	$[(1 - k/1352)^2, (1 - licznik), (1 - k/1352)^2 licznik_2]$	0.29	0.01	1000
9*	$[(1 - k/1352)^2, (1 - licznik), (1 - k/1352)^2 licznik_2]$	0.14	0.15	1500
10	$(1 - k/1352)^2 [1, (1 - licznik), licznik_2]$	0.14	0.04	1000

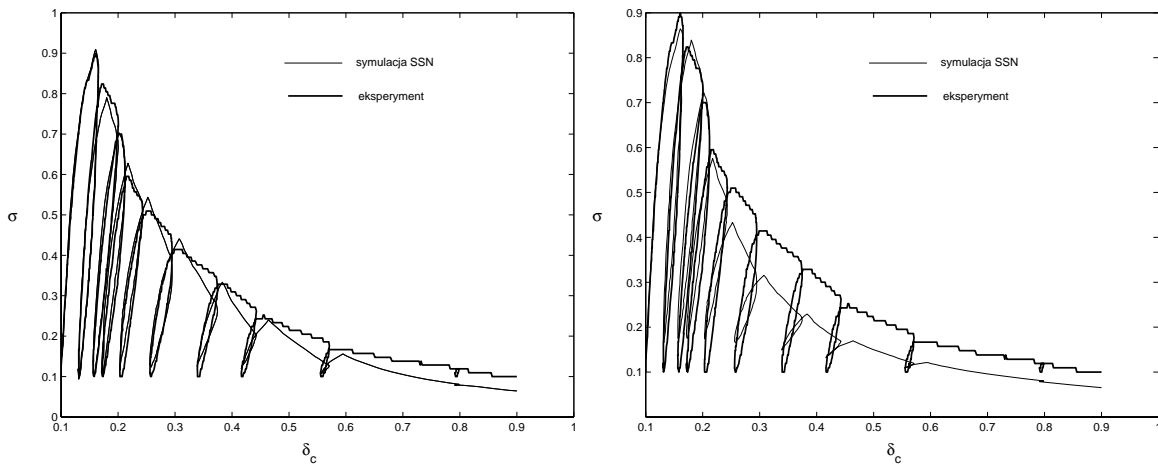
Tab. 5.12. Wyniki symulacji dla grupy 2 badań, test rozciąganie-rozciąganie.

Zaproponowano również, ze względu na niedokładności w odtworzeniu krzywych histerezy, w zakresie obwiedni testu oraz ze względu na kształt poszczególnych pętli, sygnał wejściowy do sieci, nie będący liniową, lecz kwadratową funkcją numeru bieżącego przetwarzanego wzorca. Przemnożenie sygnału wejściowego, dotychczas stosowanego, przez kwadratową funkcję znacznika postępu eksperymentu pozwoliło uzyskać znaczną poprawę uzyskiwanych wyników (por. tab.5.12, testy 6, 7, 8).

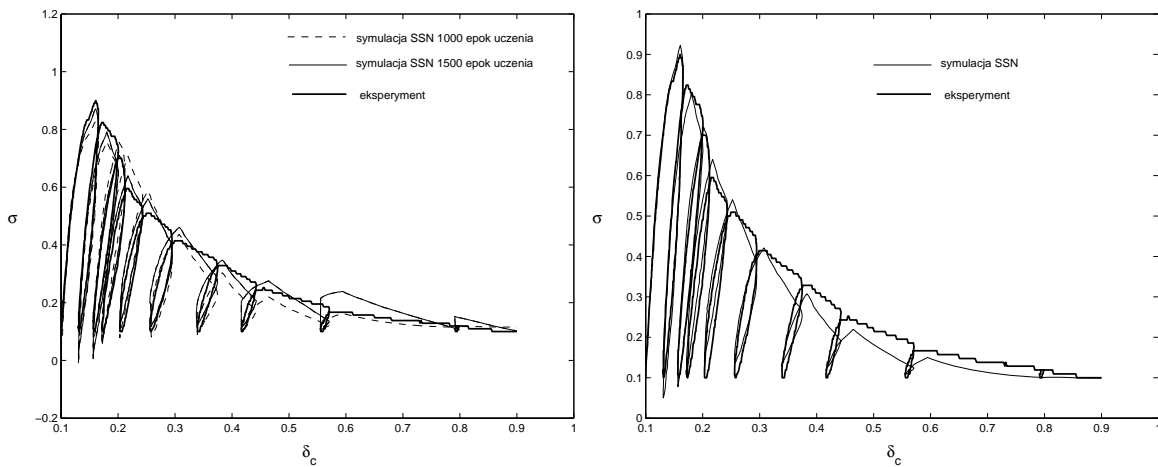


Rys.5.32.c. Test 2: $[k/1352, 1 - licznik, licznik_2]$ Test 6: $[(k/1352)^2, (k/1352)^2 obw(k), (k/1352)^2 licznik_2]$

Za najbardziej efektywne uznano wyniki uzyskane w testach 7 i 10, por. rys. 5.32.



Rys.5.32.d. Test 7: $(1 - k/1352)^2 * [1, obw(k), licznik_2]$ Test 8: $[1 - k/1352, (1 - k/1352)^2 obw(k), (1 - k/1352)^2 licznik_2]$



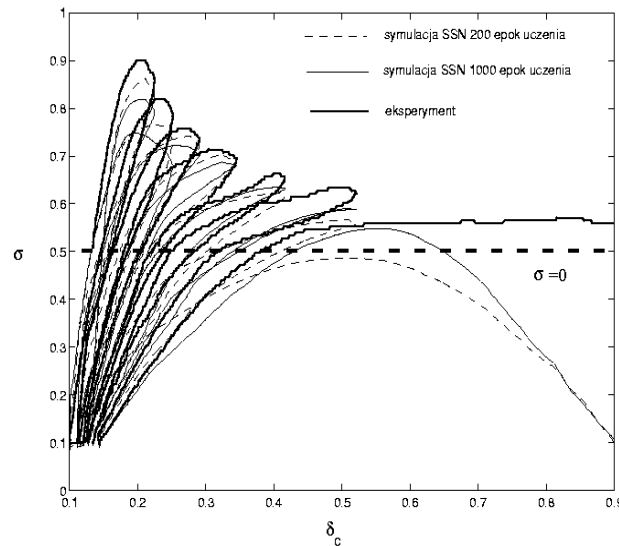
Rys.5.32.e. Test 9: $[(1 - k/1352)^2, (1 - licznik), (1 - k/1352)^2 licznik_2]$ Test 10: $(1 - k/1352)^2 * [1, (1 - licznik), licznik_2]$

7. Wyniki modelowania neuronowego testu rozciąganie-ściskanie – grupa 2 badań.

Do uczenia sieci przeznaczono 5 pierwszych pętli histerezy, na które składało się $L = 1625$ punktów pomiarowych, zdolności generalizacyjne sieci testując na 2 ostatnich pętlach, zawierających $T = 171$ punktów pomiarowych. W tab. 5.13 przedstawiono wyniki symulacji dla najefektywniejszych numerycznie wektorów wejściowych SSN, zachowując numerację związaną z poszczególnymi postaciami wejścia SSN.

Nr	Wejście SSN	$MSEU * 10^2$	$MSET * 10^2$	S
9	$[(1 - k/1352)^2, (1 - licznik), (1 - k/1352)^2 licznik_2]$	0.29	1.44	200
9*	$[(1 - k/1352)^2, (1 - licznik), (1 - k/1352)^2 licznik_2]$	0.29	1.15	1000
10	$(1 - k/1352)^2 [1, (1 - licznik), licznik_2]$	4.32	8.66	1000
10*	$(1 - k/1352)^2 [1, (1 - licznik), licznik_2]$	0.57	26.0	1400

Tab. 5.13. Wyniki symulacji dla grupy 2 badań, test rozciąganie-ściskanie.



Rys.5.33. Test 9: $[(1 - k/1352)^2, (1 - licznik), (1 - k/1352)^2 licznik_2]$

Dla tego testu uzyskano większe błędy uczenia, w porównaniu do grup poprzednich. Wyniki symulacji neuronowej przedstawiono na rys. 5.33. Prawidłowa predykcja eksperymentu występuje tylko w zakresie wydłużenia całkowitego $\delta_c \in [0.1, 0.5]$. Dla większych zakresów wydłużenia symulacja neuronowa jest niepoprawna. Można jednak wysnuć tezę, że pomimo tego faktu, odmienne zachowanie się materiału pod wpływem tej fazy programu jego obciążania zostało przez SSN wychwycone. Wygląd ostatniej, wyprognozowanej przez SSN, pętli histerezy jest odmienny od poprzednich pętli. To sugeruje jakościową zmianę w zachowaniu się materiału w tej fazie doświadczenia, aczkolwiek jej charakter oddany jest niepoprawnie.

SSN przyjęta dla poprzednich grup eksperymentów, jako jedna z najefektywniejszych (test 10), dla tej grupy badań daje oszacowania dużo mniej dokładne. Dodatkowo dla większej ilości epok uczenia występuje problem ponadprzystosowania SSN - poziom błędu $MSET$ rośnie, wraz z poprawą dokładności odwzorowania zbioru uczącego, por. tab. 5.13, test 10.

8. Wyniki modelowania neuronowego testu rozciąganie-rozciąganie – grupa 3 badań.

Dla tej grupy badań, ze względu na podobieństwo otrzymywanych krzywych histerezy

do krzywych otrzymanych dla grupy 2 badań, zastosowano najefektywniejsze, dobrze uprzednio wektory wejściowe, konstruując je wg schematu opisanego dla grupy 1 i 2 badań. Zastosowano ten sam wektor wejścia zarówno dla próbek większych, jak i mniejszych, ich konstrukcję różnicując tylko w obrębie ilości dostępnych punktów eksperymentalnych oraz ilości danych przypadających na poszczególne pętle histerezy.

8.1. *Próbki małe – naprężenia takie same w całym przekroju próbki.*

Do uczenia sieci przeznaczono 9 pierwszych pętli histerezy, na które składało się $L = 1151$ punktów pomiarowych, zdolności generalizacyjne sieci testując na 2 ostatnich pętlach, zawierających $T = 331$ punktów pomiarowych. Ze względu na podobieństwo otrzymanych wyników, do danych z testu rozciąganie-rozciąganie, przeprowadzonych w [89], otrzymano wyniki bardzo zbliżone do przedstawionych w punkcie 6.

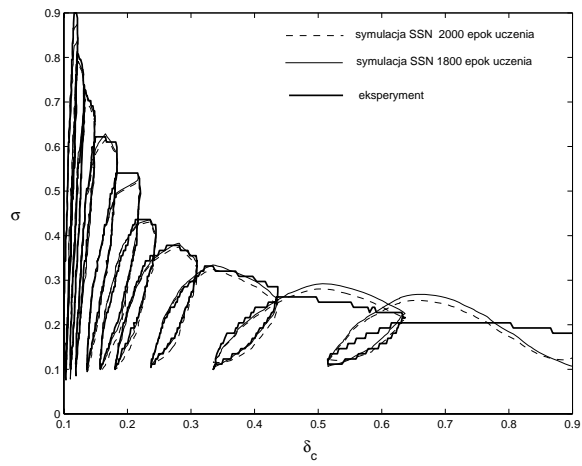
8.2.1. *Próbki duże – naprężenia zróżnicowane w przekroju próbki – testy do pełnego odciążenia.*

Do uczenia sieci przeznaczono 8 pierwszych pętli histerezy, na które składało się $L = 1250$ punktów pomiarowych, zdolności generalizacyjne sieci testując na 2 ostatnich pętlach, zawierających $T = 317$ punktów pomiarowych. Dla próbek dużych wyniki uczenia i testowania przedstawiono w tabeli 5.14., oraz na rys. 5.34.

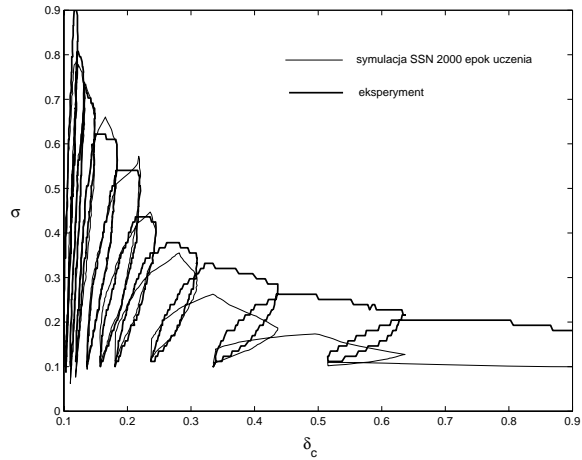
Nr	Wejście SSN	$MSEU * 10^2$	$MSET * 10^2$	S
9	$[(1 - k/1352)^2, (1 - licznik), (1 - k/1352)^2 licznik_2]$	0.08	0.15	1800
9*	$[(1 - k/1352)^2, (1 - licznik), (1 - k/1352)^2 licznik_2]$	0.06	0.08	2000
10	$(1 - k/1352)^2 [1, (1 - licznik), licznik_2]$	0.10	0.63	2000
11	$[(1 - k/1352), (1 - licznik), (1 - k/1352) licznik_2]$	0.06	0.15	500
11*	$[(1 - k/1352), (1 - licznik), (1 - k/1352) licznik_2]$	0.06	0.07	2000

Tab. 5.14. Wyniki symulacji dla grupy 3 badań, próbki duże, test rozciąganie-rozciąganie, do pełnego odciążenia.

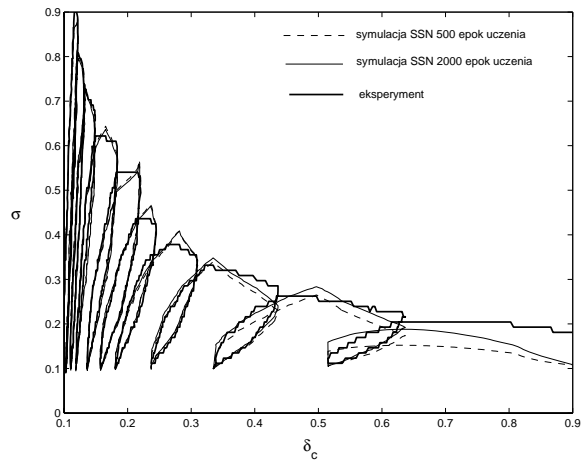
Za najefektywniejsze uznano SSN uzyskane w testach 9 i 11. Uzyskane wyniki sugerują, iż wektor wejścia, oraz ilość epok uczenia powinna być dobierana do każdego zbioru doświadczalnego osobno.



Rys.5.34.a. Test 9: $[(1 - k/1352)^2, (1 - \text{licznik}), (1 - k/1352)^2 \text{licznik}_2]$



Rys.5.34.b. Test 10 $(1 - k/1352)^2 * [1, (1 - \text{licznik}), \text{licznik}_2]$



Rys.5.34.c. Test 11: $[(1 - k/1352), (1 - \text{licznik}), (1 - k/1352) \text{licznik}_2]$

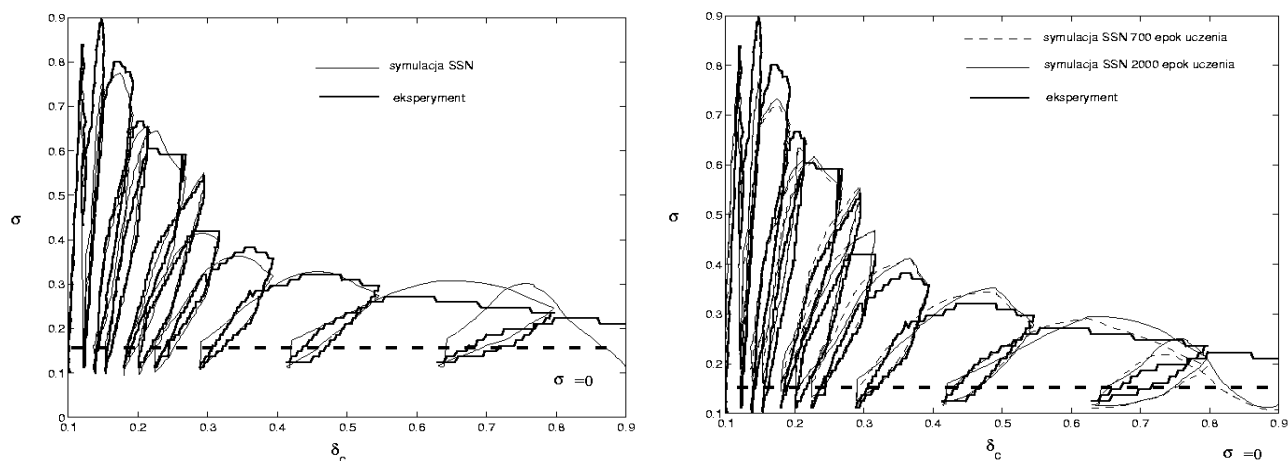
8.2.2. *Próbki duże – naprężenia zróżnicowane w przekroju próbki – testy do niepełnego odciążenia.*

Do uczenia sieci przeznaczono 9 pierwszych pętli histerezy, na które składało się $L = 1137$ punktów pomiarowych, zdolności generalizacyjne sieci testując na 2 ostatnich pętlach, zawierających $T = 205$ punktów pomiarowych. Dla próbek tych wyniki uczenia i testowania przedstawiono w tab. 5.15 oraz na rys. 5.35.

Widoczna jest silna zależność wyników testowania SSN od przyjętego wektora informacji wejściowej do sieci. Ponownie za najefektywniejsze numerycznie uznano SSN uzyskane w testach 9 i 11.

Nr	Wejście SSN	$MSEU * 10^2$	$MSET * 10^2$	S
9	$[(1 - k/1352)^2, (1 - licznik), (1 - k/1352)^2 licznik_2]$	0.31	0.18	2000
10	$(1 - k/1352)^2 [1, (1 - licznik), licznik_2]$	0.51	0.69	1800
11	$[(1 - k/1352), (1 - licznik), (1 - k/1352) licznik_2]$	0.43	0.13	700
11*	$[(1 - k/1352), (1 - licznik), (1 - k/1352) licznik_2]$	0.45	0.12	2000

Tab. 5.15. Wyniki symulacji dla grupy 3 badań, próbki duże, test rozciąganie-rozciąganie do niepełnego odciążenia.



Rys.5.35. Test 9: $[(1 - k/1352)^2, (1 - licznik), (1 - k/1352)^2 licznik_2]$ Test 11: $[(1 - k/1352), (1 - licznik), (1 - k/1352) licznik_2]$

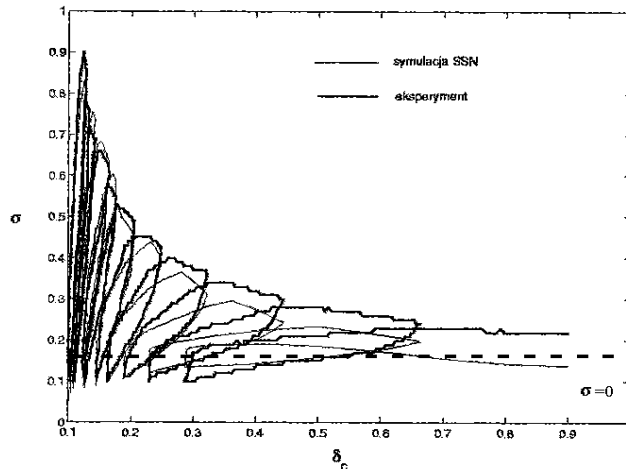
9. Wyniki modelowania neuronowego testu rozciąganie-ściskanie do 5% R_r - grupa 3 badań.

9.1. *Próbki małe.*

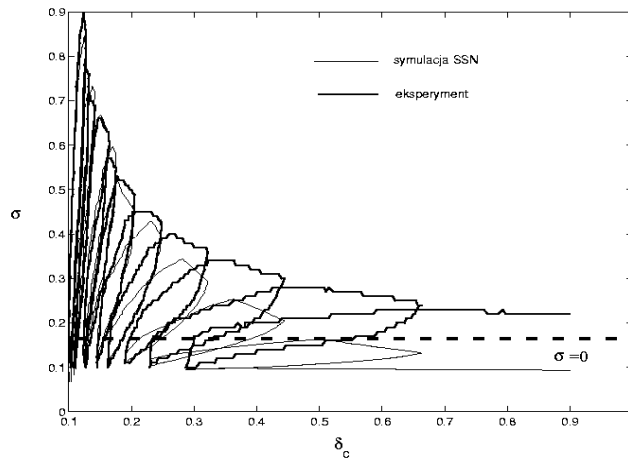
Do uczenia sieci przeznaczono 8 pierwszych pętli histerezy, na które składało się $L = 1022$ punktów pomiarowych, zdolności generalizacyjne sieci testując na pozostałych $T = 441$ punktach pomiarowych.

Nr	Wejście SSN	$MSEU * 10^2$	$MSET^{10^2}$	S
9	$[(1 - k/1352)^2, (1 - licznik), (1 - k/1352)^2 licznik_2]$	0.08	0.15	1000
10	$(1 - k/1352)^2 [1, (1 - licznik), licznik_2]$	0.07	0.81	2000
11	$[(1 - k/1352), (1 - licznik), (1 - k/1352) licznik_2]$	0.03	0.28	2000

Tab. 5.16. Wyniki symulacji dla grupy 3 badań, test rozciąganie-ściskanie.

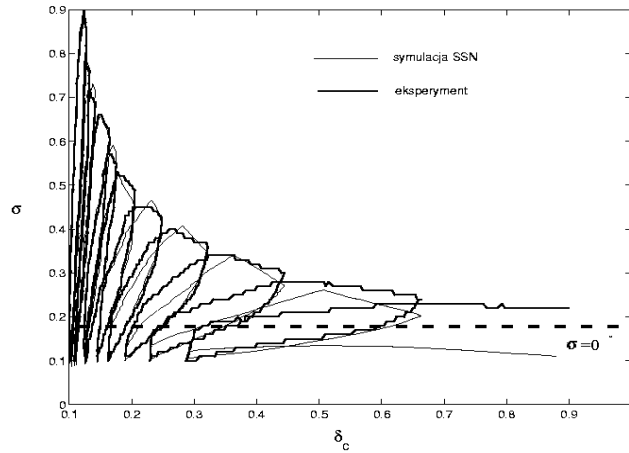


Rys.5.36.a. Test 9: $[(1 - k/1352)^2, (1 - licznik), (1 - k/1352)^2 licznik_2]$



Rys.5.36.b. Test 10: $(1 - k/1352)^2 * [1, (1 - licznik), licznik_2]$

Dla wszystkich testów otrzymano poprawną jakościowo symulację pierwszej z testowych pętli histerezy (przedostatnia pętla doświadczalna). Symulowane dla pętli ostatniej doświadczalnej wartości naprężeń są zbyt małe w porównaniu wartości doświadczalnych. Pomimo to symulacja neuronowa oddaje brak przyrostów naprężeń wraz ze wzrostem odkształceń.



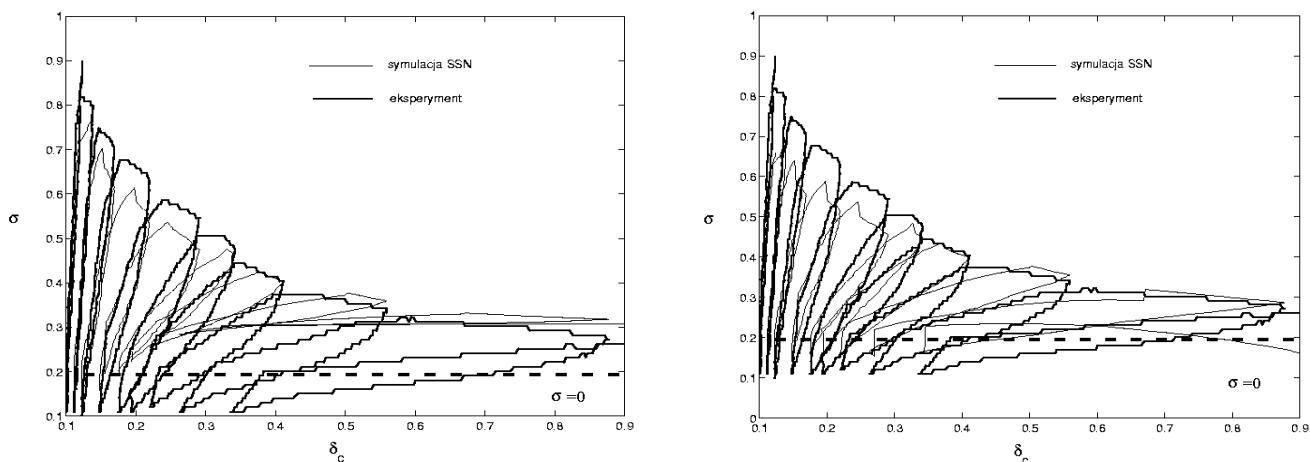
Rys.5.36.c Test 11: $[(1-k/1352), (1-licznik), (1-k/1352)licznik_2]$

9.2. Próbki duże.

Do uczenia sieci przeznaczono 7 pierwszych pętli histerezy, na które składało się $L = 1220$ punktów pomiarowych, zdolności generalizacyjne sieci testując na 2 ostatnich pętlach, zawierających $T = 576$ punktów pomiarowych.

Nr	Wejście SSN	$MSEU * 10^2$	$MSET * 10^2$	S
9	$[(1 - k/1352)^2, (1 - licznik), (1 - k/1352)^2 licznik_2]$	0.26	1.13	2000
10	$(1 - k/1352)^2 [1, (1 - licznik), licznik_2]$	0.30	0.96	1000
10*	$(1 - k/1352)^2 [1, (1 - licznik), licznik_2]$	0.27	0.94	2000
11	$[(1 - k/1352), (1 - licznik), (1 - k/1352) licznik_2]$	0.39	0.21	200

Tab. 5.17. Wyniki symulacji dla grupy 3 badań, test rozciąganie-ściskanie do 5% R_r . Dla tej grupy doświadczeń zastosowanie efektywnych jakościowo dla grup poprzednich wektorów wejścia daje oszacowania mniej dokładne.



Rys.5.37. Test 10: $(1 - k/1352)^2 * [1, (1-\text{licznik}) \text{licznik}_2]$ Test 11:
 $[(1-k/1352), (1-\text{licznik}), (1-k/1352)\text{licznik}_2]$

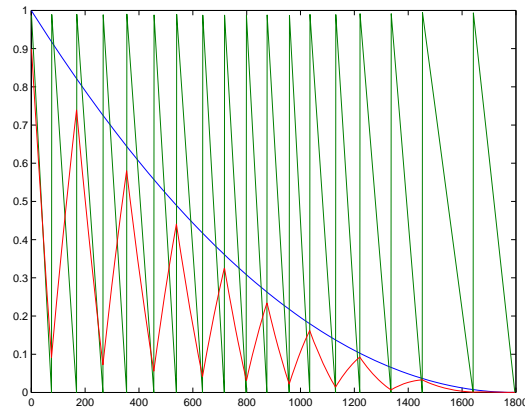
10. Analiza otrzymanych wyników pod kątem doboru wektora wejścia SSN.

10.1. Testy rozciąganie-rozciąganie, grupa 2 oraz próbki małe z grupy 3 i gupa 1.

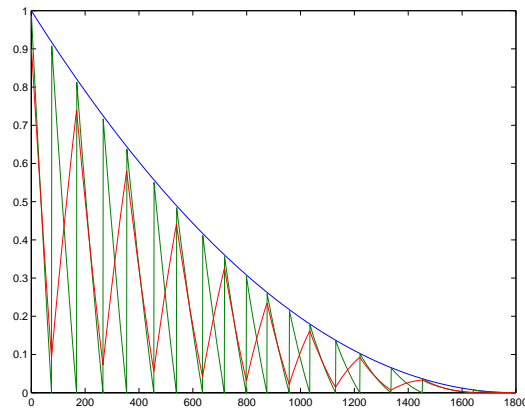
Przeprowadzone testy wykazały, że kluczowym do rozwiązania tego problemu jest dobór odpowiednich wejść do sieci neuronowej.

W zakresie symulacji testu rozciąganie-rozciąganie i predykcji zachowania się materiału w dalszej części tego testu, dla próbek małych w 2 i 3 grupie badań, najlepsze rezultaty otrzymano dla wejść z wytłumieniem sygnału wejściowego, w postaci czynnika $(1 - k/N)^2$, (por. testy: 9,10, tab. 5.12). Dla testu 9 wytłumiono tylko sygnał związany z numerem bieżącego punktu na histerezie oraz znacznik obciążenie-odciążenie. Dla testu 10 wytłumiono również sygnał, związany z numerami poszczególnych punktów, w obrębie każdej z pętli. Sygnał wejściowy do SSN, dla tych 2 testów przedstawiono na rys. 5.38 i 5.39. Dla testu 10 otrzymano lepsze odtworzenie zbioru uczącego, natomiast dla testu 9 dokładniejszą predykcję dwóch ostatnich pętli symulowanej histerezy, co sugeruje, że wytłumienie całości sygnału wejściowego było mniej korzystne, ze względu na jakość testowania.

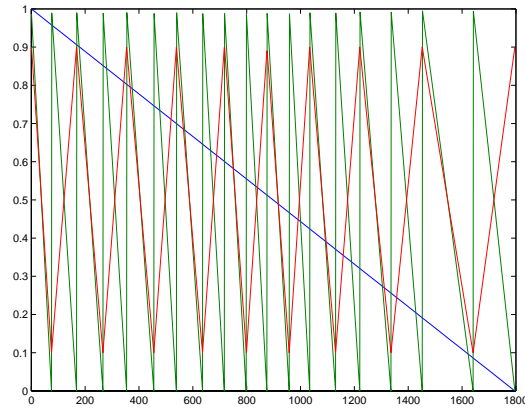
Dla grupy 1 badań, z powodu mniejszego gradientu naprężeń (w stosunku do grup 2 i 3), wytłumienie sygnału nie było konieczne. Sygnał wejściowy dla SSN, dla 1 grupy badań, dla najefektywniejszego testu (nr 3., por. tab. 5.11) przedstawiono na rys. 5.40.



Rys.5.38. Test 9, sygnał wejściowy do SSN: linia niebieska $(1 - k/N)^2$, linia czerwona $(1 - k/N)^2 \text{licznik}_2(k)$, linia zielona $1 - \text{licznik}(k)$ dla $k = 1, 2, \dots, P$



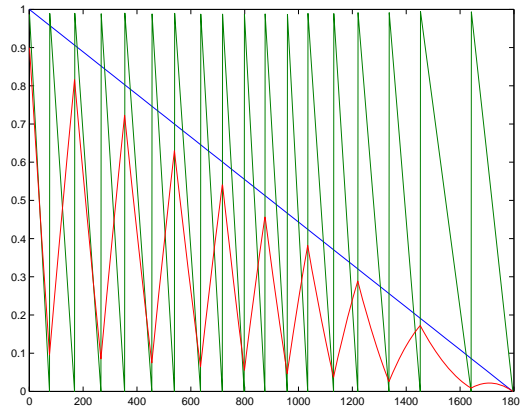
Rys.5.39. Test 10, sygnał wejściowy do SSN: linia niebieska $(1 - k/N)^2$, linia czerwona $(1 - k/N)^2 \text{licznik}_2(k)$, linia zielona $(1 - k/N)^2 (1 - \text{licznik}(k))$ dla $k = 1, 2, \dots, P$



Rys.5.40. Test 2, sygnał wejściowy do SSN: linia niebieska $(1 - k/N)$, linia czerwona $\text{licznik}_2(k)$, linia zielona $1 - \text{licznik}(k)$ dla $k = 1, 2, \dots, P$

10.2. Testy rozciąganie-rozciąganie – próbki duże z grupy 3 i testy rozciąganie-ściskanie do 5% R_r w grupie 3.

Dla próbek dużych grupy 2 silne wytłumienie sygnału wejściowego nie było korzystne. Najlepsze rezultaty otrzymano dla testu o nr 11, dla którego sygnał wejściowy do SSN jest przedstawiony na rys. 5.41. (por. tab 5.14 – 5.17). Dla tych grup badań otrzymano mniej dokładne odwzorowanie ostatniej pętli eksperymentalnej, niż dla badań omówionych w punkcie 10.a., gdyż, otrzymany z symulacji SSN, spadek wartości naprężeń dla ostatniej fazy obciążania materiału jest zbyt gwałtowny, w porównaniu z wartościami eksperymentalnymi. To sugeruje, że wytłumienie sygnału w zakresie pętli testowych być może jest zbyt silne.

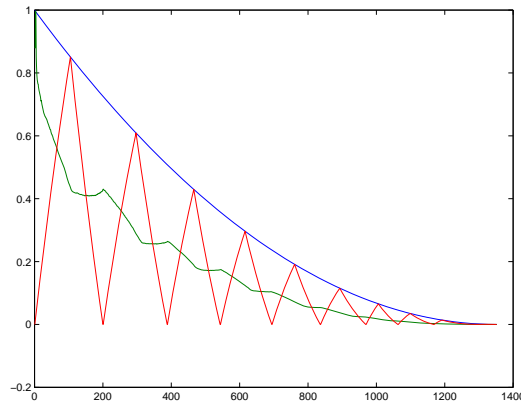


Rys.5.41. Test 11, sygnał wejściowy do SSN: linia niebieska $(1 - k/N)$, linia czerwona $(1 - k/N) * licznik_2(k)$, linia zielona $1 - licznik(k)$ dla $k = 1, 2, \dots, P$

10.3. Informacja dodatkowa, w postaci obwiedni testu monotonicznego

Na przykładzie testu rozciąganie-rozciąganie dla 2 grupy badań, testowano efektywność odtworzenia krzywej histerezy i predykcję dalszej części doświadczenia, w przypadku, gdy wprowadzano na wejście informację o obwiedni testu monotonicznego, odpowiadającej danemu testowi obciążenia cyklicznego.

Rozważano przeskalowanie sygnału wejściowego, przez wartości obwiedni (por. tab. 5.12, test 3) oraz dołączenie wartości obwiedni jako jednego z elementów wektora wejścia (por. tab. 5.12, testy 4-7). Najbardziej efektywnie numerycznie było właśnie takie podejście, gdzie dodatkowo zastosowano tłumienie sygnału wejścia, jak opisane w punkcie 10.1., por. tab. 5.12, test 7, rys. 5.37.



Rys.5.37. Test 7, sygnał wejściowy do SSN: linia niebieska $(1 - k/N)^2$, linia czerwona $(1 - k/N)^2 licznik_2$, linia zielona $(1 - k/N)^2 obw(k)$

Dla tego testu osiągnięto bardzo zbliżone rezultaty symulacji, jak dla testu nr 10 (por. tab. 5.12), dla którego zastosowano zamiast informacji o wartości obwiedni, informację o numerze punktu w obrębie pętli, pozostałe elementy wektora wejściowego pozostawiając bez zmian. Z tego punktu widzenia można stwierdzić, że te informacje wejściowe można stosować zamiennie oraz, że podawanie wartości obwiedni nie jest konieczne do odtwarzania zbioru uczącego i prawidłowego testowania SSN.

11. Wnioski

Przeprowadzone testy i analiza problemu pozwala na przedstawienie następujących wniosków:

- Dla większości rozważanych doświadczeń, dla wybranych wektorów wejścia do SSN, udało się uzyskać poprawną jakościowo symulację neuronową zachowania się materiału pod wpływem zadanego obciążenia cyklicznego. Predykcja dwóch ostatnich, danych w zakresie rozważanych testów, pętli histerezy odkształcenie-napężenie (na podstawie zbioru uczącego, stanowiącego informacje o wcześniejszej fazie doświadczenia) oddaje jakościowo charakter zachodzącego zjawiska zniszczenia materiału.
- Rozważany problem predykcji zachowania się materiału w końcowej fazie doświadczenia okazał się trudny. Efektywność numeryczna SSN oraz poprawność predykcji ostatnich pętli histerezy, w obrębie rozważanych doświadczeń, silnie zależały od przyjętego wektora wejścia SSN.
- Dla większości testów, uzyskane z symulacji SSN wartości naprężeń dla ostatniej pętli histerezy, są niższe, niż wartości uzyskane z eksperymentu.
- Uzyskane wyniki sugerują, iż wektor wejścia, oraz ilość epok uczenia powinna być dobierana do każdego zbioru doświadczalnego osobno.

6. Projektowanie SSN uczonych za pomocą FK

Niniejszy rozdział dotyczy zagadnień związanych z wyborem architektury SSN, uczonych za pomocą algorytmu NDEKF oraz modyfikacji algorytmu wyjściowego, przedstawionego w rozdz. 4. Cechy uczenia SSN właściwe tej metodzie, zostały wykorzystane w celu poprawienia efektywności jej działania.

Rozważano efektywność uczenia w zakresie zmniejszenia otrzymanego błędu uczenia i testowanie, skrócenia czasu uczenia sieci oraz usunięcia, lub zmniejszenia, niekorzystnych zjawisk towarzyszących rozwiązywaniu problemów z użyciem SSN, jako narzędzia obliczeniowego (np. 'overfittingu').

Omówione w rozdziale niniejszym procedury opierają się na ingerencji w wewnętrzną strukturę SSN na poziomie poszczególnych połączeń, lub wybranych neuronów SSN.

Opisane schematy są autorskimi modyfikacjami standartowego postępowania, przy uczeniu SSN za pomocą FK. Są wynikiem analizy przebiegu uczenia SSN, która dotyczy między innymi: wartości wag sieci podczas procesu uczenia, wartości macierzy kowariancji błędów filtracji w modelach teoretycznych, zróżnicowania ww. wielkości w obrębie poszczególnych warstw sieci oraz poszczególnych modeli teoretycznych.

6.1. Dobór architektury SSN w oparciu o algorytm obcinania (pruning).

Omawiana procedura dotyczy automatyzacji wyboru architektury SSN. Spośród różnych możliwości postępowania przy doborze architektury sieci, [3,4], zdecydowano się na tzw. 'pruning' (ang. prune - odcinać). Główną ideą pruningu jest przeprowadzana w trakcie uczenia sieci neuronowej zmiana jej architektury, poprzez redukcję wybranych połączeń pomiędzy neuronami. Po wstępnym uczeniu SSN, co pewną przyjętą ilość epok uczenia – następuje kontrola, które wagi sieci mają niewielki wpływ (według założonych kryteriów) na odwzorowanie wejścia w wyjście i zmniejszenie uzyskanych błędów uczenia i testowania. Połączenia te zostają usunięte. W ten sposób powstaje sieć o mniejszej liczbie parametrów, która uczona jest aż do kolejnej zmiany jej architektury. Podczas konstrukcji zaproponowanego algorytmu oparto się na algorytmie 'lprune' doboru architektury sieci neuronowych, uczonych klasyczną metodą Resilient-propagation [69]. Wstępne testy wykazały jednak konieczność zmodyfikowania wyjściowego algorytmu, celem przystosowania go do procedury uczenia SSN za pomocą FK. Konieczne było uwzględnienie przebiegu procesu uczenia oraz właściwych dla tej metody parametrów regularizacyjnych algorytmu.

Niech

$$\mathbf{w} = \{w^j\}_{j=1}^W \in \mathbf{R}^W \quad (6.1)$$

oznacza wektor parametrów (wag i biasów) rozpatrywanej SSN. W metodzie 'lprune',

por [69], dla każdej wagi w^j sieci neuronowej obliczana jest wartość statystyki Λ :

$$\Lambda(w^j) = \ln\left(\frac{|\sum_{p=1}^L w^j - \eta \frac{\partial E(p)}{\partial w^j}|}{\eta \sqrt{\sum_{p=1}^L (\frac{\partial E(p)}{\partial w^j} - \text{mean}_{\{p=1,2,\dots,L\}} \frac{\partial E(p)}{\partial w^j})^2}}\right) \quad (6.2)$$

gdzie sumowanie przebiega po elementach p L -elementowego zbioru wzorców uczących, η jest pewnym parametrem, $\text{mean}_{\{p=1,2,\dots,L\}}$ – średnią arytmetyczną w zbiorze wzorców uczących. $E(p)$ oznacza funkcję błędu SSN dla x^p , p -tego elementu zbioru uczącego, daną wzorem:

$$E(p) = (h(x^p, \mathbf{w}) - y^p)^2 \quad (6.3)$$

gdzie y^p oznacza znaną odpowiedź na wzorec uczący x^p , h reprezentuje strukturę SSN.

Duże wartości statystyki Λ dla danej wagi oznaczają jej duży wpływ na proces uczenia się sieci neuronowej. Wagi dla których wartości statystyki są małe, ulegają usunięciu.

Problem ile połączeń powinno zostać usunięte, jest otwarty nawet w klasycznych metodach uczenia sieci neuronowych [3,4]. W [69] zaproponowano algorytm dla procedury usuwania kolejnych wag sieci neuronowej uczonej metodą Resilient-propagation, (por. rozdz. 2). Jego skuteczność potwierdzono szeregiem testów, dla których ilość usuwanych połączeń zależała od właściwego tej metodzie uczenia parametru uczenia η . Dla sieci neuronowych uczonych za pomocą algorytmu Kalmana, metoda ta nie została wcześniej przetestowana, nie zaproponowano również jej modyfikacji, w celu dostosowania do tej właśnie metody uczenia SSN. W pracy niniejszej zostały zaproponowane modyfikacje algorytmu przedstawionego w [69], przetestowano ich przydatność oraz przydatność algorytmu oryginalnego zaczerpniętego z [69] do projektowania sieci neuronowej uczonej algorytmem Kalmana.

Przyjęto następujący schemat postępowania:

- dobór wstępnej architektury sieci o dużej liczbie parametrów,
- uczenie sieci początkowej S epok, w celu dokonania porównania efektywności numerycznej sieci początkowej oraz sieci, na której przeprowadzono procedurę usuwania wag,
- dobór ilości epok S_{start} , po których następuje włączenie procedury pruningu do procesu uczenia. Do dalszego uczenia brana jest sieć, której wartości wag pochodzą z S_{start} epoki uczenia sieci oryginalnej;
- by umożliwić szybsze zmiany wag po etapie usuwania połączeń, uczenie rozpoczęto od poziomu szumu właściwego dla wybranej epoki S_{reset} o numerze niższym, lub równym numerowi epoki S_{start} ,
- wybór ilości epok uczenia SSN pomiędzy kolejnymi redukcjami jej połączeń: dobór k , takiego, że jeśli numer bieżącej epoki uczenia jest podzielny przez k , przeprowadzana jest procedura redukcji,

- dobór wartości progowej Λ_{prog} dla statystyki Λ tj. wagi, dla których wartość statystyki Λ jest niższa niż wartość progowa Λ_{prog} , są usuwane,
- redukcja wielkości sieci następuje zgodnie z algorytmem:

- 1 ucz sieć 1 epokę,
- 2 testuj sieć,
- 3 jeżeli numer epoki uczenia jest podzielny przez k , to

3.1 oblicz wartość statystyki Λ dla wszystkich wag sieci

3.2 wyznacz wagi dla których wartość statystyki jest mniejsza, niż zadana wartość progowa, która stanowi określony procent ze średniej wartości statystyki dla wszystkich wag:

$$\Lambda_{prog} = \alpha mean(\Lambda) \quad (6.4)$$

gdzie α oznacza współczynnik procentowy, $mean(\Lambda)$ średnią wartość statystyki, czyli średnią arytmetyczną liczb $\Lambda(w^j)$ dla $j = 1, 2, \dots, W$

3.3 usuń połączenia, przypisując ich wagom wartości równe zero na cały dalszy przebieg uczenia,

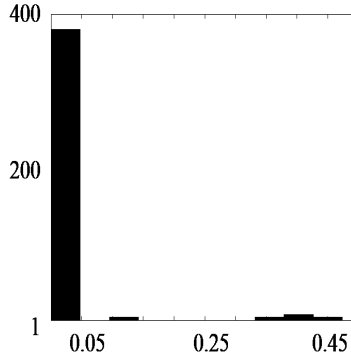
4 powrót do 1.

Mniejsza wartość parametru S_{start} oznacza, iż dokonywana jest redukcja wielkości SSN, uczonej przez większą ilość epok, niż w przypadku wyboru wyższej wartości tego parametru (sieci "niedouczony" do poziomu stabilizowania się błędu uczenia, lub nie osiągnącej satysfakcjonującej efektywności numerycznej). Wybór $S_{reset} = 1$ oznacza zezwolenie na szybkość zmian wag sieci zredukowanej, równą szybkości zmian wag sieci początkowej od początku procesu jej uczenia.

6.2. Uczenie na podstawie analizy wartości macierzy kowariancji przybliżonych błędów filtracji.

Budowa oddzielnych modeli teoretycznych dla każdego neuronu, por. rozdz. 4, powoduje, iż dla poszczególnych modeli, ze względu na inny zestaw parametrów początkowych oraz asymetryczną budowę wektora wejścia sieci, dokładność estymacji wag wyrażona przybliżoną macierzą kowariancji błędów estymacji może być w obrębie sieci zróżnicowana.

Większe, co do wartości, elementy na przekątnej głównej macierzy przybliżonych błędów estymacji oznaczają mniej dokładną estymację odpowiednich wag sieci, małe błędy – estymację dobrą. W SSN, nauczonej do akceptowalnego poziomu błędu uczenia MSE , występują wagi, dla których przybliżony błąd ich estymacji wskazuje, iż estymacja została przeprowadzona mniej dokładnie i w razie dalszego ich uczenia istnieje możliwość zwiększenia jej dokładności, por. rys. 6.1.



Rys.6.1. Przykładowy histogram wartości przybliżonej macierzy kowariancji błędów estymacji w obrębie połączeń wejście – 1 warstwa ukryta

Zaproponowany algorytm polega na dynamicznej zmianie obszarów SSN, które poddane są uczeniu. Neurony, dla których estymacja wag została przeprowadzona mniej dokładnie zostają wyseparowane, a następnie – w kolejnych epokach – następuje uczenie wyłącznie tych neuronów, przy założeniu stałych wartości wag neuronów pozostałych. Po pewnej ilości epok estymacja wag wybranych neuronów powinna być lepsza, co zaowocuje zmniejszeniem poziomu błędu estymacji. Po kilkukrotnym powtórzeniu procedury, poziom błędów estymacji w poszczególnych neuronach powinien się wyrównywać.

Niech $i = 1, 2, \dots, I$ oznacza liczbę neuronów w sieci, $I = I(1) + I(2) + \dots$ gdzie $I(1)$ odpowiada liczbie neuronów w pierwszej warstwie SSN itd. Niech $w^i \in R^{W_i}$ oznacza wektor wszystkich parametrów (wag, biasów, połączeń rekurencyjnych) i -tego neuronu sieci. Dla i -tego modelu, podczas każdej $k + 1$ -ej pętli algorytmu Kalmana, obliczana jest wartość przybliżonej macierzy kowariancji błędów estymacji postaci:

$$P_{k+1}^i = (I - G_k^i H_k) P_k^i + Q_k^i \quad (6.5)$$

przy czym

$$P_{k+1}^i \approx E(w^i - \hat{w}_{k+1}^i)(w^i - \hat{w}_{k+1}^i)^T \quad (6.6)$$

gdzie \hat{w}_{k+1}^i jest estymatorem dla w^i wyznaczonym po podaniu sieci $k + 1$ -go wzorca uczącego. W_i jest wymiarem macierzy P^i . Na jej przekątnej głównej znajdują się wielkości:

$$P_{k+1}^i(m) \approx E(w^i(m) - \hat{w}_{k+1}^i(m))(w^i(m) - \hat{w}_{k+1}^i(m))^T \quad (6.7)$$

dla $m = 1, 2, \dots, W_i$, będące przybliżonymi błędami średniokwadratowymi estymacji poszczególnych wag i -tego neuronu.

Ze względu na duże różnice w obrębie każdej z warstw SSN, porównywano jakość estymacji poszczególnych modeli w każdej z warstw oddzielnie.

Kolejne etapy zmodyfikowanego algorytmu NDEKF dla sieci o 2 warstwach ukrytych przyjęto następująco:

- wstępne uczenie SSN przez $s = 1, 2, \dots, S_0$ epok, a następnie w obrębie warstwy 1 ukrytej zastosowano postępowanie:
- wyznaczenie średniego poziomu błędu średniokwadratowego estymacji wag w całej warstwie:

$$M = \text{mean}_{\{m=1, \dots, n_i, i=1, \dots, I(1)\}} P_{k+1}^i(m) \quad (6.8)$$

- wyznaczenie średniego poziomu błędu średniokwadratowego estymacji wag w obrębie poszczególnych neuronów:

$$M(i) = \text{mean}_{\{m=1, \dots, W_i\}} P_{k+1}^i(m) \quad (6.9)$$

dla $i = 1, 2, \dots, I(1)$

- wyznaczenie neuronów w sieci, dla których poziom błędu średniokwadratowego estymacji wag jest niższy, niż ustalony procent poziomu błędu średniokwadratowego estymacji wag w całej warstwie, gdzie współczynnik procentowy oznaczamy α , czyli i takich że:

$$M(i) < \alpha M \quad (6.10)$$

- dla tych neuronów na dalsze S_{krok} epok uczenia wagi ustalono na poziomie dotychczasowym,
- dla pozostałych neuronów wagi uczono przez S_{krok} epok, po których znowu dla całej sieci obliczono wielkości M , $M(i)$
- procedurę powtarzano, aż do osiągnięcia kryterium stopu algorytmu.

Analogiczny schemat zastosowano dla drugiej warstwy ukrytej.

6.3. Dobór architektury SSN na podstawie algorytmu obcinania – wybór parametrów sterujących na podstawie testów numerycznych

W niniejszym paragrafie zostaną przedstawione wyniki testów efektywności algorytmu, zaproponowanego w 6.1, dotyczącego doboru architektury SSN, w oparciu o procedurę pruningu SSN, uczonej z wykorzystaniem algorytmu NDEKF. Zamieszczone

wyniki dotyczą omawianego w punkcie 6.2.1. niniejszej pracy problemu predykcji pętli histerezy dla kompozytu hierarchicznego, złożonego z wiązek nadprzewodników.

Celem zbadania efektywności procedury, wybrano jeden z analizowanych wektorów wejścia do SSN oraz ustalono strukturę sieci. Testy wykonano dla wektora wejścia

$$[F(k), d(k), F(k + 1)] \quad (6.11)$$

dla sieci jednokierunkowej FLNN, o architekturze wyjściowej 3-25-25-1. Rozważano problem testowania na 2 ostatnich pętlach analizowanej histerezy.

W zaproponowanym szeregu testów, badano wpływ:

- liczby usuwanych w kolejnych fazach algorytmu połączeń międzyneuralnych (wartość parametru Λ_{prog})
- długości uczenia sieci po każdym kolejnym usunięciu połączeń (parametr k)
- liczby epok, po której najkorzystniej jest włączyć do uczenia procedurę pruningu (wartość parametru S_{start})
- parametry szumów w modelach teoretycznych, które najkorzystniej jest przyjąć dla uczenia zmodyfikowanej SSN (wartość parametru S_{reset})

na efektywność obliczeniową SSN. Miarę efektywności działania SSN stanowił poziom błędów uczenia i testowania MSE .

Testy wykazały konieczność modyfikacji algorytmu przedstawionego w [69], opracowanego dla metody uczenia Rprop, por. rozdz. 2. Efektywność algorytmu, zaproponowanego w ww. pracy, nie została wcześniej przetestowana dla SSN, uczonych za pomocą algorytmu Kalmana. Nie zaproponowano również wcześniej modyfikacji oryginalnego algorytmu, w oparciu o statystykę Λ , dla innych metod uczenia sieci. Parametr η zaproponowano po wstępnych testach numerycznych na stałym poziomie, niezależnym od numeru bieżącej epoki uczenia. Zmieniono również, w stosunku do [69], kryterium stopu algorytmu. Oparto je na ustalonej odgórnie liczbie S epok uczenia sieci. Parametr S wybrano tak, by błędy uczenia i testowania stabilizowały się na poziomie, odpowiadającym oczekiwanej dokładności rozwiązania tego konkretnego problemu. Zestawienie parametrów dla przykładowych testów przedstawiono w tabeli 6.1.

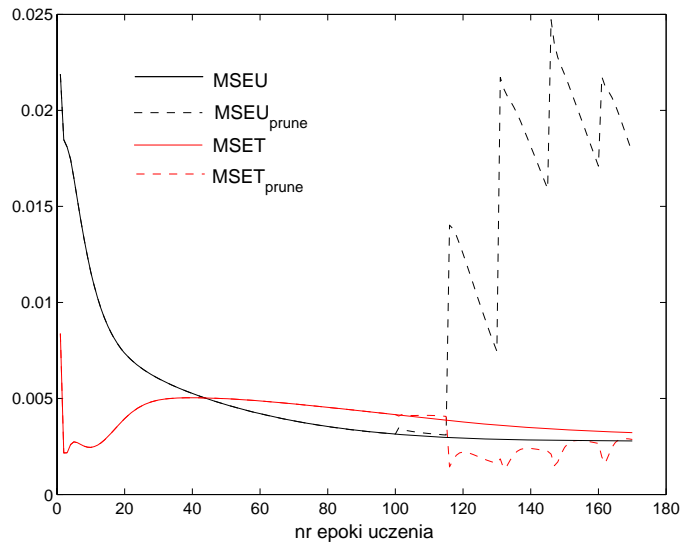
nr testu	S_{start}	S_{reset}	k	Λ_{prog}	S
1	100	30	15	$mean(\Lambda)$	170
2	200	30	30	$mean(\Lambda)$	370
3	200	30	30	$0.5mean(\Lambda)$	370
4	200	30	30	$0.75mean(\Lambda)$	370
5	200	50	10	$0.5mean(\Lambda)$	370
6	50	50	30	$0.5mean(\Lambda)$	200

Tabela 6.1. Zestawienie parametrów dla przykładowych testów algorytmu pruningu zaproponowanego w 6.1.

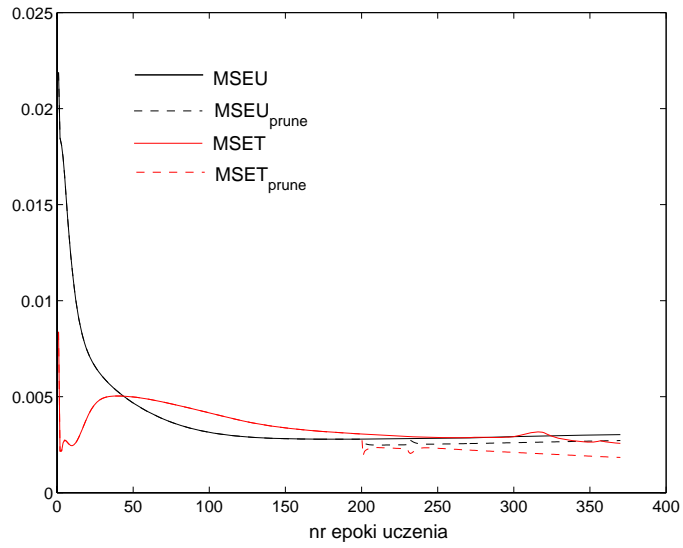
Stwierdzono następujące, jakościowo różne, przebiegi procesu uczenia i testowania sieci, w zależności od wartości parametrów, kontrolujących proces pruningu i uczenia sieci zredukowanej:

- usunięcie części neuronów sieci, powyżej pewnego progu, powoduje wzrost błędu uczenia bezpośrednio po epoce, w której następuje pruningu SSN, następnie błąd ten maleje szybciej, niż dla SSN wyjściowej. Przyjęcie zbyt dużej liczby usuwanych każdorazowo neuronów, lub zbyt krótkiego czasu uczenia sieci, bezpośrednio po zmniejszeniu jej rozmiarów, powoduje, iż wartość błędu sieci nie spada do pożądanego poziomu lub rośnie, por. test 1,2, rys. 6.2. Za parametry graniczne, dla których stwierdzono ww. niekorzystne efekty w przebiegu procesu uczenia, uznano: współczynnik procentowy dla wielkości Λ_{prog} : $\alpha \approx 1$ oraz ilość epok uczenia, pomiędzy kolejnymi redukcjami wielkości sieci, równą $k = 15$,
- przyjęcie parametru sterującego liczbą usuwanych neuronów na poziomie $\alpha \approx 0.5$ nie powoduje wzrostu błędu uczenia sieci, por. test 3. Efektem przeprowadzonej procedury jest uzyskanie SSN o niższych błędach uczenia oraz testowania, niż wyjściowa sieć pełna, rys 6.3,
- przyjęcie parametru sterującego ilością usuwanych neuronów na poziomie $\alpha \approx 0.75$ powoduje wzrost błędu uczenia sieci, w epoce bezpośrednio po usunięciu połączeń, następnie opadanie krzywej uczenia jest znacznie szybsze, niż dla SSN wyjściowej. Odpowiednio długie ($k \geq 30$) uczenie sieci zredukowanej pozwala jednak na spadek błędów uczenia do poziomu nie osiąganego przez wyjściową SSN, por. test 4, rys 6.4. W zakresie testowania także odnotowano obniżenie poziomu uzyskiwanych błędów.

Powyższe rozważania dotyczą sytuacji, gdy procesowi pruningu poddano sieć neuronową, dla której spadek poziomu błędów uczenia zaczął się stabilizować (dla tego problemu ≈ 150 epoki uczenia). Jeżeli usuwano neurony SSN tylko wstępnie nauczanej, dla której błędy uczenia maleją jeszcze bardzo gwałtownie (≈ 50 epoki uczenia), nie osiągnięto poprawy rezultatów, w stosunku do sieci wyjściowej, por. rys. 6.5, test 6.



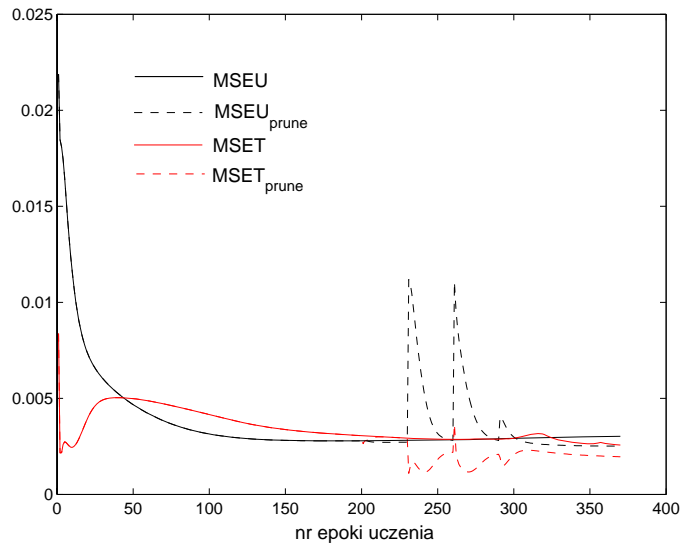
Rys.6.2. Test 1, $S_{start} = 100$, $S_{reset} = 30$, $k = 15$, $\Lambda_{prog} = mean(\Lambda)$



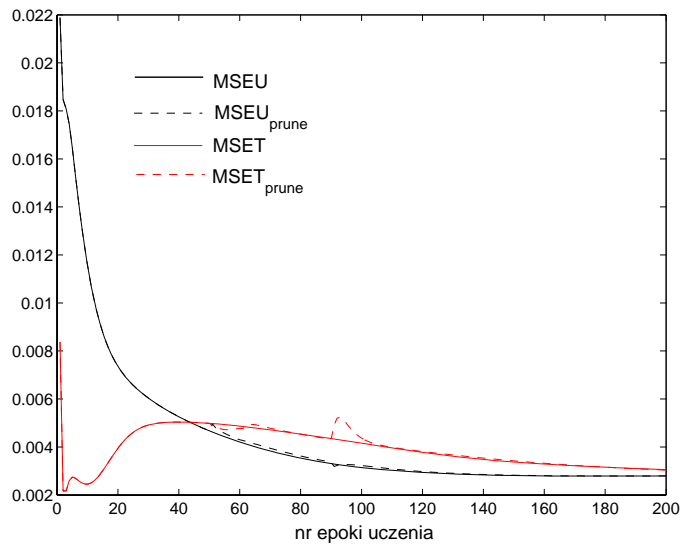
Rys.6.3. Test 3, $S_{start} = 200$, $S_{reset} = 30$, $k = 30$, $\Lambda_{prog} = 0.5mean(\Lambda)$

Nr testu	$MSEU(S)$	$MSEU_{prune}(S)$	$MSET(S)$	$MSET_{prune}(S)$
1	0.0027894	0.017279	0.0032182	0.0028317
3	0.0030258	0.0027187	0.0025696	0.0018334
4	0.0030258	0.0025201	0.0025696	0.0019575
6	0.0027951	0.0027969	0.0030559	0.0030291

Tabela 6.2. Zestawienie wyników wybranych testów po zakończeniu procedury uczenia



Rys.6.4. Test 4, $S_{start} = 200$, $S_{reset} = 30$, $k = 30$, $\Lambda_{prog} = 0.75mean(\Lambda)$



Rys.6.5. Test 6, $S_{start} = 50$, $S_{reset} = 50$, $k = 30$, $\Lambda_{prog} = 0.5mean(\Lambda)$

Zaproponowano także:

- test zachowujący opisany w [69] warunek zatrzymania algorytmu, ale z ustalonym parametrem η ,
- test z zachowanym z konstrukcji algorytmu dla metody Rprop, zmiennym parametrem η i kryterium zatrzymania, dobranym dla metody DEKF,
- wykorzystanie zaproponowanej w ww. pracy idei zmiany procenta usuwanych wag, w oparciu o wielkość zwaną utratą generalizacji SSN, por. [69], w połą-

czeniu ze zmiennym parametrem η oraz ze zmienionym w stosunku do zaproponowanego w [69] kryterium stopu procedury.

Przetestowano także efektywność oryginalnego algorytmu, zaczerpniętego z [69], zaimplementowanego dla SSN, uczonych z wykorzystaniem algorytmu DEKF.

Przeprowadzone testy nie potwierdziły celowości konstrukcji algorytmu pruningu SSN, uczonych za pomocą algorytmu DEKF, z wykorzystaniem powyższych elementów algorytmu oryginalnego.

Przeprowadzone testy pozwalają na przedstawienie następujących wniosków:

- przy odpowiednim doborze parametrów sterujących oraz wprowadzonych modyfikacjach, algorytm pruningu SSN, uczonej za pomocą DEKF, z wykorzystaniem idei kontroli ilości usuwanych połączeń w oparciu o wartości statystyki Λ , pozwala na przeprowadzenie skutecznej redukcji wielkości sieci neuronowej,
- przebieg procesu uczenia można kontrolować za pomocą zaproponowanych parametrów,
- przy odpowiednim doborze parametrów, możliwe jest skrócenie procesu uczenia SSN, uzyskanie niższych błędów zarówno uczenia, jak i testowania, dla sieci zredukowanej w stosunku do sieci wyjściowej.

6.4. Uczenie SSN na podstawie analizy wartości macierzy kowariancji przybliżonych błędów filtracji – wybór parametrów sterujących na podstawie testów numerycznych

W niniejszym paragrafie zostaną przedstawione wyniki testów efektywności algorytmu, zaproponowanego w 6.2., dotyczącego modyfikacji oryginalnego postępowania, przy uczeniu SSN z wykorzystaniem algorytmu NDEKF, w oparciu o analizę wartości macierzy kowariancji przybliżonych błędów filtracji, w modelach dotyczących poszczególnych neuronów sieci. Zamieszczone wyniki także dotyczą problemu predykcji pętli histerezy, dla kompozytu hierarchicznego złożonego z wiązek nadprzewodników, dla problemu opisanego w punkcie poprzednim. Po wstępnej analizie krzywych błędów $MSEU$ i $MSET$ dla analizowanego problemu, przyjęto w większości testów parametr S_0 , równy 50 epoce uczenia, jako na punkt, w którym krzywe błędów przestają maleć z początkową znaczną szybkością oraz zanika ich wrażliwość na wylosowane wagi startowe SSN. Parametr S_{krok} , wyznaczający liczbę epok, po której następuje kolejna modyfikacja uczonych obszarów SSN, przyjmowano w formie wielokrotności liczby 25 epok. Wtedy następowała zmiana poziomu błędów uczenia i testowania, którą uznano za znaczącą. Dla wszystkich testów przyjęto kryterium zatrzymania $S=500$ epok uczenia, jako liczbę po której następowała stabilizacja otrzymywanych błędów MSE. Z procesu uczenia nie wyłączano neuronu wyjściowego SSN.

Przykładowe parametry, w zakresie ww. to:

- TEST 0: $S_0 = 50$, $\alpha = 1$, $S_{krok} = 50$
- TEST 1: $S_0 = 50$, $\alpha = 1$, $S_{krok} = 25$
- TEST 2: $S_0 = 50$, $\alpha = 0.75$, $S_{krok} = 25$
- TEST 3: $S_0 = 50$, $\alpha = 1$, $S_{krok} = 2$
- TEST 4: $S_0 = 50$, $\alpha = 1.3$ dla neuronów warstwy 1 ukrytej $\alpha = 1$ dla neuronów warstwy 2 ukrytej, $S_{krok} = 25$
- TEST 5: $S_0 = 50$, $\alpha = 1.3$, $S_{krok} = 25$
- TEST 6: $S_0 = 50$, $\alpha = 1.7$, $S_{krok} = 25$
- TEST 7: $S_0 = 100$, $\alpha = 1.7$ od 100 do 300 epoki uczenia, $\alpha = 1$, od 300 do 500 epoki uczenia, $S_{krok} = 50$

Współczynnik α wyznacza liczbę neuronów, dla których następuje ustalenie wartości wag na dalszy przebieg procesu uczenia. Przyjęcie większej wartości współczynnika α , powoduje, iż większe obszary sieci zostaną wyłączone z adaptacji na kolejne S_{krok} epoki uczenia.

Wyniki działania zaproponowanej procedury porównywano z wynikami otrzymywanymi dla wyjściowej SSN, uczonej również przez 500 epok. Wyniki testów w zakresie otrzymanych błędów uczenia sieci oryginalnej ($MSEU$), sieci poddanej algorytmowi ingerencji ($MSEU_{ingerencja}$) oraz błędów testowania sieci oryginalnej ($MSET$) i sieci poddanej algorytmowi ingerencji ($MSET_{ingerencja}$) zestawiono w tabeli 6.3.

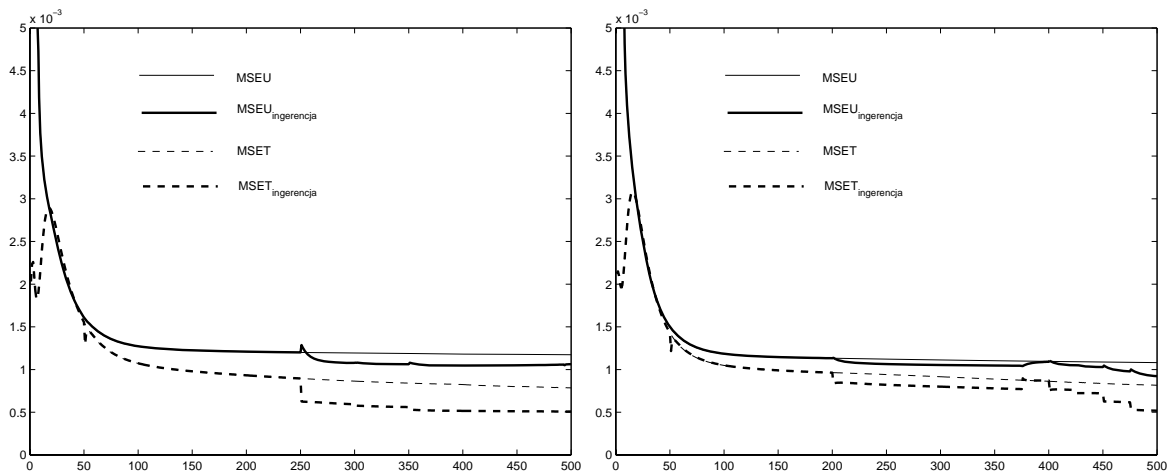
Nr testu	$MSEU * 10^3$	$MSEU_{ingerencja} * 10^3$	$MSET * 10^3$	$MSET_{ingerencja} * 10^3$
0	1.1	1.00	0.78	0.50
1	1.0	0.92	0.82	0.50
2	1.0	0.91	0.82	0.52
3	1.1	1.14	0.89	0.88
4	1.1	1.00	0.86	0.64
5	1.1	1.00	0.78	0.50
6	1.0	0.98	0.74	0.42

Tabela 6.3. Zestawienie wyników dla testowania na 2 ostatnich pętlach histerezy

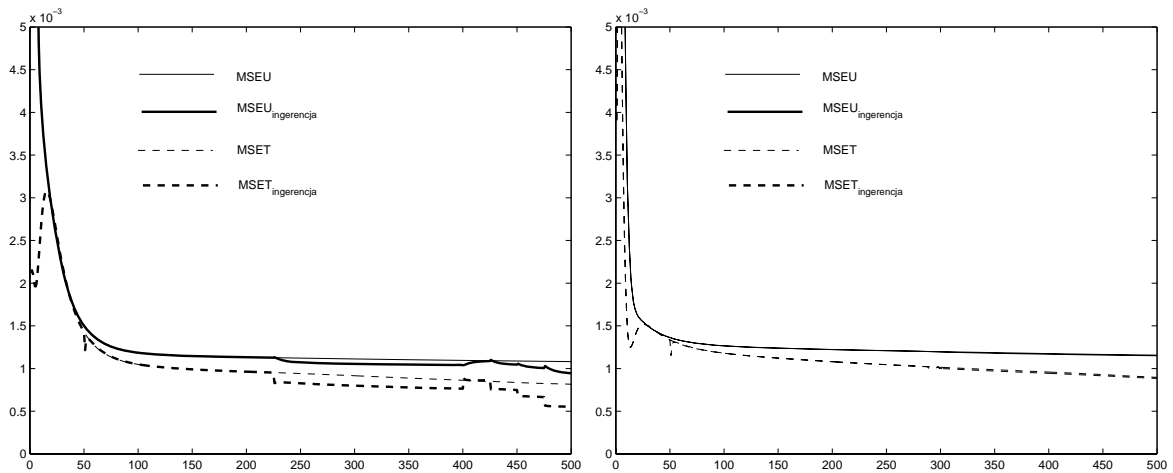
Poziom błędów uczenia i testowania przedstawiony jest dla poszczególnych testów na rys.6.6.a-d. Przyjęty warunek wyznaczania 'aktywnych', ze względu na proces uczenia

neuronów SSN powoduje, że wyjściowy algorytm ulegnie modyfikacji dopiero, gdy nastąpi zróżnicowanie poziomu błędów estymacji, pomiędzy poszczególnymi neuronami w sieci.

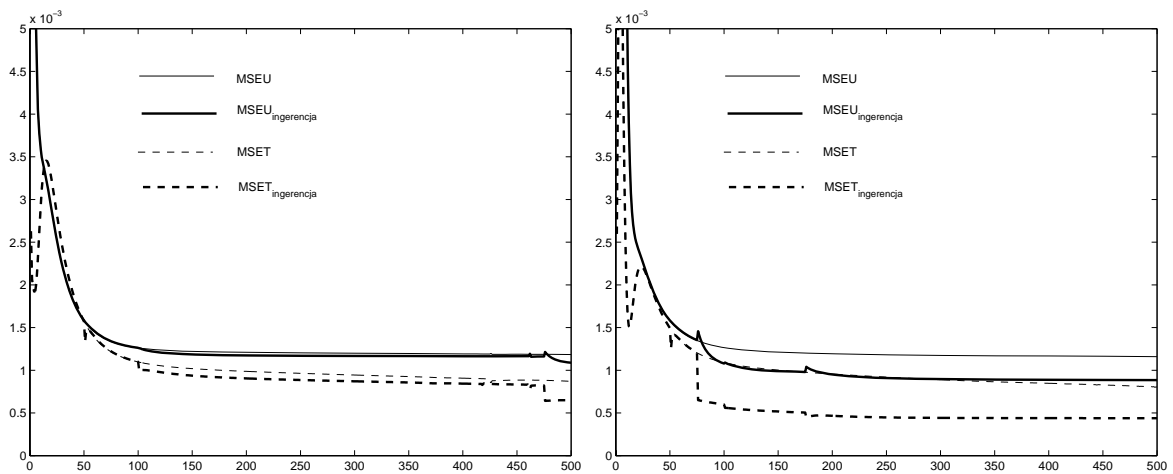
Przykładowo, dla testu 0: redukcja obszarów SSN, poddawanych uczeniu, nastąpiła po 250 epokach uczenia, gdy uczone były tylko obszary SSN związane z kolejno 37,32,27,27,27 neuronami SSN, na przyjętych 50 neuronów poddanych procedurze. Zastosowanie procedury 'douczenia' neuronów, dla których poziom błędów estymacji wag wskazywał na mniejszą dokładność estymacji, spowodowało w końcowej fazie uczenia wyrównanie poziomu ww. błędów. Redukcja poddanego uczeniu obszaru sieci wahała się od 0% (sieć pełna) do ok. 50%, przy czym aktywne obszary sieci, co każde S_{krok} epok uczenia, mogły ulec zmianie.



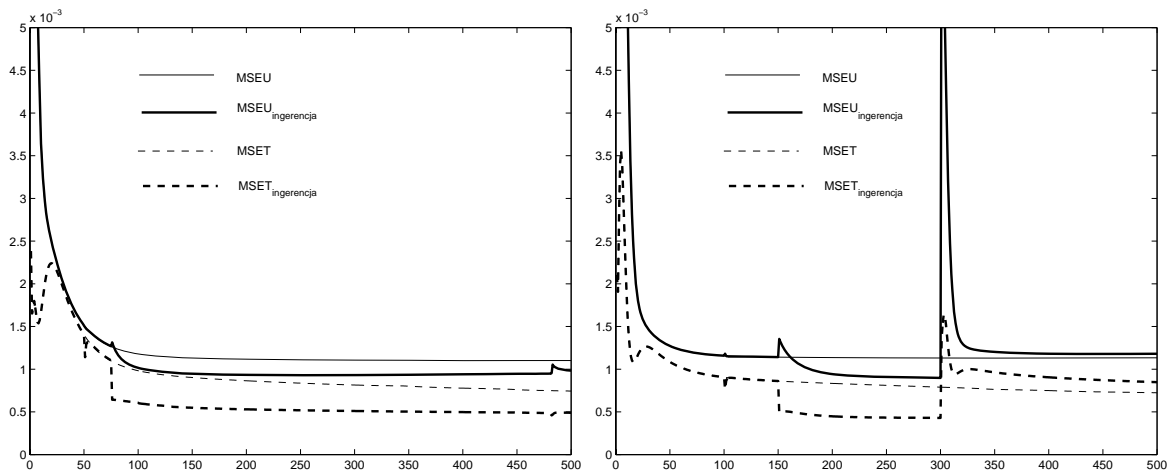
Rys. 6.6.a. Wyniki uczenia - test 0 oraz test 1



Rys. 6.6.b. Wyniki uczenia - test 2 oraz test 3



Rys. 6.6.c. Wyniki uczenia - test 4 oraz test 5



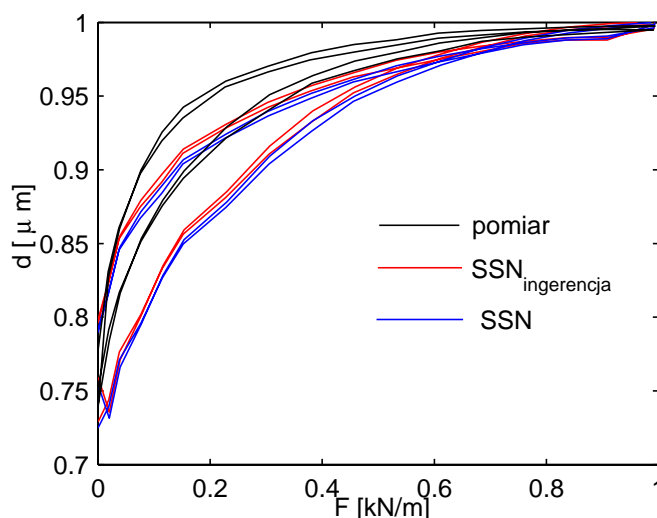
Rys. 6.6.d. Wyniki uczenia - test 6 oraz test 7

Przedstawione testy pozwalają na przedstawienie następujących wniosków:

- przyjęcie zbyt małej wartości parametru S_{krok} nie powoduje różnic w procesie uczenia, w stosunku do wyjściowej SSN,
- włączenie zaproponowanej procedury do procesu uczenia sieci, zaowocowało obniżeniem poziomu błędów uczenia, jak i testowania, w stosunku do sieci uczonej wyjściowym algorytmem Filtra Kalmana, przy czym obniżenie błędu testowania było bardziej znaczące,
- przyjęcie parametru $\alpha \approx 1$, z równoczesną zmianą obszarów uczących się sieci co 25 epok, zaowocowało stopniowym, kilkukrotnym w trakcie trwania procesu uczenia, obniżaniem się wartości błędów uczenia i testowania, przy czym sieć o najniższych ww błędach otrzymywano w końcowej fazie postępowania,

- przyjęcie parametru $\alpha \approx 1.5$ spowodowało znaczne obniżenie błędów uczenia i testowania sieci już w fazie początkowej procesu uczenia, po czym opadanie krzywych błęd było podobnie szybkie, jak dla sieci wyjściowej,
- zmiana parametrów algorytmu w czasie uczenia SSN, lub ich różnicowanie w obrębie warstw sieci, nie było korzystne dla przebiegu procesu uczenia.

Odtworzenie przykładowej petli testowej dla wybranego testu nr. 5. przedstawiono na rys.6.7.



Rys.6.7. Wyniki testowania sieci na wybranej petli histerezy, test 5

Przedstawiona procedura pozwala zatem na dalsze obniżenie błędów uczenia i testowania w sytuacji, gdy dla sieci wyjściowej błędy te stabilizują się na ustalonym poziomie. Przyjęcie zaproponowanych parametrów procedury pozwala na skrócenie czasu uczenia SSN (testy 5,6), bądź stopniowe poprawianie jej efektywności (testy 1,2).

6.5. Podsumowanie

Zaproponowano dwie metody projektowania SSN uczonych za pomocą FK:

- dobór architektury sieci, poprzez redukcję liczby połączeń międzyneuralnych wstępnie zaprojektowanej SSN, opartą o metodę statystycznej oceny istotności danej wagi sieci w procesie uczenia SSN,
- dynamiczną zmianę obszarów uczenia SSN, związaną z czasowym ustaleniem wartości wybranych wag sieci, opartą o analizę macierzy kowariancji przybliżonych błędów estymacji w modelach teoretycznych.

Obie zaproponowane procedury są autorskimi rozwiązaniami, wykorzystującymi specyfikę procesu uczenia SSN, opartego o filtr Kalmana, wraz z zastosowaniem rozprężenia modelu względem poszczególnych neuronów sieci (NDEKF).

Wykazano, iż stosując metodę redukcji połączeń międzyneuralnych sieci, realizowaną poprzez zaproponowany w 6.1 algorytm obcinania, z odpowiednio dobranymi parametrami sterującymi, możliwe jest:

- skrócenie procesu obliczeń neuronowych, poprzez uzyskanie niższych błędów uczenia, niż dla SSN wyjściowej,
- zwiększenie efektywności predykcji neuronowej, poprzez uzyskanie SSN, o niższym błędzie testowania, niż sieć niezredukowana.

Wykazano, iż stosując metodę uczenia wybranych obszarów SSN, w oparciu o algorytm zaproponowany w 6.2 możliwe jest:

- uzyskanie obniżenia błędów uczenia i testowania SSN, w sytuacji gdy dla sieci wyjściowej błędy te stabilizują się na ustalonym poziomie,
- skrócenie czasu uczenia, poprzez uzyskanie określonych poziomów błędów $MSEU$ uczenia oraz $MSET$ testowania dla mniejszej ilości epok uczenia SSN, niż dla sieci nie poddanej tej procedurze.

7. Pakiet programów do uczenia oraz projektowania jednokierunkowych i rekurencyjnych SSN uczonych za pomocą FK

Wszystkie wyniki w ramach niniejszej rozprawy, dotyczące uczenia SSN za pomocą filtru Kalmana, otrzymano w oparciu o własne oprogramowanie, napisane w środowisku do obliczeń naukowo–inżynierskich MATLAB (Matrix Laboratory), [10].

7.1. Motywacja wyboru środowiska obliczeniowego

Przepływ sygnału w sztucznych sieciach neuronowych warstwowych, stosowanych w pracy, zarówno jednokierunkowych, jak i rekurencyjnych, oparty jest o przetwarzanie wektorów. Wejścia do SSN, stosowanych w niniejszej pracy, są wektorami przynajmniej trójelementowymi, przemnożone przez odpowiednie wektory parametrów sieci (wag i biasów) dają potencjały poszczególnych neuronów. W postaci wektorowej mogą również zostać zapisane potencjały neuronów w obrębie poszczególnych warstw SSN oraz sygnał wychodzący z poszczególnych warstw sieci, por. rozdz. 2. Algorytm Kalmana również oparty jest o przetwarzanie wielkości wektorowych: wektorów parametrów sieci, wektorów błędów prezentacji kolejnych wzorców uczących, oraz macierzowych: macierzy wzmocnienia Kalmana, przybliżonej macierzy kowariancji błędów estymacji a posteriori oraz macierzy linearyzacji równania obserwacyjnego, por. rozdz. 3,4. MATLAB jest językiem programowania wysokiego poziomu, który został zaprojektowany i zoptymalizowany pod kątem operacji wektorowych i macierzowych, [96]. Dodatkowo analizę otrzymywanych wyników ułatwiają polecenia wizualizacji danych w zakresie grafiki dwuwymiarowej, natomiast analizę i obróbkę wstępną zbiorów danych umożliwia między innymi pakiet *Statistic Toolbox* [70].

Środowisko to posiada także zaimplementowany pakiet *Neural Network Toolbox*, służący do projektowania i symulacji SSN, uczonych w oparciu o opisane w rozdz. 2. klasyczne algorytmy uczenia Sztucznych Sieci Neuronowych: wstecznej propagacji błędów, Resilient-propagation, czy Levenberga- Marquarda'ta. Ten element środowiska został wykorzystany w celu porównania efektywności numerycznej SSN, uczonych za pomocą algorytmu Kalmana, z Sieciami Neuronowymi, uczonymi w oparciu o szeroko stosowane algorytmy.

7.2. Opis cech środowiska obliczeniowego MATLAB, wykorzystanych przy projektowaniu systemu obliczeń neuronowych, połączonego z zastosowaniem FK, jako algorytmu uczącego.

MATLAB, jako środowisko obliczeniowe, pozwala na wykonywanie szerokiej gamy operacji na wektorach i macierzach, na które składa się między innymi, por. [10,71,72]:

- szereg metod konstrukcji macierzy, wraz z funkcjami wspomagającymi ich konstrukcję,
- możliwość traktowania macierzy, jako wektora kolumnowego,

- operacje wykonywane na macierzach, jako podstawowych jednostkach danych (np. mnożenie macierzy, odwracanie macierzy) oraz szereg możliwości wykonywania operacji arytmetycznych na każdym elemencie, lub parze elementów macierzy (tzw. operacje tablicowe, w tym np. mnożenie tablicowe macierzy),
- szeroka biblioteka funkcji przetwarzających macierze oraz funkcji tablicowych, umożliwiających przetwarzanie poszczególnych elementów macierzy (np. potęgowanie tablicowe, tablicowe operacje trygonometryczne, wykładnicze, czy potęgowe),
- szereg funkcji logicznych, badających własności macierzy.

Taka konstrukcja środowiska obliczeniowego pozwala na:

- zarządzanie zbiorem danych wejściowych do SSN, poprzez zapis kolejno podawanych elementów zbioru uczącego i testującego w postaci wektorowej (lub macierzowej) i operowanie na zbiorze danych, o ustalonym w ten sposób porządku, jako na odrębnej jednostce danych;
- obliczanie potencjału neuronów w postaci iloczynu skalarnego wektorów, oraz sygnału wychodzącego z neuronów poszczególnych warstw z wykorzystaniem funkcji aktywacji poszczególnych neuronów jako funkcji tablicowych;
- szybkie wykonywanie operacji koniecznych do zrealizowania algorytmu Kalmana, takich jak mnożenie czy odwracanie macierzy;
- możliwość zapisu wielkości wchodzących w skład algorytmu DEKF, jako podmacierzy i operowania nimi w postaci wektora kolumnowego;
- możliwość operowania macierzą blokową białych szumów oraz macierzą blokową przybliżonych błędów filtracji w modelach teoretycznych;
- zapis algorytmu uczenia, niezależnie od wymiarów macierzy na których dokonywane są operacje, pozwalający na łatwą zmianę architektury SSN, wymiarów wektora wejścia czy zapis algorytmu uczącego, niezależnie od przyjętej ilości połączeń dochodzących do neuronu.

Pełny algorytm uczenia i testowania SSN został zapisany w szeregu M-plików funkcyjnych, o listach parametrów wejściowych będących wektorami lub macierzami. Wykorzystano możliwość umieszczania w treści M-funkcji dalszych funkcji własnych.

Wykorzystano także fakt, iż nie ma konieczności deklarowania wymiarów macierzy i wektorów, na których operują M-pliki funkcyjne systemu MATLAB.

7.3 Opis wykonanego oprogramowania.

Ze względu na zaprogramowaną architekturę Sieci Neuronowej, w obrębie SSN warstwowych, zaimplementowano następujące systemy obliczeń:

- SSN jednokierunkowe, jedno i dwuwarstwowe typu FLNN,
- SSN rekurencyjne, typu RLNN,
- w obrębie powyższych: SSN ze sprzężeniem zwrotnym wyjścia z wejściem.

Zaimplementowano SSN o sigmoidalnej, bipolarnej funkcji aktywacji neuronów, postaci (2.4) w warstwach ukrytych oraz sigmoidalną, lub liniową funkcję (2.5), dla neuronów wyjściowych.

Parametry białego szumu, podczas prezentacji k -tego wzorca uczącego, w trakcie s -tej epoki uczenia, w modelach teoretycznych sieci neuronowej przyjęto w postaci (4.25) i (4.26).

Na wejściu do systemu obliczeń neuronowych znajdowały się następujące wielkości:

- wektora danych uczących, na który składały się uporządkowane w kolejności podawania SSN elementy zbioru uczącego (2.15),
- wektora danych do testowania efektywności sieci nauczonej, na który składały się uporządkowane w kolejności podawania SSN elementy zbioru testującego (2.16),
- licznosc zbiorów uczącego L i testującego T ,
- opcjonalnie, wektor wag startowych wszystkich połączeń neuronalnych.

Na wyjściu z powyższych systemów obliczeń, otrzymywano następujące wielkości:

- wylosowane wagi startowe wszystkich połączeń neuronalnych,
- wartości wag SSN, po każdej epoce uczenia sieci,
- wartości błędów uczenia i testowania sieci, po każdej epoce uczenia,
- wartości wyjścia z SSN, dla poszczególnych elementów zbioru uczącego i testującego,
- opcjonalnie, macierz blokową, składającą się z macierzy kowariancji przybliżonych błędów estymacji w poszczególnych modelach teoretycznych.

Zestaw parametrów sterujących procesem uczenia SSN, za pomocą algorytmu Kalmana obejmował:

- liczbę epok uczenia sieci S ,
- parametry a funkcji aktywacji neuronów (2.4), (2.5),
- parametry $\alpha, \beta, \alpha', \beta'$, wyznaczające szybkość zmian białego szumu w modelach teoretycznych kolejnych warstw SSN (4.25), (4.26),
- liczbę losowań różnych wag początkowych sieci, która wyznaczała liczbę niezależnych przebiegów algorytmu uczącego.

Ze względu na wykorzystane wersje algorytmu uczenia, oprogramowano:

- nieliniowy algorytm Filtra Kalmana, rozprzężony według poszczególnych neuronów SSN – NDEKF, por. rozdz. 4. (4.13)-(4.20),
- liniowy algorytm Filtra Kalmana, rozprzężony według poszczególnych neuronów SSN – NDLKF, por. rozdz. 4. (4.21)-(4.24),
- nieliniowy algorytm Filtra Kalmana, rozprzężony według poszczególnych neuronów SSN, uwzględniający człon rekurencyjny, por. rozdz. 4. (4.1)-(4.12).

W ramach modelu teoretycznego, zawierającego równanie obserwacyjne oraz pomiarowe, zaimplementowano następujące wersje:

- liniowe równanie procesu zmiany wag SSN
- równanie obserwacyjne liniowe lub nieliniowe
- zmienna obserwowalna w modelu teoretycznym: wyjście z SSN, lub pożądany potencjał neuronu.

W zakresie opracowania nowych algorytmów uczenia SSN, bazujących na algorytmie filtra Kalmana, powstało oprogramowanie do testowania efektywności:

1. Algorytmu pruningu SSN, będącego modyfikacją algorytmu lprune, stosowanego dla SSN, uczonych za pomocą algorytmu Resilien Propagation, dostosowaną do metody uczenia opartej o algorytmu Filtra Kalmana, por. rozdz. 7,
2. Algorytmu uczenia SSN, za pomocą algorytmu Kalmana, w oparciu o analizę wartości macierzy kowariancji przybliżonych błędów filtracji, por. rozdz. 7.

W wyniku przeprowadzonych testów, w zakresie projektowania SSN, uczonych z wykorzystaniem powyższych modyfikacji algorytmu Kalmana, wykonano następujące grupy programów:

– system doboru architektury SSN w oparciu o algorytm pruningu sieci, sterowany parametrami:

1. S – liczbę epok uczenia sieci wstępnej,
2. S_{start} – liczbę epok, po których następuje włączenie procedury pruningu do procesu uczenia,
3. S_{reset} – jako parametr wyznaczający poziom szumu, od jakiego rozpoczyna się uczenia sieci zredukowanej,
4. k – jako parametr, wyznaczający częstość kolejnych redukcji połączeń wewnątrz sieci neuronowej,
5. Λ_{prog} – jako wartość progową dla statystyki Λ , por. rozdz. 7.

– system uczenia, za pomocą algorytmu Kalmana, w oparciu o analizę wartości macierzy kowariancji przybliżonych błędów filtracji, sterowany parametrami:

1. S_0 – liczbę epok uczenia sieci wstępnej
2. α – współczynnik procentowy, wyznaczający liczbę neuronów podlegających uczeniu na kolejne S_{krok}
3. kryterium stopu algorytmu, w postaci określonej S ilości epok uczenia sieci, por. rozdz. 7.

8. Zakończenie

W rozprawie przedstawiono rozwiązania wybranych problemów z dziedziny mechaniki konstrukcji i mechaniki materiałów. Do ich analizy wykorzystano Sztuczne Sieci Neuronowe (SSN), uczone w oparciu o algorytmy Filtra Kalmana. Rozważane zagadnienia, zostały uznane za trudne do rozwiązania zarówno metodami tradycyjnymi mechaniki, jak i metodami Sztucznych Sieci Neuronowych, uczonych klasycznymi algorytmami (w przypadku istniejących rozwiązań tego typu).

Przedstawione schematy postępowania stanowią propozycję zastosowania do ich rozwiązania narzędzi alternatywnych, w stosunku do dotychczas rozważanych, celem poprawiania efektywności otrzymywanych wyników i symulacji.

W pracy przedstawiono szczegółowo rozwiązania następujących zagadnień:

- symulacja Przyśpieszeniowych (Akceleracyjnych Spektrow Odpowiedzi, ASO*) od wstrząsów górniczych, w zależności od energii i odległości epicentralnej rozpatrywanego wstrząsu,
- predykcja Przemieszczeniowych Spektrow Odpowiedzi (PSO) fundamentu budynku, w zależności od PSO gruntu przed budynkiem.
- symulacja i predykcja pętli histerezy od obciążeń niskocyklicznych nadprzewodnika kriogenicznego, stali nierdzewnej oraz różnych próbek betonowych, poddanych obciążeniu cyklicznemu typu:

1. sciskania
2. rozciągania-ściskania.

* W akronimie ASO posłużono się obcojęzycznym słowem "Akceleracyjne" SO aby odróżnić je od PSO (Przyśpieszeniowych SO).

W zakresie efektywności numerycznej proponowanych rozwiązań przedstawiono porównania z:

- tradycyjnymi metodami mechaniki (metodami empirycznymi, modelowania metodami teorii sprężystości i plastyczności, modelowania metodami mechaniki zniszczenia),
- stosowanymi dotąd metodami "miękkimi", opartymi o SSN i klasyczne algorytmy ich uczenia.

8.1. Rezultaty pracy

1. Wykazano numeryczną efektywność metody uczenia SSN, opartej o filtry Kalmana, do aproksymacji doświadczalnych danych rozważanych problemów mechaniki.
2. Dla każdego z przedstawionych zagadnień zaprezentowano dobór metodologii rozwiązywania problemu za pomocą SSN. Przedstawiono dobór wektora wejścia do SSN, wybór architektury SSN, sposobu przepływu sygnału wewnątrz sieci.
3. Efektem przedstawionych badań jest powstanie kompleksowego systemu obliczeń neuronowych, pozwalającego na uczenie oraz projektowanie jednokierunkowych i rekurencyjnych SSN za pomocą FK.

8.2. Wykazane tezy oraz rezultaty w zakresie mechaniki

W wyniku przeprowadzonych analiz i testów można postawić następujące tezy:

1. Filtry Kalmana, wykorzystane jako metoda uczenia SSN, pozwalają efektywnie rozwiązywać zadania uczenia i testowania Sztucznych Sieci Neuronowych.
2. Zastosowanie filtrów Kalmana, jako metody ugruntowanej na matematycznych podstawach teorii dyskretnych procesów stochastycznych, pozwala w istotny sposób zwiększyć dokładność uczenia i predykcji Sztucznych Sieci Neuronowych w porównaniu z klasycznymi metodami heurystycznymi.
3. Wykonane programy pozwalają na efektywną analizę, z dużą dokładnością, wybranych zagadnień mechaniki, dotyczących analizy wymuszeń parasejsmicznych na budynki prefabrykowane o średniej wysokości oraz deformacji materiałów przy obciążeniach cyklicznych.

W zakresie mechaniki konstrukcji i materiałów rezultatami przeprowadzonych badań, analiz i symulacji są:

- system obliczeń neuronowych służący do symulacji Akceleracyjnych (Przyśpieszeniowych) ASO od wstrząsów górniczych, w zależności od energii i odległości epicentralnej rozpatrywanego wstrząsu, na terenie GZW (Górnośląskiego Zagłębia Węglowego),
- system obliczeń neuronowych służący predykcji PSO Przemieszczeniowych Spektrow Odpowiedzi PSO fundamentu budynku, w zależności od PSO gruntu przed budynkiem, na terenie LGOM (Legnicko-Głogowskiego Okręgu Miedziowego)

- systemy neuronowe symulacji i predykcji pętli histerezy od obciążeń niskocyklicznych.

8.3. Wnioski ogólne w zakresie metodologii stosowania SSN, uczonych za pomocą FK, do rozwiązywania rozpatrywanych problemów mechaniki

Rozpatrywane problemy różniły się zarówno tematyką w zakresie mechaniki, jak też specyfiką dostępnych do uczenia i testowania SSN danych. Rozważano dane doświadczalne, pochodzące z prób laboratoryjnych, wykonanych dla różnych materiałów (beton, stal, kompozyt) oraz wielkości wyznaczone na podstawie pomiarów terenowych (Spektra Odpowiedzi budynków). Rozpatrywano zbiory danych o licznosciach zarówno dużych (ok. 22 tys. wzorców), jak i niewielkich (ok. 250 wzorców). Zbiory uczące i testujące miały charakter zarówno interpolacyjny (symulacja), jak też ekstrapolacyjny (predykcja neuronowa).

Przeprowadzone badania pozwalają na postawienie następujących wniosków, dotyczących zastosowania SSN, uczonych z wykorzystaniem FK:

- dobór rodzaju SSN w zakresie typu jednokierunkowego bądź rekurencyjnego pomiędzy wyjściem sieci, a wejściem – powinno być rozpatrywane dla każdego problemu indywidualnie
- zastosowanie rekurencyjnych połączeń typu autoregresyjnego dla niektórych problemów może znacznie podnieść efektywność obliczeniową SSN
- niektóre z rozpatrywanych problemów wykazują silną wrażliwość wybór na wektora wejścia do SSN
- wrażliwość SSN, w zakresie otrzymywanych błędów uczenia i testowania, w zależności od wag początkowych sieci jest niska (wynika to ze stosowania algorytmu Kalmana i stosowanych tam białych szumów)
- efektywność numeryczną SSN można zwiększyć, stosując zaproponowany algorytm zmniejszania liczby połączeń międzyneuralnych SSN (pruningu).
- osiągnięte błędy uczenia i testowania, można obniżyć stosując zaproponowaną metodę uczenia SSN na podstawie zróżnicowania wartości macierzy kowariancji przybliżonych błędów filtracji w poszczególnych modelach teoretycznych

8.4. Wnioski szczegółowe

W dziedzinie efektywności rozwiązywania problemów z dziedziny mechaniki konstrukcji i materiałów, przeprowadzone testy pozwalają na przedstawienie następujących konkluzji:

- Zastosowanie FK, jako metody uczenia SSN, połączonej z odpowiednim doborem informacji wejściowej do sieci pozwoliło na poprawienie dokładności predykcji rozpatrywanych Przyśpieszeniowych i Przemieszczeniowych Spektrow Odpowiedzi, w porównaniu do dotychczas stosowanych metod empirycznych oraz neuronowych, opartych o inne niż filtrowanie Kalmana algorytmy uczące, wraz z dobranymi do nich wektorami wejścia, por. 5.1.
- Zastosowanie SSN, uczonych z wykorzystaniem filtrowania Kalmana pozwoliło modelować zachowanie się pod wpływem cyklicznie zmiennej siły, tak złożonego materiału, jak opisany w 5.2.1. kompozyt hierarchiczny, zarówno z zakresie predykcji ostatniej jak i początkowej fazy doświadczenia oraz zachowania się materiału w losowo wybranych punktach historii obciążenia, por. 5.2.1.
- Zastosowanie SSN, uczonych z wykorzystaniem filtrowania Kalmana, pozwoliło na zbudowanie niejawnego modelu stali AISI 316L, poddanej obciążeniu cyklicznemu, por. 5.2.2.
- FK, jako metoda uczenia SSN, okazała się efektywna w predykcji pętli histerezy dla rozpatrywanych obciążeń próbek betonowych, por. 5.2.3.
- Konstrukcja niejawnego, neuronowego modelu zachowania się materiału, opartego o SSN, nauczone przy pomocy FK, może stanowić konkurencyjną (prostszą i szybszą) metodę symulacji i predykcji zachodzących w materiale zjawisk, w porównaniu do modelu sprężysto-plastycznego, połączonego z koniecznością identyfikacji nieraz wielu parametrów, por. 5.2.1.
- Model zjawiska fizycznego, dany w postaci nauczonej i przetestowanej pozytywnie SSN, może stanowić alternatywę, wobec modelowania za pomocą np. modeli reologicznych. W modelach tych istnieje konieczność wyboru modelu teoretycznego i identyfikacji jego parametrów. Modelowanie za pomocą SSN jest oparte wyłącznie o dane doświadczalne, bez konieczności konstruowania modelu matematycznego zjawiska, por. 5.2.1.
- SSN, nauczona za pomocą FK, może również stanowić atrakcyjne narzędzie, w stosunku do metod modelowania w oparciu o klasyczną mechanikę zniszczenia, gdzie nieodpowiedni dobór pewnych parametrów, lub przyjęcie zbyt mało złożonego modelu, nie odzwierciedla dobrze zachodzących zjawisk, por. 5.2.2.

W zakresie metodologii rozwiązywania problemów mechaniki konstrukcji i materiałów, za pomocą SSN, uczonych z wykorzystaniem FK, można przedstawić następujące wnioski:

- do analizy wielu problemów opartych o analizę danych doświadczalnych, stosując SSN uczone w oparciu o FK – wystarczające są proste SSN jedno lub dwu warstwowe, o niewielkiej liczbie neuronów w poszczególnych warstwach, por. 5.1., 5.2.1., 5.2.2., 5.2.3.,
- do efektywnego modelowania rozpatrywanych zjawisk nie są konieczne SSN o rozbudowanych wektorach wejścia, por. 5.1.,
- celem efektywnego rozwiązania większości ww. problemów można przyjąć poziom białego szumu w modelach teoretycznych w postaci określonej w 4.2. niniejszej pracy,
- efektywność uczenia i testowania SSN, uczonych za pomocą algorytmu Kalmana, zależy od wprowadzonej informacji wejściowej do sieci, przy czym do rozwiązania tego samego problemu można zaproponować kilka efektywnych wektorów wejścia, 5.2.2.
- struktura SSN, w obrębie sieci jednokierunkowej lub rekurencyjnej powinna być dobierana zależnie od charakteru rozpatrywanego problemu, por. 5.1., 5.2.1.,
- wprowadzenie autoregresyjnego sprzężenia zwrotnego (wyjścia, z wejściem SSN) może podnieść efektywność obliczeń dla niektórych problemów, por. 5.1., 5.2.1.,
- dla niewielkich zbiorów danych uczących lub krótkich przebiegów procesów uczenia SSN (niewielka ilość epok), występuje zależność otrzymywanych błędów uczenia i testowania, od początkowych wartości wag SSN. Dla dużych zbiorów danych lub dłuższych procesów uczenia, rozrzut wartości błędów dla różnych zestawów wag początkowych sieci okazał się zaniedbywalny wobec precyzji rozwiązania wymaganej zastosowaniami,
- wstępny dobór architektury SSN, uczonej za pomocą FK, można przeprowadzić na podstawie analizy krzywych błędów uczenia i testowania podczas początkowej fazy przebiegu algorytmu pruningu, opisanego w 6.
- czas obliczeń neuronowych, opartych o omówione w pracy metody uczenia, bez straty jakości otrzymywanych wyników, można skrócić dzięki zastosowaniu uczenia, opartego na zróżnicowaniu wartości macierzy kowariancji w modelach teoretycznych, por. 6., lub (dla niektórych problemów) stosując SSN rekurencyjne, zamiast jednokierunkowych, por. 5.1.,
- jeśli nastąpiła stabilizacja błędów uczenia, dla SSN o określonej architekturze, uczonej z wykorzystaniem FK, można uzyskać dalszą poprawę rezultatów, poprzez zastosowanie postępowania opartego o analizę wartości przybliżonych

macierzy kowariancji błędów estymacji dla poszczególnych neuronów sieci, por. 6., czyli wybór do uczenia tych obszarów SSN, dla których proces filtracji został przeprowadzony z mniejszą dokładnością.

8.5. Elementy oryginalne rozprawy

SSN, uczone z wykorzystaniem filtrów Kalmana, nie były wcześniej stosowane do rozwiązania omawianych zagadnień.

W zakresie rozwoju przedstawione metody uczenia SSN, zaprezentowano autorskie rozwiązania w zakresie:

- projektowania SSN na podstawie zmodyfikowanego algorytmu zmniejszania liczby parametrów SSN (pruningu),
- uczenia SSN, opartego na analizie wartości macierzy przybliżonych błędów filtracji w modelach teoretycznych,
- dobór sposobu transmisji informacji wewnątrz SSN, w zależności od charakteru rozpatrywanego problemu.

Rezultaty obliczeń za pomocą SSN, uczonych w filtrem Kalmana, zostały otrzymane na podstawie autorskiego oprogramowania, zaimplementowanego w systemie do obliczeń naukowo – inżynierskich MATLAB.

8.6. Kierunki dalszych prac.

Perspektywy rozwijania problematyki, zawartej w niniejszej pracy, mogłyby dotyczyć:

- wobec potwierdzenia skuteczności omawianej metody, do analizy przedstawionych zagadnień - podjęcie prób rozwiązania innych, niż wymienione w pracy, problemów z dziedziny mechaniki materiałów i konstrukcji, w oparciu o metodologię uzyskaną na podstawie przeprowadzonych analiz i testów,
- kontynuacji prac, nad wybranymi z przedstawionych tematów, w szczególności – dalszych prac, dotyczących modelowania zjawisk w zakresie obciążenia cyklicznego betonu,
- analizy przydatności do rozwiązywania problemów mechaniki, innych niż wymienione w pracy, modeli teoretycznych w ramach filtrowania Kalmana, w szczególności modeli z nieliniowym równaniem procesu zmiany wag SSN,
- budowy spójnego symulatora SSN, opartego na wykonanym oprogramowaniu, służącego do projektowania, uczenia i testowania SSN. Taki moduł dodatkowy, mógłby być dostępny użytkownikowi w ramach neuronowego pakietu symulacyjnego środowiska MATLAB,

- od strony matematycznej interesująca wydaje się być analiza takich SSN, jako układów dynamicznych.

9. Literatura

1. J Żurada, M Barski, W Jędruch, Sztuczne sieci neuronowe, Wyd. Nauk. PWN, Warszawa, 1996
2. S Osowski, Sieci neuronowe do przetwarzania informacji, Ofic. Wyd. Polit. Warsz., Warszawa, 2000
3. S Haykin, Neural Networks - A Comprehensive Foundation, 2nd Ed., MacMillan College Publ., Englewood Cliffs, NJ, 1999
4. R Rojas, Neural Networks, A systematic Introduction, Springer-Verlag, New York, 1996
5. R Golden, Mathematical methods for neural network analysis and design, MIT Press, 1996
6. K Maurin, Analiza, PWN, Warszawa, 1977
7. A Plucińska, E Pluciński, Rachunek prawdopodobieństwa, statystyka matematyczna, procesy stochastyczne, Wyd. Nauk-Techn., Warszawa, 2000
8. VN Vapnik, Principles of risk minimalization for learning theory, Advances in Neural Information Processing Systems 4:831-838, (JE Moody, SJ Hanson, RP Lippman, Eds.), San Mateo, CA, Morgan Kaufman, 1993
9. Korbicz, A Obuchowicz, D Uciński, Artificial Neural Networks, Akad. Ofic. Wyd. PLJ, Warszawa, 1994
10. MATLAB, The language of technical computing, ver. 2., User's Guide, The Matlab Works Inc., 2001
11. Neural Network Toolbox for Use with MATLAB, User's Guide Version 3, The Math Works Inc., Natick MA, 1998
12. WA Fuller, Introduction to Statistical Time Series, Wiley, 1976
13. S Haykin (Ed.), Kalman Filtering and Neural Networks, John Wiley & Sons, New York, 2001
14. R Kalman, A new approach to linear filtering and prediction problems, Transactions of the ASME-Journal of Basic Engineering 82:(D)35-45, ASME, 1960
15. K Lou, R Perez, A new system identification technique using Kalman filters and multilayer neural networks, Artificial Intelligence in Engineering, 10:1-8, Elsevier, 1996

16. M Cernansky, L and Benuskova, Simple recurrent network trained by RTRL and extended Kalman filter algorithms, *Neural Network World* 13(3): 223-234, 2003
17. W Angand, C Riviere, Neural network methods for error canceling in human-machine manipulation, *Proc. 23rd Annual Intl. Conf. IEEE Engineering in Medicine and Biology Society, Canadian Journal of Civil Engineering*: 3462-3465, 2001
18. P Gil, A Dourado, JO Henriques, P Carvalho, Constrained Adaptive Nonlinear Neural Model Based Predictive Control of a Distributed Solar Collector Field, *Proceedings of the Second IHP Workshop of PSA Users, Almeria, Spain, 2002*
19. J Choi, TH Yeap, M Bouchard, Online State-Space Modeling using Recurrent Multilayer Perceptrons with Unscented Kalman Filter, *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC 2004)*: 3427-3432, The Hague, 2004
20. F Lange, Fast and Accurate Training of Multilayer Perceptrons Using an Extended Kalman Filter (EKFNet), *Internal paper of Institute of Robotics and Technetronic, German, Aerospace Research Establishment, Germany, 1995*
21. Y Huang, Application of Kalman Learning Algorithm Multilayer Neural Network to Estimates of Ore Grades, *Application of Artificial Neural Networks to Rock Engineering Problems, Doctoral Thesis 1997:16, Luleå University of Technology, Sweden , ISSN 1402-1544 / ISRN LTU-DT-97/16-SE / NR 1997, 1997*
22. MR Sahebi, F Bonn, and GB Bénié, Neural networks for the inversion of soil surface parameters from synthetic aperture radar satellite data, *Can. J. Civ. Eng.* 31(1): 95-108, 2004
23. DJ Lary, HY Mussa, Using an extended Kalman filter learning algorithm for feed-forward neural networks to describe tracer correlations, *Atmos. Chem. Phys.* 4(3): 3653-3667, 2004
24. A Forsgren, R Kling, An implementation of recurrent neural networks for prediction and control of nonlinear dynamic systems , *MSc Programmes in Engineering Computer Science and Engineering 2003:119, Luleå University of Technology, Sweden, ISSN 1402-1617 / ISRN LTU-EX-03/119-SE / NR 2003:119, 2003*
25. L Shuhui, DC Wunsch, E O'Hair, MG Giesselmann, Wind turbine power estimation by neural networks with Kalman filter training on a SIM parallel machine, *Neural Networks, IJCNN '99 International Joint Conference on Neural Networks* 5: 3430-3434, 1999

26. P Gil, J Henriques, H Duarte-Ramos, A Dourado, State-Space Neural Networks and the Unscented Kalman Filter in On-line Nonlinear System Identification, IASTED Int. Conf. on Intelligent Systems and Control (ISC'2001), Tampa, USA, 2001
27. T Sato, M Sato, Structural Identification Using Neural Network and Kalman Filter, JSCE 14: 23-32, 1997
28. J Silva, A Neto, Preliminary Testing and Analysis of an Adaptive Neural Network Training Kalman Filtering Algorithm , Proceedings of the IV Brazilian Conference on Neural Networks - IV Congresso Brasileiro de Redes Neurais: 247-251, São José dos Campos Brazil, 1999
29. D Simon, Training radial basis neural networks with the extended Kalman filter, Neurocomputing 48(1-4): 455-475, 2002
30. B Ciocoiu, RBF networks training using a dual extended Kalman filter, Neurocomputing 48(1-4): 609-622, 2002
31. W Liu, LL Yang, L Hanzo, Recurrent Neural Network Based Narrowband Channel Prediction (Speech). Proc. of IEEE VTC'06 Spring, Melbourne, Australia, 2006
32. J Choi, AC. de Lima, S Haykin, Kaman Filter-Trained Recurrent Neural Equalizers for Time-Varying Channels, IEEE Transactions on Communications 53(3): 472-480, 2005
33. C Shao and M Bouchard, Efficient Classification for Speech/Audio Applications using Neural Networks, Proceedings of IEEE-EURASIP Seventh International Symposium on Signal Processing and its Applications (ISSPA) 2003, 1: 357-360, Paris, France, 2003
34. M Bouchard, New recursive-least-squares algorithms for active control of sound and vibration using neural networks, IEEE Trans. on Neural Networks 12(1): 135-147, 2001
35. J Perez-Ortiz, FA Gers, D Eck, J Schmidhuber, Kalman filters improve LSTM network performance in problems unsolvable by traditional recurrent nets, Neural Networks 16(2), 2003
36. JA Perez-Ortiz, J Schmidhuber, F Gers, D Eck, Improving Long-Term Online Prediction with Decoupled Extended Kalman Filters. In J. Dorrnsoro, ed.: 1055-1060, Proceedings of Int. Conf. on Artificial Neural Networks ICANN'02 Madrid, Springer, Berlin, 2002
37. MC Nechyba and Y Xu, Cascade Neural Networks with Node-Decoupled Extended Kalman Filtering, Proc. IEEE Int. Symp. on Computational Intelligence in Robotics and Automation 1: 214-9, 1997

38. I Rivals, L Personnaz, A recursive algorithm based on the extended Kalman filter for the training of feedforward neural models, *Neurocomputing* 20(1-3): 279-294, 1998
39. L Feldkamp, D Prokhorov, C Eagen, and F Yuan, Enhanced Multi-Stream Kalman Filter Training for Recurrent Networks. In J. Suykens and J. Vandewalle (Eds.) *Nonlinear Modeling: Advanced Black-Box Techniques*,: 29-53, Kluwer Academic Publishers, 1998
40. E Maciąg, *Wstrząsy górnicze i ich wpływ na budynki*, Mat. Sympozjum Wpływy sejsmiczne i parasejsmiczne na budowle, Kraków, 1994
41. E Maciąg, Interakcja układu budynek podłoże podlegającego działaniom sejsmicznym i parasejsmicznym (praca przeglądowa), *Mech. Teoret. i Stos.* 17(4): 497-536, 1979,
42. Eurocode 8, Design provisions for earthquake resistance of structures. European Prestandard, European Committee for Standardization, Brussels, 1994
43. Wymagania techniczne dla obiektów budowlanych wznoszonych na szkodach górniczych, ITB, Oddział w Gliwicach, 2000
44. K Kuźniar, Analiza drgań budynków ścianowych o średniej wysokości podlegających wstrząsom górniczym z wykorzystaniem sieci neuronowych, monografia 310, wyd. Politechniki Krakowskiej, Kraków 2004
45. K Kuźniar, E Maciąg, Wyznaczanie spektrów odpowiedzi od wstrząsów górniczych z wykorzystaniem sieci neuronowych, *Zeszyty Nauk. Polit. Rzeszowskiej, Mechanika* 17(60): 367-370, Rzeszów 2002
46. J Ghaboussi, ChJ Lin, New method of generating spectrum compatible accelerograms using neural networks, *Earth. Eng. Struct. Dyn.* 27: 377-396, 1998
47. K Cheng, N Popplewell, Neural network for earthquake selection in structural time history analysis, *Earth. Eng. Stru. Dyn.* 23: 303-396, 1994
48. A Zell, Ed. *SNNS- Stuttgart Neural Network Simulator, User's Manual, Ver.4.2*, Stuttgart, 1998
49. K Kuźniar, E Maciąg, Z Waszczyszyn, Computation of response spectra form mining tremors using neural networks, *Soil Dyn. and Earth. Eng.* 25: 331-339, 2005
50. K Kuźniar, E Maciąg, Z Waszczyszyn, Zastosowanie SSN do analizy przekazywania się na budynki drgań wzbudzanych wstrząsami górniczymi, *XLIX Konf. Nauk. KILiW PAN i KN PZITB 'Krynica2003' (II)*: 73-80, 2003

51. K Kuźniar, E Maciąg, Z Waszczyszyn, Neuronowa analiza interakcji budynek - podłoże w przypadku wstrząsów górniczych, II Problemowa Konf. Geotechniki: Współpraca konstrukcji z podłożem gruntowym, Białystok-Białowieża, 2004
52. R Hongg, E Tanis, Probability and Statistical Interference, Prentice Hall, New Jersey, 2001
53. JS Jang, CHT Sun, E Mizutani, Neuro-fuzzy and soft computing. A computational approach to learning and machine intelligence, Prentice Hall Int. Inc. Upper Saddle River, New Jork, 1997
54. Fuzzy Logic Toolbox for use with Matlab, User's Guide, Ver.2. The matlab Works Inc., 2001
55. www.iter.org, oficjalna witryna internetowa projektu ITER
56. M Lefik, Zastosowania sztucznych sieci neuronowych w mechanice i w inżynierii, Zeszyty Nauk. Polit. Łódzkiej, 2005
57. A Nijhuis, NHW Noordman, HHJ Kate, Effects of Transverse Cycling Loading on the Mechanical and Electromagnetic Behavior of two ITER CSMC Conductors in Cryogenic Press, Final Report No.UT-NET 99-1(1), University of Twente, Netherlands, 1999
58. U Galvanetto, V Naumov, V Palmov, BA Schlefler, Analysis of mechanical behavior of cable-in-conduit superconductors under transverse cycling loading, Int. J. Comput. Civ. Struct. Eng.1(2): 1-10, 2000
59. M Lefik, Schlefler B.A., Artificial neural networks for parameter identifications for elasto-plastic model of superconducting cable under cycling loading, ComputersStructures, 80(2002): 1699-1713, 2002
60. J Lemaitre, A course on damage mechanics, Springer-Verlag, 1992
61. A Ganczarski, L Barwacz, Low cycle fatigue based on unilateral damage evolution, Int. J. Damage Mech, (in press), 2006,
62. Adobe Photoshop 6.0, Owner's Manual, Adobe Systems Inc., 2004
63. A Ylinen, A method of determining the buckling stress and required cross-sectional area for centrally loaded straight columns in elastic and inelastic range, Association Int. Des Ponts et Charpents 16: 529-550, Zurich, 1956
64. Sz Woliński, Właściwości betonu rozciąganego i ich zastosowania w mechanice pęknięcia betonu, Zesz. Nauk. Politechniki Rzeszowskiej, Rzeszów, 1991
65. HW Reinhardt, Fracture mechanics of an elastic softening material like concrete, Heron Public. 29(2), 1994

66. BP Sinha, KH Gerstle, LG Tulin, Stress-strain relations for concrete under cyclic loading, Journ. of American Concrete Inst. 61-12, 1964
67. Sz Wolinski, Influence of aggregate size and loading type on the tensile softening behavior of concrete, Delft Univ. of Tech Rapport No. 5-86-16, 1986
68. Graph Digitizer Version 1.9, For digitization of scientific charts,
69. L Prechelt, Connection Pruning with Static and Adaptive Pruning Schedules. Neurocomputing 16(1): 49-61, 1997
70. Statistic Toolbox for Use with MATLAB, User's Guide Version 3, The Math Works Inc., Natick MA, 1998
71. J Brzózka, L Dorobczyński, Programowanie w Matlab, ZNI Mikom, 1998
72. A Zalewski, R Cegiela, Matlab - obliczenia numeryczne i ich zastosowania, Wyd. Nakom, Poznań, 2000
73. W Mlak, Wstęp do teorii przestrzeni Hilberta, PWN, Warszawa, 1982
74. JP Drecourt, Kalman Filtering in hydrological modelling, DHI Water Environment, Danish Hydraulic Institute, Raport No.2003-1, Horsholm, Denmark, 2003
75. T Chmielwski, Z Zembaty, Podstawy dynamiki budowli, Wyd. Arkady, Warszawa, 1998

10. Dodatki

10.1. Wyprowadzenie równań liniowego FK

Definiujemy zadanie filtracji dyskretnego procesu stochastycznego \mathbf{x} , jako wyznaczenie estymatora $\hat{\mathbf{x}}_k$ dla \mathbf{x}_k , minimalizującego błąd średniokwadratowy estymacji w postaci:

$$\mathbf{E}(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T \quad (10.1)$$

gdzie dany jest zbiór obserwacji procesu \mathbf{y} : $Y(k) = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k\}$, por. [13].

Rozwiązanie tego problemu równoważne jest znalezieniu:

$$\hat{\mathbf{x}}_k = \operatorname{argmin} \mathbf{E}[(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T | Y(k)] \quad (10.2)$$

gdzie $\mathbf{E}(\cdot | Y(k))$ oznacza wartość oczekiwaną warunkową przy założeniu znajomości wielkości $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k$, gdzie *argmin* oznacza element realizujący minimum, por. [52].

Rozwiązanie tak postawionego problemu, w postaci danej równaniami rekurencyjnymi, dla modelu nie uwzględniającego szumu ν_k w równaniu obserwacyjnym, zaproponowane zostało przez R.E. Kalmana w pracy [14]. W [13] przedstawiono schemat dowodu, w przypadku występowania czynnika ν_k w równaniu obserwacyjnym. Poniżej zostanie on sformalizowany w oparciu o właściwości projekcji ortogonalnych w przestrzeni zmiennych losowych z iloczynem skalarnym, wyznaczonym przez wartość oczekiwaną.

By znaleźć rozwiązanie problemu dla równań w postaci (3.1)-(3.2) wykażemy najpierw następujący lemat:

Lemat:

Wielkością minimalizującą wyrażenie $\mathbf{E}[(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T | Y(k)]$ względem zmiennej $\hat{\mathbf{x}}_k$ jest wartość oczekiwana warunkowa zmiennej losowej \mathbf{x}_k względem $Y(k)$.

dowód:

$$\begin{aligned} & \mathbf{E}[(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T | Y(k)] = \\ & \mathbf{E}[\hat{\mathbf{x}}_k \hat{\mathbf{x}}_k^T - \hat{\mathbf{x}}_k \mathbf{x}_k^T - \mathbf{x}_k \hat{\mathbf{x}}_k^T + \mathbf{x}_k \mathbf{x}_k^T | Y(k)] = \\ & = \mathbf{E}[\hat{\mathbf{x}}_k \hat{\mathbf{x}}_k^T | Y(k)] - \mathbf{E}[\hat{\mathbf{x}}_k \mathbf{x}_k^T | Y(k)] - \mathbf{E}[\mathbf{x}_k \hat{\mathbf{x}}_k^T | Y(k)] + \mathbf{E}[\mathbf{x}_k \mathbf{x}_k^T | Y(k)] = \end{aligned} \quad (10.3)$$

ze względu na liniowość wartości oczekiwanej warunkowej, por. [7],

$$\mathbf{E}[\hat{\mathbf{x}}_k \hat{\mathbf{x}}_k^T | Y(k)] - \hat{\mathbf{x}}_k \mathbf{E}[\mathbf{x}_k^T | Y(k)] - \mathbf{E}[\mathbf{x}_k | Y(k)] \hat{\mathbf{x}}_k^T + \mathbf{E}[\mathbf{x}_k \mathbf{x}_k^T | Y(k)] = \quad (10.4)$$

po dodaniu i odjęciu w powyższej sumie składnika $\mathbf{E}(\mathbf{x}_k^T | Y(k)) \mathbf{E}(\mathbf{x}_k | Y(k))$, otrzymujemy:

$$\mathbf{E}[\mathbf{x}_k \mathbf{x}_k^T | Y(k)] - \mathbf{E}(\mathbf{x}_k^T | Y(k)) \mathbf{E}(\mathbf{x}_k | Y(k)) + [\hat{\mathbf{x}}_k - \mathbf{E}(\mathbf{x}_k | Y(k))] [\hat{\mathbf{x}}_k - \mathbf{E}(\mathbf{x}_k | Y(k))]^T \quad (10.5)$$

które to wyrażenie należy zminimalizować względem $\hat{\mathbf{x}}_k$. Pierwsze dwa składniki są niezależne od $\hat{\mathbf{x}}_k$, więc jest to problem równoważny minimalizacji wyrażenia:

$$[\hat{\mathbf{x}}_k - \mathbf{E}(\mathbf{x}_k|Y(k))][\hat{\mathbf{x}}_k - \mathbf{E}[\mathbf{x}_k|Y(k)]^T] \quad (10.6)$$

którego zależność od $\hat{\mathbf{x}}_k$ jest kwadratowa, więc minimum jest osiągnięte gdy $[\hat{\mathbf{x}}_k - \mathbf{E}(\mathbf{x}_k|Y(k))] = 0$, czyli w punkcie $\mathbf{E}(\mathbf{x}_k|Y(k))$, co kończy *dowód lematu*.

Ponieważ \mathbf{x}_0 ma rozkład normalny oraz równania (3.1)-(3.2) są liniowe, więc zmienne losowe \mathbf{x}_k oraz \mathbf{y}_k mają również rozkłady normalne z uwagi na fakt, iż przeszczałcenia liniowe zachowują rozkład normalny [7].

Z lematu powyższego otrzymana została postać poszukiwanego estymatora, poniżej podana zostanie jego charakterystyka, pomocna przy wyznaczaniu równań *explicite*.

Niech X oznacza przestrzeń zmiennych losowych, takich że $\mathbf{E}uu^T < \infty$ dla $u \in X$, iloczynem skalarnym $\langle u, v \rangle = \mathbf{E}(uv^T)$. Wtedy $\mathbf{x}_k, \mathbf{y}_k, \omega_k, \nu_k \in X$ dla $k=1,2,\dots$. Niech

$$\mathbf{Y}(k) := \left\{ \sum_{i=1}^k a_i \mathbf{y}_i \mid a_1, a_2, \dots, a_k \in R \right\} \quad (10.7)$$

oraz

$$\mathbf{Y}^\perp(k) = \{x \in X : \forall \mathbf{y} \in \mathbf{Y}(k) \mathbf{E}(\mathbf{x}\mathbf{y}^T) = 0\} \quad (10.8)$$

będzie przestrzenią ortogonalną do $\mathbf{Y}(k)$ w X [73]. Wtedy z definicji wartości oczekiwanej warunkowej: $\mathbf{E}(\mathbf{x}_k|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k) = Proj_{\mathbf{Y}(k)}(\mathbf{x}_k)$, gdzie $Proj_{\mathbf{Y}(k)}$ oznacza projekcję ortogonalną na $\mathbf{Y}(k)$, por. [7], [73].

Niech $\hat{\mathbf{x}}_k^- = \mathbf{E}(\mathbf{x}_k|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1})$ będzie estymatorem *a priori* dla \mathbf{x}_k , znanym w chwili $k-1$, por. [13].

Ponieważ

$$\hat{\mathbf{x}}_k = \mathbf{E}(\mathbf{x}_k|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k) = Proj_{\mathbf{Y}(k)}(\mathbf{x}_k) \in \mathbf{Y}(k) \quad (10.9)$$

więc estymator $\hat{\mathbf{x}}_k$ jest liniowy względem elementów zbioru $\mathbf{Y}(k)$ i możemy zapisać:

$$\hat{\mathbf{x}}_k = \mathbf{K}_k^- \hat{\mathbf{x}}_k^- + \mathbf{K}_k \mathbf{y}_k \quad (10.10)$$

gdzie \mathbf{K}_k^- oraz \mathbf{F}_k są pewnymi macierzami o wymiarach $n \times n$ oraz $n \times p$ odpowiednio.

Skoro $\hat{\mathbf{x}}_k$, jest projekcją ortogonalną na $\mathbf{Y}(k)$ to: $\hat{\mathbf{x}}_k \in \mathbf{Y}(k)$, czyli $\mathbf{x}_k - \hat{\mathbf{x}}_k \in \mathbf{Y}(k)^\perp$, z jednoznaczności rozkładu przestrzeni X na sumę prostą

$$X = \mathbf{Y}(k) + \mathbf{Y}^\perp(k) \quad (10.11)$$

wynika

$$\mathbf{E}(\mathbf{x}_k - \hat{\mathbf{x}}_k) \mathbf{y}_i^T = 0 \quad (10.12)$$

dla $i = 1, 2, \dots, k - 1$. por. [73].

Podstawiając do (10.12) równość (10.10), otrzymujemy:

$$\mathbf{E}(\mathbf{x}_k - \mathbf{K}_k^- \widehat{\mathbf{x}}_k^- - \mathbf{K}_k \mathbf{y}_k) \mathbf{y}_i^T = 0 \quad (10.13)$$

a następnie, podstawiając (3.1):

$$\mathbf{E}(\mathbf{x}_k - \mathbf{K}_k^- \widehat{\mathbf{x}}_k^- - \mathbf{K}_k (\mathbf{H}_k \mathbf{x}_k + \nu_k)) \mathbf{y}_i^T = 0 \quad (10.14)$$

$$\mathbf{E}(\mathbf{x}_k - \mathbf{K}_k^- \widehat{\mathbf{x}}_k^- - \mathbf{K}_k (\mathbf{H}_k \mathbf{x}_k)) \mathbf{y}_i^T + \mathbf{K}_k \mathbf{E} \nu_k \mathbf{y}_i^T = 0 \quad (10.15)$$

$$\mathbf{E}((\mathbf{I} + \mathbf{K}_k \mathbf{E}_k + \mathbf{K}_k^-) \mathbf{x}_k \mathbf{y}_i^T + \mathbf{K}_k^- (\mathbf{x}_k - \widehat{\mathbf{x}}_k^-) \mathbf{y}_i^T) + \mathbf{K}_k \mathbf{E} \nu_k \mathbf{y}_i^T = 0. \quad (10.16)$$

Ponieważ ν_k i \mathbf{y}_i są niezależne dla $i=1,2,\dots,k-1$ oraz $\mathbf{E} \nu_k = 0$ więc:

$$\mathbf{E}((\mathbf{I} - \mathbf{K}_k \mathbf{E}_k - \mathbf{K}_k^-) \mathbf{x}_k \mathbf{y}_i^T + \mathbf{F}_k^- (\mathbf{x}_k - \widehat{\mathbf{x}}_k^-) \mathbf{y}_i^T) = 0. \quad (10.17)$$

$\mathbf{E}(\mathbf{x}_k - \widehat{\mathbf{x}}_k^-) \mathbf{y}_i^T = 0$ dla $i = 1, 2, \dots, k - 1$ więc powyższa równość redukuje się do:

$$(\mathbf{I} - \mathbf{K}_k \mathbf{E}_k - \mathbf{K}_k^-) \mathbf{E} \mathbf{x}_k \mathbf{y}_i^T = 0. \quad (10.18)$$

Równanie to zachodzi dla wszystkich $i = 1, 2, \dots, k - 1$, niezależnie od wartości obserwacji \mathbf{y}_k , tylko gdy $(\mathbf{I} - \mathbf{K}_k \mathbf{E}_k - \mathbf{K}_k^-) = 0$, czyli gdy: $\mathbf{K}_k^- = (\mathbf{I} - \mathbf{K}_k \mathbf{E}_k \mathbf{H}_k)$. Teraz (10.10) można zapisać jako:

$$\widehat{\mathbf{x}}_k = \widehat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \widehat{\mathbf{x}}_k^-) \quad (10.19)$$

Ponieważ z (10.10) oraz z faktu niezależności ω_k od $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1}$ mamy

$$\widehat{\mathbf{x}}_k^- = \mathbf{E}(\mathbf{F}_{k,k-1} \mathbf{x}_{k-1} + \omega_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1}) = \quad (10.20)$$

$$\mathbf{F}_{k,k-1} \mathbf{E}(\mathbf{x}_{k-1} | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1}) + \mathbf{E}(\omega_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1}) = \mathbf{F}_{k,k-1} \widehat{\mathbf{x}}_{k-1}$$

więc, by w sposób rekurencyjny wyznaczyć $\widehat{\mathbf{x}}_k$ konieczna jest znajomość macierzy \mathbf{K}_k .

Niech $\widehat{\mathbf{y}}_k = \mathbf{E}(\mathbf{y}_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1})$. Wtedy:

$$\begin{aligned} \mathbf{y}_k - \widehat{\mathbf{y}}_k &= \mathbf{y}_k - \mathbf{E}(\mathbf{H}_k \mathbf{x}_k + \nu_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1}) = \\ &= \mathbf{y}_k - \mathbf{H}_k \mathbf{E}(\mathbf{x}_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1}) + \mathbf{E}(\nu_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1}) = \end{aligned} \quad (10.21)$$

a z faktu niezależności ν_k od $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1}$ i założenia $\mathbf{E} \nu_k = 0$:

$$= \mathbf{y}_k - \mathbf{H}_k \widehat{\mathbf{x}}_k^- = \mathbf{H}_k \mathbf{x}_k + \nu_k - \mathbf{H}_k \widehat{\mathbf{x}}_k^- = \mathbf{H}_k (\mathbf{x}_k - \widehat{\mathbf{x}}_k^-) + \nu_k. \quad (10.22)$$

Podobnie z (10.19):

$$\begin{aligned}
\mathbf{x}_k - \widehat{\mathbf{x}}_k &= \mathbf{x}_k - \widehat{\mathbf{x}}_k^- - \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k \widehat{\mathbf{x}}_k^-) = \\
&= \mathbf{x}_k - \widehat{\mathbf{x}}_k^- - \mathbf{K}_k(\mathbf{H}_k \mathbf{x}_k + \nu_k - \mathbf{H}_k \widehat{\mathbf{x}}_k^-) = \\
&= (\mathbf{I} - \mathbf{K}_k \mathbf{E}_k)(\mathbf{x}_k - \widehat{\mathbf{x}}_k^-) - \mathbf{K}_k \nu_k.
\end{aligned} \tag{10.23}$$

Ponieważ $\widehat{\mathbf{y}}_k \in Y(k-1) \subset Y(k)$ oraz $\mathbf{x}_k - \widehat{\mathbf{x}}_k \in Y(k)^\perp$ więc $\mathbf{E}(\mathbf{x}_k - \widehat{\mathbf{x}}_k) \widehat{\mathbf{y}}_k^T = 0$.

Dodając ostatnią równość do $\mathbf{E}(\mathbf{x}_k - \widehat{\mathbf{x}}_k) \mathbf{y}_i^T = 0$, otrzymujemy:

$$\mathbf{E}(\mathbf{x}_k - \widehat{\mathbf{x}}_k)(\mathbf{y}_k - \widehat{\mathbf{y}}_k)^T = 0. \tag{10.24}$$

Po podstawieniu do (10.24) wyrażeń (10.22) do (10.23) otrzymujemy:

$$\mathbf{E}((\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)(\mathbf{x}_k - \widehat{\mathbf{x}}_k^-) - \mathbf{K}_k \nu_k)(\mathbf{H}_k(\mathbf{x}_k - \widehat{\mathbf{x}}_k^-) + \nu_k)^T = 0 \tag{10.25}$$

$$(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{E}((\mathbf{x}_k - \widehat{\mathbf{x}}_k^-)((\mathbf{x}_k - \widehat{\mathbf{x}}_k^-)^T) \mathbf{K}_k^T - \mathbf{K}_k \mathbf{E} \nu_k \nu_k^T) = 0. \tag{10.26}$$

Definiując macierz kowariancji błędów estymacji a priori postaci:

$$\mathbf{P}_k^- := \mathbf{E}(\mathbf{x}_k - \widehat{\mathbf{x}}_k^-)((\mathbf{x}_k - \widehat{\mathbf{x}}_k^-)^T) \tag{10.27}$$

otrzymujemy:

$$(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \mathbf{H}_k^T - \mathbf{K}_k R_k = 0 \tag{10.28}$$

a stąd

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + R_k]^{-1}. \tag{10.29}$$

Definiując następnie macierz kowariancji błędów estymacji a posteriori jako:

$$\mathbf{P}_k := \mathbf{E}(\mathbf{x}_k - \widehat{\mathbf{x}}_k)(\mathbf{x}_k - \widehat{\mathbf{x}}_k)^T \tag{10.30}$$

otrzymujemy:

$$\mathbf{P}_k = \mathbf{E}((\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)(\mathbf{x}_k - \widehat{\mathbf{x}}_k^-) - \mathbf{K}_k \nu_k)((\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)(\mathbf{x}_k - \widehat{\mathbf{x}}_k^-) - \mathbf{K}_k \nu_k)^T = \tag{10.31}$$

$$(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{E}[(\mathbf{x}_k - \widehat{\mathbf{x}}_k^-)(\mathbf{x}_k - \widehat{\mathbf{x}}_k^-)^T] (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{E} \nu_k \nu_k^T$$

$$= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{E} \nu_k \nu_k^T =$$

$$= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- - (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{K}_k^T + \mathbf{K}_k \mathbf{E} \nu_k \nu_k^T.$$

Ostatecznie dostajemy:

$$\mathbf{K}_k [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + R_k] = \mathbf{P}_k^- \mathbf{H}_k^T \tag{10.32}$$

$$\mathbf{K}_k[\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T] + \mathbf{K}_k R_k = \mathbf{P}_k^-\mathbf{H}_k^T \quad (10.33)$$

a po wymnożeniu powyższej równości przez \mathbf{K}_k^T :

$$\mathbf{K}_k[\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T]\mathbf{K}_k^T + \mathbf{K}_k R_k \mathbf{K}_k^T = \mathbf{P}_k^-\mathbf{H}_k^T \mathbf{K}_k^T. \quad (10.34)$$

Podstawiając (10.34) do (10.31):

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-. \quad (10.35)$$

Z (10.10) oraz ze wzoru na $\hat{\mathbf{x}}_k^-$:

$$\mathbf{x}_k - \hat{\mathbf{x}}_k^- = \mathbf{F}_{k,k-1}\mathbf{x}_{k-1} + \omega_{k-1} - \mathbf{F}_{k,k-1}\hat{\mathbf{x}}_{k-1}^- = \mathbf{F}_{k,k-1}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^-) + \omega_{k-1} \quad (10.36)$$

więc z definicji \mathbf{P}_k^- oraz z (10.10) otrzymujemy:

$$\mathbf{P}_k^- = \mathbf{E}(\mathbf{F}_{k,k-1}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^-) + \omega_{k-1})(\mathbf{F}_{k,k-1}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^-) + \omega_{k-1})^T = \quad (10.37)$$

$$\mathbf{F}_{k,k-1}\mathbf{P}_k^-\mathbf{F}_{k,k-1}^T + \mathbf{E}\omega_{k-1}\omega_{k-1}^T = \mathbf{F}_{k,k-1}\mathbf{P}_k^-\mathbf{F}_{k,k-1}^T + Q_{k-1}.$$

Równości powyższe pozwalają rekurencyjnie wyznaczyć \mathbf{K}_k , po uprzednim wyznaczeniu: $\hat{\mathbf{x}}_k^-$ oraz \mathbf{P}_k^- .

Filtr Kalmana propaguje więc rozkład normalny warunkowy względem $Y(k)$ wg. schematu:

$$\mathbf{x}_k \sim N(\hat{\mathbf{x}}_k, \mathbf{P}_k)$$

tak więc wyznaczenie $\hat{\mathbf{x}}_k$ i \mathbf{P}_k , pozwala w pełni scharakteryzować nieobserwowalny proces \mathbf{x}_k , przy założeniu znajomości obserwacji $\mathbf{y}_1, \dots, \mathbf{y}_k$.

Estymator $\hat{\mathbf{x}}_k$ jest nieobciążonym estymatorem wartości oczekiwanej \mathbf{x}_k , czyli:

$$\mathbf{E}(\hat{\mathbf{x}}_k) = \mathbf{E}(\mathbf{x}_k) \quad (10.38)$$

oraz jest estymatorem o minimalnej wariancji [104].

10.2. Wyprowadzenie równań rozszerzonego FK

By wyznaczyć wartości $\hat{\mathbf{x}}_{k+1}^-$, $\hat{\mathbf{x}}_{k+1}$ na podstawie $\hat{\mathbf{x}}_k^-$, $\hat{\mathbf{x}}_k$, stosujemy wzór Taylora dla funkcji wielu zmiennych, linearyzując funkcje f oraz h , z pominięciem reszty wzoru Taylora, [6]:

$$f_{k+1}(\mathbf{x}_k) \approx f_{k+1}(\hat{\mathbf{x}}_k) + \mathbf{F}_{k+1,k}(\mathbf{x}_k - \hat{\mathbf{x}}_k) \quad (10.39)$$

gdzie $\mathbf{F}_{k+1,k} = \left[\frac{\partial f^i(\mathbf{x})}{\partial x^j} \right]_{j=1,\dots,n}^{i=1,\dots,n} \Big|_{\mathbf{x}=\widehat{\mathbf{x}}_k}$, $f = [f^1, \dots, f^n]$, $\mathbf{x} = [x^1, \dots, x^n]$
oraz:

$$h_k(\mathbf{x}_k) \approx f_k(\widehat{\mathbf{x}}_k^-) + \mathbf{H}_k(\mathbf{x}_k - \widehat{\mathbf{x}}_k^-) \Big|_{\mathbf{x}=\widehat{\mathbf{x}}_k^-} \quad (10.40)$$

gdzie $\mathbf{H}_k = \left[\frac{\partial h^i(\mathbf{x})}{\partial x^j} \right]_{j=1,\dots,n}^{i=1,\dots,p}$.

Po podstawieniu linearyzacji do równań (3.14)-(3.15) otrzymujemy:

$$\mathbf{x}_{k+1} = \mathbf{F}_{k+1,k}\mathbf{x}_k + f_{k+1}(\widehat{\mathbf{x}}_k) - \mathbf{F}_{k+1,k}\widehat{\mathbf{x}}_k + \omega_k \quad (10.41)$$

$$\mathbf{y}_k - [h_k(\widehat{\mathbf{x}}_k^-) - \mathbf{H}_k\widehat{\mathbf{x}}_k^-] = \mathbf{H}_k\mathbf{x}_k + \nu_k. \quad (10.42)$$

Wprowadzając oznaczenia:

$$d_k = f_{k+1}(\widehat{\mathbf{x}}_k) - \mathbf{F}_{k+1,k}\widehat{\mathbf{x}}_k \quad (10.43)$$

$$\widetilde{\mathbf{y}}_k = \mathbf{y}_k - [h_k(\widehat{\mathbf{x}}_k^-) - \mathbf{H}_k\widehat{\mathbf{x}}_k^-] \quad (10.44)$$

otrzymujemy model postaci:

$$\mathbf{x}_{k+1} = \mathbf{F}_{k+1,k}\mathbf{x}_k + d_k + \omega_k \quad (10.45)$$

$$\widetilde{\mathbf{y}}_k = \mathbf{H}_k\mathbf{x}_k + \nu_k. \quad (10.46)$$

Przy wyznaczaniu estymatora $\widehat{\mathbf{x}}_{k+1}$, wartość d_k jest znana (jest pewną liczbą), wartość $\widetilde{\mathbf{y}}_k$ także jest znana i staje się w tym modelu zmienną obserwowalną.

Zastosowanie do modelu (10.45)-(10.46) postępowania analogicznego, jak dla modelu (3.1)-(3.2) daje dla $k = 1, 2, \dots$ następujące równania:

– estymator wektora stanu a priori :

$$\widehat{\mathbf{x}}_k^- = f_k(\widehat{\mathbf{x}}_{k-1}) \quad (10.47)$$

– przybliżenie macierzy kowariancji błędów estymacji a priori:

$$\mathbf{P}_k^- = \mathbf{F}_{k,k-1}\mathbf{P}_{k-1}\mathbf{F}_{k,k-1}^T + \mathbf{Q}_{k-1} \quad (10.48)$$

– macierz wzmocnienia Kalmana:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k]^{-1} \quad (10.49)$$

– estymator wektora stanu a posteriori, po uwzględnieniu obserwacji \mathbf{y}_k :

$$\widehat{\mathbf{x}}_k = \widehat{\mathbf{x}}_k^- + \mathbf{K}_k[\mathbf{y}_k - h_k(\widehat{\mathbf{x}}_k^-)] \quad (10.50)$$

– przybliżenie macierzy kowariancji błędów estymacji a posteriori:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-. \quad (10.51)$$

Estymator $\hat{\mathbf{x}}_k$ w modelu nieliniowym nie musi być nieobciążony, czyli równość:

$$\mathbf{E}(\hat{\mathbf{x}}_k) = \mathbf{E}(\mathbf{x}_k) \quad (10.52)$$

niekoniecznie zachodzi. Istnieją jednak metody na wyznaczenie obciążenia tego estymatora, lub takiej korekcji działania filtru, by wyznaczyć estymator nieobciążony w przypadku, gdy problem obciążenia estymatora jest istotny [74].

10.3. Definicje akceleracyjnego (przyśpieszeniowego) i przemieszczeniowego spektrum odpowiedzi oraz wykorzystanie spektrum odpowiedzi do projektowania budynków i oceny szkodliwości drgań parasejsmicznych

1. Definicja spektrum odpowiedzi

Spektrum Odpowiedzi wyznacza się na podstawie pojedynczego zarejestrowanego akceleroqramu. Określa ono maksymalne wartości przyspieszenia (Przyspieszeniowe Spektrum Odpowiedzi) lub przemieszczenia (Przemieszczeniowe Spektrum Odpowiedzi) układu o jednym stopniu swobody, poddanego wymuszeniu kinematycznemu, [75]. Celem wyznaczenia spektrum, należy rozwiązać równanie ruchu takiego układu o masie m , parametrze tłumienia c , sztywności nieważkich więzi sprężystej k :

$$m \frac{d^2 y}{dt^2} + c \frac{dy}{dt} + ky = -m \frac{d^2 y_g}{dt^2} \quad (10.53)$$

gdzie $y(t)$ jest przemieszczeniem względnym układu w czasie (czyli przemieszczeniem liczonym w stosunku do położenia konstrukcji przed wstąpieniem ruchu podłoża), $\frac{d^2 y_g}{dt^2}$ jest przyśpieszeniem podłoża, które zwykle dane jest w postaci akceleroqramu zarejestrowanego dla pojedynczego wstrząsu.

Po wprowadzeniu częstości kołowej drgań własnych $\omega_0 = \sqrt{k/m}$ oraz ułamka tłumienia $\xi = c/2\sqrt{km}$ równanie to przyjmuje postać:

$$\frac{d^2 y}{dt^2} + 2\xi\omega_0 \frac{dy}{dt} + \omega_0^2 y = \frac{d^2 y_g}{dt^2} \quad (10.54)$$

Jest to równanie liniowe różniczkowe zwyczajne drugiego rzędu, o stałych współczynnikach. Jego rozwiązanie otrzymuje się w postaci całki:

$$y(t, \omega_0, \xi) = \frac{1}{\omega_0 \sqrt{1 - \xi^2}} \int_0^t \frac{d^2 y_g}{dt^2}(\tau) e^{-\xi\omega_0(t-\tau)} \sin(\omega_0 \sqrt{1 - \xi^2}(t - \tau)) d\tau \quad (10.55)$$

Przemieszczeniowe Spektrum Odpowiedzi definiuje się jako maksymalne wartości przemieszczeń:

$$S_d(\omega_0, \xi) = \max_t (|y(t, \omega_0, \xi)|) \quad (10.56)$$

natomiast maksymalne wartości przyspieszeń:

$$S_a(\omega_0, \xi) = \max_t \left(\left| \frac{d^2 y(t, \omega_0, \xi)}{d^2 t} \right| \right) \quad (10.57)$$

jako Przyspieszeniowe Spektrum Odpowiedzi. Przyspieszeniowe Spektrum Odpowiedzi podzielone przez maksymalną wartość przyspieszenia podłoża $a_{max} = \max_t \left(\frac{d^2 y_g}{d^2 t} \right)$, czyli wielkość

$$\beta = \frac{S_a(\omega_0, \xi)}{a_{max}} \quad (10.58)$$

nazywamy Względny Przyspieszeniowym Spektrum Odpowiedzi.

2. Wykorzystanie spektrum odpowiedzi do projektowania budynków i oceny szkodliwości drgań parasejsmicznych

Jeśli budynek modelowany jest za pomocą układu o jednym stopniu swobody otrzymujemy maksymalną siłę działającą na układ przy zadanym wymuszeniu postaci:

$$\max_t f(t) = \max_t ky(t) = kS_d(\omega_0, \xi) \quad (10.59)$$

Można w ten sposób określić maksymalną odpowiedź układu dla różnych częstości drgań własnych (ω_0) budynku oraz dla różnych liczb tłumienia (ξ), dla zadanego zapisu ruchu podłoża, opisanego jednym akcelerogramem [75].

Jeśli dla danego terenu poddanego wstrząsom parasejsmicznym wyznaczono Spektra Odpowiedzi od występujących tam wstrząsów, to dla budowli modelowanych układami dyskretnymi o wielu stopniach swobody, określa się na ich podstawie maksymalną siłę sejsmiczną działającą na j -ty element konstrukcji o masie m_j przy drganiach i -ta częstością własną. Oblicza się też całkowitą maksymalną siłę sejsmiczną obciążającą j -ty element konstrukcji metodą Spektrum Odpowiedzi dla układów dyskretnych.

Niech

$$B \frac{d^2 y}{d^2 t} + C \frac{dy}{dt} + Ky = -B \mathbf{r} \frac{d^2 y_g}{d^2 t} \quad (10.60)$$

będzie równaniem ruchu dla układu o n -stopniach swobody, o macierzy bezwładności B , macierzy sztywności K oraz macierzy tłumienia C , gdzie \mathbf{r} jest wektorem wpływu. (W przypadku płaskiego układu, poddanego wymuszeniu kinematycznemu poziomemu, \mathbf{r} jest wektorem o n elementach równych jeden.)

Rozwiązanie tego równania (wyznaczone za pomocą metody transformacji własnej) jest funkcją o postaci:

$$y(t) = \sum_i \mathbf{w}_i y_i(t) \quad (10.61)$$

gdzie \mathbf{w}_i jest i -tym wektorem własnym oraz:

$$y_i(t) = \frac{\mathbf{w}_i^T B \mathbf{r}}{\mathbf{w}_i^T B \mathbf{w}_i \omega_i \sqrt{1 - \xi_i^2}} \int_0^t \frac{d^2 y_g}{d^2 t}(\tau) e^{-\xi_i \omega_i (t-\tau)} \sin(\omega_i \sqrt{1 - \xi_i^2} (t - \tau)) d\tau. \quad (10.62)$$

Udział i -tej postaci drgań własnych w reakcji j -tej współrzędnej uogólnionej wynosi:

$$y_{ji}(t) = \mathbf{w}_i(j) \frac{\mathbf{w}_i^T B \mathbf{r}}{\mathbf{w}_i^T B \mathbf{w}_i \omega_i \sqrt{1 - \xi_i^2}} \frac{d^2 y_g}{d^2 t}(\tau) e^{-\xi_i \omega_i (t-\tau)} \sin(\omega_i \sqrt{1 - \xi_i^2} (t - \tau)) d\tau \quad (10.63)$$

gdzie $\mathbf{w}_i(j)$ jest j -tą współrzędną i -tego wektora własnego. Maksymalny udział i -tej postaci drgań własnych w reakcji j -tej współrzędnej uogólnionej wynosi więc:

$$\max_t y_{ji}(t) = |\mathbf{w}_i(j)| \frac{\mathbf{w}_i^T B \mathbf{r}}{\mathbf{w}_i^T B \mathbf{w}_i \omega_i \sqrt{1 - \xi_i^2}} \|S_d(\xi_i, \omega_i)\| \quad (10.64)$$

Dla diagonalnej macierzy bezwładności $-$, maksymalna siła sejsmiczna działająca na j -ty element konstrukcji o masie m_j , przy drganiach i -tą częstotścią własną wynosi:

$$\max_t P_{ji}(t) = |m_j \mathbf{w}_i(j)| \frac{\mathbf{w}_i^T B \mathbf{r}}{\mathbf{w}_i^T B \mathbf{w}_i \omega_i \sqrt{1 - \xi_i^2}} S_d(\xi_i, \omega_i) \quad (10.65)$$

Całkowitą siłę sejsmiczną obciążającą j -ty element wyznacza się np. na podstawie metody SRSS (*ang. Square Root of Sum Squares*) w postaci:

$$\max_t P_j(t) = \sqrt{\sum_i \max(P_{ji})^2}. \quad (10.66)$$