

Politechnika Wrocławska  
Wydział Informatyki i Zarządzania  
Instytut Informatyki

Rozprawa doktorska

IDENTYFIKACJA DYNAMICZNYCH  
SYSTEMÓW ZŁOŻONYCH Z  
WYKORZYSTANIEM SIECI NEURONOWYCH

**Jarosław Drapała**

Promotor: dr hab inż. Jerzy Świątek, prof. PWr

Wrocław 2009

*Profesorowi Jerzemu Świątkowi składam serdeczne podziękowania za przekazaną wiedzę oraz pomoc w pracy naukowej. Rodzicom: Marii i Edwardowi, wyrażam wdzięczność za otrzymane wsparcie.*

# Spis treści

Spis treści	i
<b>1 Identyfikacja systemów złożonych z wykorzystaniem sieci neuronowych – sformułowanie problemu pracy</b>	<b>1</b>
1.1 Wstęp . . . . .	1
1.2 Aktualny stan badań . . . . .	5
1.3 Sieci neuronowe . . . . .	9
1.3.1 Opis neuronu statycznego . . . . .	9
1.3.2 Statyczna sieć wielowarstwowa . . . . .	11
1.3.3 Opis neuronu rekurencyjnego . . . . .	14
1.3.4 Diagonalna rekurencyjna sieć neuronowa (DRNN) . . . . .	15
1.4 Identyfikacja obiektu dynamicznego . . . . .	19
1.4.1 Algorytm uczenia statycznej sieci neuronowej . . . . .	22
1.4.2 Algorytm identyfikacji z wykorzystaniem modelu liniowego . . . . .	27
1.4.3 Algorytm uczenia rekurencyjnej sieci neuronowej . . . . .	34
1.5 Identyfikacja systemów złożonych . . . . .	37
1.5.1 Modele lokalnie i globalnie optymalne . . . . .	41
1.5.2 Wykorzystanie sieci neuronowych . . . . .	47
1.6 Teza, cel i zakres pracy . . . . .	48
<b>2 Algorytmy uczenia rekurencyjnych sieci neuronowych w modelowaniu systemu złożonego</b>	<b>50</b>
2.1 Model lokalnie optymalny . . . . .	50
2.2 Model globalnie optymalny . . . . .	56

2.3	Model globalnie optymalny z uwzględnieniem jakości modeli lokalnych . . . . .	81
2.3.1	Syntetyczny wskaźnik jakości . . . . .	82
2.3.2	Zadawalające modele lokalne . . . . .	88
<b>3</b>	<b>Analiza zbieżności algorytmów uczenia sieci neuronowych w systemie złożonym</b>	<b>100</b>
3.1	Zbieżność algorytmu uczenia sieci lokalnej . . . . .	101
3.2	Zbieżność algorytmu uczenia sieci złożonej przy globalnym wskaźniku jakości . . . . .	104
3.3	Zbieżność przy syntetycznym wskaźniku jakości . . . . .	109
3.4	Zbieżność przy zewnętrznej funkcji kary . . . . .	112
3.5	Zbieżność przy wewnętrznej funkcji kary . . . . .	118
<b>4</b>	<b>Badania symulacyjne</b>	<b>121</b>
4.1	Plan i zakres badań . . . . .	121
4.2	Wyniki badań symulacyjnych – system liniowy . . . . .	126
4.3	Wyniki badań symulacyjnych – reaktor chemiczny . . . . .	135
4.4	Omówienie wyników . . . . .	142
<b>5</b>	<b>Podsumowanie</b>	<b>145</b>
5.1	Oryginalny wkład pracy w dziedzinie identyfikacji systemów oraz sieci neuronowych . . . . .	145
5.2	Plan dalszych badań . . . . .	147
5.3	Uwagi końcowe . . . . .	147
	<b>Bibliografia</b>	<b>151</b>
	<b>Spis symboli i skrótów</b>	<b>163</b>
	<b>Spis rysunków</b>	<b>165</b>
	<b>Spis tabel</b>	<b>168</b>
	<b>Skorowidz</b>	<b>169</b>

## Rozdział 1

# Identyfikacja systemów złożonych z wykorzystaniem sieci neuronowych – sformułowanie problemu pracy

### 1.1 Wstęp

Modelowanie matematyczne i będąca jego częścią identyfikacja, polegająca na ustalaniu modeli obiektów na podstawie badań eksperymentalnych, stanowią ważne narzędzia o charakterze nie tylko poznawczym, ale również praktycznym. Z uwagi na uniwersalną naturę opisu matematycznego mają one charakter interdyscyplinarny, dzięki czemu znalazły wiele zastosowań w różnych gałęziach nauki i techniki, [50]. Z jednej strony opracowane metody identyfikacji dostarczają informatyce narzędzi do rozwiązywania szerokiej klasy problemów, z drugiej strony nieustanny rozwój technik komputerowych powiększa zakres metod modelowania i identyfikacji, które mogą być zrealizowane w praktyce, inspirując tym samym powstawanie nowych algorytmów. Ponadto sam wzrost prędkości obliczeń komputerów umożliwia stosowanie algorytmów identyfikacji, których użycia wcześniej nie rozpatrywano ze względu na ograniczone zasoby pamięciowe oraz wymaganą prędkość obliczeń (algorytmy opracowywane ramach niniejszej rozprawy można zaliczyć do tej kategorii).

Tworzenie modelu nie jest celem samym w sobie. Skonstruowany model służy dalej do takich celów jak, [79], [130]:

- zrozumienie działania systemu;
- projektowanie algorytmu sterowania modelowanym procesem/obiektem;
- symulacja zjawisk zachodzących w badanym systemie;
- predykcja;
- diagnostyka;
- optymalizacja procesu; itp.

Oznacza to, że jakość modelowania determinuje wielkość górnego ograniczenia na jakość rozwiązania docelowego problemu [130]. Stąd wynika ważność technik identyfikacji oraz potrzeba ich rozwijania.

Konieczność rozpatrywania coraz bardziej złożonych obiektów, wielopoziomowych systemów sterowania, zdecentralizowanego sterowania złożonymi procesami, stwarzają potrzebę konstruowania modeli systemów złożonych. Przez system złożony rozumiemy tu obiekt, w którym można wyróżnić powiązane ze sobą elementy składowe o określonych wielkościach wejściowych i wyjściowych. Powiązania oznaczają, że wyjścia pewnych elementów są jednocześnie wejściami innych elementów. Modelowanie systemów złożonych wiąże się wobec tego z identyfikacją pojedynczych elementów z uwzględnieniem powiązań między nimi. Generuje to nowe zadania identyfikacji systemów złożonych wymagające nowych metod ich rozwiązywania. Z głównych kierunków badawczych tej dziedziny należy wymienić:

- identyfikację przy ograniczonych możliwościach pomiarowych, [123];
- identyfikację wielostopniową, [119];
- identyfikację globalną [17].

Ostatnie z wymienionych zagadnień jest głównym przedmiotem rozważań. Pojawia się ono na skutek możliwości określenia różnego rodzaju wskaźników jakości identyfikacji w systemie złożonym: *lokalnych wskaźników jakości* oraz *wskaźnika*

*globalnego*, [121]. Lokalny wskaźnik jakości ocenia błąd modelowania każdego elementu niezależnie od powiązań. Globalny wskaźnik jakości bierze pod uwagę błędy związane z wyróżnionymi ze względu na cel tworzenia modelu wyjściami oraz uwzględnia połączenia między elementami. Tak więc z jednej strony możliwe jest wyznaczenie modeli optymalnych dla każdego z elementów osobno, niezależnie od struktury systemu, z drugiej zaś ocena jakości identyfikacji może dotyczyć systemu jako całości. W pierwszym przypadku otrzymujemy *model lokalnie optymalny*, natomiast drugi prowadzi do *modelu globalnie optymalnego*, [121]. Złożenie modeli lokalnych zgodnie ze strukturą systemu na ogół prowadzi do modelu systemu złożonego różnego od uzyskanego w drodze optymalizacji globalnego wskaźnika jakości.

Istnieje również możliwość takiego sformułowania wskaźnika jakości identyfikacji, w którym obie oceny, lokalna i globalna, są uwzględniane w takim stopniu, jakiego życzy sobie użytkownik odpowiedniego algorytmu identyfikacji systemu złożonego [124].

W pracy badane są dynamiczne obiekty wejściowo-wyjściowe, [47], i z takich komponentów składać się będą omawiane systemy złożone. W sytuacji, gdy modele obiektów budujemy jedynie na podstawie pomiarów przebiegów jego wejść i wyjść, wygodną w zastosowaniu klasą modeli są sieci neuronowe, zwłaszcza w przypadku modelowania obiektów nieliniowych, [12], [20], [68], [126]. Cenną właściwością sieci neuronowych jest cecha „uniwersalnej aproksymacji” oznaczająca zdolność do przybliżania każdej gładkiej funkcji z dowolną dokładnością, rosnącą wraz liczbą tzw. neuronów ukrytych (w przypadku sieci wielowarstwowych), [23], [54], [107], [64]. Jednak cechę „uniwersalnej aproksymacji” posiadają nie tylko sieci neuronowe ale również m.in. modele wielomianowe, szeregi Fouriera, falki, pewne rodzaje modeli rozmytych i rozmyto-neuronowych, [109], [94], [114].

Za użyciem sieci neuronowych przemawiają, [53], [38]:

- możliwość równoległego prowadzenia obliczeń związanych zarówno z wyznaczeniem wartości wyjść modelu neuronowego jak i doбором jego parametrów w procesie identyfikacji, zwanym najczęściej „uczeniem sieci” [13];
- stosunkowo duża odporność na przypadkowe błędy w danych pomiarowych użytych do uczenia, wynikająca z postaci nieliniowej tzw. funkcji przejścia

stosowanej w równaniach opisujących przebieg obliczeń w sieci neuronowej [143], [38], [97];

- gładka charakterystyka nieliniowa, dzięki czemu nawet przy dużej liczbie parametrów łatwiej jest kontrolować jakość interpolacji modelu neuronowego (w odróżnieniu od np. modeli wielomianowych wysokiego stopnia, które mogą zbyt często i gwałtownie zmieniać wartości wyjść przy niewielkim przedziale zmienności sygnałów wejściowych, [60]).

Wymienione wyżej zalety czynią sieci neuronowe dogodnym narzędziem do modelowania złożonych, nieliniowych obiektów. Wśród zalet sieci neuronowych wymienia się często zdolność do modelowania obiektów na podstawie danych pomiarowych oraz do automatycznej korekty wartości swoich parametrów (pomijając konieczność doboru odpowiednich wartości tzw. „współczynników uczenia” przez projektanta sieci). Należy jednak zaznaczyć, że jest to nie tyle cecha sieci neuronowych, co samych procedur identyfikacji, w których – ze względu na brak możliwości określenia dokładnej relacji wiążącej wejście obiektu z jego wyjściem – dobiera się parametry modelu o z góry założonej postaci (tzw. „wybór optymalnego modelu”, [16]) w taki sposób, aby minimalizować błąd modelowania wyliczany w oparciu o dane pomiarowe.

Z wad sieci neuronowych należy zwrócić uwagę na:

- trudności w interpretacji wartości parametrów sieci uzyskanych w rezultacie uczenia;
- nieliniowy (najczęściej) sposób występowania parametrów w modelu neuronowym, powodujący konieczność użycia numerycznych metod optymalizacji w uczeniu sieci, [97].

Pomimo, że sieci neuronowe są narzędziem ogólnego zastosowania, to – w związku wyżej wymienionymi ich wadami i zaletami – w konkretnych zadaniach ich jakość i efektywność jako modeli może być zróżnicowana, [4].

W ramach pracy autor zamierza opracować algorytmy identyfikacji nieliniowych dynamicznych systemów złożonych o strukturze kaskadowej z wykorzystaniem rekurencyjnych sieci neuronowych (tj. sieci ze sprzężeniami zwrotnymi) jako:

- modeli lokalnie optymalnych;



- modelu globalnie optymalnego;
- modelu globalnie optymalnego, w którym uwzględniono jakości modeli lokalnych.

Układ pracy jest następujący. Dalsza część rozdziału pierwszego zawiera przegląd aktualnego stanu badań w dziedzinie identyfikacji globalnej systemów złożonych oraz rekurencyjnych sieci neuronowych. Przedstawione zostały podstawowe algorytmy uczenia wielowarstwowej statycznej sieci neuronowej oraz sieci ze sprzężeniami zwrotnymi, modelujące pojedynczy, wejściowo-wyjściowy obiekt dynamiczny. W podpunkcie 1.5 wyjaśniono istotę modelowania lokalnego i globalnego systemu złożonego oraz przedstawiono rolę rekurencyjnych sieci neuronowych w rozwiązywaniu zadań modelowania systemów złożonych. W podpunkcie 1.6 podano tezę, cel i zakres pracy.

W dwóch kolejnych rozdziałach zawarte są najważniejsze wyniki pracy. W rozdziale drugim przedstawiono oryginalne algorytmy uczenia rekurencyjnych sieci neuronowych realizujących model globalny systemu złożonego o strukturze kaskadowej oraz model globalny, w którym uwzględniono jakość modeli lokalnych. Analizę zbieżności opracowanych algorytmów przeprowadzono w rozdziale trzecim.

W rozdziale czwartym wykazano skuteczność działania algorytmów w drodze badań symulacyjnych. Modelowano złożony system liniowy oraz reaktor chemiczny z ciągłym przepływem składników.

Rozdział piąty podsumowuje osiągnięte rezultaty i wskazuje obszar dalszych badań.

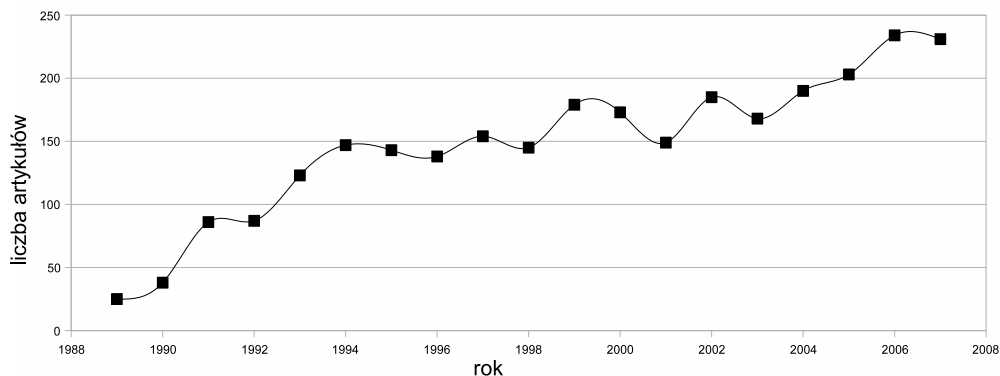
## 1.2 Aktualny stan badań

Problemy identyfikacji globalnej systemów złożonych zostały wstępnie sformułowane w pracy [16], a następnie rozwijane w pracach [119], [122], [28], [32] oraz [51]. Potrzebę modelowania systemów złożonych dla celów sterowania sygnalizowano już we wcześniejszych publikacjach, np. [73]. Pewne problemy identyfikacji systemów złożonych sformułowano w [128] i [129]. W pracy [118] opracowano i zastosowano algorytmy identyfikacji globalnej do rozwiązania praktycznego zadania modelowania procesów hemodynamicznych w układzie krążenia. W pracy [51]

przedstawiono algorytmy identyfikacji statycznych systemów złożonych dla klasy modeli liniowych. Rozpatrzono m. in. zadanie identyfikacji parametrycznej oraz wyboru optymalnego modelu, dla sytuacji deterministycznej oraz w warunkach losowych. Przedstawiono także przykład zastosowania praktycznego polegającego na modelowaniu procesu wzbogacania rudy miedzi. W artykule [44] pokazano podejście lokalne do modelowania systemu złożonego. Podejście wykorzystujące programowanie dynamiczne do identyfikacji globalnej statycznego systemu złożonego o strukturze kaskadowej opracowano w [122]. Kolejnym krokiem było rozwiązanie problemu dla nieliniowych systemów złożonych. W takim przypadku nie jest możliwe uzyskanie rozwiązań analitycznych o charakterze ogólnym, dlatego posłużono się sieciami neuronowymi w celu uzyskania odpowiednich modeli. W pracach [25] i [120] opracowano algorytmy uczenia jednokierunkowych sieci wielowarstwowych do modelowania statycznych systemów złożonych. Udało się je zastosować do modelowania złożonego procesu produkcji azotynu amonu, [24]. Praca [26] przedstawia algorytm uczenia sieci neuronowych modelujących system złożony w taki sposób, aby uwzględniać jednocześnie jakość modelu globalnego i jakość modeli lokalnych. W [27] pokazano, jak algorytmy identyfikacji globalnej z użyciem jednokierunkowych sieci wielowarstwowych można dostosować do celów modelowania złożonego systemu dynamicznego (niniejsza praca stanowi kontynuację tego nurtu badań). Modelowanie systemu złożonego na zasadzie składania wielu sieci w sieć złożoną zastosowane zostało również w pracy [89]. Artykuły [99] i [7] przedstawiają zastosowania rekurencyjnej sieci neuronowej do modelowania procesu złożonego, nie wyróżniając jednak poszczególnych elementów wejściowo-wyjściowych. Podobne podejście, lecz wykorzystujące sieci neuronowo-rozmyte i modele Takagi-Sugeno zaprezentowano w artykułach [72] i [105]. Ocenę lokalną i globalną w identyfikacji modeli rozmytych Takagi-Sugeno wprowadzono w pracy [3], następnie w artykule [5] rozszerzono rozważania o ocenę wielokryterialną, a w pracy [6] rozpatrzono przypadek systemu o strukturze kaskadowej. Użycie klasycznych modeli lokalnych w identyfikacji systemu złożonego zademonstrowano w pracy [15]. W pracach [48], [52] rozwijany jest kierunek badań polegający na rozpatrywaniu nieliniowych dynamicznych obiektów jako systemu złożonego z szeregowego połączenia elementów liniowego dynamicznego oraz nieliniowego statycznego tj. systemy Hammersteina i Wienera. W artykule [55] zaprezentowano zastosowanie adaptacji przez identyfikację do sterowania złożonym systemem połączonych szeregowo odwróconych wahadeł. W artykule [142] opracowano al-

gorytm uczenia rekurencyjnych sieci neuronowych do sterowania dynamicznym systemem złożonym. Sieci zorganizowano w dwupoziomą strukturę hierarchiczną, w której sieć na wyższym poziomie pełni rolę koordynatora. Takie podejścia dla modeli klasycznych znane są w polskiej literaturze, głównie z monografii [41] i [43]. Najnowsze podejścia do modelowania systemów złożonych można znaleźć w [14], [104] i [57].

Rekurencyjnym sieciom neuronowym jako modelom dynamicznych systemów wejściowo-wyjściowych poświęca się obecnie coraz więcej uwagi, [87], [49]. Wykres na rys. 1.1 ilustruje wzrost liczby artykułów dotyczących zastosowania rekurencyjnych sieci do identyfikacji systemów dynamicznych w czasopiśmie *IEEE Trans. on Neural Network* od roku 1988.



Rysunek 1.1: Liczba artykułów nt. rekurencyjnych sieci neuronowych w czasopiśmie *IEEE Trans. on Neural Networks* w latach 1988–2008 (badania własne)

Opracowano wiele rodzajów sieci rekurencyjnych, począwszy od sieci w pełni rekurencyjnych, [137], w których każdy neuron pobudzany jest przez poprzednie sygnały wszystkich neuronów sieci, przez – rzadziej używane – sieci z połączeniami rekurencyjnymi pomiędzy warstwami lecz nie pomiędzy neuronami z tej samej warstwy, aż po sieci, w których rekurencyjne połączenia istnieją jedynie w obrębie samych neuronów, [108]. W pracy skupimy się na ostatnim z wymienionych rodzajów sieci, gdyż posiada najmniej parametrów związanych z połączeniami rekurencyjnymi, umożliwiając względnie dużą szybkość uczenia (w pracy [78] uzasadnia się, że wpływ losowych zaburzeń na model wzrasta wraz z liczbą jego parametrów). Pomimo tego w większości zadań modelowania obiektów możliwości aproksymacji charakterystyk dynamicznych tego typu sieci są wystarczająco

jące [140]. Znajdują bardzo dużo zastosowań w sterowaniu, [134], [91], [106], [93]. Sieci tego typu nazywa się *lokalnie rekurencyjnymi*, *globalnie jednokierunkowymi* (LRGF – *Locally Recurrent, Globally Feedforward*), [94], [127]. Dokładniej, użyta została tzw. diagonalna rekurencyjna sieć neuronowa (DRNN – *Diagonal Recurrent Neural Network*), [69], [70], opracowana głównie na potrzeby sterowania adaptacyjnego [140]. Jej nazwa bierze się stąd, że w kwadratowej macierzy zerojedynkowej, której element o współrzędnych  $(i, j)$  wskazuje na istnienie lub nieistnienie połączenia rekurencyjnego pomiędzy neuronem  $i$ -tym a  $j$ -tym danej warstwy, jedynie elementy na diagonalu mają wartość 1. Macierz ta reprezentuje warstwę neuronów, w której pętle sprzężenia zwrotnego występują tylko w obrębie pojedynczych neuronów. Wyczerpującego przeglądu różnych typów sieci LRGF oraz neuronów z różnego rodzaju sprzężeniami zwrotnymi (obejmującymi tylko część liniową neuronu, wyjście części nieliniowej lub z filtrami o nieskończonej odpowiedzi impulsowej w miejscu wag) dokonano w pracy [127]. Z ważniejszych typów sieci lokalnie rekurencyjnych wymienić należy sieci Elmana i Jordana, [97].

Opracowano różne podejścia do wyznaczania wartości „gradientów” potrzebnych w uczeniu sieci rekurencyjnych. Najważniejsze z nich przedstawiono w pracach [137], [135], [92], [115] [45] oraz zestawiono w przeglądowym artykule [102]. Wielu autorów wskazuje na podobieństwa między rekurencyjnymi neuronami a filtrami adaptacyjnymi, lub wręcz między filtrami a całymimi sieciami, [103]. Dotyczy to zarówno formalnego opisu jak i wspólnych algorytmów uczenia/adaptacji, [95], [110], [2], [87], [100]. Przedmiotem ostatniej z wymienionych prac jest algorytm uczenia rekurencyjnej sieci neuronowej oparty o metodę aproksymacji stochastycznej, a w [101] podjęto tematykę doboru sekwencji uczącej dla sieci rekurencyjnej. W [76] pokazano, że rekurencyjna sieć neuronowa realizuje proces NARX. Warto odnotować zastosowanie w pracy [8] sieci samoorganizujących się typu SOM do identyfikacji i sterowania systemów dynamicznych.

Na koniec warto zaznaczyć, że istnieje wiele typów sieci ze sprzężeniami zwrotnymi mających zastosowania w innych dziedzinach niż identyfikacja, np. sieci BAM, Hopfielda, maszyny Boltzmana, [143], [75].

## 1.3 Sieci neuronowe

W niniejszym punkcie podano dokładne opisy:

- neuronu statycznego;
- statycznej sieci wielowarstwowej;
- neuronu rekurencyjnego;
- diagonalnej rekurencyjnej sieci neuronowej (DRNN);

w formie zależności funkcyjnych i równań różnicowych. Dla sieci neuronowych podano również opisy w postaci odwzorowań wektorowych z przestrzeni wejść na przestrzeń wyjść sieci.

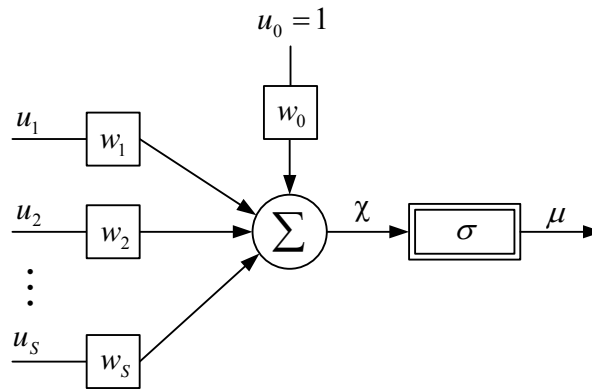
### 1.3.1 Opis neuronu statycznego

Neuronem statycznym nazywamy element o wielu wejściach i jednym wyjściu (rys. 1.2), opisanym funkcją postaci:

$$\mu = \sigma(\chi), \quad \chi = \sum_{s=0}^S w_s u_s, \quad u_0 = 1, \quad (1.1)$$

gdzie  $u_s$  ( $s = 1, 2, \dots, S$ ) – sygnały wejściowe,  $\mu$  – wyjście neuronu,  $w_s$  ( $s = 0, 1, \dots, S$ ) – parametry neuronu, zwane "wagami",  $\chi$  – wyjście części liniowej neuronu, natomiast  $\sigma : \mathcal{R} \rightarrow (a, b)$  jest funkcją sigmoidalną spełniającą następujące warunki, [87]:

- i)  $\sigma(\chi)$  jest funkcją ciągłą i różniczkowalną;
- ii)  $\sigma'(\chi) = \frac{d\sigma(\chi)}{d\chi} > 0$  dla wszystkich  $\chi \in \mathcal{R}$ ;
- iii)  $\lim_{\chi \rightarrow \pm\infty} \sigma'(\chi) = 0$ ;
- iv)  $\sigma'(\chi)$  ma globalne optimum w jednoznacznie określonym punkcie  $\chi = 0$ ;
- v)  $\sigma(\chi)$  posiada jeden punkt przegięcia (najlepiej w  $\chi = 0$ );
- vi)  $\sigma(\chi)$  spełnia warunek Lipschitza, tj. istnieje taka stała  $L > 0$ , że
 
$$\forall_{\chi_1, \chi_2 \in \mathcal{R}} \quad \|\sigma(\chi_1) - \sigma(\chi_2)\| \leq L \|\chi_1 - \chi_2\|.$$



Rysunek 1.2: Statyczny neuron

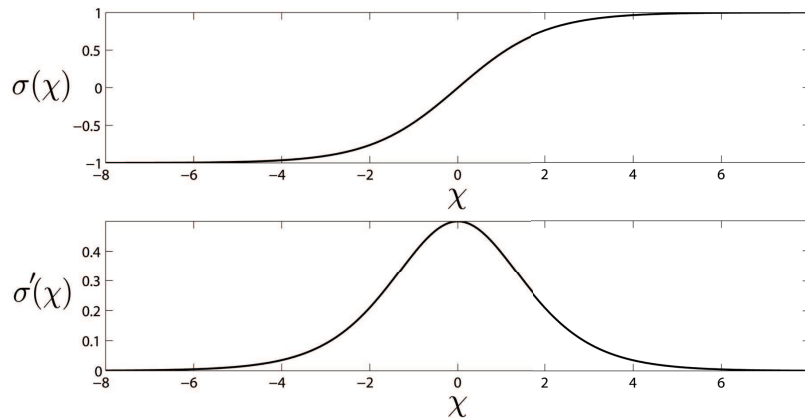
W pracy korzystać będziemy z funkcji  $\sigma$  o postaci:

$$\sigma(\chi) = \frac{1 - e^{-\chi}}{1 + e^{-\chi}}, \quad (1.2)$$

która przyjmuje wartości od  $-1$  dla  $\chi \rightarrow -\infty$  do  $1$  dla  $\chi \rightarrow +\infty$ . Pochodna tej funkcji ma postać dogodną przy wykonywaniu obliczeń:

$$\frac{d\sigma(\chi)}{d\chi} \equiv \sigma'(\chi) = \frac{1}{2} (1 - \sigma^2(\chi)). \quad (1.3)$$

Wykres funkcji  $\sigma$  i jej pochodnej przedstawia rys. 1.3.



Rysunek 1.3: Wykres funkcji sigmoidalnej (1.3) i jej pochodnej

Zwróćmy uwagę, iż stała wartość  $u_0 = 1$  powoduje, że waga  $w_0$  spełnia rolę parametru swobodnego części liniowej neuronu. Neuron posiada więc  $S$  wejść i  $S + 1$  parametrów.

### 1.3.2 Statyczna sieć wielowarstwowa

Sieć neuronowa wielowarstwowa jest modelem, którego postać funkcyjna składa się z elementów opisanych wzorem (1.1). Elementy te, czyli neurony, ułożone są w szereg warstw, z których każda składa się z równolegle umieszczonych neuronów, na które trafiają te same sygnały wejściowe, [126]. Pojedyncza warstwa sieci realizuje nieliniowe odwzorowanie przestrzeni wektorów wejściowych o wymiarze  $S$  w przestrzeń wektorów wyjściowych o wymiarze  $L$ .

Warstwy połączone są w ten sposób, że wszystkie wyjścia neuronów warstwy poprzedzającej podawane są na wejścia każdego neuronu warstwy kolejnej. Oznacza to, że liczba wejść neuronów danej warstwy jest równa liczbie neuronów warstwy poprzedzającej, lub – jeśli jest to warstwa pierwsza w kolejności – liczbie wejść sieci neuronowej. Ostatnią warstwę nazywamy *warstwą wyjściową*, [126], gdyż jej wyjścia są jednocześnie wyjściami całego modelu neuronowego. Liczba neuronów tej warstwy jest równa  $L$ . Pozostałe warstwy nazywamy *warstwami ukrytymi*, [126], a liczba neuronów w tych warstwach jest dobierana dowolnie przez projektanta sieci.

Obliczenia konieczne do wyznaczenia wartości wyjść modelu neuronowego w odpowiedzi na zadane sygnały wejściowe można zilustrować w formie grafu (rys. 1.4), którego węzeł (neuron) realizuje równanie:

$$\mu_k^{(j)} = \sigma(\chi_k^{(j)}), \quad \chi_k^{(j)} = \sum_{l=0}^{L_j-1} w_{kl}^{(j)} \mu_l^{(j-1)}, \quad \mu_0^{(j)} = 1, \quad (1.4)$$

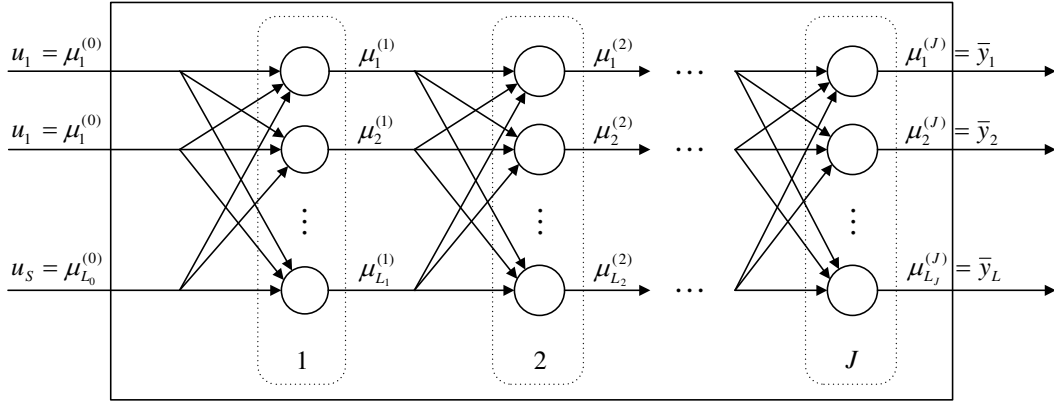
gdzie  $j$  – oznacza numer warstwy,  $J$  – jest liczbą warstw sieci ( $j = 0, 1, 2, \dots, J$ ), przy czym  $J$ -ta warstwa jest warstwą wyjściową a wejścia sieci traktujemy jako wyjścia warstwy o numerze  $j = 0$ ,  $L_j$  – liczba neuronów składających się na  $j$ -tą warstwę, przy czym dla  $j = 0$  mamy  $L_0 = S$ ,  $\mu_k^{(j)}$  ( $k = 0, 1, \dots, L_j$ ;  $j = 0, 1, \dots, J$ ) – jest wyjściem  $k$ -tego neuronu  $j$ -tej warstwy,  $w_{kl}^{(j)}$  ( $l = 0, 1, \dots, L_{j-1}$ ;  $j = 1, 2, \dots, J$ ) – to waga połączenia między  $k$ -tym neuronem  $j$ -tej warstwy a  $l$ -tym neuronem warstwy o numerze  $j - 1$ . Podobnie jak w (1.1), wagi  $w_{k0}^{(j)}$  ( $k = 1, 2, \dots, L_j$ ;  $j = 1, 2, \dots, J$ ) pełnią rolę wyrazów wolnych części liniowych neuronów, gdyż  $\mu_0^{(j)}$  są stale równe 1. Jak wspomniano wcześniej, wyjścia warstwy  $J$ -tej są sygnałami wyjściowymi sieci neuronowej, tzn.  $\bar{y}_l = \mu_l^{(J)}$  ( $l = 1, 2, \dots, L$ ;  $L_j = L$ ).

Zapiszmy odwzorowanie realizowane przez sieć w formie wektorowej. W tym celu zbierzemy wejścia i wyjścia wszystkich warstw w wektory z odpowiednich

przestrzeni:

$$\mathbf{u} = \begin{bmatrix} 1 \\ u_1 \\ u_2 \\ \vdots \\ u_S \end{bmatrix} \in \mathcal{R}^{S+1}, \quad \boldsymbol{\mu}^{(j)} = \begin{bmatrix} 1 \\ \mu_1^{(j)} \\ \mu_2^{(j)} \\ \vdots \\ \mu_{L_j}^{(j)} \end{bmatrix} \in [-1, 1]^{L_j+1}$$

$$\boldsymbol{\chi}^{(j)} = \begin{bmatrix} \chi_1^{(j)} \\ \chi_2^{(j)} \\ \vdots \\ \chi_{L_j}^{(j)} \end{bmatrix} \in \mathcal{R}^{L_j}, \quad \bar{\mathbf{y}} = \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \vdots \\ \bar{y}_L \end{bmatrix} \in [-1, 1]^L.$$



Rysunek 1.4: Statyczna sieć wielowarstwowa

Wektor parametrów  $k$ -tego neuronu  $j$ -tej warstwy przyjmuje postać:

$$\mathbf{w}_k^{(j)} = \begin{bmatrix} w_{k0}^{(j)} \\ w_{k1}^{(j)} \\ \vdots \\ w_{kL_{j-1}}^{(j)} \end{bmatrix} \in \mathcal{R}^{L_{j-1}+1}.$$

Parametry neuronów  $j$ -tej warstwy złożyć możemy w macierz postaci:

$$\mathbf{W}^{(j)} = \begin{bmatrix} \mathbf{w}_1^{(j)} & \mathbf{w}_2^{(j)} & \dots & \mathbf{w}_{L_j}^{(j)} \end{bmatrix} \in \mathcal{R}^{(L_{j-1}+1) \times L_j},$$

lub w wektor:

$$\mathbf{w}^{(j)} = \begin{bmatrix} \mathbf{w}_1^{(j)} \\ \mathbf{w}_2^{(j)} \\ \vdots \\ \mathbf{w}_{L_j}^{(j)} \end{bmatrix} \in \mathcal{R}^{(L_{j-1}+1) \cdot L_j}.$$



Wektor wszystkich parametrów sieci neuronowej ma postać:

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}^{(1)} \\ \mathbf{w}^{(2)} \\ \vdots \\ \mathbf{w}^{(J)} \end{bmatrix} \in \mathcal{R}^{\sum_{j=1}^J (L_{j-1}+1) \cdot L_j}. \quad (1.5)$$

Równanie (1.4) możemy teraz zapisać w równoważnej postaci:

$$\mu_k^{(j)} = \sigma \left( \left[ \mathbf{w}_k^{(j)} \right]^T \boldsymbol{\mu}^{(j-1)} \right). \quad (1.6)$$

Przyjmując dodatkowo oznaczenie wektorowej funkcji  $\sigma$ :

$$\sigma(\boldsymbol{\chi}^{(j)}) = \begin{bmatrix} 1 \\ \sigma(\chi_1^{(j)}) \\ \sigma(\chi_2^{(j)}) \\ \vdots \\ \sigma(\chi_{L_j}^{(j)}) \end{bmatrix},$$

działanie  $j$ -tej warstwy zapiszemy następująco:

$$\boldsymbol{\mu}^{(j)} = \sigma(\boldsymbol{\chi}^{(j)}) = \sigma \left( \left[ \mathbf{W}^{(j)} \right]^T \boldsymbol{\mu}^{(j-1)} \right), \quad (1.7)$$

a ponieważ warstwy ułożone są szeregowo, mamy kaskadę odwzorowań wektorowych:

$$\begin{aligned} \bar{\mathbf{y}} = \boldsymbol{\mu}^{(J)} &= \sigma \left( \left[ \mathbf{W}^{(J)} \right]^T \boldsymbol{\mu}^{(J-1)} \right), \\ \boldsymbol{\mu}^{(J-1)} &= \sigma \left( \left[ \mathbf{W}^{(J-1)} \right]^T \boldsymbol{\mu}^{(J-2)} \right), \\ &\vdots \\ \boldsymbol{\mu}^{(j)} &= \sigma \left( \left[ \mathbf{W}^{(j)} \right]^T \boldsymbol{\mu}^{(j-1)} \right), \\ &\vdots \\ \boldsymbol{\mu}^{(1)} &= \sigma \left( \left[ \mathbf{W}^{(1)} \right]^T \boldsymbol{\mu}^{(0)} \right) = \sigma \left( \left[ \mathbf{W}^{(1)} \right]^T \mathbf{u} \right). \end{aligned} \quad (1.8)$$

Działanie wielowarstwowej statycznej sieci neuronowej sprowadza się do odwzorowania postaci:

$$\bar{\mathbf{y}} = \sigma \left( \left[ \mathbf{W}^{(J)} \right]^T \sigma \left( \left[ \mathbf{W}^{(J-1)} \right]^T \sigma \left( \dots \sigma \left( \left[ \mathbf{W}^{(1)} \right]^T \mathbf{u} \right) \dots \right) \right) \right). \quad (1.9)$$

Jest to model neuronowy  $\Phi$ , którego wyjście  $\bar{y}$  zależy od wejścia  $\mathbf{u}$  oraz parametrów  $\mathbf{w}$ :

$$\bar{y} = \Phi(\mathbf{u}; \mathbf{w}^{(J)}, \mathbf{w}^{(J-1)}, \dots, \mathbf{w}^{(1)}) = \Phi(\mathbf{u}, \mathbf{w}). \quad (1.10)$$

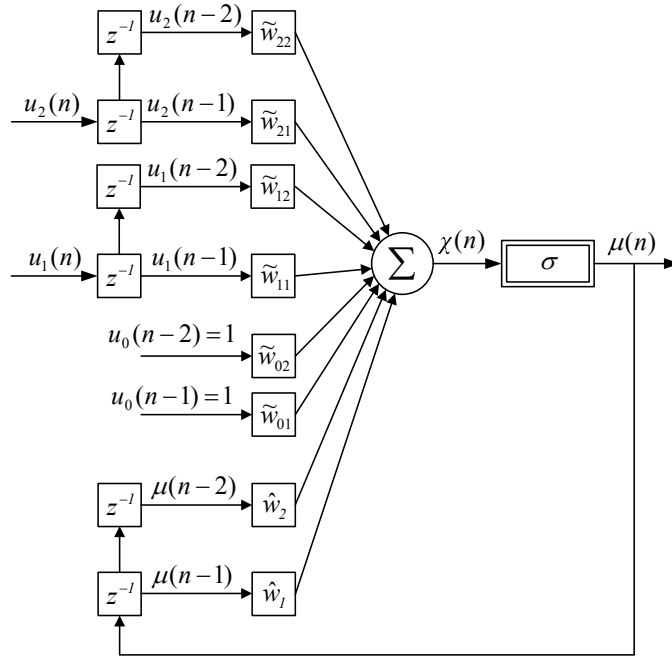
### 1.3.3 Opis neuronu rekurencyjnego

Neuron rekurencyjny o  $S$  wejściach i jednym wyjściu opisujemy równaniem różnicowym, [87]:

$$\mu(n) = \sigma(\chi(n)), \quad \chi(n) = \sum_{s=0}^S \sum_{d=1}^{D_u} \tilde{w}_{sd} u_s(n-d) + \sum_{d=1}^{D_\mu} \hat{w}_d \mu(n-d), \quad (1.11)$$

gdzie  $n$  – numer taktu,  $u_s(n)$  ( $s = 0, 1, \dots, S$ ) – wartość  $s$ -tego wejścia neuronu w  $n$ -tym takcie, przy czym  $u_0(n) = 1$  dla każdego  $n$ ,  $D_u$  – liczba kolejnych wartości wejść z poprzednich taktów brana pod uwagę przy wyliczaniu wartości sygnału wyjściowego  $\mu(n)$  neuronu,  $D_\mu$  – liczba kolejnych wartości wyjść z poprzednich taktów brana pod uwagę przy wyliczaniu wartości sygnału wyjściowego neuronu,  $\tilde{w}_{sd}$  ( $s = 0, 1, \dots, S$ ;  $d = 1, 2, \dots, D_u$ ) – parametr wagowy określający wpływ wartości  $s$ -tego wejścia sprzed  $d$  taktów na bieżącą wartość wyjścia neuronu,  $\hat{w}_d$  ( $d = 1, 2, \dots, D_\mu$ ) – parametr wagowy określający wpływ wartości wyjścia neuronu sprzed  $d$  taktów na bieżącą,  $n$ -tą wartość tego wyjścia.

Wagami spełniającymi rolę parametrów swobodnych części liniowej neuronu są  $\tilde{w}_{0d}$  ( $d = 1, 2, \dots, D_u$ ). Na rysunku 1.5 przedstawiono przykład neuronu rekurencyjnego o dwóch wejściach i dla  $D_u = D_\mu = 2$ . Operator opóźnienia  $z^{-d}$  z rysunku zdefiniowany jest jako  $z^{-d}u(n) \triangleq u(n-d)$ . Opisany neuron nazywany jest w literaturze modelem z dynamiką wewnętrzną, ponieważ – jak widać ze wzoru (1.11) oraz rysunku 1.5 – wartości poprzednich wyjść pobierane są z modelu a nie z obiektu poprzez tzw. linie opóźniające. Wyjście  $\mu(n)$  neuronu jest więc wyznaczane w odpowiedzi na sygnały  $u_s(n-1), u_s(n-2), \dots, u_s(n-D_u)$  ( $s = 0, 2, \dots, S$ ) oraz  $\mu(n-1), \mu(n-2), \dots, \mu(n-D_\mu)$ , czyli jest realizacją modelu NARX, [76], [87].


 Rysunek 1.5: Rekurencyjny neuron dla  $D_u = D_\mu = 2$ 

### 1.3.4 Diagonalna rekurencyjna sieć neuronowa (DRNN)

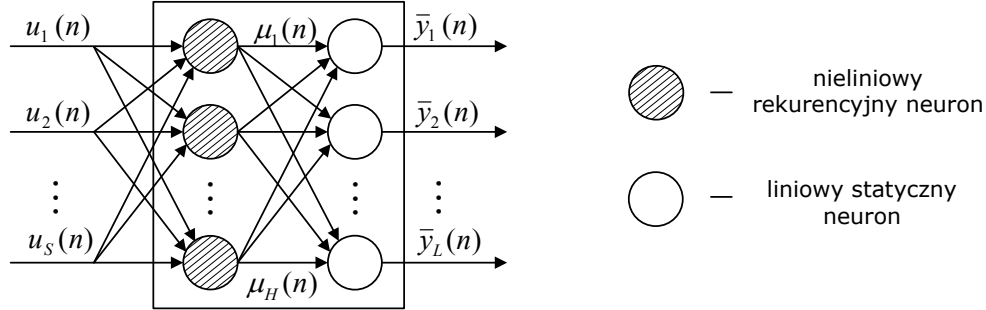
Sieć DRNN, [70], [77], [132], [140], (przedstawiona na rys. 1.6 z pominięciem oznaczeń numerów taktu) złożona ze statycznej warstwy wyjściowej  $L$  neuronów liniowych oraz warstwy ukrytej zawierającej  $H$  rekurencyjnych neuronów przedstawionych w poprzednim podpunkcie, realizuje odwzorowanie opisane następującym równaniem różnicowym:

$$\bar{y}_l(n) = \sum_{h=0}^H \bar{w}_{lh} \mu_h(n), \quad \mu_h(n) = \sigma(\chi_h(n)), \quad (1.12)$$

$$\chi_h(n) = \sum_{s=0}^S \sum_{d=1}^{D_u} \tilde{w}_{hsd} u_s(n-d) + \sum_{d=1}^{D_\mu} \hat{w}_{hd} \mu_h(n-d), \quad (1.13)$$

gdzie  $\bar{y}_l(n)$  ( $l = 1, 2, \dots, L$ ) – jest  $l$ -tym wyjściem modelu neuronowego w takcie  $n$ -tym, będące zarazem wyjściem  $l$ -tego neuronu warstwy wyjściowej,  $\mu_h(n)$  ( $h = 0, 1, \dots, H$ ) – wyjście  $h$ -tego neuronu warstwy ukrytej w  $n$ -tym takcie, przy czym  $\mu_0(n) = 1$  dla każdego  $n$ ,  $\bar{w}_{lh}$  ( $l = 1, 2, \dots, L$ ;  $h = 0, 1, \dots, H$ ) – waga połączenia między  $l$ -tym neuronem warstwy wyjściowej a  $h$ -tym neuronem warstwy ukrytej,  $\tilde{w}_{hsd}$  ( $h = 1, 2, \dots, H$ ;  $s = 0, 1, \dots, S$ ;  $d = 1, 2, \dots, D_u$ ) – waga określająca wpływ wartości  $s$ -tego wejścia sieci neuronowej sprzed  $d$  taktów na wyjście

$h$ -tego neuronu warstwy ukrytej,  $\hat{w}_{hd}$  ( $h = 1, 2, \dots, H$ ;  $d = 1, 2, \dots, D_\mu$ ) – waga określająca wpływ wartości wyjścia  $h$ -tego neuronu warstwy ukrytej sprzed  $d$  taktów na swoje własne wyjście w takcie bieżącym.



Rysunek 1.6: Rekurencyjna sieć DRNN

Wagi  $\bar{w}_{l0}$  ( $l = 1, 2, \dots, L$ ) oraz  $\tilde{w}_{h0d}$  ( $h = 1, 2, \dots, H$ ;  $d = 1, 2, \dots, D_u$ ) pełnią rolę parametrów swobodnych, odpowiednio części liniowej neuronów warstwy wyjściowej oraz części liniowej neuronów warstwy ukrytej.

Podobnie jak w przypadku statycznej sieci wielowarstwowej, zapiszemy odwzorowanie realizowane przez sieć rekurencyjną w formie wektorowej. Wprowadźmy oznaczenia wektorów:

$$\mathbf{u}(n) = \begin{bmatrix} 1 \\ u_1(n) \\ u_2(n) \\ \vdots \\ u_S(n) \end{bmatrix} \in \mathcal{R}^{S+1}, \quad \boldsymbol{\mu}(n) = \begin{bmatrix} 1 \\ \mu_1(n) \\ \mu_2(n) \\ \vdots \\ \mu_H(n) \end{bmatrix} \in [-1, 1]^{H+1},$$

$$\boldsymbol{\chi}(n) = \begin{bmatrix} \chi_1(n) \\ \chi_2(n) \\ \vdots \\ \chi_H(n) \end{bmatrix} \in \mathcal{R}^H, \quad \bar{\mathbf{y}}(n) = \begin{bmatrix} \bar{y}_1(n) \\ \bar{y}_2(n) \\ \vdots \\ \bar{y}_L(n) \end{bmatrix} \in \mathcal{R}^L.$$

Wektory wag  $l$ -tego neuronu warstwy wyjściowej i  $h$ -tego neuronu warstwy ukry-

tej to, odpowiednio:

$$\bar{\mathbf{w}}_l = \begin{bmatrix} \bar{w}_{l0} \\ \bar{w}_{l1} \\ \vdots \\ \bar{w}_{lH} \end{bmatrix} \in \mathcal{R}^{H+1}, \quad \hat{\mathbf{w}}_h = \begin{bmatrix} \hat{w}_{h1} \\ \hat{w}_{h2} \\ \vdots \\ \hat{w}_{hD_\mu} \end{bmatrix} \in \mathcal{R}^{D_\mu},$$

$$\tilde{\mathbf{w}}_h = \begin{bmatrix} \tilde{w}_{h0} \\ \tilde{w}_{h1} \\ \vdots \\ \tilde{w}_{hS} \end{bmatrix} \in \mathcal{R}^{(S+1) \cdot D_u}, \quad \text{gdzie} \quad \tilde{\mathbf{w}}_{hs} = \begin{bmatrix} \tilde{w}_{hs1} \\ \tilde{w}_{hs2} \\ \vdots \\ \tilde{w}_{hsD_u} \end{bmatrix} \in \mathcal{R}^{D_u}.$$

Wszystkie wagi warstwy wyjściowej i ukrytej można reprezentować przy użyciu poniższych wektorów:

$$\bar{\mathbf{w}} = \begin{bmatrix} \bar{\mathbf{w}}_1 \\ \bar{\mathbf{w}}_2 \\ \vdots \\ \bar{\mathbf{w}}_L \end{bmatrix} \in \mathcal{R}^{L \cdot (H+1)}, \quad \hat{\mathbf{w}} = \begin{bmatrix} \hat{\mathbf{w}}_1 \\ \hat{\mathbf{w}}_2 \\ \vdots \\ \hat{\mathbf{w}}_H \end{bmatrix} \in \mathcal{R}^{H \cdot D_\mu}, \quad \tilde{\mathbf{w}} = \begin{bmatrix} \tilde{\mathbf{w}}_1 \\ \tilde{\mathbf{w}}_2 \\ \vdots \\ \tilde{\mathbf{w}}_H \end{bmatrix} \in \mathcal{R}^{H \cdot (S+1) \cdot D_u},$$

a wówczas wszystkie parametry wagowe sieci można umieścić w jednym wektorze:

$$\mathbf{w} = \begin{bmatrix} \bar{\mathbf{w}} \\ \hat{\mathbf{w}} \\ \tilde{\mathbf{w}} \end{bmatrix} \in \mathcal{R}^{L \cdot (H+1) + H \cdot D_\mu + H \cdot (S+1) \cdot D_u}.$$

W celu zapisania działania sieci DRNN w formie odwzorowania wektorowego, wygodnie jest zebrać wagi warstwy wyjściowej oraz wagi warstwy ukrytej związane z wejściami sieci i wagi sprzężeń zwrotnych w osobne macierze:

$$\bar{\mathbf{W}} = \begin{bmatrix} \bar{w}_{10} & \bar{w}_{11} & \cdots & \bar{w}_{1H} \\ \bar{w}_{20} & \bar{w}_{21} & \cdots & \bar{w}_{2H} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{w}_{L0} & \bar{w}_{L1} & \cdots & \bar{w}_{LH} \end{bmatrix} \in \mathcal{R}^{L \times (H+1)},$$

$$\hat{\mathbf{W}} = \begin{bmatrix} \hat{w}_{11} & \hat{w}_{12} & \cdots & \hat{w}_{1D_\mu} \\ \hat{w}_{21} & \hat{w}_{22} & \cdots & \hat{w}_{2D_\mu} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{w}_{H1} & \hat{w}_{H2} & \cdots & \hat{w}_{HD_\mu} \end{bmatrix} \in \mathcal{R}^{H \times D_\mu},$$

$$\widetilde{\mathbf{W}} = \left[ \widetilde{\mathbf{W}}_1 \quad \widetilde{\mathbf{W}}_2 \quad \cdots \quad \widetilde{\mathbf{W}}_{D_u} \right] \in \mathcal{R}^{H \times ((S+1) \cdot D_u)},$$

gdzie podmacierze  $\widetilde{\mathbf{W}}_d$  ( $d = 1, 2, \dots, D_u$ ) zdefiniowane są jako:

$$\widetilde{\mathbf{W}}_d = \begin{bmatrix} \tilde{w}_{10d} & \tilde{w}_{11d} & \cdots & \tilde{w}_{1Sd} \\ \tilde{w}_{20d} & \tilde{w}_{21d} & \cdots & \tilde{w}_{2Sd} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{w}_{H0d} & \tilde{w}_{H1d} & \cdots & \tilde{w}_{HSd} \end{bmatrix} \in \mathcal{R}^{H \times (S+1)}.$$

Dodatkowo wprowadźmy pomocnicze wektor i macierz:

$$\tilde{\mathbf{u}}(n) = \begin{bmatrix} \mathbf{u}(n-1) \\ \mathbf{u}(n-2) \\ \vdots \\ \mathbf{u}(n-D_u) \end{bmatrix}, \quad \mathbf{\Upsilon}(n) = \begin{bmatrix} \mu_1(n-1) & \mu_1(n-2) & \cdots & \mu_1(n-D_\mu) \\ \mu_2(n-1) & \mu_2(n-2) & \cdots & \mu_2(n-D_\mu) \\ \vdots & \vdots & \ddots & \vdots \\ \mu_H(n-1) & \mu_H(n-2) & \cdots & \mu_H(n-D_\mu) \end{bmatrix}.$$

Zauważmy, że macierz  $\mathbf{\Upsilon}$  zbudowana jest z tych samych elementów, które wchodzi w skład wektora  $\tilde{\mathbf{u}}$ .

Teraz, posługując się oznaczeniem wektorowej funkcji  $\sigma$  z podpunktu 1.3.2, możemy zapisać obliczenia wykonywane przez sieć DRNN w postaci wektorowej:

$$\bar{\mathbf{y}}(n) = \overline{\mathbf{W}} \boldsymbol{\mu}(n), \quad \boldsymbol{\mu}(n) = \sigma(\boldsymbol{\chi}(n)), \quad (1.14)$$

$$\boldsymbol{\chi}(n) = \widetilde{\mathbf{W}} \tilde{\mathbf{u}}(n) + \text{tr}(\widehat{\mathbf{W}} \bullet [\mathbf{\Upsilon}(n)]^T), \quad (1.15)$$

gdzie  $\mathbf{A} \bullet \mathbf{B}$  – oznacza iloczyn Hadamarda macierzy  $\mathbf{A}$  i  $\mathbf{B}$ ,  $\text{tr}(\mathbf{A})$  – jest śladem macierzy  $\mathbf{A}$ , lub równoważnie:

$$\bar{\mathbf{y}}(n) = \overline{\mathbf{W}} \sigma \left( \widetilde{\mathbf{W}} \tilde{\mathbf{u}}(n) + \text{tr}(\widehat{\mathbf{W}} \bullet [\mathbf{\Upsilon}(n)]^T) \right). \quad (1.16)$$

Jest to model neuronowy  $\Phi$ , którego wyjście  $\bar{\mathbf{y}}(n)$  zależy od wejść  $\tilde{\mathbf{u}}(n)$  oraz parametrów  $\mathbf{w}$ :

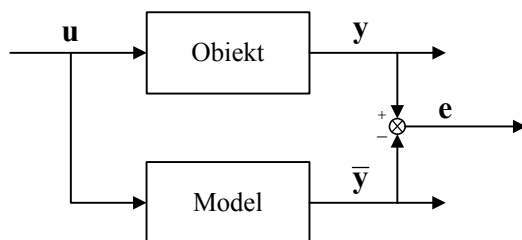
$$\bar{\mathbf{y}}(n) = \Phi \left( \tilde{\mathbf{u}}(n); \overline{\mathbf{W}}, \widetilde{\mathbf{W}}, \widehat{\mathbf{W}} \right) = \Phi(\tilde{\mathbf{u}}(n), \mathbf{w}). \quad (1.17)$$

## 1.4 Identyfikacja obiektu dynamicznego

Rysunek 1.7 przedstawia istotę identyfikacji obiektu wejściowo-wyjściowego, [113], [80]. Dany jest obiekt, w którym wyróżniono pewne wielkości wejściowe  $\mathbf{u}$  i wyjściowe  $\mathbf{y}$ . Następnie, na podstawie znajomości charakterystyki obiektu lub w oparciu o analizę, proponujemy – z dokładnością do parametrów  $\mathbf{w}$  – ogólną postać modelu obiektu:

$$\bar{\mathbf{y}}(n) = \Phi(\mathbf{u}(n-1), \dots, \mathbf{u}(n-D); \mathbf{w}),$$

(tzw. klasę modeli) gdzie  $D$  jest rzędem modelu. Sygnał  $\bar{\mathbf{y}}(n)$  jest wyjściem modelu. Identyfikacja polega na ustaleniu dokładnej postaci modelu  $\Phi$  na podstawie pomiarów wielkości  $\mathbf{u}(n)$  i  $\mathbf{y}(n)$  z obiektu, a konkretnie – na doborze wartości parametrów  $\mathbf{w}$  tego modelu w taki sposób aby, po podaniu na obiekt i model tego samego sygnału  $\mathbf{u}(n)$ , ocena różnicy  $\mathbf{e}(n)$  między sygnałem  $\mathbf{y}(n)$  zmierzonym na wyjściu obiektu a sygnałem  $\bar{\mathbf{y}}(n)$  wyliczonym z modelu była minimalna (dla zadanego zakresu wartości sygnału wejściowego).



Rysunek 1.7: Ogólny schemat identyfikacji

Przyjmijmy, że modelowany obiekt ma  $S$  wejść i  $L$  wyjść, czyli  $\mathbf{u}(n) \in \mathcal{R}^S$ ,  $\mathbf{y}(n), \bar{\mathbf{y}}(n) \in \mathcal{R}^L$ . Zakładamy dyskretny model obiektu\* dynamicznego, [84], [85], [40]. Wykonano  $N$  pomiarów wejść  $\mathbf{u}(n)$  i wyjść  $\mathbf{y}(n)$  obiektu ( $n = 1, 2, \dots, N$ ), które zebrano w macierzach:

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}(1) & \mathbf{u}(2) & \cdots & \mathbf{u}(N) \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} \mathbf{y}(1) & \mathbf{y}(2) & \cdots & \mathbf{y}(N) \end{bmatrix}.$$

przy czym znany jest stan początkowy obiektu  $\mathbf{u}(0)$  i  $\mathbf{y}(0)$ .

W przypadku eksperymentu aktywnego, tzn. gdy możliwe jest kontrolowanie wejść obiektu,  $\mathbf{U}$  nazywamy serią identyfikującą a  $\mathbf{Y}$  wynikiem eksperymentu, [130]. Na podstawie powyższych pomiarów można ocenić jakość modelu  $\Phi$  dla różnych zestawów wartości jego parametrów  $\mathbf{w}$ . Dla ustalonych wartości  $\mathbf{w}$

\*Rozważania dla modeli dynamicznych ciągłych oraz dla modeli statycznych są analogiczne.

wyznaczamy odpowiedzi modelu  $\bar{\mathbf{y}}(n)$  na *serię identyfikującą*  $\mathbf{U}$  i zbieramy je w macierzy

$$\bar{\mathbf{Y}} = \begin{bmatrix} \bar{\mathbf{y}}(1) & \bar{\mathbf{y}}(2) & \cdots & \bar{\mathbf{y}}(N) \end{bmatrix}.$$

Określić należy wskaźnik (kryterium) jakości identyfikacji  $Q$  o postaci:

$$Q(\mathbf{U}, \mathbf{Y}; \mathbf{w}) \triangleq \|\mathbf{Y} - \bar{\mathbf{Y}}\|, \quad (1.18)$$

gdzie  $\|\cdot\|$  oznacza pewną normę, a składowe macierzy  $\bar{\mathbf{Y}}$  wyznacza się z zależności

$$\bar{\mathbf{y}}(n) = \Phi(\mathbf{u}(n-1), \mathbf{u}(n-2), \dots, \mathbf{u}(n-D_u), \mathbf{y}(n-1), \mathbf{y}(n-2), \dots, \mathbf{y}(n-D_y); \mathbf{w}) \quad (1.19)$$

w przypadku modelu z dynamiką zewnętrzną [94], lub

$$\bar{\mathbf{y}}(n) = \Phi(\mathbf{u}(n-1), \mathbf{u}(n-2), \dots, \mathbf{u}(n-D_u), \bar{\mathbf{y}}(n-1), \bar{\mathbf{y}}(n-2), \dots, \bar{\mathbf{y}}(n-D_y); \mathbf{w}) \quad (1.20)$$

w przypadku modelu z dynamiką wewnętrzną. Wskaźnik jakości identyfikacji przyjmuje wartości rzeczywiste nieujemne. Wartość wskaźnika zależy od pomiarów oraz od przyjętych wartości parametrów modelu. W dalszej części pracy pomijając będziemy  $\mathbf{U}$  i  $\mathbf{Y}$  w zapisie  $Q$ , pisząc krótko –  $Q(\mathbf{w})$ , zamiast  $Q(\mathbf{U}, \mathbf{Y}; \mathbf{w})$ . Zwykle korzysta się ze wskaźnika jakości o postaci:

$$Q(\mathbf{U}, \mathbf{Y}; \mathbf{w}) \triangleq \sum_{n=1}^N q(\mathbf{y}(n), \bar{\mathbf{y}}(n)), \quad (1.21)$$

gdzie  $q(\mathbf{y}(n), \bar{\mathbf{y}}(n))$  jest funkcją określającą różnicę między  $\mathbf{y}(n)$  a  $\bar{\mathbf{y}}(n)$ , np. normą Euklidesową wektora  $\mathbf{y}(n) - \bar{\mathbf{y}}(n)$ . Zadanie identyfikacji polega na znalezieniu najlepszego modelu z zadanej klasy, co sprowadza się do wyznaczenia takich wartości parametrów modelu, dla których wskaźnik jakości identyfikacji przyjmuje wartość minimalną. Dokładniej, dane są:

- wskaźnik jakości identyfikacji  $Q$ ;
- postać modelu  $\Phi$  z dokładnością do parametrów  $\mathbf{w}$ ;
- seria pomiarowa  $\mathbf{U}, \mathbf{Y}$ .

Wyznaczyć należy optymalny zestaw  $\mathbf{w}^*$  parametrów modelu, czyli taki, dla którego:

$$Q(\mathbf{w}^*) = \min_{\mathbf{w}} Q(\mathbf{w}). \quad (1.22)$$



Algorytm identyfikacji  $\Psi$ , będący wynikiem rozwiązania powyższego zadania ma postać:

$$\mathbf{w}^* = \Psi(\mathbf{U}, \mathbf{Y}). \quad (1.23)$$

Podkreślmy, że wyznaczone wartości parametrów  $\mathbf{w}^*$  są optymalne dla przyjętego kryterium i zmierzonych sygnałów  $\mathbf{U}, \mathbf{Y}$ . Powtarzając pomiary na tym samym obiekcie, możemy w rezultacie otrzymać nieco inne wyniki algorytmu identyfikacji. Dobór serii identyfikującej  $\mathbf{U}$  ma istotny wpływ na jakość uzyskanego modelu. Zagadnieniu temu poświęcono wiele prac z dziedziny identyfikacji, np. [83], [46]. W niniejszej rozprawie nie będzie ono poruszane.

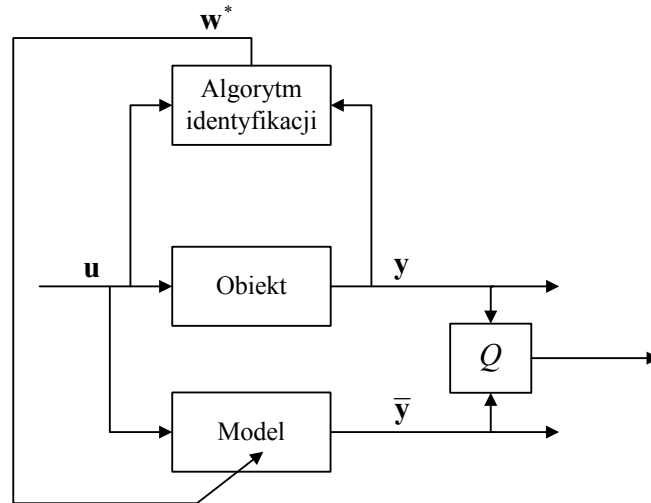
W przypadku modeli, których parametry związane są w sposób nieliniowy lub dla których kryterium jakości jest funkcją wielomodalną, uzyskanie optymalnych wartości parametrów  $\mathbf{w}^*$  może okazać się niemożliwe. Należy wówczas posłużyć się numerycznym algorytmem w celu przybliżenia ich optymalnych wartości, [42], [136], [139], [9], [96]. Gdy klasą modeli jest sieć neuronowa, proces identyfikacji nazywamy uczeniem sieci.

Możliwe jest również wyznaczanie wartości parametrów modelu nie na podstawie przygotowanej uprzednio serii pomiarowej, lecz w trakcie działania obiektu. Wówczas po upływie określonej liczby taktów (często co jeden takt) uaktualnia się wartości parametrów modelu uzyskane w poprzednim kroku na podstawie nowej pary pomiarów  $(\mathbf{u}(n), \mathbf{y}(n))$ . Algorytm identyfikacji na bieżąco, [19], [141] (*on-line*) obiektu dynamicznego przyjmuje w takim przypadku postać:

$$\mathbf{w}(n) = \Psi(\mathbf{u}(n), \mathbf{y}(n); \mathbf{w}(n-1)), \quad (1.24)$$

przy czym początkowy zestaw wartości  $\mathbf{w}(0)$  jest wybierany dowolnie (najczęściej losowo), natomiast we wskaźniku jakości (1.18) mamy  $N = 1$ . Dla obiektu stacjonarnego oraz przebiegu sygnału wejściowego spełniającego warunki określone w literaturze anglojęzycznej nazwą *persistence of excitation conditions*, [1], [58], ciąg kolejnych wartości  $\mathbf{w}(n)$  jest zbieżny do  $\mathbf{w}^*$ . W przypadku wielomodalnej funkcji kryterialnej mówić możemy, oczywiście, jedynie o zbieżności lokalnej. Algorytmy uczenia sieci neuronowych opracowane i przedstawione w ramach pracy są algorytmami identyfikacji na bieżąco. Niemniej, można je stosować również do uczenia *off-line*, tj. w sytuacji gdy wszystkie pomiary dane są w formie serii pomiarowej. Wystarczy w tym celu wybierać pojedyncze pomiary z serii i używać algorytmu tak, jakby były to pomiary pojawiające się na bieżąco.

Obiekt identyfikacji, model oraz algorytm identyfikacji tworzą system identyfikacji zilustrowany na rysunku 1.8.



Rysunek 1.8: System identyfikacji

W kolejnych podpunktach przedstawiono numeryczne algorytmy identyfikacji obiektu dynamicznego z wykorzystaniem jako klasy modeli:

- statycznej sieci wielowarstwowej, opisanej w podpunkcie 1.3.2;
- klasycznego modelu liniowego dynamicznego z jednym wejściem i jednym wyjściem, [56];
- rekurencyjnej sieci neuronowej.

Styczna sieć wielowarstwowa jest przykładem modelu z dynamiką zewnętrzną. Kolejne dwa modele posiadają dynamikę wewnętrzną, co komplikuje postać algorytmu identyfikacji. Użycie drugiego z wymienionych modeli pozwoli na czytelne przedstawienie głównej idei algorytmu doboru parametrów modelu z dynamiką wewnętrzną. Natomiast z modeli w postaci sieci DRNN konstruowany będzie w rozdziale 2 model systemu złożonego.

#### 1.4.1 Algorytm uczenia statycznej sieci neuronowej

Rozważmy obiekt dynamiczny o  $S$  wejściami i  $L$  wyjściami, modelowany przy użyciu statycznej sieci wielowarstwowej. Od modelu wymagamy, żeby uwzględniał wpływ  $D_u$  poprzednich wejść i  $D_y$  poprzednich wyjść obiektu na bieżące wyjście.

Aby sieć statyczna mogła modelować charakterystyki dynamicznego obiektu, konieczne jest podawanie na jej wejścia wartości wszystkich sygnałów z wejścia obiektu z momentów czasowych od  $n-1$  do  $n-D_u$  oraz z wyjścia obiektu z momentów czasowych od  $n-1$  do  $n-D_y$ , gdzie  $n$  jest bieżącym taktom, [74], [59], [61]. Rys. 1.9 przedstawia przykład takiego modelu neuronowego dla  $D_u = 3, D_y = 2$ :

$$\bar{\mathbf{y}}(n) = \Phi(\mathbf{u}(n-1), \mathbf{u}(n-2), \mathbf{u}(n-3), \mathbf{y}(n-1), \mathbf{y}(n-2); \mathbf{w}), \quad (1.25)$$

przy czym wektor parametrów  $\mathbf{w}$  jest postaci (1.5).

Oznaczenia w niniejszym podpunkcie są takie, jak wprowadzone w podpunkcie 1.3.2. Na rysunku linią przerywaną zaznaczono linie opóźniające na zewnątrz sieci dla podkreślenia faktu, że jest to model z dynamiką zewnętrzną. Człon  $\mathbf{z}^{-1}$  jest wielowymiarowym odpowiednikiem operatora opóźnienia  $z^{-1}$ .

Wszystkie wejścia sieci neuronowej można teraz zebrać w jeden pomocniczy wektor:

$$\mathbf{u}_E(n) = \begin{bmatrix} \bar{\mathbf{y}}(n-1) \\ \bar{\mathbf{y}}(n-2) \\ \vdots \\ \bar{\mathbf{y}}(n-D_y) \\ \mathbf{u}(n-1) \\ \mathbf{u}(n-2) \\ \vdots \\ \mathbf{u}(n-D_u) \end{bmatrix} \in \mathcal{R}^{\bar{S}}, \quad \bar{S} = L \cdot D_y + S \cdot D_u = L_0.$$

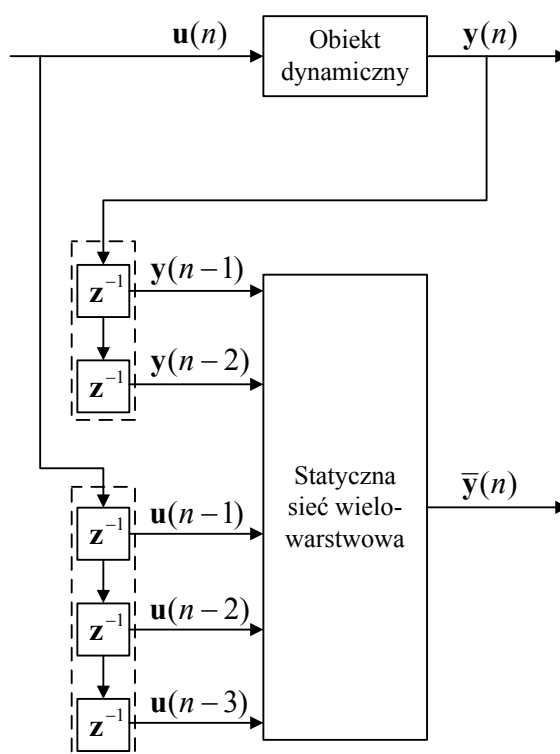
Jako model obiektu dynamicznego o wejściach  $\mathbf{u}$  i wyjściach  $\mathbf{y}$  proponujemy więc model statyczny o wejściach  $\mathbf{u}_E$  i wyjściach  $\bar{\mathbf{y}}$ :

$$\bar{\mathbf{y}}(n) = \Phi(\mathbf{u}_E(n), \mathbf{w}). \quad (1.26)$$

Mając do dyspozycji w każdym  $n$ -tym taktom dane pomiarowe  $\mathbf{u}_E(n)$ ,  $\bar{\mathbf{y}}(n)$  oraz przyjmując kwadratowy wskaźnik jakości identyfikacji:

$$Q(n, \mathbf{w}) = \frac{1}{2} \sum_{l=1}^L [e_l(n)]^2, \quad e_l(n) = y_l(n) - \bar{y}_l(n), \quad (1.27)$$

należy skonstruować algorytm uczenia sieci neuronowej, dający w wyniku wartości parametrów sieci będące przybliżeniem optymalnego zestawu  $\mathbf{w}^*$ , tj. takiego, dla którego wskaźnik (1.27) przyjmuje wartość najmniejszą. Ze względu na nieliniowość modelu neuronowego oraz – będącą tego konsekwencją – wielomodalność funkcji kryterialnej (1.27), dokładny rezultat jest na ogół nieosiągalny, [111].



Rysunek 1.9: Stacyczna sieć wielowarstwowa modelująca obiekt dynamiczny

Dlatego do rozwiązania postawionego zadania trzeba użyć numerycznej metody optymalizacji. Tutaj posłużymy się metodą gradientu prostego, [22], w której początkowy zestaw wartości wag sieci  $\mathbf{w}(0)$  jest aktualizowany każdorazowo po przetworzeniu kolejnej pary pomiarowej  $(\mathbf{u}_E(n), \mathbf{y}(n))$ , zgodnie z iteracyjną formułą, [112]:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \eta \nabla_{\mathbf{w}} Q(n, \mathbf{w})|_{\mathbf{w}=\mathbf{w}(n)}, \quad (1.28)$$

gdzie  $\eta$  jest parametrem algorytmu określającym wielkość poprawki w jednym kroku, nazywanym współczynnikiem uczenia. Dobór wartości  $\eta$  ma decydujący wpływ na zbieżność algorytmu uczenia, [138], [11]. Wzór (1.28) można zapisać dla każdej wagi oddzielnie:

$$w_{ki}^{(j)}(n+1) = w_{ki}^{(j)}(n) - \eta \left. \frac{\partial Q(n, \mathbf{w})}{\partial w_{ki}^{(j)}} \right|_{w_{ki}^{(j)} = w_{ki}^{(j)}(n)}, \quad (1.29)$$

przy czym  $k = 1, 2, \dots, L_j$ ,  $i = 1, 2, \dots, L_{j-1}$ ,  $j = 1, 2, \dots, J$ .

Korzystając z metody gradientu prostego, w kolejnych krokach działania algorytmu otrzymujemy ciąg wartości parametrów sieci neuronowej  $\mathbf{w}(1), \mathbf{w}(2), \dots$ ,

które – zakładając, że obiekt jest stacjonarny – po pewnej liczbie kroków uczenia są przybliżeniem optymalnych wartości  $\mathbf{w}^*$ .

W celu uzyskania kompletnej postaci algorytmu uczenia sieci, trzeba wyznaczyć gradient  $\nabla_{\mathbf{w}}Q(n, \mathbf{w})$  występujący we wzorze (1.28) co jest równoważne wyliczeniu pochodnych cząstkowych  $\frac{\partial Q(n, \mathbf{w})}{\partial w_{ki}^{(j)}}$  ze wzoru (1.29). Ogólna postać tych pochodnych jest następująca:

$$\frac{\partial Q(n, \mathbf{w})}{\partial w_{ki}^{(j)}} = \frac{1}{2} \sum_{l=1}^L 2e_l(n) \frac{\partial e_l(n)}{\partial w_{ki}^{(j)}}. \quad (1.30)$$

Wiedząc, że z (1.27) wynika równość:

$$\frac{\partial e_l(n)}{\partial w_{ki}^{(j)}} = -\frac{\partial \bar{y}_l(n)}{\partial w_{ki}^{(j)}}, \quad (1.31)$$

możemy ogólną postać pochodnych cząstkowych zapisać jako:

$$\frac{\partial Q(n, \mathbf{w})}{\partial w_{ki}^{(j)}} = -\sum_{l=1}^L e_l(n) \frac{\partial \bar{y}_l(n)}{\partial w_{ki}^{(j)}}. \quad (1.32)$$

Następnie wyznaczmy pochodne cząstkowe w warstwie wyjściowej:

$$\frac{\partial \bar{y}_k(n)}{\partial w_{ki}^{(j)}} = \frac{d\sigma(\chi_k^{(j)}(n))}{d\chi_k^{(j)}(n)} \frac{\partial \chi_k^{(j)}(n)}{\partial w_{ki}^{(j)}} = \sigma'(\chi_k^{(j)}(n)) \cdot \mu_i^{(j-1)}(n). \quad (1.33)$$

Pochodne cząstkowe w  $j$ -tej warstwie ( $j = 1, 2, \dots, J-1$ ) wyprowadzimy w drodze następujących przekształceń:

$$\begin{aligned} \frac{\partial \bar{y}_k(n)}{\partial w_{ki}^{(j)}} &= \frac{d\sigma(\chi_l^{(j)}(n))}{d\chi_l^{(j)}(n)} \frac{\partial \chi_l^{(j)}(n)}{\partial w_{ki}^{(j)}} = \sigma'(\chi_l^{(j)}(n)) \sum_{l_{j-1}=1}^{L_{j-1}} w_{l, l_{j-1}}^{(j)} \frac{\partial \mu_{l_{j-1}}^{(j-1)}(n)}{\partial w_{ki}^{(j)}} = \\ &= \sigma'(\chi_l^{(j)}(n)) \sum_{l_{j-1}=1}^{L_{j-1}} w_{l, l_{j-1}}^{(j)} \frac{d\sigma(\chi_{l_{j-1}}^{(j-1)}(n))}{d\chi_{l_{j-1}}^{(j-1)}(n)} \frac{\partial \chi_{l_{j-1}}^{(j-1)}(n)}{\partial w_{ki}^{(j)}}, \end{aligned} \quad (1.34)$$

gdzie:

$$\frac{\partial \chi_{l_{j-1}}^{(j-1)}(n)}{\partial w_{ki}^{(j)}} = \sum_{l_{j-2}=1}^{L_{j-2}} w_{l_{j-1}, l_{j-2}}^{(j-1)} \frac{d\sigma(\chi_{l_{j-2}}^{(j-2)}(n))}{d\chi_{l_{j-2}}^{(j-2)}(n)} \frac{\partial \chi_{l_{j-2}}^{(j-2)}(n)}{\partial w_{ki}^{(j)}} \quad (1.35)$$

Wstawiając (1.35) do (1.34) oraz kontynuując ten sposób przekształceń aż do warstwy  $j$ -tej, otrzymujemy:

$$\begin{aligned} \frac{\partial \bar{y}_l(n)}{\partial w_{ki}^{(j)}(n)} &= \sigma'(\chi_l^{(J)}(n)) \sum_{l_{J-1}=1}^{L_{J-1}} w_{l,l_{J-1}}^{(J)} \sigma'(\chi_{l_{J-1}}^{(J-1)}(n)) \sum_{l_{J-2}=1}^{L_{J-2}} w_{l_{J-1},l_{J-2}}^{(J-1)} \cdot \\ &\cdot \sigma'(\chi_{l_{J-2}}^{(J-2)}(n)) \cdot \dots \cdot \sum_{l_{j+1}=1}^{L_{j+1}} w_{l_{j+2},l_{j+1}}^{(j+2)} \frac{d\sigma(\chi_{l_{j+1}}^{(j+1)}(n))}{d\chi_{l_{j+1}}^{(j+1)}(n)} \frac{\partial \chi_{l_{j+1}}^{(j+1)}(n)}{\partial w_{ki}^{(j)}}, \end{aligned} \quad (1.36)$$

gdzie:

$$\begin{aligned} \frac{\partial \chi_{l_{j+1}}^{(j+1)}(n)}{\partial w_{ki}^{(j)}} &= \sum_{l_j=1}^{L_j} w_{l_{j+1},l_j}^{(j+1)} \frac{\partial \mu_{l_j}^{(j)}(n)}{\partial w_{ki}^{(j)}} = \sum_{l_j=1}^{L_j} w_{l_{j+1},l_j}^{(j+1)} \frac{d\sigma(\chi_{l_j}^{(j)}(n))}{d\chi_{l_j}^{(j)}(n)} \frac{\partial \chi_{l_j}^{(j)}(n)}{\partial w_{ki}^{(j)}} = \\ &= w_{l_{j+1},k}^{(j+1)} \cdot \sigma'(\chi_k^{(j)}(n)) \frac{\partial \chi_k^{(j)}(n)}{\partial w_{ki}^{(j)}} = w_{l_{j+1},k}^{(j+1)} \cdot \sigma'(\chi_k^{(j)}(n)) \mu_i^{(j-1)}(n). \end{aligned} \quad (1.37)$$

Podsumowując, pochodne cząstkowe w  $j$ -tej warstwie, przy czym jest to warstwa ukryta ( $j = 1, 2, \dots, J - 1$ ), mają postać:

$$\begin{aligned} \frac{\partial \bar{y}_k(n)}{\partial w_{ki}^{(j)}} &= \sigma'(\chi_l^{(J)}(n)) \sum_{l_{J-1}=1}^{L_{J-1}} w_{l,l_{J-1}}^{(J)} \sigma'(\chi_{l_{J-1}}^{(J-1)}(n)) \sum_{l_{J-2}=1}^{L_{J-2}} w_{l_{J-1},l_{J-2}}^{(J-1)} \sigma'(\chi_{l_{J-2}}^{(J-2)}(n)) \cdot \\ &\cdot \dots \cdot \sum_{l_{j+1}=1}^{L_{j+1}} w_{l_{j+2},l_{j+1}}^{(j+2)} \sigma'(\chi_{l_{j+1}}^{(j+1)}(n)) w_{l_{j+1},k}^{(j+1)} \cdot \sigma'(\chi_k^{(j)}(n)) \mu_i^{(j-1)}(n). \end{aligned} \quad (1.38)$$

Po wstawieniu formuł (1.33) i (1.38) do (1.32), a rezultaty tych podstawień do (1.29), otrzymamy dwa wzory – jeden dla neuronów warstwy wyjściowej, drugi dla neuronów warstwy ukrytej. W zasadzie dają nam one kompletną postać algorytmu uczenia statycznej sieci wielowarstwowej. Istnienie odrębnego wzoru dla korekty wag warstwy wyjściowej wynika z faktu, że dla wyjść neuronów tej warstwy w sposób bezpośredni zadane są wymagane wartości, równe zmierzonym wyjściom obiektu. Dla warstw ukrytych (o numerach od 1 do  $J - 1$ ) takich bezpośrednich wymagań nie ma.

Korzystanie wprost z wyprowadzonych wzorów może wymagać dosyć dużego nakładu obliczeniowego, ponieważ niektóre wyrażenia obliczane są w nich kilkakrotnie. W celu zminimalizowania liczby operacji obliczeniowych, przekształcimy uzyskane formuły, wprowadzając oznaczenia:

$$\delta_k^{(J)}(n) = \sigma' \left( \chi_k^{(J)}(n) \right) e_k(n), \quad (1.39)$$

$$\delta_k^{(j)}(n) = \sigma' \left( \chi_k^{(j)}(n) \right) \sum_{l_{j+1}=1}^{L_{j+1}} \delta_{l_{j+1}}^{(j+1)}(n) w_{l_{j+1},k}^{(j+1)}. \quad (1.40)$$

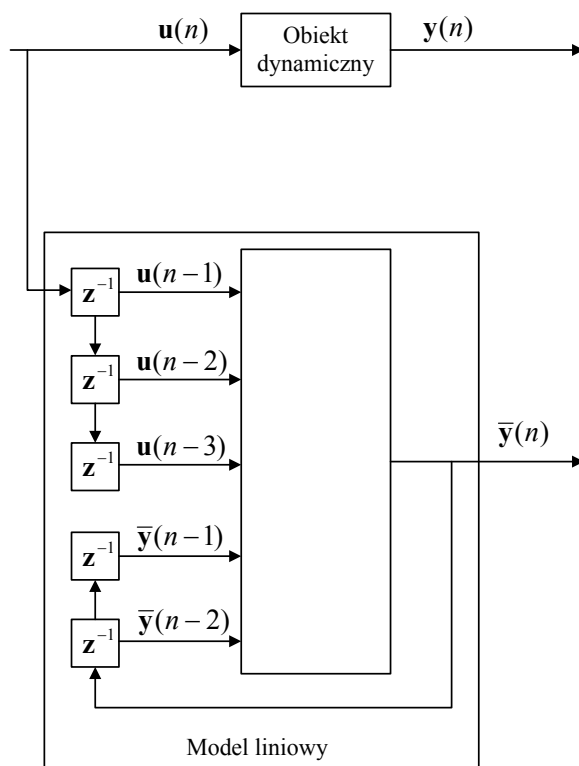
Teraz wzory (1.33) i (1.38) upraszczają się do tego stopnia, że formułę uczenia (1.29) można zapisać w prostej postaci (zwanej często *regulą delta*, [126], [97], [75]):

$$w_{ki}^{(j)}(n+1) = w_{ki}^{(j)}(n) - \eta \cdot \delta_k^{(j)}(n) \cdot \mu_i^{(j-1)}(n). \quad (1.41)$$

Zaletą tego zapisu jest możliwość równoległego prowadzenia obliczeń w neuronach poszczególnych warstw sieci w trakcie uczenia. Wystarczy w grafie z rys. 1.4, reprezentującym proces obliczania sygnałów wyjściowych sieci, odwrócić kierunek prowadzenia obliczeń, na wejścia odwróconego grafu podać błędy  $e_l(n)$  ( $l = 1, 2, \dots, L$ ), rozgałęzienia zastąpić elementami sumacyjnymi, elementy sumacyjne rozgałęzieniami, nieliniowości (1.2) ich pochodnymi (1.3), aby na węzłach odwróconego grafu odczytać wartości wszystkich zmiennych  $\delta_k^{(j)}$  ( $k = 1, 2, \dots, L_j$ ;  $j = 1, 2, \dots, J$ ), [97]. Stąd interpretacja wielkości  $\delta_k^{(j)}$  jako błędów propagowanych z neuronów warstwy wyjściowej na neurony wszystkich warstw ukrytych. Przedstawiony algorytm uczenia nosi nazwę „metody wstecznej propagacji błędów”, [126], (BP – *BackPropagation*).

#### 1.4.2 Algorytm identyfikacji z wykorzystaniem modelu liniowego

Ideę dwóch podstawowych algorytmów uczenia rekurencyjnych sieci neuronowych przedstawimy na przykładzie identyfikacji obiektu dynamicznego o jednym wejściu i jednym wyjściu z wykorzystaniem klasycznego modelu liniowego z dynamiką wewnętrzną, [56]. Bieżąca wartość wyjścia modelu wyliczana jest na podstawie  $D_y$  swoich poprzednich wyjść oraz  $D_u$  poprzednich wejść modelu. Na rysunku 1.10 pokazano przykładowy wielowymiarowy model liniowy dla  $D_u = 3$ ,  $D_y = 2$ . W odróżnieniu od modelu o dynamice zewnętrznej z rys. 1.9, człony opóźniające zamknięte są wewnątrz modelu.


 Rysunek 1.10: Model liniowy z dynamiką wewnętrzną dla  $D_u = 3$  i  $D_\mu = 2$ 

Równanie różnicowe opisujące model liniowy SISO z dynamiką wewnętrzną jest następujące:

$$\bar{y}(n) = \sum_{d=1}^{D_u} \tilde{w}_d u(n-d) + \sum_{d=1}^{D_y} \hat{w}_d \bar{y}(n-d). \quad (1.42)$$

Parametry modelu można umieścić w wektorze  $\mathbf{w}$ :

$$\mathbf{w} = \left[ \tilde{w}_1 \quad \tilde{w}_2 \quad \dots \quad \tilde{w}_{D_u} \quad \hat{w}_1 \quad \hat{w}_2 \quad \dots \quad \hat{w}_{D_y} \right]^T \in \mathcal{R}^{D_u + D_y}.$$

Przyjmując, że w każdym,  $n$ -tym takcie uzyskujemy nowe pomiary  $(u(n), y(n))$  z obiektu oraz biorąc kwadratowy wskaźnik jakości identyfikacji:

$$Q(n, \mathbf{w}) = \frac{1}{2} [e(n)]^2, \quad e(n) = y(n) - \bar{y}(n), \quad (1.43)$$

konstruujemy algorytm doboru parametrów modelu (1.42) w drodze kolejnych przybliżeń, korzystając – podobnie jak w podpunkcie 1.4.1 – z metody gradientu prostego, opisanej wzorem (1.28). Oznaczając przez  $w_b$  ( $b = 1, 2, \dots, D_u + D_y$ ) elementy wektora  $\mathbf{w}$ , wzór (1.28) można zapisać dla pojedynczej wagi:

$$w_b(n+1) = w_b(n) - \eta \left. \frac{\partial Q(n, \mathbf{w})}{\partial w_b} \right|_{w_b=w_b(n)}, \quad (1.44)$$



Gradient z równania (1.28) przyjmuje ogólną postać:

$$\nabla_{\mathbf{w}}Q(n, \mathbf{w}) = e(n) \cdot \nabla_{\mathbf{w}}e(n) = -e(n) \cdot \nabla_{\mathbf{w}}\bar{y}(n), \quad (1.45)$$

a jego  $b$ -ta składowa jest równa:

$$\frac{\partial Q(n, \mathbf{w})}{\partial w_b} = -e(n) \frac{\partial \bar{y}(n)}{\partial w_b}. \quad (1.46)$$

Na potrzeby dalszych rozważań przyjmijmy uproszczoną notację:

$$\frac{\partial \bar{y}(n)}{\partial w_b(n)} \triangleq \left. \frac{\partial \bar{y}(n)}{\partial w_b} \right|_{w_b=w_b(n)}. \quad (1.47)$$

Zapis ten w czytelny sposób wskazuje na moment czasowy  $n$ , z którego wartość danego parametru jest używana do wyliczenia wartości pochodnych cząstkowych po tym parametrze.

Dla wag typu  $\tilde{w}_i$  ( $i = 1, 2, \dots, D_u$ ) policzmy wyrażenia  $\frac{\partial \bar{y}(n)}{\partial \tilde{w}_i(n)}$ :

$$\frac{\partial \bar{y}(n)}{\partial \tilde{w}_i(n)} = u(n-i) + \sum_{d=1}^{D_y} \hat{w}_d \frac{\partial \bar{y}(n-d)}{\partial \tilde{w}_i(n)}, \quad (1.48)$$

oraz dla wag typu  $\hat{w}_i$  ( $i = 1, 2, \dots, D_y$ ) policzmy wyrażenia  $\frac{\partial \bar{y}(n)}{\partial \hat{w}_i(n)}$ :

$$\frac{\partial \bar{y}(n)}{\partial \hat{w}_i(n)} = \bar{y}(n-i) + \sum_{d=1}^{D_y} \hat{w}_d \frac{\partial \bar{y}(n-d)}{\partial \hat{w}_i(n)}. \quad (1.49)$$

Pojawia się tu problem z wyliczaniem pochodnych cząstkowych  $\frac{\partial \bar{y}(n-d)}{\partial \tilde{w}_i(n)}$  i  $\frac{\partial \bar{y}(n-d)}{\partial \hat{w}_i(n)}$ . Rekurencyjne wywołania w formułach (1.48) i (1.49) kończą się dopiero na warunkach początkowych  $\frac{\partial \bar{y}(0)}{\partial \tilde{w}_i(n)} = 0$  i  $\frac{\partial \bar{y}(0)}{\partial \hat{w}_i(n)} = 0$ . Oznacza to, że do uzyskania dokładnych wartości  $\frac{\partial \bar{y}(n)}{\partial \tilde{w}_i(n)}$  i  $\frac{\partial \bar{y}(n)}{\partial \hat{w}_i(n)}$  trzeba by w każdym takcie identyfikacji wyliczać wartości pochodnych cząstkowych z wyjść modelu we wszystkich poprzedzających momentach czasowych, po bieżących wartościach parametrów:  $\frac{\partial \bar{y}(n-1)}{\partial \tilde{w}_i(n)}$ ,  $\frac{\partial \bar{y}(n-2)}{\partial \tilde{w}_i(n)}$ ,  $\dots$ ,  $\frac{\partial \bar{y}(1)}{\partial \tilde{w}_i(n)}$  oraz  $\frac{\partial \bar{y}(n-1)}{\partial \hat{w}_i(n)}$ ,  $\frac{\partial \bar{y}(n-2)}{\partial \hat{w}_i(n)}$ ,  $\dots$ ,  $\frac{\partial \bar{y}(1)}{\partial \hat{w}_i(n)}$ . Wymaga to bardzo dużego nakładu obliczeniowego, rosnącego wraz z każdym taktem identyfikacji. Problem staje się jeszcze poważniejszy w przypadku wielowymiarowym. W literaturze, [87], [56], [60], [94], znanych jest kilka podejść pozwalających uzyskać przybliżone wartości pochodnych cząstkowych  $\frac{\partial \bar{y}(n)}{\partial \tilde{w}_i(n)}$  i  $\frac{\partial \bar{y}(n)}{\partial \hat{w}_i(n)}$ . Przedstawimy tu dwa podstawowe, na których bazują algorytmy „rekurencyjnego uczenia w czasie rzeczywistym” (RTRL – *Real Time Recurrent Learning*) i „wstecznej propagacji błędów w czasie” (BPTT – *BackPropagation Through Time*).

**RTRL.** Podejście, na którym opiera się algorytm uczenia RTRL, [137], oparte jest na założeniu, że parametry modelu nie podlegają zbyt dużym zmianom w trakcie uczenia, tj.:

$$\tilde{w}_i(n) \approx \tilde{w}_i(n-1) \approx \tilde{w}_i(n-2) \approx \dots \approx \tilde{w}_i(n-D_u), \quad (1.50)$$

$$\hat{w}_i(n) \approx \hat{w}_i(n-1) \approx \hat{w}_i(n-2) \approx \dots \approx \hat{w}_i(n-D_y), \quad (1.51)$$

co uprawnia do napisania:

$$\frac{\partial \bar{y}(n-d)}{\partial \tilde{w}_i(n)} \approx \frac{\partial \bar{y}(n-d)}{\partial \hat{w}_i(n-d)}. \quad (1.52)$$

Powyższe przybliżenie zachodzi przynajmniej dla wartości  $d = 1, 2, \dots, D_u$  w przypadku parametrów  $\tilde{w}_i$  związanych z wejściem oraz dla wartości  $d = 1, 2, \dots, D_y$  dla parametrów  $\hat{w}_i$  sprzężenia zwrotnego. Pozwala to zapisać wzory (1.48) i (1.49) jako

$$\frac{\partial \bar{y}(n)}{\partial \tilde{w}_i(n)} \approx u(n-i) + \sum_{d=1}^{D_y} \hat{w}_d \frac{\partial \bar{y}(n-d)}{\partial \tilde{w}_i(n-d)}, \quad (1.53)$$

$$\frac{\partial \bar{y}(n)}{\partial \hat{w}_i(n)} \approx \bar{y}(n-i) + \sum_{d=1}^{D_y} \hat{w}_d \frac{\partial \bar{y}(n-d)}{\partial \hat{w}_i(n-d)}. \quad (1.54)$$

Po przyjęciu oznaczeń

$$\Lambda_i(n) \triangleq \frac{\partial \bar{y}(n)}{\partial \tilde{w}_i(n)}, \quad \Gamma_i(n) \triangleq \frac{\partial \bar{y}(n)}{\partial \hat{w}_i(n)}, \quad (1.55)$$

wzory (1.53) i (1.54) można zapisać w postaci:

$$\Lambda_i(n) \approx u(n-i) + \sum_{d=1}^{D_y} \hat{w}_d \Lambda_i(n-d), \quad (1.56)$$

$$\Gamma_i(n) \approx \bar{y}(n-i) + \sum_{d=1}^{D_y} \hat{w}_d \Gamma_i(n-d), \quad (1.57)$$

przy czym  $\Lambda_i(0) = 0$  i  $\Gamma_i(0) = 0$ . Po wstawieniu formuł (1.56) i (1.57) do (1.46), a rezultat tych podstawień dalej do (1.44), otrzymujemy numeryczny algorytm identyfikacji dynamicznego obiektu SISO.

Wzory (1.56) i (1.57) można uważać za realizację pewnego procesu dynamicznego z zerowymi warunkami początkowymi. Niestety, stabilność tego procesu uzależniona jest od spełnienia warunków (1.50) i (1.51), które ograniczają jego

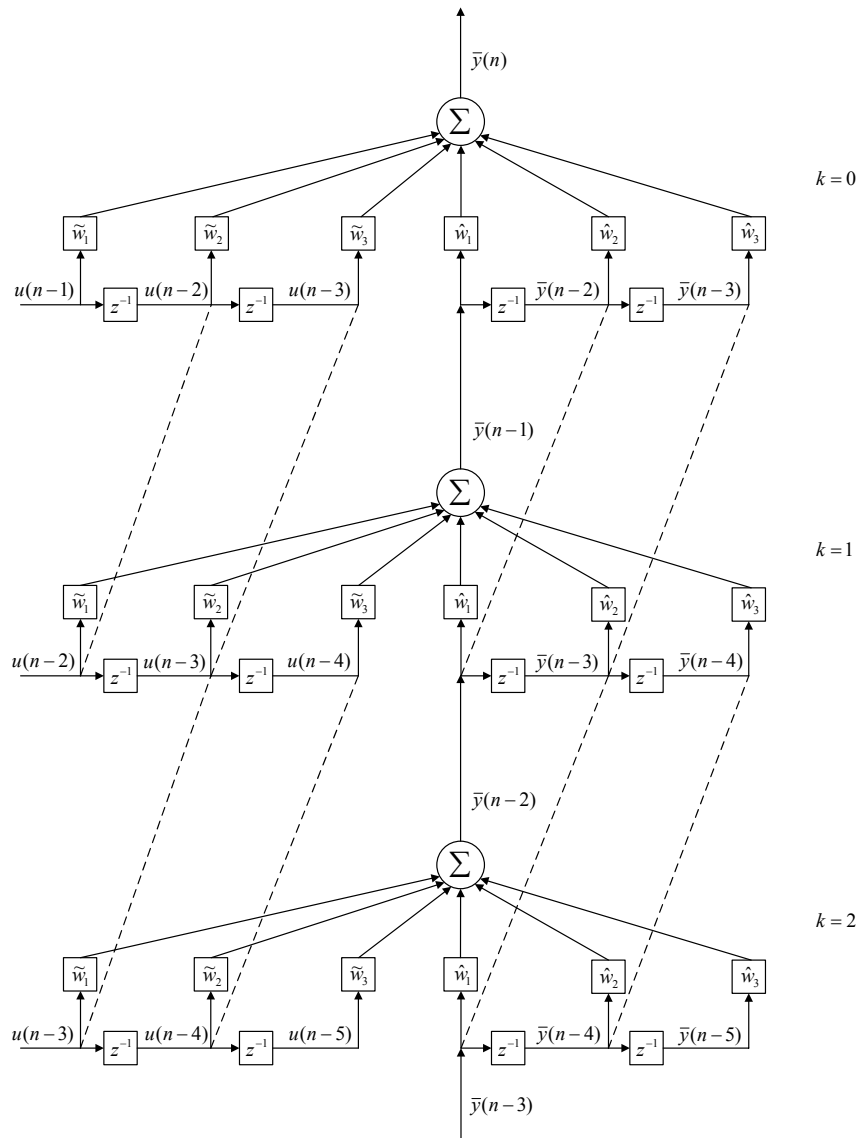
zastosowanie do obiektów o niedużych szybkościach zmian sygnałów wejściowych i wyjściowych w początkowej fazie identyfikacji. W przeciwnym przypadku wartości pochodnych cząstkowych mogą zmierzać do nieskończoności. Istnieją sposoby radzenia sobie z tym problemem, jak np. opisana w pracy [86] metoda polegająca na odpowiednim manipulowaniu w każdym takcie wartością współczynnika uczenia  $\eta$ , tak aby stabilizować proces opisany formułami (1.56), (1.57). Wiąże się to jednak z dodatkowym nakładem obliczeniowym, a sami autorzy pracy [87] twierdzą, że w modelowaniu złożonych procesów trudno jest uniknąć problemów ze stabilnością przy stosowaniu algorytmu RTRL.

**BPTT.** Drugie podejście, na którym opiera się metoda uczenia BPTT, [135], wynika ze spojrzenia na model ze sprzężeniem zwrotnym jak na szeregowe połączenie modeli nie mających tego sprzężenia, lecz posiadających te same parametry o identycznych wartościach, z tą jednak różnicą, że działających w różnych momentach czasowych. Ideę tę ilustruje rys. 1.11, na którym rozważany model został rozwinięty (ang. *unfold*) dwa kroki wstecz w czasie. Wartości wszystkich wag na każdym poziomie rozwinięcia  $k$  pochodzą z tego samego,  $n$ -tego taktu.

Model rozwinięty wstecz w czasie jest równoważny wyjściowemu tylko wtedy, gdy liczba rozwinięć w takcie  $n$ -tym jest równa  $n$ . Prowadzi to – podobnie jak poprzednio – do bardzo dużej liczby obliczeń przy wyznaczaniu pochodnych cząstkowych. Aproxymacja, którą tu się stosuje, polega na ograniczeniu liczby rozwinięć modelu w czasie do pewnej ustalonej liczby  $K$ . W większości przypadków dostatecznie dokładne przybliżenia uzyskuje się już dla  $K = 4$ , [60]. Wówczas pochodne cząstkowe wyznacza się ze wzoru, [135]:

$$\left. \frac{\partial^+ \bar{y}(n)}{\partial w_b} \right|_{w_b=w_b(n)} = \left. \frac{\partial \bar{y}(n)}{\partial w_b} \right|_{w_b=w_b(n)} + \sum_{k=1}^K \frac{\partial^+ \bar{y}(n)}{\partial \bar{y}(n-k)} \left. \frac{\partial \bar{y}(n-k)}{\partial w_b} \right|_{w_b=w_b(n)}, \quad (1.58)$$

gdzie  $\left. \frac{\partial^+ \bar{y}(n)}{\partial w_b} \right|_{w_b=w_b(n)} \triangleq \frac{\partial^+ \bar{y}(n)}{\partial w_b(n)}$  oznaczają tzw. „uszeregowane pochodne cząstkowe (*ordered partial derivatives*) (warto porównać wzór (1.58) z wyrażeniami (1.53) i (1.54)). Jest to pojęcie wprowadzone przez Werbosa [135]. Dostarcza ono systematycznego, ujednoliczonego sposobu wyznaczania pochodnych cząstkowych w uszeregowanych układach równań, tj. takich, w których w równaniu  $n$ -tym używa się zmiennych występujących w równaniach o numerach mniejszych. Modele z dynamiką wewnętrzną w istocie stanowią uszeregowany układ równań. Przykła-



Rysunek 1.11: Model liniowy rozwinięty wstecz w czasie

dowo, model liniowy można przedstawić jako:

$$\left\{ \begin{array}{l} \bar{y}(0) = 0, \quad u(0) = 0 \\ \bar{y}(1) = \tilde{w}_1 u(0) + \hat{w}_1 \bar{y}(0) \\ \bar{y}(2) = \tilde{w}_1 u(1) + \tilde{w}_2 u(0) + \hat{w}_1 \bar{y}(1) + \hat{w}_2 \bar{y}(0) \\ \bar{y}(3) = \tilde{w}_1 u(2) + \tilde{w}_2 u(1) + \hat{w}_1 \bar{y}(2) + \hat{w}_2 \bar{y}(1) \quad , \\ \vdots \\ \bar{y}(n) = \sum_{d=1}^{D_u} \tilde{w}_d u(n-d) + \sum_{d=1}^{D_y} \hat{w}_d \bar{y}(n-d) \end{array} \right.$$

gdzie zmiennymi są  $\bar{y}(0), \bar{y}(1), \dots, \bar{y}(0)$ .

Pochodne cząstkowe związane z wagami  $\tilde{w}_i$  ( $i = 1, 2, \dots, D_u$ ) wyliczymy, korzystając z (1.58):

$$\begin{aligned} \frac{\partial^+ \bar{y}(n)}{\partial \tilde{w}_i(n)} &= \frac{\partial \bar{y}(n)}{\partial \tilde{w}_i(n)} + \sum_{k=1}^K \frac{\partial^+ \bar{y}(n)}{\partial \bar{y}(n-k)} \frac{\partial \bar{y}(n-k)}{\partial \tilde{w}_i(n)} = \\ &= u(n-i) + \sum_{k=1}^K \frac{\partial^+ \bar{y}(n)}{\partial \bar{y}(n-k)} u(n-i-k). \end{aligned} \quad (1.59)$$

W podobny sposób otrzymujemy pochodne cząstkowe związane z wagami  $\hat{w}_i$  ( $i = 1, 2, \dots, D_y$ ):

$$\begin{aligned} \frac{\partial^+ \bar{y}(n)}{\partial \hat{w}_i(n)} &= \frac{\partial \bar{y}(n)}{\partial \hat{w}_i(n)} + \sum_{k=1}^K \frac{\partial^+ \bar{y}(n)}{\partial \bar{y}(n-k)} \frac{\partial \bar{y}(n-k)}{\partial \hat{w}_i(n)} = \\ &= \bar{y}(n-i) + \sum_{k=1}^K \frac{\partial^+ \bar{y}(n)}{\partial \bar{y}(n-k)} \bar{y}(n-i-k). \end{aligned} \quad (1.60)$$

Pojawiające się w powyższych równaniach uszeregowane pochodne cząstkowe  $\frac{\partial^+ \bar{y}(n)}{\partial \bar{y}(n-k)}$  wyznaczamy – analogicznie jak z (1.58) – następująco:

$$\begin{aligned} \frac{\partial^+ \bar{y}(n)}{\partial \bar{y}(n-k)} &= \frac{\partial \bar{y}(n)}{\partial \bar{y}(n-k)} + \sum_{\tau=1}^{\min\{D_y, k-1\}} \frac{\partial^+ \bar{y}(n)}{\partial \bar{y}(n-k+\tau)} \frac{\partial \bar{y}(n-k+\tau)}{\partial \bar{y}(n-k)} = \\ &= \hat{w}_k \cdot \mathbf{1}(k \leq D_y) + \sum_{\tau=1}^{\min\{D_y, k-1\}} \frac{\partial^+ \bar{y}(n)}{\partial \bar{y}(n-k+\tau)} \hat{w}_\tau, \end{aligned} \quad (1.61)$$

gdzie:

$$\mathbf{1}(k \leq \kappa) = \begin{cases} 1 & \text{dla } k \leq \kappa \\ 0 & \text{w przeciwnym przypadku} \end{cases}. \quad (1.62)$$

Po wstawieniu formuł (1.59) i (1.60) do (1.28), gdzie składowymi gradientu są – zamiast zwykłych pochodnych cząstkowych  $\frac{\partial \bar{y}(n)}{\partial w_b}$  – odpowiednie uszeregowane pochodne cząstkowe  $\frac{\partial^+ \bar{y}(n)}{\partial w_b(n)}$ , otrzymujemy algorytm identyfikacji. Umieszczenie obliczeń wykonanych w ramach algorytmu w grafie z rys. 1.11 wskazuje na analogię między algorytmem BP a BPTT. Otóż gdyby na model przedstawiony na rys. 1.11 spojrzeć jak na statyczną sieć wielowarstwową z rys. 1.4, okazałoby

się, że działania opisane wzorami (1.59), (1.60), (1.28) można interpretować jako przesyłanie sygnałów błędu wstecz do poprzednich warstw rozwiniętego wstecz w czasie modelu (1.42). Tutaj poprzednie warstwy odpowiadają poprzednim momentom czasowym, stąd nazwa metody „wstecznej propagacji błędów w czasie”.

Związek, jaki istnieje między metodami BPTT a RTRL pokazano w artykułach [10] i [131]. Zwróćmy jedynie uwagę, że dla  $K = 0$  algorytm BPTT jest równoważny zwykłemu algorytmowi wstecznej propagacji błędów BP.

### 1.4.3 Algorytm uczenia rekurencyjnej sieci neuronowej

Ponownie rozważmy obiekt dynamiczny o  $S$  wejściach i  $L$  wyjściach, jak w podpunkcie 1.4.1, lecz tym razem modelowany przy użyciu sieci DRNN. Przyjmijmy oznaczenia takie, jak wprowadzone w podpunkcie 1.3.4. Od modelu neuronowego wymagamy uwzględniania wpływu  $D_\mu$  poprzednich wejść i  $D_y$  swoich poprzednich wyjść na bieżące wyjście (liczby te określają zarazem rzędy neuronów warstwy ukrytej). Sytuacja jest identyczna, jak przedstawiona na rys. 1.10, w którym blok liniowy zostałby zastąpiony blokiem DRNN. W tym podpunkcie ograniczymy się do wyprowadzenia algorytmu BPTT do uczenia sieci DRNN. Dotychczas do uczenia sieci tego typu opracowano jedynie algorytm RTRL, [70].

Mając do dyspozycji w każdym  $n$ -tym takcie nowe dane pomiarowe ( $\mathbf{u}(n)$ ,  $\mathbf{y}(n)$ ) z obiektu oraz przyjmując kwadratowy wskaźnik jakości identyfikacji (1.27), należy skonstruować algorytm uczenia sieci DRNN, dający w wyniku wartości parametrów będące przybliżeniem optymalnego zestawu  $\mathbf{w}^*$ , tj. minimalizującego kryterium jakości identyfikacji (1.27). Skorzystamy ponownie z metody gradientu prostego (1.28). Oznaczając przez  $w_b$  ( $b = 1, 2, \dots, L \cdot (H + 1) + H \cdot D_\mu + H \cdot (S + 1) \cdot D_u$ ) elementy wektora parametrów sieci  $\mathbf{w}$ , wzór (1.28) można zapisać dla pojedynczej wagi w postaci formuły (1.44).

Gradient z równania (1.28) przyjmuje ogólną postać:

$$\nabla_{\mathbf{w}} Q(n, \mathbf{w}) = \frac{1}{2} \sum_{l=1}^L 2e_l(n) \cdot \nabla_{\mathbf{w}} e_l(n) = - \sum_{l=1}^L e_l(n) \cdot \nabla_{\mathbf{w}} \bar{y}_l(n), \quad (1.63)$$

a jego  $b$ -ta składowa jest równa:

$$\frac{\partial Q(n, \mathbf{w})}{\partial w_b} = - \sum_{l=1}^L e_l(n) \frac{\partial \bar{y}_l(n)}{\partial w_b}. \quad (1.64)$$

W dalszej części pracy przez zapis  $\frac{\partial \bar{y}_l(n)}{\partial w_b}$  będziemy domyślnie rozumieć  $\frac{\partial \bar{y}_l(n)}{\partial w_b(n)}$ ,

czyli pochodne cząstkowe względem parametrów domyślnie wyliczane są dla ich wartości z bieżącego taktu  $n$ .

W szczególności, pochodne cząstkowe względem wag neuronów warstwy wyjściowej  $\bar{w}_{ij}$  ( $i = 1, 2, \dots, L$ ,  $j = 0, 1, \dots, H$ ) wyliczamy następująco:

$$\frac{\partial \bar{y}_l(n)}{\partial \bar{w}_{ij}} = \begin{cases} \sigma(\chi_j(n)) = \mu_j(n) & \text{dla } l = i \\ 0 & \text{w przeciwnym przypadku} \end{cases} \quad (1.65)$$

Pochodne cząstkowe wag  $\tilde{w}_{ijp}$  ( $i = 1, 2, \dots, H$ ,  $j = 0, 1, \dots, S$ ,  $p = 1, 2, \dots, D_u$ ) są równe:

$$\frac{\partial \bar{y}_l(n)}{\partial \tilde{w}_{ijp}} = \bar{w}_{li} \frac{\partial^+ \mu_i(n)}{\partial \tilde{w}_{ijp}}, \quad (1.66)$$

natomiast uszeregowane pochodne cząstkowe z powyższego równania wyznaczamy w drodze następujących przekształceń:

$$\begin{aligned} \frac{\partial^+ \mu_i(n)}{\partial \tilde{w}_{ijp}} &= \frac{\partial \mu_i(n)}{\partial \tilde{w}_{ijp}} + \sum_{k=1}^K \frac{\partial^+ \mu_i(n)}{\partial \mu_i(n-k)} \frac{\partial \mu_i(n-k)}{\partial \tilde{w}_{ijp}} = \\ &= \frac{d\sigma(\chi_i(n))}{d\chi_i(n)} \frac{\partial \chi_i(n)}{\partial \tilde{w}_{ijp}} + \sum_{k=1}^K \frac{\partial^+ \mu_i(n)}{\partial \mu_i(n-k)} \frac{d\sigma(\chi_i(n-k))}{d\chi_i(n-k)} \frac{\partial \chi_i(n-k)}{\partial \tilde{w}_{ijp}} = \\ &= \sigma'(\chi_i(n)) u_j(n-p) + \sum_{k=1}^K \frac{\partial^+ \mu_i(n)}{\partial \mu_i(n-k)} \sigma'(\chi_i(n-k)) u_j(n-p-k) \end{aligned} \quad (1.67)$$

Pochodne cząstkowe wag  $\hat{w}_{ij}$  ( $i = 1, 2, \dots, H$ ,  $j = 1, 2, \dots, D_\mu$ ) są równe:

$$\frac{\partial \bar{y}_l(n)}{\partial \hat{w}_{ij}} = \bar{w}_{li} \frac{\partial^+ \mu_i(n)}{\partial \hat{w}_{ij}} \quad (1.68)$$

natomiast uszeregowane pochodne cząstkowe z tego równania wyznaczamy w drodze poniższych przekształceń:

$$\begin{aligned} \frac{\partial^+ \mu_i(n)}{\partial \hat{w}_{ij}} &= \frac{\partial \mu_i(n)}{\partial \hat{w}_{ij}} + \sum_{k=1}^K \frac{\partial^+ \mu_i(n)}{\partial \mu_i(n-k)} \frac{\partial \mu_i(n-k)}{\partial \hat{w}_{ij}} = \\ &= \frac{d\sigma(\chi_i(n))}{d\chi_i(n)} \frac{\partial \chi_i(n)}{\partial \hat{w}_{ij}} + \sum_{k=1}^K \frac{\partial^+ \mu_i(n)}{\partial \mu_i(n-k)} \frac{d\sigma(\chi_i(n-k))}{d\chi_i(n-k)} \frac{\partial \chi_i(n-k)}{\partial \hat{w}_{ij}} = \\ &= \sigma'(\chi_i(n)) \mu_i(n-j) + \sum_{k=1}^K \frac{\partial^+ \mu_i(n)}{\partial \mu_i(n-k)} \sigma'(\chi_i(n-k)) \mu_i(n-j-k). \end{aligned} \quad (1.69)$$

Pojawiające się w wyprowadzeniach (1.67) i (1.69) dodatkowe uszeregowane pochodne cząstkowe  $\frac{\partial^+ \bar{y}(n)}{\partial \bar{y}(n-k)}$  wyznaczamy następująco:

$$\begin{aligned} \frac{\partial^+ \mu_i(n)}{\partial \mu_i(n-k)} &= \frac{\partial \mu_i(n)}{\partial \mu_i(n-k)} + \sum_{\tau=1}^{\min\{D_\mu, k-1\}} \frac{\partial^+ \mu_i(n)}{\partial \mu_i(n-k+\tau)} \frac{\partial \mu_i(n-k+\tau)}{\partial \mu_i(n-k)} = \\ &= \frac{d\sigma(\chi_i(n))}{d\chi_i(n)} \frac{\partial \chi_i(n)}{\partial \mu_i(n-k)} + \\ &+ \sum_{\tau=1}^{\min\{D_\mu, k-1\}} \frac{\partial^+ \mu_i(n)}{\partial \mu_i(n-k+\tau)} \frac{d\sigma(\chi_i(n-k+\tau))}{d\chi_i(n-k+\tau)} \frac{\partial \chi_i(n-k+\tau)}{\partial \mu_i(n-k)} = \\ &= \sigma'(\chi_i(n)) \cdot \hat{w}_{ik} \cdot \mathbf{1}(k \leq D_\mu) + \end{aligned} \quad (1.70)$$

$$+ \sum_{\tau=1}^{\min\{D_\mu, k-1\}} \frac{\partial^+ \mu_i(n)}{\partial \mu_i(n-k+\tau)} \sigma'(\chi_i(n-k+\tau)) \cdot \hat{w}_{i\tau} \quad .$$

Po uwzględnieniu (1.70) w wynikach przekształceń (1.67) i (1.69), wstawionych następnie do, odpowiednio (1.66) i (1.68), dostajemy wyrażenia na pochodne cząstkowe  $\frac{\partial \bar{y}_l(n)}{\partial \tilde{w}_{ijp}}$  i  $\frac{\partial \bar{y}_l(n)}{\partial \hat{w}_{ij}}$ . Reguły aktualizacji wag są – według wzorów (1.63) i (1.64) – następujące:

$$\tilde{w}_{ijp}(n+1) = \tilde{w}_{ijp}(n) + \eta \sum_{l=1}^L e_l(n) \frac{\partial \bar{y}_l(n)}{\partial \tilde{w}_{ijp}}, \quad (1.71)$$

$$\hat{w}_{ij}(n+1) = \hat{w}_{ij}(n) + \eta \sum_{l=1}^L e_l(n) \frac{\partial \bar{y}_l(n)}{\partial \hat{w}_{ij}}, \quad (1.72)$$

natomiast – według wzoru (1.65) – reguła aktualizacji wag warstwy wyjściowej ma postać:

$$\bar{w}_{ij}(n+1) = \bar{w}_{ij}(n) + \eta \cdot e_i(n) \frac{\partial \bar{y}_i(n)}{\partial \bar{w}_{ij}}. \quad (1.73)$$

Powyższe reguły, po wstawieniu do nich wzorów na pochodne cząstkowe, tworzą algorytm BPTT uczenia sieci DRNN.



## 1.5 Identyfikacja systemów złożonych

Jak wspomniano wcześniej, w systemie złożonym wyodrębnia się elementy składowe z określonymi wielkościami wejściowymi i wyjściowymi oraz wskazuje się powiązania elementów. Rozważmy system złożony z  $M$  dającymi się wyodrębnić elementami wejściowo-wyjściowymi  $O_1, O_2, \dots, O_M$ . Każdy z nich posiada  $S_m$  wejść i  $L_m$  wyjść ( $m = 1, 2, \dots, M$ ), które zebrać można w wektory:

$$\mathbf{u}^{(m)}(n) = \begin{bmatrix} u_1^{(m)}(n) \\ u_2^{(m)}(n) \\ \vdots \\ u_{S_m}^{(m)}(n) \end{bmatrix} \in \mathcal{R}^{S_m}, \quad \mathbf{y}^{(m)}(n) = \begin{bmatrix} y_1^{(m)}(n) \\ y_2^{(m)}(n) \\ \vdots \\ y_{L_m}^{(m)}(n) \end{bmatrix} \in \mathcal{R}^{L_m}. \quad (1.74)$$

Z kolei wektory wszystkich wejść i wyjść w całym systemie złożonym są następujące:

$$\mathbf{u}(n) = \begin{bmatrix} \mathbf{u}^{(1)}(n) \\ \mathbf{u}^{(2)}(n) \\ \vdots \\ \mathbf{u}^{(M)}(n) \end{bmatrix} \in \mathcal{R}^{\bar{S}}, \quad \mathbf{y}(n) = \begin{bmatrix} \mathbf{y}^{(1)}(n) \\ \mathbf{y}^{(2)}(n) \\ \vdots \\ \mathbf{y}^{(M)}(n) \end{bmatrix} \in \mathcal{R}^{\bar{L}}, \quad (1.75)$$

gdzie  $\bar{S} = \sum_{m=1}^M S_m$ ,  $\bar{L} = \sum_{m=1}^M L_m$ .

Pewna część wejść występujących w systemie złożonym stanowi zarazem wyjścia innych elementów. Pozostałe wejścia, które nie są wyjściami innych elementów, nazywamy *wejściami zewnętrznymi*, [121]. Oznaczmy przez  $S$  ich liczbę oraz zbierzmy je w wektor postaci:

$$\mathbf{x}(n) = \begin{bmatrix} x_1(n) \\ x_2(n) \\ \vdots \\ x_S(n) \end{bmatrix} \in \mathcal{R}^S.$$

Dodatkowo, w związku z celem tworzenia modelu, wyróżniono  $L$  wyjść, które zbierzemy w wektor postaci:

$$\mathbf{v}(n) = \begin{bmatrix} v_1(n) \\ v_2(n) \\ \vdots \\ v_L(n) \end{bmatrix} \in \mathcal{R}^L.$$

Wyjścia te nazwiemy *wyjściami globalnymi*, [121].

Strukturę systemu złożonego opisuje zależność, [121]:

$$\begin{cases} \mathbf{u} = \mathbf{A}\mathbf{y} + \mathbf{B}\mathbf{x} \\ \mathbf{v} = \mathbf{C}\mathbf{y} \end{cases}, \quad (1.76)$$

gdzie zerojedynkowa macierz  $\mathbf{A} \in \{0, 1\}^{\bar{S} \times \bar{L}}$  – definiuje powiązania pomiędzy elementami, tj. wystąpienie wartości 1 w wierszu  $i$ -tym, kolumnie  $j$ -tej oznacza, że wielkość wejściowa będąca  $i$ -tą składową wektora  $\mathbf{u}$  jest jednocześnie wielkością wyjściową będącą  $j$ -tą składową wektora  $\mathbf{y}$ , zerojedynkowa macierz  $\mathbf{B} \in \{0, 1\}^{\bar{S} \times S}$  – wskazuje na wyjścia zewnętrzne, zerojedynkowa macierz  $\mathbf{C} \in \{0, 1\}^{L \times \bar{L}}$  – wskazuje na wyjścia globalne. W opisie struktury pominięto oznaczenia numeru taktu, gdyż zakładamy, że nie podlega ona zmianom w czasie. Model systemu złożonego można zapisać jako

$$\bar{\mathbf{v}}(n) = \Phi(\mathbf{x}(n-1), \dots, \mathbf{x}(n-D); \mathbf{w}), \quad (1.77)$$

gdzie  $D$  jest rzędem modelu a  $\mathbf{w}$  to wektor parametrów modeli wszystkich elementów:

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}^{(1)} \\ \mathbf{w}^{(2)} \\ \vdots \\ \mathbf{w}^{(M)} \end{bmatrix}, \quad (1.78)$$

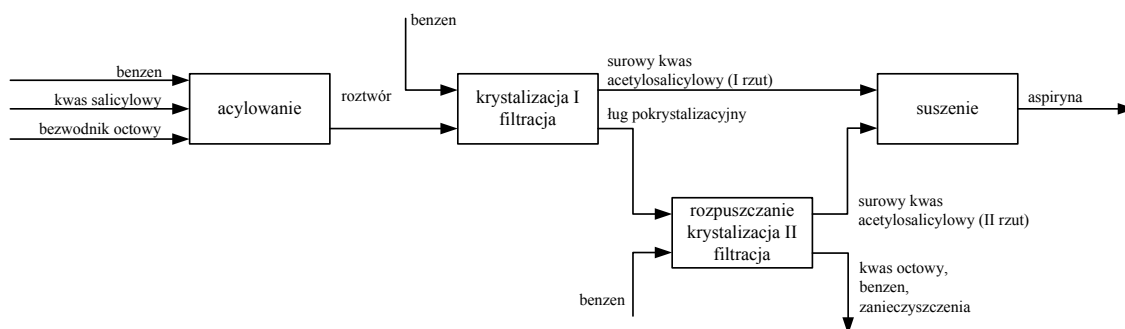
a  $\mathbf{w}^{(m)}$  ( $m = 1, 2, \dots, M$ ) jest wektorem parametrów  $m$ -tego modelu.

Sama czynność ustalania struktury systemu może być zadaniem trudnym. Wchodzi ona w zakres analizy systemowej i nie jest przedmiotem rozważań w niniejszej pracy.

**Przykład rzeczywistego systemu złożonego.** Na rys. 1.12 przedstawiono ideowy schemat produkcji aspiryny, [71], w którym wyodrębnić można kilka elementarnych procesów:

- acylowanie (element  $O_1$ ),
- krystalizacja I i filtracja (potraktowane tu łącznie jako jeden element  $O_2$ ),
- rozpuszczanie, krystalizacja II i filtracja (potraktowane jako jeden element  $O_3$ ),
- suszenie (element  $O_4$ ).

Podczas acylowania z benzenu, kwasu salicylowego i bezwodnika octowego powstaje roztwór, który następnie krystalizuje. Przemycanie benzenem i filtracja dają w rezultacie surowy kwas acetylosalicylowy oraz odpad w postaci ługu krystalicznego. Wysuszenie surowego kwasu acetylosalicylowego daje produkt końcowy – aspirynę. Z ługu pokrystalicznego można natomiast uzyskać jeszcze pewną ilość produktu. W tym celu należy ług rozpuścić, ponownie używając benzenu, poddać krystalizacji i filtracji. W efekcie, oprócz odpadów w postaci benzenu, kwasu octowego i zanieczyszczeń, otrzymamy jeszcze pewną ilość surowego kwasu acetylosalicylowego, który po wysuszeniu jest aspiryną.



Rysunek 1.12: Schemat procesu produkcji aspiryny

Opisany proces jest systemem złożonym, w skład którego wchodzi cztery elementy  $O_1, O_2, O_3, O_4$  o liczbach wejść odpowiednio  $S_1 = 3, S_2 = 2, S_3 = 2, S_4 = 2$  i liczbach wyjść  $L_1 = 1, L_2 = 2, L_3 = 2, L_4 = 1$ . Wejściami pierwszego elementu są:  $u_1^{(1)}$  – ilość benzenu,  $u_2^{(1)}$  – ilość kwasu salicylowego,  $u_3^{(1)}$  – ilość bezwodnika octowego, natomiast wyjściem jest  $y_1^{(1)}$  – ilość roztworu powstałego wskutek acylowania. Wejściami drugiego elementu są:  $u_1^{(2)}$  – ilość benzenu,  $u_2^{(2)} = y_1^{(1)}$  – czyli drugie wejście elementu  $O_2$  jest wyjściem elementu  $O_1$ . Wyjściami drugiego elementu to:  $y_1^{(2)}$  – ilość surowego kwasu acetylosalicylowego (I rzut),  $y_2^{(2)}$  – ilość ługu pokrystalicznego. Wejściami elementu trzeciego to:  $u_1^{(3)} = y_2^{(2)}$  oraz  $u_2^{(3)}$  – ilość benzenu. Wyjściami są:  $y_1^{(3)}$  – ilość surowego kwasu acetylosalicylowego (II rzut),  $y_2^{(3)}$  – łączna ilość kwasu octowego, benzenu i zanieczyszczeń. Wejściami ostatniego elementu to:  $u_1^{(4)} = y_1^{(2)}$  oraz  $u_2^{(4)} = y_1^{(3)}$  a jego wyjściem jest  $y_1^{(4)}$  – końcowa ilość aspiryny. Widzimy, że powiązania między elementami są konsekwencją tego, że produkty uzyskane w jednych procesach stanowią wejścia kolejnych procesów. Wejściami zewnętrznymi – z uwagi na możliwość bezpośredniego kontrolowania ich wartości – są:  $x_1 = u_1^{(2)}, x_2 = u_1^{(1)}, x_3 = u_2^{(1)}, x_4 = u_3^{(1)}, x_5 = u_2^{(3)}$ . Interesuje

nas ostateczna ilość wyprodukowanej aspiryny oraz ilość produktów ubocznych procesu, wobec czego wyjściami globalnymi są  $v_1 = y_2^{(3)}$  i  $v_2 = y_1^{(4)}$ . Modelowane mogą być przebiegi zmian ilości produktów w obrębie każdego podprocesu (modele dynamiczne), lub też charakterystyki statyczne, tzn. wartości wybranych zmiennych w stanie ustalonym.

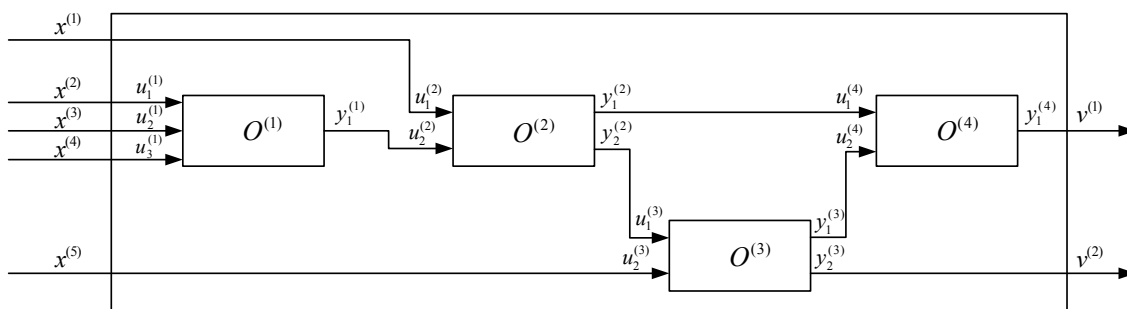
Równania (1.76) opisujące strukturę systemu przyjmują postać:

$$\begin{bmatrix} u_1^{(1)} \\ u_2^{(1)} \\ u_3^{(1)} \\ u_1^{(2)} \\ u_2^{(2)} \\ u_1^{(3)} \\ u_2^{(3)} \\ u_1^{(4)} \\ u_2^{(4)} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_1^{(1)} \\ y_1^{(2)} \\ y_2^{(2)} \\ y_1^{(3)} \\ y_2^{(3)} \\ y_1^{(4)} \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix},$$

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_1^{(1)} \\ y_1^{(2)} \\ y_2^{(2)} \\ y_1^{(3)} \\ y_2^{(3)} \\ y_1^{(4)} \end{bmatrix}.$$

Na system złożony można więc patrzeć jak na obiekt o wejściach  $\mathbf{x}$  i wyjściach  $\mathbf{v}$ .

Wiele przykładów chemicznych procesów złożonych opisano w [81].



Rysunek 1.13: Przykład systemu złożonego – proces produkcji aspiryny

### 1.5.1 Modele lokalnie i globalnie optymalne

Zakładając znajomość struktury modelowanego systemu złożonego, zadanie identyfikacji sprowadza się do identyfikacji poszczególnych jego elementów z wykorzystaniem serii pomiarów (1.74) wykonanych dla każdego elementu niezależnie:

$$\mathbf{U}^{(m)} = \begin{bmatrix} \mathbf{u}^{(m)}(1) & \mathbf{u}^{(m)}(2) & \cdots & \mathbf{u}^{(m)}(N_m) \end{bmatrix}, \quad (1.79)$$

$$\mathbf{Y}^{(m)} = \begin{bmatrix} \mathbf{y}^{(m)}(1) & \mathbf{y}^{(m)}(2) & \cdots & \mathbf{y}^{(m)}(N_m) \end{bmatrix}, \quad (1.80)$$

przy czym znane są stany początkowe  $\mathbf{u}^{(m)}(0)$ ,  $\mathbf{y}^{(m)}(0)$  wszystkich obiektów ( $m = 1, 2, \dots, M$ ),  $N_m$  – jest długością serii pomiarowej  $m$ -tego obiektu, przy czym w identyfikacji na bieżąco  $N_m = 1$  dla każdego  $m$  oraz zamiast stanu początkowego mamy stan w takcie poprzedzającym. Dla  $m$ -tego elementu proponujemy model (tu z dynamiką wewnętrzną):

$$\bar{\mathbf{y}}^{(m)}(n) = \Phi_m \left( \mathbf{u}^{(m)}(n-1), \mathbf{u}^{(m)}(n-2), \dots, \mathbf{u}^{(m)}(n - D_u^{(m)}); \mathbf{w}^{(m)} \right), \quad (1.81)$$

gdzie  $D_u^{(m)}$  – liczba kolejnych wartości wejść  $m$ -tego elementu z poprzednich taktów brana pod uwagę przy wyliczaniu wartości sygnału wyjściowego tego elementu.

Następnie należy określić kryterium jakości identyfikacji. Występuje tu nowy problem, który nie pojawia się w przypadku pojedynczego obiektu. Otóż w systemie złożonym ocena jakości modelowania może być przeprowadzana zarówno dla każdego modelu  $\Phi_m$  niezależnie, w oderwaniu od struktury systemu, jak i dla systemu  $\Phi$  jako całości.

W pierwszym przypadku formułujemy *lokalne kryteria jakości identyfikacji*:

$$Q_L^{(m)} \left( \mathbf{U}^{(m)}, \mathbf{Y}^{(m)}; \mathbf{w}^{(m)} \right) \triangleq \left\| \mathbf{Y}^{(m)} - \bar{\mathbf{Y}}^{(m)} \right\|, \quad (1.82)$$

gdzie

$$\bar{\mathbf{Y}}^{(m)} = \begin{bmatrix} \bar{\mathbf{y}}^{(m)}(1) & \bar{\mathbf{y}}^{(m)}(2) & \cdots & \bar{\mathbf{y}}^{(m)}(N_m) \end{bmatrix},$$

przy czym sygnały  $\bar{\mathbf{y}}^{(m)}(n)$  są wyznaczane z modelu (1.81) jako odpowiedź na sygnały wejściowe zmierzone z  $m$ -tego obiektu wchodzącego w skład systemu złożonego. W pracy korzystać będziemy z kryterium postaci:

$$Q_L^{(m)} \left( \mathbf{U}^{(m)}, \mathbf{Y}^{(m)}; \mathbf{w}^{(m)} \right) \triangleq \sum_{n=1}^{N_m} q_m \left( \mathbf{y}^{(m)}(n), \bar{\mathbf{y}}^{(m)}(n) \right), \quad (1.83)$$

gdzie  $q_m(\mathbf{y}^{(m)}(n), \bar{\mathbf{y}}^{(m)}(n))$  jest funkcją oceniającą różnicę między  $\mathbf{y}^{(m)}(n)$  a  $\bar{\mathbf{y}}^{(m)}(n)$ . Dalej pomijając będziemy oznaczenie serii pomiarowej w zapisie kryterium, pisząc krótko  $Q_L^{(m)}(\mathbf{w}^{(m)})$ .

Zadanie identyfikacji systemu złożonego polega – w omawianym przypadku z lokalnymi kryteriami – na rozwiązaniu  $M$  zadań identyfikacji pojedynczych elementów, z których każde sprowadza się do wyznaczenia takich wartości parametrów modelu  $\Phi_m$ , dla których wskaźnik jakości identyfikacji  $Q_L^{(m)}$  przyjmuje wartość minimalną. Dokładniej, w zadaniu *identyfikacji lokalnej*  $m$ -tego elementu dane są pomiary wejść i wyjść  $\mathbf{U}^{(m)}$ ,  $\mathbf{Y}^{(m)}$ , określony jest wskaźnik jakości  $Q_L^{(m)}$  oraz zadana jest postać modelu  $\Phi_m$  z dokładnością do parametrów  $\mathbf{w}^{(m)}$ . Wyznaczyć należy optymalny zestaw  $\mathbf{w}_L^{(m)*}$  parametrów modelu, czyli taki, dla którego zachodzi

$$Q_L^{(m)}(\mathbf{w}_L^{(m)*}) = \min_{\mathbf{w}^{(m)}} Q_L^{(m)}(\mathbf{w}^{(m)}). \quad (1.84)$$

Algorytmy identyfikacji  $\Psi_L^{(m)}$  ( $m = 1, 2, \dots, M$ ), będące wynikiem rozwiązania powyższych lokalnych zadań, mają postać:

$$\mathbf{w}_L^{(m)*} = \Psi_L^{(m)}(\mathbf{U}^{(m)}, \mathbf{Y}^{(m)}). \quad (1.85)$$

Model (1.81) z parametrami  $\mathbf{w}_L^{(m)*}$  jest optymalnym, w sensie lokalnego wskaźnika jakości, modelem  $m$ -tego elementu:

$$\bar{\mathbf{y}}^{(m)}(n) = \Phi_m(\mathbf{u}^{(m)}(n-1), \mathbf{u}^{(m)}(n-2), \dots, \mathbf{u}^{(m)}(n-D_u^{(m)}); \mathbf{w}_L^{(m)*}). \quad (1.86)$$

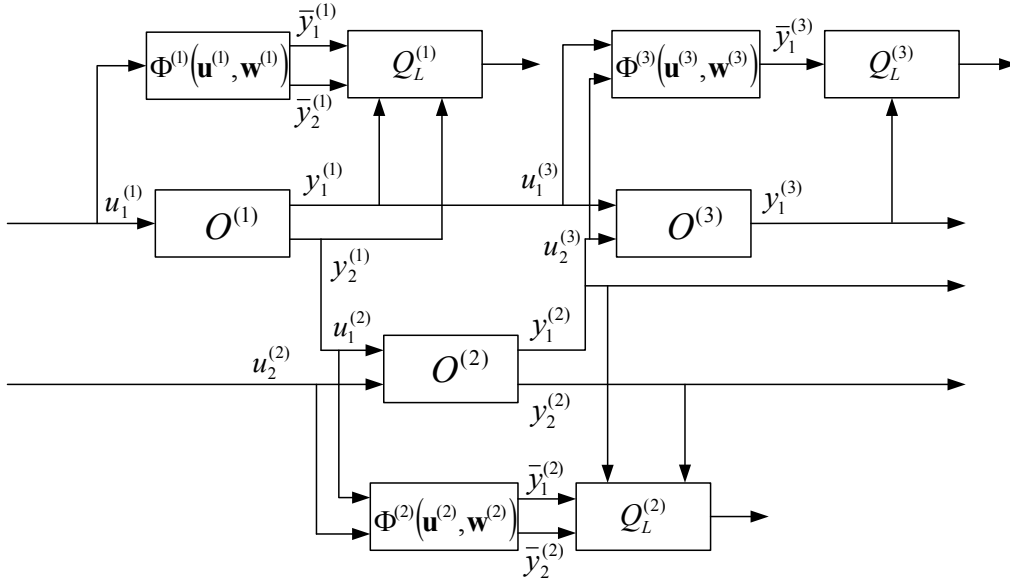
Wykonanie zadania identyfikacji dla wszystkich elementów daje nam wektor lokalnie optymalnych parametrów:

$$\mathbf{w}_L^* = \begin{bmatrix} \mathbf{w}_L^{(1)*} \\ \mathbf{w}_L^{(2)*} \\ \vdots \\ \mathbf{w}_L^{(M)*} \end{bmatrix}, \quad (1.87)$$

Składając modele  $\Psi_L^{(m)}$  zgodnie ze strukturą systemu, otrzymujemy model lokalnie optymalny systemu złożonego (1.77), [124]:

$$\bar{\mathbf{v}}(n) = \Phi(\mathbf{x}(n-1), \dots, \mathbf{x}(n-D); \mathbf{w}_L^*). \quad (1.88)$$

Model ten jest oczywiście optymalny dla zadanej klasy i ta uwaga dotyczy kolejnych wprowadzanych rodzajów modeli. Rysunek 1.14 przedstawia ideę lokalnej



Rysunek 1.14: Przykład systemu złożonego z modelami lokalnymi

oceny jakości modelowania w systemie złożonym. Aby zachować przejrzystość rysunku, pominięto oznaczenia taktu  $n$ .

Spojrzenie na modelowany system jak na pewną całość i skupienie się na wyróżnionych wyjściach globalnych prowadzi do sformułowania *globalnego wskaźnika jakości identyfikacji*. Załóżmy, że wykonano (lub wyodrębniono z serii (1.79), (1.80)) pomiary wejść zewnętrznych  $\mathbf{x}$  oraz wyjść globalnych  $\mathbf{y}$ :

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}(1) & \mathbf{x}(2) & \cdots & \mathbf{x}(N) \end{bmatrix}, \quad (1.89)$$

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}(1) & \mathbf{v}(2) & \cdots & \mathbf{v}(N) \end{bmatrix}, \quad (1.90)$$

gdzie  $N$  jest długością serii pomiarowej oraz znany jest stan początkowy  $\mathbf{x}(0)$ ,  $\mathbf{v}(0)$ . Dla każdego elementu proponujemy model:

$$\bar{\mathbf{v}}^{(m)}(n) = \Phi_m \left( \bar{\mathbf{u}}^{(m)}(n-1), \bar{\mathbf{u}}^{(m)}(n-2), \dots, \bar{\mathbf{u}}^{(m)}(n-D_u^{(m)}); \mathbf{w}^{(m)} \right), \quad (1.91)$$

gdzie  $\bar{\mathbf{u}}^{(m)}(n)$  jest wektorem wejść  $m$ -tego modelu w  $n$ -tym takcie.

Jego składowe mają na ogół wartości różne od składowych wektora wejść  $m$ -tego obiektu, ponieważ mogą być rezultatem wyliczeń sygnałów wyjściowych powiązanych modeli. Ocena składającego się z nich modelu systemu złożonego dokonywana jest przy pomocy *globalnego kryterium jakości identyfikacji*, [124]:

$$Q_G(\mathbf{X}, \mathbf{V}; \mathbf{w}) \triangleq \|\mathbf{V} - \bar{\mathbf{V}}\|, \quad (1.92)$$

gdzie:

$$\bar{\mathbf{V}} = \begin{bmatrix} \bar{\mathbf{v}}(1) & \bar{\mathbf{v}}(2) & \cdots & \bar{\mathbf{v}}(N) \end{bmatrix}$$

jest serią odpowiedzi  $\bar{\mathbf{v}}(n)$  ( $n = 1, 2, \dots, N$ ) modelu (1.77) z bieżącymi wartościami parametrów  $\mathbf{w}$  na sygnały wejściowe z serii identyfikującej  $\mathbf{X}$ . W pracy korzystać będziemy z kryterium postaci:

$$Q_G(\mathbf{X}, \mathbf{V}; \mathbf{w}) \triangleq \sum_{n=1}^N q(\mathbf{v}(n), \bar{\mathbf{v}}(n)), \quad (1.93)$$

gdzie  $q(\mathbf{v}(n), \bar{\mathbf{v}}(n))$  jest funkcją oceniającą różnicę między wektorami  $\mathbf{v}(n)$  a  $\bar{\mathbf{v}}(n)$ . Dalej pomijając będziemy oznaczenie serii pomiarowej w zapisie kryterium, pisząc krótko  $Q_G(\mathbf{w})$ .

W zadaniu *identyfikacji globalnej* dane są więc pomiary wejść i wyjść  $\mathbf{X}$ ,  $\mathbf{V}$ , określony jest wskaźnik jakości  $Q_G$  oraz zadane są postacie modeli  $\Phi_m$  ( $m = 1, 2, \dots, M$ ) z dokładnością do parametrów  $\mathbf{w}^{(m)}$ , zebranych w jednym wektorze  $\mathbf{w}$ . Wyznaczyć należy optymalny zestaw  $\mathbf{w}_G^*$  tych parametrów, czyli taki, dla którego zachodzi:

$$Q_G(\mathbf{w}_G^*) = \min_{\mathbf{w}} Q_G(\mathbf{w}). \quad (1.94)$$

Algorytm identyfikacji  $\Psi_G$ , będący wynikiem rozwiązania powyższego zadania, ma postać:

$$\mathbf{w}_G^* = \Psi_G(\mathbf{X}, \mathbf{V}). \quad (1.95)$$

Model (1.77) z parametrami  $\mathbf{w}_G^*$  nazywamy *globalnie optymalnym modelem* systemu złożonego:

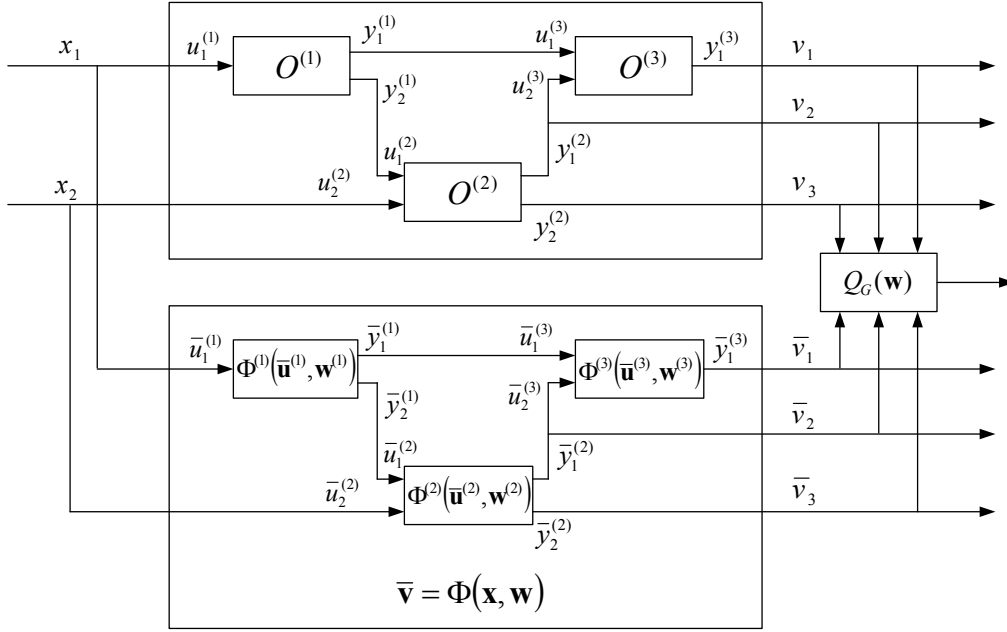
$$\bar{\mathbf{v}}(n) = \Phi(\mathbf{x}(n-1), \dots, \mathbf{x}(n-D); \mathbf{w}_G^*). \quad (1.96)$$

Rysunek 1.15 przedstawia ideę globalnej oceny jakości modelowania w systemie złożonym. Z podobnych względów, jak poprzednio, pominięto oznaczenia taktu  $n$ .

Oba przedstawione podejścia do identyfikacji systemu złożonego prowadzą do modeli o różnych wartościach parametrów. W zależności od celu budowy modelu, różnie oceniać będziemy ich przydatność. Przykładowo, do projektowania lokalnych układów regulacji korzystać będziemy z modelu lokalnie optymalnego, natomiast projektując system sterowania nadrzędnego wybierzemy model globalnie optymalny, [41].

Istnieją również sytuacje, w których chcemy korzystać z zalet obu rodzajów modeli. Na przykład w dwupoziomowym systemie sterowania istotna jest zarówno jakość modelu globalnego (na potrzeby sterowania nadrzędnego) jak i jakość





Rysunek 1.15: Przykład systemu złożonego z modelem globalnym

modeli lokalnych (na potrzeby podejmowania decyzji na dolnym poziomie), [18]. Takie postawienie problemu prowadzi do wielokryterialnej oceny modelu systemu złożonego, uwzględniającej zarówno jakość modeli lokalnych jak i jakość modelu globalnego. Jednym ze sposobów jednoczesnego uwzględnienia obu ocen: lokalnej i globalnej, jest zdefiniowanie *syntetycznego wskaźnika jakości identyfikacji*, [124], jako funkcji poszczególnych wskaźników:

$$Q_S(\mathbf{X}, \mathbf{V}, \mathbf{U}, \mathbf{Y}; \mathbf{w}) \triangleq E \left( Q_G(\mathbf{X}, \mathbf{V}; \mathbf{w}), Q_L^{(1)}(\mathbf{U}^{(1)}, \mathbf{Y}^{(1)}; \mathbf{w}^{(1)}), \right. \\
 \left. Q_L^{(2)}(\mathbf{U}^{(2)}, \mathbf{Y}^{(2)}; \mathbf{w}^{(2)}), \dots, Q_L^{(M)}(\mathbf{U}^{(M)}, \mathbf{Y}^{(M)}; \mathbf{w}^{(M)}) \right),$$

co zapisać można krócej jako:

$$Q_S(\mathbf{X}, \mathbf{V}, \mathbf{U}, \mathbf{Y}; \mathbf{w}) \triangleq E \left( Q_G(\mathbf{w}), Q_L^{(1)}(\mathbf{w}^{(1)}), Q_L^{(2)}(\mathbf{w}^{(2)}), \dots, Q_L^{(M)}(\mathbf{w}^{(M)}) \right), \quad (1.97)$$

gdzie  $\mathbf{U}$  i  $\mathbf{Y}$  oznaczają:

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}^{(1)} & \mathbf{U}^{(2)} & \dots & \mathbf{U}^{(M)} \end{bmatrix}, \quad (1.98)$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}^{(1)} & \mathbf{Y}^{(2)} & \dots & \mathbf{Y}^{(M)} \end{bmatrix}, \quad (1.99)$$

$\mathbf{X}$  i  $\mathbf{V}$  zawierają część pomiarów z serii  $\mathbf{U}$  i  $\mathbf{Y}$ , odpowiadającą wejściom zewnętrznym i wyjściom globalnym, natomiast  $E$  jest funkcją  $M + 1$  zmiennych, wypukłą

ze względu na każdą ze składowych (w najprostszym przypadku funkcją liniową). Minimalizacja syntetycznego wskaźnika (1.97) prowadzi do uzyskania optymalnego zestawu wartości parametrów  $\mathbf{w}_S^*$  modelu systemu złożonego, tj. takiego, dla którego spełniony jest warunek:

$$Q_S(\mathbf{w}_S^*) = \min_{\mathbf{w}} Q_S(\mathbf{w}). \quad (1.100)$$

Algorytm identyfikacji  $\Psi_S$  ma tu postać:

$$\mathbf{w}_S^* = \Psi_S(\mathbf{U}, \mathbf{Y}, \mathbf{X}, \mathbf{V}), \quad (1.101)$$

a model (1.77) z parametrami  $\mathbf{w}_S^*$  nazywamy globalnie optymalnym modelem systemu złożonego z syntetycznym wskaźnikiem jakości, [124]:

$$\bar{\mathbf{v}}(n) = \Phi(\mathbf{x}(n-1), \dots, \mathbf{x}(n-D); \mathbf{w}_S^*). \quad (1.102)$$

Jakość modeli lokalnych przy tworzeniu modelu globalnego może być uwzględniona również w inny sposób. Polega on na minimalizacji globalnego wskaźnika jakości przy jednoczesnym założeniu, że jakości modeli lokalnych są zadowalające, [124]. Dokładniej, wartości parametrów  $\mathbf{w}^{(m)}$  ( $m = 1, 2, \dots, M$ ) poszczególnych modeli powinny spełniać warunek:

$$Q_L^{(m)}(\mathbf{w}^{(m)}) \leq \lambda_m, \quad \text{przy czym} \quad \lambda_m \geq Q_L^{(m)}(\mathbf{w}_L^{*(m)}), \quad (1.103)$$

gdzie  $\lambda_m$  interpretuje się jako "miarę zadowolenia" z wartości wskaźnika jakości  $Q_L^{(m)}$  dla modelu  $m$ -tego elementu. Wymaganie na wartość tego wskaźnika nie może być oczywiście zbyt duże, tzn. nie można wymagać, by wskaźnik przyjmował wartości mniejsze od optymalnych  $Q_L^{(m)}(\mathbf{w}_L^{*(m)})$ . Zestaw  $M$  warunków (1.103) określa zbiór  $\Omega$  dopuszczalnych parametrów  $\mathbf{w}$  modelu systemu złożonego. Minimalizacja globalnego wskaźnika  $Q_G$  (1.92) na tym zbiorze prowadzi do uzyskania takiego zestawu wartości parametrów  $\mathbf{w}_{GL}^*$  modelu systemu złożonego, dla którego spełnione jest:

$$Q_G(\mathbf{w}_{GL}^*) = \min_{\mathbf{w} \in \Omega} Q_G(\mathbf{w}). \quad (1.104)$$

Odpowiedni algorytm identyfikacji  $\Psi_{GL}$  ma postać:

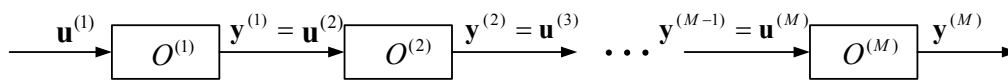
$$\mathbf{w}_{GL}^* = \Psi_{GL}(\mathbf{U}, \mathbf{Y}, \mathbf{X}, \mathbf{V}), \quad (1.105)$$

a model (1.77) z parametrami  $\mathbf{w}_{GL}^*$  nazywamy globalnie optymalnym modelem systemu złożonego z zadowalająco dobrymi modelami lokalnymi, [124]:

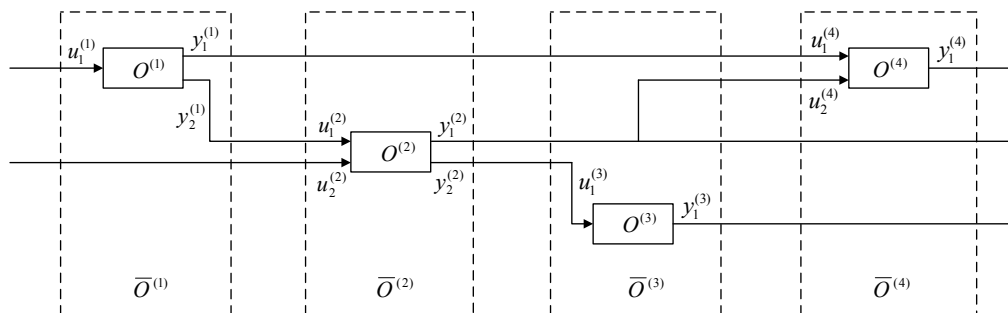
$$\bar{\mathbf{v}}(n) = \Phi(\mathbf{x}(n-1), \dots, \mathbf{x}(n-D); \mathbf{w}_{GL}^*). \quad (1.106)$$

### 1.5.2 Wykorzystanie sieci neuronowych

W pracy, jako modele elementów systemu złożonego, przyjęto rekurencyjne sieci neuronowe typu DRNN. Przyjęto również, że modelowanym systemem będzie dynamiczny system o strukturze kaskadowej, w którym wejścia kolejnego elementu są wyjściami poprzedniego (patrz rys. 1.16). Ważność struktury kaskadowej wynika z jej częstego występowania w procesach przemysłowych [81], [122], oraz z faktu, że można do niej sprowadzić wiele innych struktur, w których nie występują sprzężenia zwrotne pomiędzy elementami. Oczywiście, sprzężenia w obrębie pojedynczego elementu systemu złożonego nie stanowią tu przeszkody. Rys. 1.17 przedstawia system złożony z elementów  $O_1, O_2, O_3, O_4$ , który można sprowadzić do równoważnej postaci o strukturze kaskadowej z elementami zastępczymi  $\bar{O}_1, \bar{O}_2, \bar{O}_3, \bar{O}_4$ .



Rysunek 1.16: System złożony o strukturze kaskadowej



Rysunek 1.17: Przykład systemu złożonego dającego się sprowadzić do systemu o strukturze kaskadowej

Model systemu o strukturze kaskadowej w postaci sieci DRNN składany jest z sieci modelujących poszczególne elementy. Powstałą po ich złączeniu sieć neuronową nazywać będziemy *siecią złożoną*. Parametry złożonej sieci DRNN dobierane są w procesie uczenia opartym o odpowiednie dane pomiarowe. Algorytmy uczenia konstruowane są w oparciu o metodę optymalizacji gradientu prostego, podobnie jak w uczeniu sieci pojedynczych. W przypadku sieci złożonej dodatkowo należy uwzględnić fakt, że dobór parametrów  $m$ -tej sieci składowej ma wpływ

na jakość działania  $(m + 1)$ -szej sieci a także na następujące po niej w systemie kaskadowym, co komplikuje znacznie zadanie optymalizacji przyjętego kryterium jakości. Istota rozważanego problemu polega na tym, że sieci optymalne dla każdego elementu z osobna, po połączeniu ze sobą na ogół nie prowadzą do modelu optymalnego dla całości, jaką jest system złożony. Dlatego konieczne jest opracowanie nowych algorytmów identyfikacji/uczenia, pozwalających w rezultacie uzyskać sieci, które po połączeniu ze sobą tworzyłyby model systemu złożonego wystarczająco dokładny nie tylko na poziomie pojedynczych elementów, ale również na poziomie globalnym.

## 1.6 Teza, cel i zakres pracy

Dotychczas dla potrzeb systemów złożonych opracowano algorytmy uczenia statycznych sieci neuronowych. Natomiast – pomimo, że dla pojedynczych obiektów dynamicznych znane są dobrze algorytmy uczenia rekurencyjnych sieci neuronowych – brak jest opracowań dotyczących modeli dynamicznych systemów złożonych wykorzystujących połączone ze sobą rekurencyjne sieci neuronowe.

Korzystając z opisów modeli neuronowych oraz sformułowań zadań identyfikacji systemów złożonych, podanych w poprzednich punktach, możemy – dokładniej niż we wstępie – sformułować problem pracy.

**Celem pracy** jest opracowanie algorytmów uczenia rekurencyjnych sieci neuronowych na potrzeby identyfikacji dynamicznych systemów złożonych o strukturze kaskadowej, z uwzględnieniem lokalnych ocen jakości modeli (algorytm  $\Psi_L^{(m)}$  (1.85)), globalnej oceny jakości modelu (algorytm  $\Psi_G$  (1.95)), oraz przypadku wielokryterialnego, w którym osiąga się kompromis pomiędzy jakością modeli lokalnych oraz jakością modelu globalnego (algorytmy  $\Psi_S$  (1.101) i  $\Psi_{GL}$  (1.105)). Algorytmy konstruowane będą w oparciu o metodę numeryczną gradientu prostego. Rozpatrzone zostanie również ważne zagadnienie zbieżności uzyskanych algorytmów identyfikacji.

W **zakres pracy** wchodzi następujące elementy:

1. Opracowanie metod tworzenia modeli systemów złożonych o strukturze kaskadowej w postaci sieci DRNN (opisanych równaniami (1.12) i (1.12)), tj.:

- modeli lokalnie optymalnych (1.81):

$$\bar{\mathbf{y}}^{(m)}(n) = \Phi_m \left( \mathbf{u}^{(m)}(n-1), \mathbf{u}^{(m)}(n-2), \dots, \mathbf{u}^{(m)}(n-D_u^{(m)}); \mathbf{w}_L^{(m)*} \right),$$

- modelu globalnie optymalnego (1.96):

$$\bar{\mathbf{v}}(n) = \Phi(\mathbf{x}(n-1), \dots, \mathbf{x}(n-D); \mathbf{w}_G^*),$$

- modelu globalnie optymalnego z uwzględnieniem jakości modeli lokalnych (z użyciem syntetycznego wskaźnika jakości oraz przy zadowalającej jakości modeli lokalnych) (1.102) i (1.106):

$$\bar{\mathbf{v}}(n) = \Phi(\mathbf{x}(n-1), \dots, \mathbf{x}(n-D); \mathbf{w}_S^*),$$

$$\bar{\mathbf{v}}(n) = \Phi(\mathbf{x}(n-1), \dots, \mathbf{x}(n-D); \mathbf{w}_{GL}^*),$$

2. komputerowa realizacja odpowiednich algorytmów identyfikacji, tj.  $\Psi_L^{(m)}$ ,  $\Psi_S$ ,  $\Psi_G$ ,  $\Psi_{GL}$ , opartych o uszeregowane pochodne cząstkowe (1.58), pozwalająca na wykorzystanie ich do modelowania obiektu złożonego na podstawie danych pomiarowych (1.79) i (1.80) oraz (1.89) i (1.99),
3. teoretyczna analiza zbieżności procedury uczenia połączonych ze sobą kaskadowo sieci DRNN,
4. przeprowadzenie badań symulacyjnych pozwalających ocenić skuteczność opracowanych algorytmów na przykładzie identyfikacji dynamicznego systemu złożonego o strukturze kaskadowej.

Prezentowana praca poszerza aktualny stan wiedzy i zakres dostępnych technik zarówno w dziedzinie identyfikacji dynamicznych systemów złożonych, jak i w dziedzinie sieci neuronowych. Rezultaty pracy będą przydatne do opracowywania komputerowych systemów sterowania systemami złożonymi.

W pracy stawiana jest następująca **teza**:

*”Zastosowanie uszeregowanych pochodnych cząstkowych w uczeniu rekurencyjnych sieci neuronowych prowadzi do zbieżnych algorytmów identyfikacji dynamicznych systemów złożonych o strukturze kaskadowej.”*

## Rozdział 2

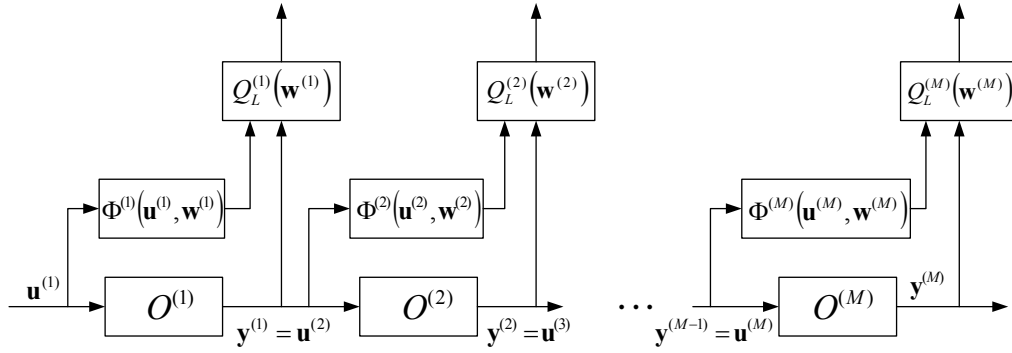
# Algorytmy uczenia rekurencyjnych sieci neuronowych w modelowaniu systemu złożonego

Rozdział zawiera oryginalne rezultaty pracy, tzn. algorytmy uczenia sieci DRNN na potrzeby identyfikacji dynamicznych systemów złożonych o strukturze kaskadowej.

### 2.1 Model lokalnie optymalny

Rozważmy system złożony o strukturze kaskadowej, w skład którego wchodzi  $M$  elementów wejściowo-wyjściowych – każdy ma  $S_m$  wejść i  $L_m$  wyjść ( $m = 1, 2, \dots, M$ ) i jest modelowany przez sieć DRNN (rys. 2.1).

Sieć  $m$ -ta w konsekwencji posiada również  $S_m$  wejść i  $L_m$  wyjść, a z uwagi na to, że wyjścia  $m$ -tej sieci są wyjściami sieci poprzedniej, zachodzi równość  $L_{m-1} = S_m$ , przy czym przyjmujemy  $L_0 = S_1$ . Sieć  $m$ -ta, będąca modelem  $m$ -tego elementu systemu złożonego, dostaje na wejścia ten sam zestaw sygnałów, które faktycznie podano na  $m$ -ty element (patrz rozdział 1.5.1). Wyliczone przez nią wyjścia



Rysunek 2.1: System złożony o strukturze kaskadowej z modelami lokalnymi

zbieramy w formie wektora:

$$\bar{\mathbf{y}}^{(m)}(n) = \begin{bmatrix} \bar{y}_1^{(m)}(n) \\ \bar{y}_2^{(m)}(n) \\ \vdots \\ \bar{y}_{L_m}^{(m)}(n) \end{bmatrix} \in \mathcal{R}^{L_m},$$

którego  $l$ -ta ( $l = 1, 2, \dots, L_m$ ) składowa wyliczana jest z układu nieliniowych równań różnicowych:

$$\bar{y}_l^{(m)}(n) = \sum_{h=0}^{H_m} \bar{w}_{lh}^{(m)} \mu_h^{(m)}(n), \quad \mu_h^{(m)}(n) = \sigma(\chi_h^{(m)}(n)), \quad (2.1)$$

$$\chi_h^{(m)}(n) = \sum_{s=0}^{S_m} \sum_{d=1}^{D_u^{(m)}} \tilde{w}_{hsd}^{(m)} u_s^{(m)}(n-d) + \sum_{d=1}^{D_\mu^{(m)}} \hat{w}_{hd}^{(m)} \mu_h^{(m)}(n-d), \quad (2.2)$$

gdzie  $\bar{y}_l^{(m)}(n)$  ( $l = 1, 2, \dots, L_m$ ) – jest  $l$ -tym wyjściem lokalnego modelu neuronowego w takcie  $n$ -tym, będące zarazem wyjściem  $l$ -tego neuronu warstwy wyjściowej,  $\mu_h^{(m)}(n)$  ( $h = 0, 1, \dots, H_m$ ) – wyjście  $h$ -tego neuronu warstwy ukrytej w  $n$ -tym takcie, przy czym  $\mu_0^{(m)}(n) = 1$  dla każdego  $n$ ,  $\bar{w}_{lh}^{(m)}$  ( $l = 1, 2, \dots, L_m$ ;  $h = 0, 1, \dots, H_m$ ) – waga połączenia między  $l$ -tym neuronem warstwy wyjściowej a  $h$ -tym neuronem warstwy ukrytej,  $\tilde{w}_{hsd}^{(m)}$  ( $h = 1, 2, \dots, H_m$ ;  $s = 0, 1, \dots, S_m$ ;  $d = 1, 2, \dots, D_u^{(m)}$ ) – waga określająca wpływ wartości  $s$ -tego wejścia sieci neuronowej sprzed  $d$  taktów na wyjście  $h$ -tego neuronu warstwy ukrytej,  $\hat{w}_{hd}^{(m)}$  ( $h = 1, 2, \dots, H_m$ ;  $d = 1, 2, \dots, D_\mu^{(m)}$ ) – waga określająca wpływ wartości wyjścia  $h$ -tego neuronu warstwy ukrytej sprzed  $d$  taktów na swoje własne wyjście w takcie bieżącym. Wagi  $\bar{w}_{l0}^{(m)}$  ( $l = 1, 2, \dots, L_m$ ) oraz  $\tilde{w}_{h0d}$  ( $h = 1, 2, \dots, H_m$ ;

$d = 1, 2, \dots, D_u^{(m)}$ ) pełnią rolę parametrów swobodnych, odpowiednio części liniowej neuronów warstwy wyjściowej oraz części liniowej neuronów warstwy ukrytej.

Wektory wag  $l$ -tego neuronu warstwy wyjściowej i  $h$ -tego neuronu warstwy ukrytej  $m$ -tej sieci to, odpowiednio:

$$\bar{\mathbf{w}}_l^{(m)} = \begin{bmatrix} \bar{w}_{l0}^{(m)} \\ \bar{w}_{l1}^{(m)} \\ \vdots \\ \bar{w}_{lH_m}^{(m)} \end{bmatrix} \in \mathcal{R}^{H_m+1}, \quad \hat{\mathbf{w}}_h^{(m)} = \begin{bmatrix} \hat{w}_{h1}^{(m)} \\ \hat{w}_{h2}^{(m)} \\ \vdots \\ \hat{w}_{hD_\mu}^{(m)} \end{bmatrix} \in \mathcal{R}^{D_\mu^{(m)}},$$

$$\tilde{\mathbf{w}}_h^{(m)} = \begin{bmatrix} \tilde{w}_{h0}^{(m)} \\ \tilde{w}_{h1}^{(m)} \\ \vdots \\ \tilde{w}_{hS_m}^{(m)} \end{bmatrix} \in \mathcal{R}^{(S_m+1) \cdot D_u^{(m)}}, \quad \text{gdzie} \quad \tilde{\mathbf{w}}_{hs}^{(m)} = \begin{bmatrix} \tilde{w}_{hs1}^{(m)} \\ \tilde{w}_{hs2}^{(m)} \\ \vdots \\ \tilde{w}_{hsD_u}^{(m)} \end{bmatrix} \in \mathcal{R}^{D_u^{(m)}}.$$

Wszystkie wagi warstwy wyjściowej i ukrytej  $m$ -tej sieci można reprezentować przy użyciu poniższych wektorów:

$$\bar{\mathbf{w}}^{(m)} = \begin{bmatrix} \bar{\mathbf{w}}_1^{(m)} \\ \bar{\mathbf{w}}_2^{(m)} \\ \vdots \\ \bar{\mathbf{w}}_{L_m}^{(m)} \end{bmatrix} \in \mathcal{R}^{L_m \cdot (H_m+1)}, \quad \hat{\mathbf{w}}^{(m)} = \begin{bmatrix} \hat{\mathbf{w}}_1^{(m)} \\ \hat{\mathbf{w}}_2^{(m)} \\ \vdots \\ \hat{\mathbf{w}}_{H_m}^{(m)} \end{bmatrix} \in \mathcal{R}^{H_m \cdot D_\mu^{(m)}},$$

$$\tilde{\mathbf{w}}^{(m)} = \begin{bmatrix} \tilde{\mathbf{w}}_1^{(m)} \\ \tilde{\mathbf{w}}_2^{(m)} \\ \vdots \\ \tilde{\mathbf{w}}_{H_m}^{(m)} \end{bmatrix} \in \mathcal{R}^{H_m \cdot (S_m+1) \cdot D_u^{(m)}},$$

a wówczas wszystkie parametry wagowe  $m$ -tej sieci można umieścić w jednym wektorze:

$$\mathbf{w}^{(m)} = \begin{bmatrix} \bar{\mathbf{w}}^{(m)} \\ \hat{\mathbf{w}}^{(m)} \\ \tilde{\mathbf{w}}^{(m)} \end{bmatrix} \in \mathcal{R}^{L_m \cdot (H_m+1) + H_m \cdot D_\mu^{(m)} + H_m \cdot (S_m+1) \cdot D_u^{(m)}}.$$

Po połączeniu wszystkich sieci w strukturę kaskadową, jak na rys. 2.1, otrzymujemy sieć złożoną, której parametry zbieramy w postaci wektora, składającego



się z parametrów wszystkich  $M$  sieci wchodzących w skład sieci złożonej:

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}^{(1)} \\ \mathbf{w}^{(2)} \\ \vdots \\ \mathbf{w}^{(M)} \end{bmatrix} \in \mathcal{R}^{\sum_{m=1}^M L_m \cdot (H_m + 1) + H_m \cdot D_\mu^{(m)} + H_m \cdot (S_m + 1) \cdot D_u^{(m)}}. \quad (2.3)$$

Jak wspomniano wcześniej, skupiamy się na zadaniu identyfikacji na bieżąco, w związku z tym zakładamy znajomość pomiarów wejść i wyjść w  $n$ -tym takcie wszystkich elementów systemu złożonego:

$$\mathbf{u}^{(m)}(n), \quad \mathbf{y}^{(m)}(n), \quad m = 1, 2, \dots, M. \quad (2.4)$$

W zadaniu identyfikacji lokalnej oceniamy każdą sieć niezależnie, przyjmując w kryterium (1.83) kwadratową miarę błędu:

$$Q_L^{(m)}(n, \mathbf{w}^{(m)}) = \frac{1}{2} \sum_{l=1}^{L_m} [e_l^{(m)}(n)]^2, \quad e_l^{(m)}(n) = y_l^{(m)}(n) - \bar{y}_l^{(m)}(n). \quad (2.5)$$

Wyprowadzimy teraz algorytm  $\Psi_L^{(m)}$  uczenia  $m$ -tej ( $m = 1, 2, \dots, M$ ) sieci, przybliżający w kolejnych iteracjach optymalne wartości wag  $\mathbf{w}_L^{(m)*}$ , minimalizujące wskaźnik jakości (2.5). Zgodnie z metodą gradientu prostego, w  $n$ -tym takcie, po dokonaniu kolejnych pomiarów, aktualizujemy wartości wag  $m$ -tej sieci według następującej reguły:

$$\mathbf{w}^{(m)}(n+1) = \mathbf{w}^{(m)}(n) - \eta^{(m)} \nabla_{\mathbf{w}^{(m)}} Q_L^{(m)}(n, \mathbf{w}^{(m)}) \Big|_{\mathbf{w}^{(m)} = \mathbf{w}^{(m)}(n)}, \quad (2.6)$$

gdzie  $\eta^{(m)}$  jest współczynnikiem uczenia dobranym dla  $m$ -tej sieci. Wartości  $\mathbf{w}^{(m)}(0)$  zwykle wybierane są losowo.

Gradient z równania (2.6) przyjmuje ogólną postać:

$$\begin{aligned} \nabla_{\mathbf{w}^{(m)}} Q_L^{(m)}(n, \mathbf{w}^{(m)}) &= \frac{1}{2} \sum_{l=1}^{L_m} 2e_l^{(m)}(n) \cdot \nabla_{\mathbf{w}^{(m)}} e_l^{(m)}(n) = \\ &= - \sum_{l=1}^{L_m} e_l^{(m)}(n) \cdot \nabla_{\mathbf{w}^{(m)}} \bar{y}_l^{(m)}(n). \end{aligned} \quad (2.7)$$

Składowe tego gradientu, czyli pochodne cząstkowe względem odpowiednich wag, mają różną postać, w zależności od typu wagi. Wagi neuronów warstwy wyjściowej  $\bar{w}_{ij}^{(m)}$  ( $i = 1, 2, \dots, L_m, j = 0, 1, \dots, H_m$ ) wyliczamy następująco:

$$\frac{\partial \bar{y}_l^{(m)}(n)}{\partial \bar{w}_{ij}^{(m)}} = \begin{cases} \sigma(\chi_j^{(m)}(n)) = \mu_j^{(m)}(n) & \text{dla } l = i \\ 0 & \text{w przec.} \end{cases} \quad (2.8)$$

Pochodne cząstkowe wag  $\tilde{w}_{ijp}^{(m)}$  ( $i = 1, 2, \dots, H_m, j = 0, 1, \dots, S_m, p = 1, 2, \dots, D_u^{(m)}$ ) są równe:

$$\frac{\partial \bar{y}_l^{(m)}(n)}{\partial \tilde{w}_{ijp}^{(m)}} = \bar{w}_{li}^{(m)} \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \tilde{w}_{ijp}^{(m)}} \quad (2.9)$$

natomiast uszeregowane pochodne cząstkowe z powyższego równania wyznaczamy w drodze następujących przekształceń:

$$\begin{aligned} \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \tilde{w}_{ijp}^{(m)}} &= \frac{\partial \mu_i^{(m)}(n)}{\partial \tilde{w}_{ijp}^{(m)}} + \sum_{k=1}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} \frac{\partial \mu_i^{(m)}(n-k)}{\partial \tilde{w}_{ijp}^{(m)}} = \\ &= \frac{d\sigma(\chi_i^{(m)}(n))}{d\chi_i^{(m)}(n)} \frac{\partial \chi_i^{(m)}(n)}{\partial \tilde{w}_{ijp}^{(m)}} + \sum_{k=1}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} \frac{d\sigma(\chi_i^{(m)}(n-k))}{d\chi_i^{(m)}(n-k)} \frac{\partial \chi_i^{(m)}(n-k)}{\partial \tilde{w}_{ijp}^{(m)}}, \end{aligned}$$

co ostatecznie daje wzór:

$$\frac{\partial^+ \mu_i^{(m)}(n)}{\partial \tilde{w}_{ijp}^{(m)}} = \sum_{k=0}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} \sigma'(\chi_i^{(m)}(n-k)) u_j^{(m)}(n-p-k). \quad (2.10)$$

Pochodne cząstkowe wag  $\hat{w}_{ij}^{(m)}$  ( $i = 1, 2, \dots, H_m, j = 1, 2, \dots, D_\mu^{(m)}$ ) są równe:

$$\frac{\partial \bar{y}_l^{(m)}(n)}{\partial \hat{w}_{ij}^{(m)}} = \bar{w}_{li}^{(m)} \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \hat{w}_{ij}^{(m)}}, \quad (2.11)$$

natomiast uszeregowane pochodne cząstkowe z tego równania wyznaczamy w drodze poniższych przekształceń:

$$\begin{aligned} \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \hat{w}_{ij}^{(m)}} &= \frac{\partial \mu_i^{(m)}(n)}{\partial \hat{w}_{ij}^{(m)}} + \sum_{k=1}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} \frac{\partial \mu_i^{(m)}(n-k)}{\partial \hat{w}_{ij}^{(m)}} = \\ &= \frac{d\sigma(\chi_i^{(m)}(n))}{d\chi_i^{(m)}(n)} \frac{\partial \chi_i^{(m)}(n)}{\partial \hat{w}_{ij}^{(m)}} + \sum_{k=1}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} \frac{d\sigma(\chi_i^{(m)}(n-k))}{d\chi_i^{(m)}(n-k)} \frac{\partial \chi_i^{(m)}(n-k)}{\partial \hat{w}_{ij}^{(m)}}, \end{aligned}$$

co ostatecznie daje wzór:

$$\frac{\partial^+ \mu_i^{(m)}(n)}{\partial \hat{w}_{ij}^{(m)}} = \sum_{k=0}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} \sigma'(\chi_i^{(m)}(n-k)) \mu_i^{(m)}(n-j-k). \quad (2.12)$$

W powyższych wyprowadzeniach pojawiły się dodatkowo uszeregowane pochodne cząstkowe  $\frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)}$ . Wyznaczamy je następująco:

$$\begin{aligned} \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} &= \frac{\partial \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} + \sum_{\tau=1}^{\min\{D_\mu^{(m)}, k-1\}} \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k+\tau)} \frac{\partial \mu_i^{(m)}(n-k+\tau)}{\partial \mu_i^{(m)}(n-k)} = \\ &= \frac{d\sigma(\chi_i^{(m)}(n))}{d\chi_i^{(m)}(n)} \frac{\partial \chi_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} + \sum_{\tau=1}^{\min\{D_\mu^{(m)}, k-1\}} \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k+\tau)} \cdot \\ &\cdot \frac{d\sigma(\chi_i^{(m)}(n-k+\tau))}{d\chi_i^{(m)}(n-k+\tau)} \frac{\partial \chi_i^{(m)}(n-k+\tau)}{\partial \mu_i^{(m)}(n-k)}, \end{aligned}$$

co ostatecznie daje wzór:

$$\begin{aligned} \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} &= \sigma'(\chi_i^{(m)}(n)) \hat{w}_{ik}^{(m)} \cdot \mathbf{1}(k \leq D_\mu^{(m)}) + \\ &+ \sum_{\tau=1}^{\min\{D_\mu^{(m)}, k-1\}} \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k+\tau)} \sigma'(\chi_i^{(m)}(n-k+\tau)) \hat{w}_{i\tau}^{(m)}. \end{aligned} \quad (2.13)$$

Uwzględniając (2.13) w wynikach przekształceń (2.10) i (2.12), wstawionych następnie do, odpowiednio (2.9) i (2.11), dostajemy wyrażenia na pochodne cząstkowe  $\frac{\partial \bar{y}_l^{(m)}(n)}{\partial \tilde{w}_{ijp}^{(m)}}$  i  $\frac{\partial \bar{y}_l^{(m)}(n)}{\partial \hat{w}_{ij}^{(m)}}$ . Reguły aktualizacji wag są – według wzoru (2.7) rozpisane na poszczególne składowe – następujące:

$$\tilde{w}_{ijp}^{(m)}(n+1) = \tilde{w}_{ijp}^{(m)}(n) + \eta^{(m)} \sum_{l=1}^{L_m} e_l^{(m)}(n) \frac{\partial \bar{y}_l^{(m)}(n)}{\partial \tilde{w}_{ijp}^{(m)}}, \quad (2.14)$$

$$\hat{w}_{ij}^{(m)}(n+1) = \hat{w}_{ij}^{(m)}(n) + \eta^{(m)} \sum_{l=1}^{L_m} e_l^{(m)}(n) \frac{\partial \bar{y}_l^{(m)}(n)}{\partial \hat{w}_{ij}^{(m)}}, \quad (2.15)$$

natomiast – według wzoru (2.8) – reguła aktualizacji wag warstwy wyjściowej ma postać:

$$\bar{w}_{ij}^{(m)}(n+1) = \bar{w}_{ij}^{(m)}(n) + \eta^{(m)} e_i^{(m)}(n) \frac{\partial \bar{y}_i^{(m)}(n)}{\partial \bar{w}_{ij}^{(m)}}. \quad (2.16)$$

Powyższe reguły, po wstawieniu do nich wzorów na pochodne cząstkowe, tworzą algorytm BPTT uczenia  $m$ -tej sieci DRNN modelu lokalnie optymalnego systemu złożonego o strukturze kaskadowej:

$$\begin{aligned} \tilde{w}_{ijp}^{(m)}(n+1) &= \tilde{w}_{ijp}^{(m)}(n) + \eta^{(m)} \sum_{l=1}^{L_m} e_l^{(m)}(n) \cdot \bar{w}_{li}^{(m)} \sigma'(\chi_i^{(m)}(n)) u_j^{(m)}(n-p) + \\ &+ \sum_{k=1}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} \sigma'(\chi_i^{(m)}(n-k)) u_j^{(m)}(n-p-k), \end{aligned} \quad (2.17)$$

$$\begin{aligned} \hat{w}_{ij}^{(m)}(n+1) &= \hat{w}_{ij}^{(m)}(n) + \eta^{(m)} \sum_{l=1}^{L_m} e_l^{(m)}(n) \cdot \bar{w}_{li}^{(m)} \sigma'(\chi_i^{(m)}(n)) \mu_i^{(m)}(n-j) + \\ &+ \sum_{k=1}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} \sigma'(\chi_i^{(m)}(n-k)) \mu_i^{(m)}(n-j-k), \end{aligned} \quad (2.18)$$

$$\bar{w}_{ij}^{(m)}(n+1) = \bar{w}_{ij}^{(m)}(n) + \eta^{(m)} e_i^{(m)}(n) \mu_j^{(m)}(n). \quad (2.19)$$

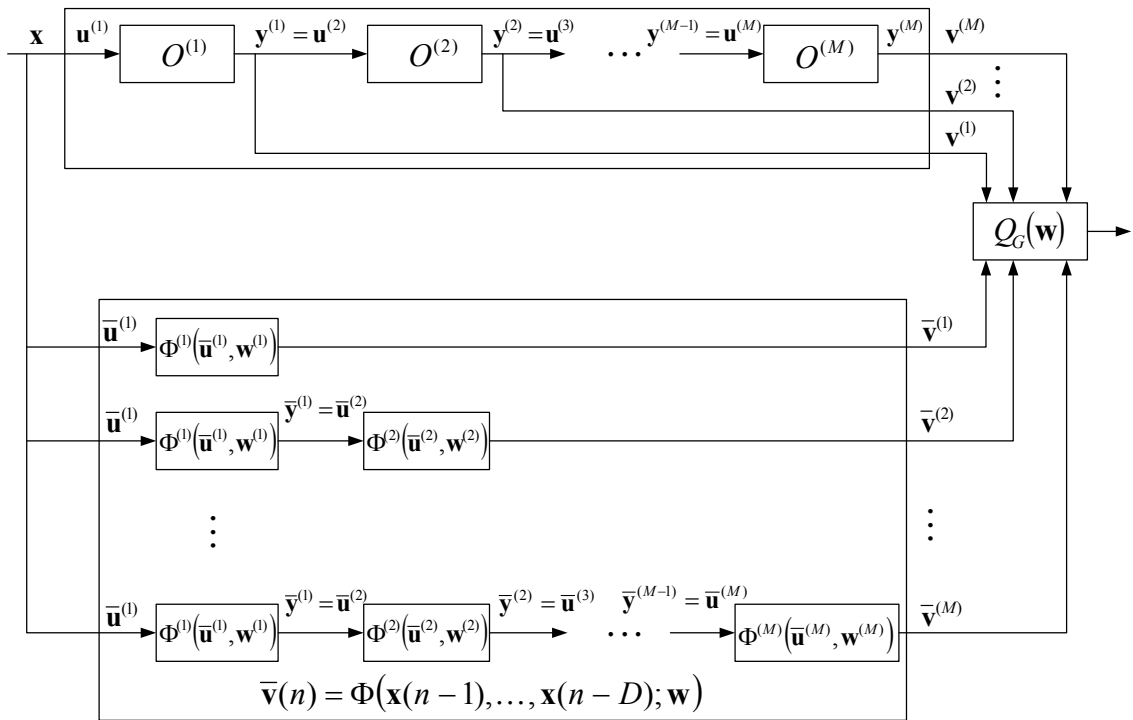
Wykonanie tych obliczeń dla  $m = 1, 2, \dots, M$  daje nam przybliżenie wektora parametrów  $\mathbf{w}_L^*$  lokalnie optymalnego modelu neuronowego systemu złożonego.

## 2.2 Model globalnie optymalny

Rozważmy ponownie system złożony o strukturze kaskadowej modelowany w poprzednim punkcie. Tym razem utworzymy model globalny, tj. taki, którego struktura i sposób przesyłania sygnałów są dokładnie takie jak w modelowanym systemie. Oznacza to uwzględnienie, że  $m$ -ta sieć w kaskadzie (za wyjątkiem pierwszej) otrzymuje sygnał wejściowy nie z  $m$ -tego elementu, jak poprzednio, ale z  $(m-1)$ -szej sieci. Używamy więc do oznaczenia wartości wyróżnionych wyjść  $m$ -tej sieci, składającej się na model globalny, symbolu  $\bar{\mathbf{v}}^{(m)}$ . Natomiast oznaczenie tych wyjść przez  $\bar{\mathbf{y}}^{(m)}$  wskazywać będzie, że wartości te uzyskano przez podanie na wejścia modelu dokładnie tych samych sygnałów  $\mathbf{u}^{(m)}$ , którymi pobudzono  $m$ -ty element systemu złożonego. Na ogół  $\bar{\mathbf{v}}^{(m)} \neq \bar{\mathbf{y}}^{(m)}$ . Rysunek 2.2 przedstawia ideę modelu globalnie optymalnego dla systemu o strukturze kaskadowej.

Jako wyjścia globalne  $\mathbf{v}$  – w najogólniejszym przypadku – wybierzemy wyjścia wszystkich elementów systemu złożonego (macierz  $\mathbf{C}$  z (1.76) jest macierzą jednostkową). Wejściami zewnętrznymi  $\mathbf{x}$  są wszystkie wejścia pierwszego elementu w kaskadzie. Model  $m$ -tego ( $m = 1, 2, \dots, M$ ) elementu w systemie kaskadowym możemy przedstawić w postaci rekurencyjnej:

$$\bar{\mathbf{v}}^{(m)} = \Phi_m \left( \bar{\mathbf{v}}^{(m-1)}, \mathbf{w}^{(m)} \right), \quad \bar{\mathbf{v}}^{(0)} \equiv \mathbf{x}. \quad (2.20)$$



Rysunek 2.2: System złożony o strukturze kaskadowej z modelem globalnym

Przyjmując oznaczenie  $\bar{\mathbf{u}}^{(m)} = \bar{\mathbf{v}}^{(m-1)}$ , model  $m$ -tego elementu przyjmuje postać:

$$\bar{\mathbf{v}}^{(m)}(n) = \Phi_m \left( \bar{\mathbf{u}}^{(m)}(n-1), \dots, \bar{\mathbf{u}}^{(m)}(n-D); \mathbf{w}^{(m)} \right), \quad \bar{\mathbf{u}}^{(1)} \equiv \mathbf{x}. \quad (2.21)$$

W zadaniu identyfikacji globalnej na bieżąco zakładamy znajomość pomiarów wejść zewnętrznych i wyjść globalnych w  $n$ -tym takcie:

$$\mathbf{x}(n), \quad \mathbf{v}(n). \quad (2.22)$$

Wyprowadzenie algorytmu uczenia sieci rekurencyjnych opartego o BPTT do modelowania globalnie optymalnego jest zadaniem skomplikowanym. Aby czytelnie przedstawić własności opracowanego algorytmu, przedstawimy go najpierw

dla prostego przypadku systemu złożonego z dwóch elementów SISO, modelowanego z wykorzystaniem dwóch klasycznych modeli liniowych .

**Przykład - złożony system liniowy.** Rozpatrzmy system złożony o strukturze kaskadowej, w skład którego wchodzi dwa elementy SISO ( $M = 2$ ). Wejściem zewnętrznym systemu jest wejście pierwszego elementu, a jako wyjścia globalne przyjmujemy wyjścia obu elementów:

$$\mathbf{v} = \begin{bmatrix} v^{(1)} \\ v^{(2)} \end{bmatrix}.$$

Modelem pierwszego elementu jest liniowe równanie różnicowe:

$$\bar{v}^{(1)}(n) = \sum_{d=1}^{D_u^{(1)}} \tilde{w}_d^{(1)} x(n-d) + \sum_{d=1}^{D_y^{(1)}} \hat{w}_d^{(1)} \bar{v}^{(1)}(n-d). \quad (2.23)$$

Model drugiego elementu pochodzi z tej samej klasy:

$$\bar{v}^{(2)}(n) = \sum_{d=1}^{D_u^{(2)}} \tilde{w}_d^{(2)} \bar{v}^{(1)}(n-d) + \sum_{d=1}^{D_y^{(2)}} \hat{w}_d^{(2)} \bar{v}^{(2)}(n-d). \quad (2.24)$$

Parametry  $m$ -tego ( $m = 1, 2$ ) modelu umieszczamy w wektorze  $\mathbf{w}^{(m)}$ :

$$\mathbf{w}^{(m)} = \left[ \tilde{w}_1^{(m)} \quad \dots \quad \tilde{w}_{D_u^{(m)}}^{(m)} \quad \hat{w}_1^{(m)} \quad \dots \quad \hat{w}_{D_y^{(m)}}^{(m)} \right]^T,$$

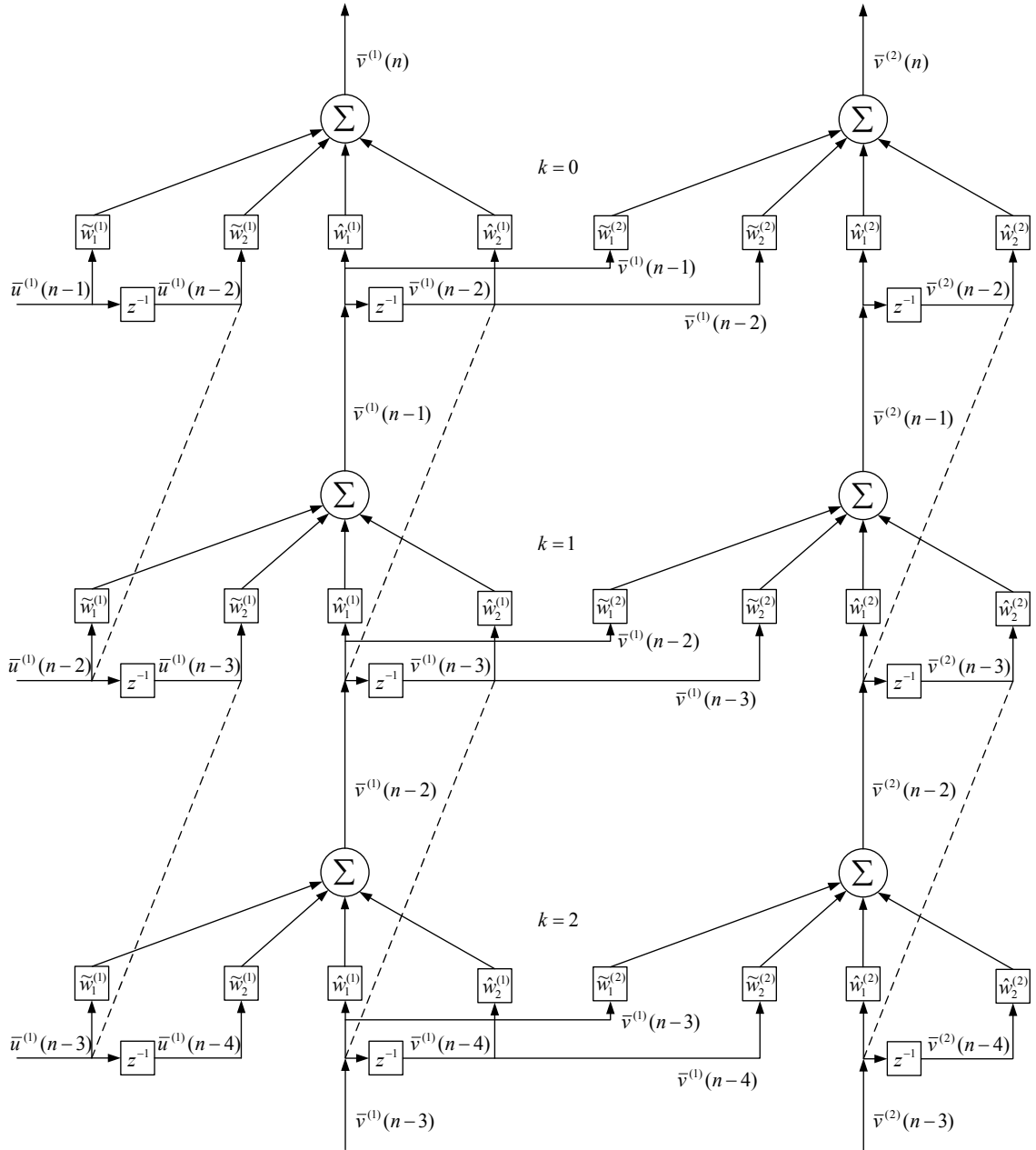
a parametry modelu globalnego zbieramy w wektor:  $\mathbf{w} = \begin{bmatrix} \mathbf{w}^{(1)} \\ \mathbf{w}^{(2)} \end{bmatrix}$ .

Model systemu złożonego powstały z szeregowego połączenia tych dwóch modeli, oceniamy przy pomocy globalnego wskaźnika jakości (1.93), któremu tutaj nadajemy postać:

$$Q_G(n, \mathbf{w}) = \frac{1}{2} \sum_{m=1}^2 \beta_m \left[ \varepsilon^{(m)}(n) \right]^2, \quad \varepsilon^{(m)}(n) = v^{(m)}(n) - \bar{v}^{(m)}(n), \quad (2.25)$$

gdzie  $\beta_1, \beta_2$  to współczynniki przyjmujące wartości rzeczywiste z przedziału  $[0, 1]$  i spełniające warunek  $\beta_1 + \beta_2 = 1$ . Określają one wagę, jaką w globalnym wskaźniku jakości mają błędy na wyjściach poszczególnych modeli. Wprowadzenie nowego oznaczenia błędu na wyjściu tzn.  $\varepsilon^{(m)}$ , motywowane jest tym, że w modelu globalnym na wielkość błędu  $m$ -tego modelu pośredni wpływ mają błędy na wyjściach modeli poprzedzających. W przypadku modelu lokalnego, gdzie wartości

wyjść wyliczane są jako odpowiedź na dokładnie te same sygnały wejściowe, które podawano na  $m$ -ty obiekt, na błąd  $e^{(m)}$  nie mają żadnego wpływu błędy popełniane przez inne modele. Model globalny, rozwinięty dwa kroki wstecz w czasie, zilustrowany jest na rys. 2.3.



Rysunek 2.3: Model złożony systemu liniowego rozwinięty wstecz w czasie dla  $K = 2$

Algorytm identyfikacji  $\Psi_G$ , przybliżający w kolejnych iteracjach optymalne

wartości parametrów  $\mathbf{w}_G^*$  modeli liniowych, dla których wskaźnik (2.25) przyjmuje wartość najmniejszą, skonstruujemy w oparciu o metodę gradientu prostego. Zgodnie z nią w  $n$ -tym takcie, po dokonaniu kolejnych pomiarów, aktualizujemy wartości parametrów  $m$ -tego ( $m = 1, 2$ ) modelu liniowego według wzoru:

$$\mathbf{w}^{(m)}(n+1) = \mathbf{w}^{(m)}(n) - \eta^{(m)} \nabla_{\mathbf{w}^{(m)}} Q_G(n, \mathbf{w})|_{\mathbf{w}^{(m)}=\mathbf{w}^{(m)}(n)}, \quad (2.26)$$

Gradient  $\nabla_{\mathbf{w}^{(m)}} Q_G(n, \mathbf{w})$  ma ogólną postać:

$$\begin{aligned} \nabla_{\mathbf{w}^{(m)}} Q_G(n, \mathbf{w}) &= \sum_{r=1}^2 \beta_r \varepsilon^{(r)}(n) \nabla_{\mathbf{w}^{(m)}} \varepsilon^{(r)}(n) = \\ &= - \sum_{r=1}^2 \beta_r \varepsilon^{(r)}(n) \nabla_{\mathbf{w}^{(m)}} \bar{v}^{(r)}(n), \end{aligned} \quad (2.27)$$

natomiast jego składowe, czyli pochodne cząstkowe względem poszczególnych parametrów mają postać zależną od typu tych parametrów. Z uwagi na dynamikę wewnętrzną modelu są to uszeregowane pochodne cząstkowe. Pochodne wyjścia pierwszego obiektu po jego parametrach  $\tilde{w}_i^{(1)}$  ( $i = 1, 2, \dots, D_u^{(1)}$ ) wyznaczamy następująco:

$$\begin{aligned} \frac{\partial^+ \bar{v}^{(1)}(n)}{\partial \tilde{w}_i^{(1)}} &= \frac{\partial \bar{v}^{(1)}(n)}{\partial \tilde{w}_i^{(1)}} + \sum_{k=1}^K \frac{\partial^+ \bar{v}^{(1)}(n)}{\partial \bar{v}^{(1)}(n-k)} \frac{\partial \bar{v}^{(1)}(n-k)}{\partial \tilde{w}_i^{(1)}} = \\ &= x(n-i) + \sum_{k=1}^K \frac{\partial^+ \bar{v}^{(1)}(n)}{\partial \bar{v}^{(1)}(n-k)} x(n-i-k). \end{aligned} \quad (2.28)$$

Uszeregowane pochodne cząstkowe ze względu na drugi typ parametrów, tj.  $\hat{w}_i^{(1)}$  ( $i = 1, 2, \dots, D_y^{(1)}$ ) wyznaczamy podobnie:

$$\begin{aligned} \frac{\partial^+ \bar{v}^{(1)}(n)}{\partial \hat{w}_i^{(1)}} &= \frac{\partial \bar{v}^{(1)}(n)}{\partial \hat{w}_i^{(1)}} + \sum_{k=1}^K \frac{\partial^+ \bar{v}^{(1)}(n)}{\partial \bar{v}^{(1)}(n-k)} \frac{\partial \bar{v}^{(1)}(n-k)}{\partial \hat{w}_i^{(1)}} = \\ &= \bar{v}^{(1)}(n-i) + \sum_{k=1}^K \frac{\partial^+ \bar{v}^{(1)}(n)}{\partial \bar{v}^{(1)}(n-k)} \bar{v}^{(1)}(n-i-k). \end{aligned} \quad (2.29)$$

Uszeregowane pochodne cząstkowe  $\frac{\partial^+ \bar{v}^{(1)}(n)}{\partial \bar{v}^{(1)}(n-k)}$ , które pojawiły się w powyższych



przekształceniach, wyliczamy następująco:

$$\begin{aligned}
 \frac{\partial^+ \bar{v}^{(1)}(n)}{\partial \bar{v}^{(1)}(n-k)} &= \frac{\partial \bar{v}^{(1)}(n)}{\partial \bar{v}^{(1)}(n-k)} + \sum_{\tau=1}^{\min\{D_y^{(1)}, k-1\}} \frac{\partial^+ \bar{v}^{(1)}(n)}{\partial \bar{v}^{(1)}(n-k+\tau)} \frac{\partial \bar{v}^{(1)}(n-k+\tau)}{\partial \bar{v}^{(1)}(n-k)} = \\
 &= \hat{w}_k^{(1)} \cdot \mathbf{1}(k \leq D_y^{(1)}) + \sum_{\tau=1}^{\min\{D_y^{(1)}, k-1\}} \frac{\partial^+ \bar{v}^{(1)}(n)}{\partial \bar{v}^{(1)}(n-k+\tau)} \cdot \hat{w}_\tau^{(1)}.
 \end{aligned} \tag{2.30}$$

Analogiczne przekształcenia wykonamy dla parametrów modelu drugiego elementu. Uszeregowane pochodne cząstkowe wyjścia drugiego modelu po jego parametrach  $\tilde{w}_i^{(2)}$ , ( $i = 1, 2, \dots, D_u^{(2)}$ ) wyznaczamy następująco:

$$\begin{aligned}
 \frac{\partial^+ \bar{v}^{(2)}(n)}{\partial \tilde{w}_i^{(2)}} &= \frac{\partial \bar{v}^{(2)}(n)}{\partial \tilde{w}_i^{(2)}} + \sum_{k=1}^K \frac{\partial^+ \bar{v}^{(2)}(n)}{\partial \bar{v}^{(2)}(n-k)} \frac{\partial \bar{v}^{(2)}(n-k)}{\partial \tilde{w}_i^{(2)}} = \\
 &= \bar{v}^{(1)}(n-i) + \sum_{k=1}^K \frac{\partial^+ \bar{v}^{(2)}(n)}{\partial \bar{v}^{(2)}(n-k)} \bar{v}^{(1)}(n-i-k).
 \end{aligned} \tag{2.31}$$

Uszeregowane pochodne cząstkowe ze względu na drugi typ parametrów, tj.  $\hat{w}_i^{(2)}$  ( $i = 1, 2, \dots, D_y^{(2)}$ ) wyznaczamy podobnie:

$$\begin{aligned}
 \frac{\partial^+ \bar{v}^{(2)}(n)}{\partial \hat{w}_i^{(2)}} &= \frac{\partial \bar{v}^{(2)}(n)}{\partial \hat{w}_i^{(2)}} + \sum_{k=1}^K \frac{\partial^+ \bar{v}^{(2)}(n)}{\partial \bar{v}^{(2)}(n-k)} \frac{\partial \bar{v}^{(2)}(n-k)}{\partial \hat{w}_i^{(2)}} = \\
 &= \bar{v}^{(2)}(n-i) + \sum_{k=1}^K \frac{\partial^+ \bar{v}^{(2)}(n)}{\partial \bar{v}^{(2)}(n-k)} \bar{v}^{(2)}(n-i-k).
 \end{aligned} \tag{2.32}$$

Występujące w powyższych przekształceniach uszeregowane pochodne cząstkowe  $\frac{\partial^+ \bar{v}^{(2)}(n)}{\partial \bar{v}^{(2)}(n-k)}$  wyliczamy następująco:

$$\begin{aligned}
 \frac{\partial^+ \bar{v}^{(2)}(n)}{\partial \bar{v}^{(2)}(n-k)} &= \frac{\partial \bar{v}^{(2)}(n)}{\partial \bar{v}^{(2)}(n-k)} + \sum_{\tau=1}^{\min\{D_y^{(2)}, k-1\}} \frac{\partial^+ \bar{v}^{(2)}(n)}{\partial \bar{v}^{(2)}(n-k+\tau)} \frac{\partial \bar{v}^{(2)}(n-k+\tau)}{\partial \bar{v}^{(2)}(n-k)} = \\
 &= \hat{w}_k^{(2)} \cdot \mathbf{1}(k \leq D_y^{(2)}) + \sum_{\tau=1}^{\min\{D_y^{(2)}, k-1\}} \frac{\partial^+ \bar{v}^{(2)}(n)}{\partial \bar{v}^{(2)}(n-k+\tau)} \cdot \hat{w}_\tau^{(2)}.
 \end{aligned} \tag{2.33}$$

Wyprowadzone dotychczas formuły są odpowiednikami reguł uczenia wypro-  
 wadzonych w poprzednim punkcie dla modelu lokalnie optymalnego. Z ogólnej  
 postaci gradientu (2.27) wynika, że należy obliczyć jeszcze pochodne wyjścia dru-  
 giego modelu po parametrach modelu pierwszego (pochodne cząstkowe wyjścia  
 pierwszego modelu po parametrach modelu drugiego są równe 0). Tego typu  
 pochodne, nie pojawiające się w algorytmie identyfikacji lokalnej, są charaktery-  
 stycznym elementem algorytmu identyfikacji globalnej.

Wyznamy najpierw postać uszeregowanych pochodnych cząstkowych wyj-  
 ścia drugiego modelu po parametrach  $\tilde{w}_i^{(1)}$  ( $i = 1, 2, \dots, D_u^{(1)}$ ):

$$\begin{aligned} \frac{\partial^{+\bar{v}}(2)(n)}{\partial \tilde{w}_i^{(1)}} &= \frac{\partial \bar{v}^{(2)}(n)}{\partial \tilde{w}_i^{(1)}} + \sum_{k=1}^{K-1} \frac{\partial^{+\bar{v}}(2)(n)}{\partial \bar{v}^{(2)}(n-k)} \frac{\partial \bar{v}^{(2)}(n-k)}{\partial \tilde{w}_i^{(1)}} = \\ &= \sum_{d=1}^{\min\{D_u^{(2)}, K\}} \tilde{w}_d^{(2)} \frac{\partial \bar{v}^{(1)}(n-d)}{\partial \tilde{w}_i^{(1)}} + \sum_{k=1}^{K-1} \frac{\partial^{+\bar{v}}(2)(n)}{\partial \bar{v}^{(2)}(n-k)} \sum_{d=1}^{\min\{D_u^{(2)}, K-k\}} \tilde{w}_d^{(2)} \frac{\partial \bar{v}^{(1)}(n-d-k)}{\partial \tilde{w}_i^{(1)}}, \end{aligned}$$

co prowadzi do wzoru:

$$\begin{aligned} \frac{\partial^{+\bar{v}}(2)(n)}{\partial \tilde{w}_i^{(1)}} &= \sum_{d=1}^{\min\{D_u^{(2)}, K\}} \tilde{w}_d^{(2)} x(n-i-d) + \\ &+ \sum_{k=1}^{K-1} \frac{\partial^{+\bar{v}}(2)(n)}{\partial \bar{v}^{(2)}(n-k)} \sum_{d=1}^{\min\{D_u^{(2)}, K-k\}} \tilde{w}_d^{(2)} x(n-i-d-k) \end{aligned} \quad (2.34)$$

Podobnie wyznaczmy uszeregowane pochodne cząstkowe wyjścia drugiego mo-  
 delu po parametrach  $\hat{w}_i^{(1)}$  ( $i = 1, 2, \dots, D_y^{(1)}$ ):

$$\begin{aligned} \frac{\partial^{+\bar{v}}(2)(n)}{\partial \hat{w}_i^{(1)}} &= \frac{\partial \bar{v}^{(2)}(n)}{\partial \hat{w}_i^{(1)}} + \sum_{k=1}^{K-1} \frac{\partial^{+\bar{v}}(2)(n)}{\partial \bar{v}^{(2)}(n-k)} \frac{\partial \bar{v}^{(2)}(n-k)}{\partial \hat{w}_i^{(1)}} = \\ &= \sum_{d=1}^{\min\{D_u^{(2)}, K\}} \hat{w}_d^{(2)} \frac{\partial \bar{v}^{(1)}(n-d)}{\partial \hat{w}_i^{(1)}} + \sum_{k=1}^{K-1} \frac{\partial^{+\bar{v}}(2)(n)}{\partial \bar{v}^{(2)}(n-k)} \sum_{d=1}^{\min\{D_u^{(2)}, K-k\}} \hat{w}_d^{(2)} \frac{\partial \bar{v}^{(1)}(n-d-k)}{\partial \hat{w}_i^{(1)}}, \end{aligned}$$

co w rezultacie daje formułę:

$$\begin{aligned} \frac{\partial^+ \bar{v}^{(2)}(n)}{\partial \hat{w}_i^{(1)}} &= \sum_{d_2=1}^{\min\{D_u^{(2)}, K\}} \hat{w}_{d_2}^{(2)} \bar{v}^{(1)}(n-i-d_2) + \\ &+ \sum_{k=1}^{K-1} \frac{\partial^+ \bar{v}^{(2)}(n)}{\partial \bar{v}^{(2)}(n-k)} \sum_{d=1}^{\min\{D_u^{(2)}, K-k\}} \hat{w}_d^{(2)} \bar{v}^{(1)}(n-i-d-k). \end{aligned} \quad (2.35)$$

Wstawienie wzoru (2.30) do formuł na uszeregowane pochodne cząstkowe (2.28), (2.29) oraz wzoru (2.33) do formuł na uszeregowane pochodne cząstkowe (2.31), (2.32), a następnie wstawienie formuł (2.28), (2.29), (2.31), (2.32) oraz (2.34), (2.35) jako odpowiednie składowe wektora gradientu we wzorze (2.27), pozwala używać reguły (2.26) jako algorytmu identyfikacji globalnej systemu złożonego modelowanego z wykorzystaniem dwóch modeli liniowych.

W dalszej części pracy pochodne cząstkowe wyjść modelu względem jego własnych parametrów lub wyjść z poprzednich taktów określać będziemy mianem *lokalnych pochodnych cząstkowych*, natomiast pochodne wyjść modelu względem parametrów innego modelu nazywać będziemy *globalnymi pochodnymi cząstkowymi*

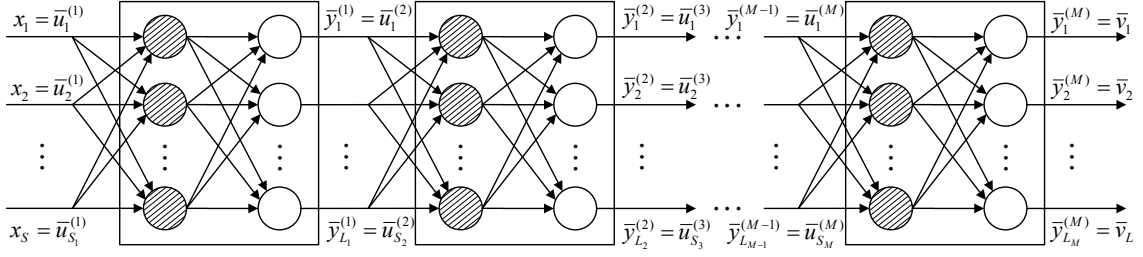
**Rekurencyjna sieć neuronowa.** Powróćmy do systemu złożonego z wielowymiarowych obiektów dynamicznych o strukturze kaskadowej, którego modelami są sieci DRNN. Układ równań różnicowych opisujących  $l$ -te ( $l = 1, 2, \dots, L_m$ ) wyjście  $m$ -tej sieci modelu złożonego jest podobny do (2.1) i (2.2):

$$\bar{v}_l^{(m)}(n) = \sum_{h=0}^{H_m} \bar{w}_{lh}^{(m)} \mu_h^{(m)}(n), \quad \mu_h^{(m)}(n) = \sigma \left( \chi_h^{(m)}(n) \right), \quad (2.36)$$

$$\chi_h^{(m)}(n) = \sum_{s=0}^{S_m} \sum_{d=1}^{D_u^{(m)}} \tilde{w}_{hsd}^{(m)} \bar{u}_s^{(m)}(n-d) + \sum_{d=1}^{D_\mu^{(m)}} \hat{w}_{hd}^{(m)} \mu_h^{(m)}(n-d), \quad (2.37)$$

tzn. w miejsce wyjścia  $\bar{y}_l^{(m)}(n)$  modelu lokalnego pojawia się wyliczane w inny sposób wyjście  $\bar{v}_l^{(m)}(n)$  oraz zamiast wejść lokalnych  $\mathbf{u}^{(m)}$  występują wejścia  $\bar{\mathbf{u}}^{(m)}(n) = \bar{\mathbf{v}}^{(m-1)}(n)$  będące wyjściami sieci poprzedniej. Sieć złożoną dla systemu o strukturze kaskadowej przedstawia rys. 2.4.

Wagi sieci złożonej można zebrać w wektor  $\mathbf{w}$  o postaci (2.3), z identycznymi składowymi jak wprowadzony w rozdziale 2.1.



Rysunek 2.4: Złożona sieć DRNN modelująca system o strukturze kaskadowej

Globalny wskaźnik jakości ma teraz postać:

$$Q_G(n, \mathbf{w}) = \frac{1}{2} \sum_{m=1}^M \beta_m \sum_{l=1}^{L_m} [\varepsilon_l^{(m)}(n)]^2, \quad (2.38)$$

gdzie  $\varepsilon_l^{(m)}(n) = v_l^{(m)}(n) - \bar{v}_l^{(m)}(n)$ ,  $\beta_m$  ( $m = 1, 2, \dots, M$ ) – to współczynniki liczbowe z przedziału  $[0, 1]$  spełniające warunek:

$$\sum_{m=1}^M \beta_m = 1. \quad (2.39)$$

Przedstawimy teraz pełne wyprowadzenie algorytmu  $\Psi_G$  uczenia na bieżąco złożonej sieci DRNN, dający w kolejnych iteracjach przybliżenia optymalnych wartości wag  $\mathbf{w}_G^*$ , tj. takie, dla których wskaźnik jakości (2.38) przyjmuje wartość najmniejszą. Zgodnie z metodą gradientu prostego w  $n$ -tym takcie, po dokonaniu kolejnych pomiarów (2.22), aktualizujemy wartości parametrów wagowych  $m$ -tej ( $m = 1, 2, \dots, M$ ) sieci według reguły (2.40):

$$\mathbf{w}^{(m)}(n+1) = \mathbf{w}^{(m)}(n) - \eta^{(m)} \nabla_{\mathbf{w}^{(m)}} Q_G(n, \mathbf{w})|_{\mathbf{w}^{(m)} = \mathbf{w}^{(m)}(n)}. \quad (2.40)$$

Gradient w (2.40) ma ogólną postać:

$$\nabla_{\mathbf{w}^{(m)}} Q_G(n, \mathbf{w}) = -\frac{1}{2} \sum_{r=1}^M \beta_r \sum_{l=1}^{L_r} 2\varepsilon_l^{(r)}(n) \nabla_{\mathbf{w}^{(m)}} \varepsilon_l^{(r)}(n) = \quad (2.41)$$

$$= -\sum_{r=1}^M \beta_r \sum_{l=1}^{L_r} \varepsilon_l^{(r)}(n) \nabla_{\mathbf{w}^{(m)}} \bar{v}_l^{(r)}(n).$$

Zauważmy, że dla  $m > r$  zachodzi

$$\nabla_{\mathbf{w}^{(m)}} \bar{v}_l^{(r)}(n) = \mathbf{0}, \quad (2.42)$$

co upraszcza (2.41) do postaci:

$$\nabla_{\mathbf{w}^{(m)}} Q_G(n, \mathbf{w}) = - \sum_{r=m}^M \beta_r \sum_{l=1}^{L_r} \varepsilon_l^{(r)}(n) \nabla_{\mathbf{w}^{(m)}} \bar{v}_l^{(r)}(n). \quad (2.43)$$

Równość (2.42) jest zapisem faktu, że w systemie kaskadowym parametry  $m$ -tego modelu neuronowego nie mają wpływu na wyjścia modeli poprzednich, czyli o numerach mniejszych od  $m$ .

Składowe wektora gradientu  $\nabla_{\mathbf{w}^{(m)}} \bar{v}_l^{(r)}(n)$ , czyli pochodne cząstkowe wyjść  $r$ -tego modelu względem poszczególnych parametrów  $m$ -tego modelu, mają postaci:

$$\frac{\partial \bar{v}_l^{(r)}(n)}{\partial \hat{w}_{ij}^{(m)}} = \sum_{h=1}^{H_r} \bar{w}_{lh}^{(r)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \hat{w}_{ij}^{(m)}}, \quad (2.44)$$

$$\frac{\partial \bar{v}_l^{(r)}(n)}{\partial \tilde{w}_{ijp}^{(m)}} = \sum_{h=1}^{H_r} \bar{w}_{lh}^{(r)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \tilde{w}_{ijp}^{(m)}}, \quad (2.45)$$

$$\frac{\partial \bar{v}_l^{(r)}(n)}{\partial \bar{w}_{ij}^{(m)}} = \sum_{h=1}^{H_r} \bar{w}_{lh}^{(r)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \bar{w}_{ij}^{(m)}}. \quad (2.46)$$

Zacniemy od wyznaczenia lokalnych uszeregowanych pochodnych cząstkowych występujących w powyższych równaniach. Jest to przypadek, w którym zachodzi równość  $m = r$ . Dotyczy to właściwie tylko dwóch równań: (2.44) i (2.45), gdyż dla (2.46) mamy wówczas:

$$\frac{\partial \bar{v}_l^{(m)}(n)}{\partial \bar{w}_{ij}^{(m)}} = \mu_j^{(m)}(n), \quad l = i. \quad (2.47)$$

Uszeregowane pochodne wyjścia  $i$ -tego ( $i = 1, 2, \dots, H_m$ ) neuronu warstwy ukrytej  $m$ -tej sieci względem jego własnych parametrów  $\hat{w}_{ij}^{(m)}$  ( $j = 1, 2, \dots, D_\mu^{(m)}$ ) wyznaczamy następująco:

$$\frac{\partial^+ \mu_i^{(m)}(n)}{\partial \hat{w}_{ij}^{(m)}} = \frac{\partial \mu_i^{(m)}(n)}{\partial \hat{w}_{ij}^{(m)}} + \sum_{k=1}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} \frac{\partial \mu_i^{(m)}(n-k)}{\partial \hat{w}_{ij}^{(m)}} = \quad (2.48)$$

$$= \sum_{k=0}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} \sigma'(\chi_i^{(m)}(n-k)) \mu_i^{(m)}(n-j-k).$$

Uszeregowane pochodne wyjścia  $i$ -tego ( $i = 1, 2, \dots, H_m$ ) neuronu warstwy ukrytej  $m$ -tej sieci względem jego własnych parametrów  $\tilde{w}_{ijp}^{(m)}$  ( $j = 0, 1, \dots, D_u^{(m)}$ ,  $p = 1, 2, \dots, D_\mu^{(m)}$ ) wyznaczamy przekształcając:

$$\begin{aligned}
 \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \tilde{w}_{ijp}^{(m)}} &= \frac{\partial \mu_i^{(m)}(n)}{\partial \tilde{w}_{ijp}^{(m)}} + \sum_{k=1}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} \frac{\partial \mu_i^{(m)}(n-k)}{\partial \tilde{w}_{ijp}^{(m)}} = \\
 &= \sum_{k=0}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} \sigma' \left( \chi_i^{(m)}(n-k) \right) \bar{u}_j^{(m)}(n-p-k),
 \end{aligned} \tag{2.49}$$

gdzie

$$\begin{aligned}
 \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} &= \frac{\partial \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} \cdot \mathbf{1} \left( k \leq D_\mu^{(m)} \right) + \sum_{\tau=1}^{k-1} \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k+\tau)} \frac{\partial \mu_i^{(m)}(n-k+\tau)}{\partial \mu_i^{(m)}(n-k)} = \\
 &= \sigma' \left( \chi_i^{(m)}(n) \right) \hat{w}_{ik}^{(m)} \cdot \mathbf{1} \left( k \leq D_\mu^{(m)} \right) + \\
 &+ \sum_{\tau=1}^{\min\{D_\mu^{(m)}, k-1\}} \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k+\tau)} \sigma' \left( \chi_i^{(m)}(n-k+\tau) \right) \cdot \hat{w}_{i\tau}^{(m)}.
 \end{aligned} \tag{2.50}$$

Rozpatrzmy teraz trudniejszą sytuację, tzn. taką, w której zachodzi nierówność  $m < r$ . Oznacza to, że wyznaczmy postaci globalnych uszeregowanych pochodnych cząstkowych z równań (2.44), (2.45) i (2.46). Pozwalają one określić wpływ wag  $m$ -tej sieci na wyjścia sieci następujących po niej w kaskadzie, czyli o numerach od  $m+1$  do  $M$ .

Zacniemy od wyznaczenia odpowiedniej formuły dla parametrów  $\hat{w}_{ij}^{(m)}$  ( $i = 1, 2, \dots, H_m$ ,  $j = 1, 2, \dots, D_\mu^{(m)}$ ,  $m = 1, 2, \dots, M$ ) sprzężenia zwrotnego  $h$ -tego neuronu warstwy  $r$ -tej:

$$\begin{aligned}
 \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \hat{w}_{ij}^{(m)}} &= \frac{\partial \mu_h^{(r)}(n)}{\partial \hat{w}_{ij}^{(m)}} + \sum_{k=1}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n-k)} \frac{\partial \mu_h^{(r)}(n-k)}{\partial \hat{w}_{ij}^{(m)}} = \\
 &= \frac{d\sigma \left( \chi_h^{(r)}(n) \right)}{d\chi_h^{(r)}(n)} \frac{\partial \chi_h^{(r)}(n)}{\partial \hat{w}_{ij}^{(m)}} + \sum_{k=1}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n-k)} \frac{d\sigma \left( \chi_h^{(r)}(n-k) \right)}{d\chi_h^{(r)}(n-k)} \frac{\partial \chi_h^{(r)}(n-k)}{\partial \hat{w}_{ij}^{(m)}} =
 \end{aligned}$$

$$\begin{aligned}
 &= \sigma' \left( \chi_h^{(r)}(n) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\min\{D_u^{(r)}, K-(r-m)+1\}} \tilde{w}_{hs_r d_r}^{(r)} \frac{\partial \bar{u}_{s_r}^{(r)}(n-d_r)}{\partial \hat{w}_{ij}^{(m)}} + \\
 &+ \sum_{k=1}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n-k)} \sigma' \left( \chi_h^{(r)}(n-k) \right) \cdot \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\min\{D_u^{(r)}, K-(r-m)+1-k\}} \tilde{w}_{hs_r d_r}^{(r)} \frac{\partial \bar{u}_{s_r}^{(r)}(n-d_r-k)}{\partial \hat{w}_{ij}^{(m)}}.
 \end{aligned}$$

Wprowadźmy oznaczenia:

$$\varrho_1 = \min \left\{ D_u^{(r)}, K - (r - m) + 1 \right\},$$

$$\rho_1 = \min \left\{ D_u^{(r)}, K - (r - m) + 1 - k \right\}.$$

Wówczas otrzymamy:

$$\begin{aligned}
 \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \hat{w}_{ij}^{(m)}} &= \sigma' \left( \chi_h^{(r)}(n) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\varrho_1} \tilde{w}_{hs_r d_r}^{(r)} \frac{\partial \bar{v}_{s_r}^{(r-1)}(n-d_r)}{\partial \hat{w}_{ij}^{(m)}} + \\
 &+ \sum_{k=1}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n-k)} \sigma' \left( \chi_h^{(r)}(n-k) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\rho_1} \tilde{w}_{hs_r d_r}^{(r)} \frac{\partial \bar{v}_{s_r}^{(r-1)}(n-d_r-k)}{\partial \hat{w}_{ij}^{(m)}} = \\
 &= \sigma' \left( \chi_h^{(r)}(n) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\varrho_1} \tilde{w}_{hs_r d_r}^{(r)} \sum_{h_{r-1}=1}^{H_{r-1}} \bar{w}_{s_r h_{r-1}}^{(r-1)} \frac{d\sigma \left( \chi_{h_{r-1}}^{(r-1)}(n-d_r) \right)}{d\chi_{h_{r-1}}^{(r-1)}(n-d_r)} \cdot \\
 &\cdot \frac{\partial \chi_{h_{r-1}}^{(r-1)}(n-d_r)}{\partial \hat{w}_{ij}^{(m)}} + \sum_{k=1}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n-k)} \sigma' \left( \chi_h^{(r)}(n-k) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\rho_1} \tilde{w}_{hs_r d_r}^{(r)} \cdot \\
 &\cdot \sum_{h_{r-1}=1}^{H_{r-1}} \bar{w}_{s_r h_{r-1}}^{(r-1)} \frac{d\sigma \left( \chi_{h_{r-1}}^{(r-1)}(n-d_r-k) \right)}{d\chi_{h_{r-1}}^{(r-1)}(n-d_r-k)} \cdot \frac{\partial \chi_{h_{r-1}}^{(r-1)}(n-d_r-k)}{\partial \hat{w}_{ij}^{(m)}} =
 \end{aligned}$$

$$\begin{aligned}
 &= \sigma' \left( \chi_h^{(r)}(n) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\varrho_1} \tilde{w}_{hs_r d_r}^{(r)} \sum_{h_{r-1}=1}^{H_{r-1}} \bar{w}_{s_r h_{r-1}}^{(r-1)} \sigma' \left( \chi_{h_{r-1}}^{(r-1)}(n - d_r) \right) \cdot \\
 &\cdot \sum_{s_{r-1}=1}^{S_{r-1}} \sum_{d_{r-1}=1}^{\min\{D_u^{(r-1)}, K - (r-1-m) + 1 - d_r\}} \tilde{w}_{h_{r-1} s_{r-1} d_{r-1}}^{(r-1)} \frac{\partial \bar{u}_{s_{r-1}}^{(r-1)}(n - d_r - d_{r-1})}{\partial \hat{w}_{ij}^{(m)}} + \\
 &+ \sum_{k=1}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n-k)} \sigma' \left( \chi_h^{(r)}(n-k) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\rho_1} \tilde{w}_{hs_r d_r}^{(r)} \sum_{h_{r-1}=1}^{H_{r-1}} \bar{w}_{s_r h_{r-1}}^{(r-1)} \cdot \\
 &\cdot \sigma' \left( \chi_{h_{r-1}}^{(r-1)}(n - d_r - k) \right) \sum_{s_{r-1}=1}^{S_{r-1}} \sum_{d_{r-1}=1}^{\min\{D_u^{(r-1)}, K - (r-1-m) + 1 - d_r - k\}} \tilde{w}_{h_{r-1} s_{r-1} d_{r-1}}^{(r-1)} \cdot \\
 &\cdot \frac{\partial \bar{u}_{s_{r-1}}^{(r-1)}(n - d_r - d_{r-1} - k)}{\partial \hat{w}_{ij}^{(m)}}.
 \end{aligned}$$

Wprowadźmy oznaczenia:  $\varrho_2 = \min \{ D_u^{(r)}, K - (r-1-m) + 1 - d_r \}$ ,  
 $\rho_2 = \min \{ D_u^{(r)}, K - (r-1-m) + 1 - d_r - k \}$  i wówczas:

$$\begin{aligned}
 &\frac{\partial^+ \mu_h^{(r)}(n)}{\partial \hat{w}_{ij}^{(m)}} = \sigma' \left( \chi_h^{(r)}(n) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\varrho_1} \tilde{w}_{hs_r d_r}^{(r)} \sum_{h_{r-1}=1}^{H_{r-1}} \bar{w}_{s_r h_{r-1}}^{(r-1)} \sigma' \left( \chi_{h_{r-1}}^{(r-1)}(n - d_r) \right) \cdot \\
 &\cdot \sum_{s_{r-1}=1}^{S_{r-1}} \sum_{d_{r-1}=1}^{\varrho_2} \tilde{w}_{h_{r-1} s_{r-1} d_{r-1}}^{(r-1)} \frac{\partial \bar{v}_{s_{r-1}}^{(r-2)}(n - d_r - d_{r-1})}{\partial \hat{w}_{ij}^{(m)}} + \\
 &+ \sum_{k=1}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n-k)} \sigma' \left( \chi_h^{(r)}(n-k) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\rho_1} \tilde{w}_{hs_r d_r}^{(r)} \sum_{h_{r-1}=1}^{H_{r-1}} \bar{w}_{s_r h_{r-1}}^{(r-1)} \cdot \\
 &\cdot \sigma' \left( \chi_{h_{r-1}}^{(r-1)}(n - d_r - k) \right) \sum_{s_{r-1}=1}^{S_{r-1}} \sum_{d_{r-1}=1}^{\rho_2} \tilde{w}_{h_{r-1} s_{r-1} d_{r-1}}^{(r-1)} \frac{\partial \bar{v}_{s_{r-1}}^{(r-2)}(n - d_r - d_{r-1} - k)}{\partial \hat{w}_{ij}^{(m)}} =
 \end{aligned}$$



$$\begin{aligned}
 &= \sigma' \left( \chi_h^{(r)}(n) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\varrho_1} \tilde{w}_{hs_r d_r}^{(r)} \sum_{h_{r-1}=1}^{H_{r-1}} \bar{w}_{s_r h_{r-1}}^{(r-1)} \sigma' \left( \chi_{h_{r-1}}^{(r-1)}(n - d_r) \right) \cdot \\
 &\cdot \sum_{s_{r-1}=1}^{S_{r-1}} \sum_{d_{r-1}=1}^{\varrho_2} \tilde{w}_{h_{r-1} s_{r-1} d_{r-1}}^{(r-1)} \sum_{h_{r-2}=1}^{H_{r-2}} \bar{w}_{s_{r-1} h_{r-2}}^{(r-2)} \frac{d\sigma \left( \chi_{h_{r-2}}^{(r-2)}(n - d_r - d_{r-1}) \right)}{d\chi_{h_{r-2}}^{(r-2)}(n - d_r - d_{r-1})} \cdot \\
 &\cdot \frac{\partial \chi_{h_{r-2}}^{(r-2)}(n - d_r - d_{r-1})}{\partial \hat{w}_{ij}^{(m)}} + \sum_{k=1}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n - k)} \sigma' \left( \chi_h^{(r)}(n - k) \right) \cdot \\
 &\cdot \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\varrho_1} \tilde{w}_{hs_r d_r}^{(r)} \sum_{h_{r-1}=1}^{H_{r-1}} \bar{w}_{s_r h_{r-1}}^{(r-1)} \sigma' \left( \chi_{h_{r-1}}^{(r-1)}(n - d_r - k) \right) \cdot \\
 &\cdot \sum_{s_{r-1}=1}^{S_{r-1}} \sum_{d_{r-1}=1}^{\varrho_2} \tilde{w}_{h_{r-1} s_{r-1} d_{r-1}}^{(r-1)} \sum_{h_{r-2}=1}^{H_{r-2}} \bar{w}_{s_{r-1} h_{r-2}}^{(r-2)} \cdot \\
 &\cdot \frac{d\sigma \left( \chi_{h_{r-2}}^{(r-2)}(n - d_r - d_{r-1} - k) \right)}{d\chi_{h_{r-2}}^{(r-2)}(n - d_r - d_{r-1} - k)} \frac{\partial \chi_{h_{r-2}}^{(r-2)}(n - d_r - d_{r-1} - k)}{\partial \hat{w}_{ij}^{(m)}} = \\
 &= \sigma' \left( \chi_h^{(r)}(n) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\varrho_1} \tilde{w}_{hs_r d_r}^{(m)} \sum_{h_{r-1}=1}^{H_{r-1}} \bar{w}_{s_r h_{r-1}}^{(r-1)} \sigma' \left( \chi_{h_{r-1}}^{(r-1)}(n - d_r) \right) \cdot \\
 &\cdot \sum_{s_{r-1}=1}^{S_{r-1}} \sum_{d_{r-1}=1}^{\varrho_2} \tilde{w}_{h_{r-1} s_{r-1} d_{r-1}}^{(r-1)} \sum_{h_{r-2}=1}^{H_{r-2}} \bar{w}_{s_{r-1} h_{r-2}}^{(r-2)} \sigma' \left( \chi_{h_{r-2}}^{(r-2)}(n - d_r - d_{r-1}) \right) \cdot \\
 &\cdot \dots \cdot \sum_{h_{m+1}}^{H_{m+1}} \bar{w}_{s_{m+2} h_{m+1}}^{(m+1)} \frac{d\sigma \left( \chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2}) \right)}{d\chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2})} \cdot \\
 &\cdot \frac{\partial \chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2})}{\partial \hat{w}_{ij}^{(m)}} + \sum_{k=1}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n - k)} \sigma' \left( \chi_h^{(r)}(n - k) \right) \cdot
 \end{aligned}$$

$$\begin{aligned}
 & \cdot \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\rho_1} \tilde{w}_{hs_r d_r}^{(r)} \sum_{h_{r-1}=1}^{H_{r-1}} \bar{w}_{s_r h_{r-1}}^{(r-1)} \sigma' \left( \chi_{h_{r-1}}^{(r-1)}(n - d_r - k) \right) \cdot \\
 & \cdot \sum_{s_{r-1}=1}^{S_{r-1}} \sum_{d_{r-1}=1}^{\rho_2} \tilde{w}_{h_{r-1} s_{r-1} d_{r-1}}^{(r-1)} \sum_{h_{r-2}=1}^{H_{r-2}} \bar{w}_{s_{r-1} h_{r-2}}^{(r-2)} \sigma' \left( \chi_{h_{r-2}}^{(r-2)}(n - d_r - d_{r-1} - k) \right) \cdot \\
 & \cdot \dots \cdot \sum_{h_{m+1}}^{H_{m+1}} \bar{w}_{s_{m+2} h_{m+1}}^{(m+1)} \frac{d\sigma \left( \chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2} - k) \right)}{d\chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2} - k)} \cdot \\
 & \cdot \frac{\partial \chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2} - k)}{\partial \tilde{w}_{ij}^{(m)}} = \\
 & = \sigma' \left( \chi_h^{(r)}(n) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\rho_1} \tilde{w}_{hs_r d_r}^{(r)} \sum_{h_{r-1}=1}^{H_{r-1}} \bar{w}_{s_r h_{r-1}}^{(r-1)} \sigma' \left( \chi_{h_{r-1}}^{(r-1)}(n - d_r) \right) \cdot \\
 & \cdot \sum_{s_{r-1}=1}^{S_{r-1}} \sum_{d_{r-1}=1}^{\rho_2} \tilde{w}_{h_{r-1} s_{r-1} d_{r-1}}^{(r-1)} \sum_{h_{r-2}=1}^{H_{r-2}} \bar{w}_{s_{r-1} h_{r-2}}^{(r-2)} \sigma' \left( \chi_{h_{r-2}}^{(r-2)}(n - d_r - d_{r-1}) \right) \cdot \\
 & \cdot \dots \cdot \sum_{h_{m+1}=1}^{H_{m+1}} \bar{w}_{s_{m+2} h_{m+1}}^{(m+1)} \sigma' \left( \chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2}) \right) \cdot \\
 & \cdot \sum_{s_{m+1}=1}^{S_{m+1}} \sum_{d_{m+1}=1}^{\min\{D_u^{(m+1)}, K - d_r - d_{r-1} - \dots - d_{m+2}\}} \tilde{w}_{h_{m+1}, s_{m+1}, d_{m+1}}^{(m+1)} \cdot \\
 & \cdot \frac{\partial \bar{w}_{s_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+1})}{\partial \tilde{w}_{ij}^{(m)}} + \sum_{k=1}^{K - (r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n - k)} \sigma' \left( \chi_h^{(r)}(n - k) \right) \cdot \\
 & \cdot \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\rho_1} \tilde{w}_{hs_r d_r}^{(r)} \sum_{h_{r-1}=1}^{H_{r-1}} \bar{w}_{s_r h_{r-1}}^{(r-1)} \sigma' \left( \chi_{h_{r-1}}^{(r-1)}(n - d_r - k) \right) \cdot
 \end{aligned} \tag{2.51}$$

$$\begin{aligned}
 & \cdot \sum_{s_{r-1}=1}^{S_{r-1}} \sum_{d_{r-1}=1}^{\rho_2} \tilde{w}_{h_{r-1}s_{r-1}d_{r-1}}^{(r-1)} \sum_{h_{r-2}=1}^{H_{r-2}} \bar{w}_{s_{r-1}h_{r-2}}^{(r-2)} \sigma' \left( \chi_{h_{r-2}}^{(r-2)}(n - d_r - d_{r-1} - k) \right) \cdot \\
 & \cdot \dots \cdot \sum_{h_{m+1}=1}^{H_{m+1}} \bar{w}_{s_{m+2}h_{m+1}}^{(m+1)} \sigma' \left( \chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2} - k) \right) \cdot \\
 & \cdot \sum_{s_{m+1}=1}^{S_{m+1}} \sum_{d_{m+1}=1}^{\min\{D_u^{(m+1)}, K - d_r - d_{r-1} - \dots - d_{m+2} - k\}} \tilde{w}_{h_{m+1}, s_{m+1}, d_{m+1}}^{(m+1)} \cdot \\
 & \cdot \frac{\partial \bar{w}_{s_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+1} - k)}{\partial \hat{w}_{ij}^{(m)}}.
 \end{aligned}$$

Wprowadzając oznaczenia:  $\varrho_{r-m} = \min\{D_u^{(m+1)}, K - d_r - d_{r-1} - \dots - d_{m+2}\}$ ,  
 $\rho_{r-m} = \min\{D_u^{(r)}, K - d_r - d_{r-1} - \dots - d_{m+2} - k\}$  otrzymujemy:

$$\begin{aligned}
 & \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \hat{w}_{ij}^{(m)}} = \sigma' \left( \chi_h^{(r)}(n) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\varrho_1} \tilde{w}_{hs_r d_r}^{(r)} \sum_{h_{r-1}=1}^{H_{r-1}} \bar{w}_{s_r h_{r-1}}^{(r-1)} \sigma' \left( \chi_{h_{r-1}}^{(r-1)}(n - d_r) \right) \cdot \\
 & \cdot \sum_{s_{r-1}=1}^{S_{r-1}} \sum_{d_{r-1}=1}^{\varrho_2} \tilde{w}_{h_{r-1}s_{r-1}d_{r-1}}^{(r-1)} \sum_{h_{r-2}=1}^{H_{r-2}} \bar{w}_{s_{r-1}h_{r-2}}^{(r-2)} \sigma' \left( \chi_{h_{r-2}}^{(r-2)}(n - d_r - d_{r-1}) \right) \cdot \\
 & \cdot \dots \cdot \sum_{h_{m+1}=1}^{H_{m+1}} \bar{w}_{s_{m+2}h_{m+1}}^{(m+1)} \sigma' \left( \chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2}) \right) \cdot \\
 & \cdot \sum_{s_{m+1}=1}^{S_{m+1}} \sum_{d_{m+1}=1}^{\varrho_{r-m}} \tilde{w}_{h_{m+1}, s_{m+1}, d_{m+1}}^{(m+1)} \frac{\partial \bar{v}_{s_{m+1}}^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1})}{\partial \hat{w}_{ij}^{(m)}} + \\
 & + \sum_{k=1}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n-k)} \sigma' \left( \chi_h^{(r)}(n-k) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\rho_1} \tilde{w}_{hs_r d_r}^{(r)} \sum_{h_{r-1}=1}^{H_{r-1}} \bar{w}_{s_r h_{r-1}}^{(r-1)} \cdot \\
 & \cdot \sigma' \left( \chi_{h_{r-1}}^{(r-1)}(n - d_r - k) \right) \sum_{s_{r-1}=1}^{S_{r-1}} \sum_{d_{r-1}=1}^{\rho_2} \tilde{w}_{h_{r-1}s_{r-1}d_{r-1}}^{(r-1)} \sum_{h_{r-2}=1}^{H_{r-2}} \bar{w}_{s_{r-1}h_{r-2}}^{(r-2)} \cdot
 \end{aligned}$$

$$\begin{aligned}
 & \cdot \sigma' \left( \chi_{h_{r-2}}^{(r-2)}(n - d_r - d_{r-1} - k) \right) \cdot \dots \cdot \sum_{h_{m+1}=1}^{H_{m+1}} \bar{w}_{s_{m+2}h_{m+1}}^{(m+1)}. \\
 & \cdot \sigma' \left( \chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2} - k) \right) \sum_{s_{m+1}=1}^{S_{m+1}} \sum_{d_{m+1}=1}^{\rho_{r-m}} \tilde{w}_{h_{m+1}, s_{m+1}, d_{m+1}}^{(m+1)}. \\
 & \cdot \frac{\partial \bar{v}_{s_{m+1}}^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - k)}{\partial \tilde{w}_{ij}^{(m)}} = \\
 & = \sigma' \left( \chi_h^{(r)}(n) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\varrho_1} \tilde{w}_{h s_r d_r}^{(r)} \sum_{h_{r-1}=1}^{H_{r-1}} \bar{w}_{s_r h_{r-1}}^{(r-1)} \sigma' \left( \chi_{h_{r-1}}^{(r-1)}(n - d_r) \right). \\
 & \cdot \sum_{s_{r-1}=1}^{S_{r-1}} \sum_{d_{r-1}=1}^{\varrho_2} \tilde{w}_{h_{r-1} s_{r-1} d_{r-1}}^{(r-1)} \sum_{h_{r-2}=1}^{H_{r-2}} \bar{w}_{s_{r-1} h_{r-2}}^{(r-2)} \sigma' \left( \chi_{h_{r-2}}^{(r-2)}(n - d_r - d_{r-1}) \right). \\
 & \cdot \dots \cdot \sum_{h_{m+1}=1}^{H_{m+1}} \bar{w}_{s_{m+2} h_{m+1}}^{(m+1)} \sigma' \left( \chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2}) \right). \\
 & \cdot \sum_{s_{m+1}=1}^{S_{m+1}} \sum_{d_{m+1}=1}^{\varrho_{r-m}} \tilde{w}_{h_{m+1}, s_{m+1}, d_{m+1}}^{(m+1)} \bar{w}_{s_{m+1} i}^{(m)} \frac{d\sigma \left( \chi_i^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1}) \right)}{d\chi_i^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1})}. \\
 & \cdot \frac{\partial \chi_i^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1})}{\partial \tilde{w}_{ij}^{(m)}} + \\
 & + \sum_{k=1}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n - k)} \sigma' \left( \chi_h^{(r)}(n - k) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\rho_1} \tilde{w}_{h s_r d_r}^{(r)} \sum_{h_{r-1}=1}^{H_{r-1}} \bar{w}_{s_r h_{r-1}}^{(r-1)}. \\
 & \cdot \sigma' \left( \chi_{h_{r-1}}^{(r-1)}(n - d_r - k) \right) \sum_{s_{r-1}=1}^{S_{r-1}} \sum_{d_{r-1}=1}^{\rho_2} \tilde{w}_{h_{r-1} s_{r-1} d_{r-1}}^{(r-1)} \sum_{h_{r-2}=1}^{H_{r-2}} \bar{w}_{s_{r-1} h_{r-2}}^{(r-2)}.
 \end{aligned}$$

$$\begin{aligned}
 & \cdot \sigma' \left( \chi_{h_{r-2}}^{(r-2)}(n - d_r - d_{r-1} - k) \right) \cdot \dots \cdot \sum_{h_{m+1}=1}^{H_{m+1}} \bar{w}_{s_{m+2}h_{m+1}}^{(m+1)} \cdot \\
 & \cdot \sigma' \left( \chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2} - k) \right) \sum_{s_{m+1}=1}^{S_{m+1}} \sum_{d_{m+1}=1}^{\rho_{r-m}} \tilde{w}_{h_{m+1}, s_{m+1}, d_{m+1}}^{(m+1)} \cdot \\
 & \cdot \bar{w}_{s_{m+1}i}^{(m)} \frac{d\sigma \left( \chi_i^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - k) \right)}{d\chi_i^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - k)} \cdot \\
 & \cdot \frac{\partial \chi_i^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - k)}{\partial \tilde{w}_{ij}^{(m)}} = \\
 & = \sigma' \left( \chi_h^{(r)}(n) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\varrho_1} \tilde{w}_{hs_r, d_r}^{(r)} \sum_{h_{r-1}=1}^{H_{r-1}} \bar{w}_{s_r h_{r-1}}^{(r-1)} \sigma' \left( \chi_{h_{r-1}}^{(r-1)}(n - d_r) \right) \cdot \\
 & \cdot \sum_{s_{r-1}=1}^{S_{r-1}} \sum_{d_{r-1}=1}^{\varrho_2} \tilde{w}_{h_{r-1} s_{r-1} d_{r-1}}^{(r-1)} \sum_{h_{r-2}=1}^{H_{r-2}} \bar{w}_{s_{r-1} h_{r-2}}^{(r-2)} \sigma' \left( \chi_{h_{r-2}}^{(r-2)}(n - d_r - d_{r-1}) \right) \cdot \\
 & \cdot \dots \cdot \sum_{h_{m+1}=1}^{H_{m+1}} \bar{w}_{s_{m+2} h_{m+1}}^{(m+1)} \sigma' \left( \chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2}) \right) \cdot \\
 & \cdot \sum_{s_{m+1}=1}^{S_{m+1}} \sum_{d_{m+1}=1}^{\varrho_{r-m}} \tilde{w}_{h_{m+1}, s_{m+1}, d_{m+1}}^{(m+1)} \bar{w}_{s_{m+1}i}^{(m)} \cdot \sigma' \left( \chi_i^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1}) \right) \cdot \\
 & \cdot \mu_i^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - j) + \\
 & + \sum_{k=1}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n - k)} \sigma' \left( \chi_h^{(r)}(n - k) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\rho_1} \tilde{w}_{hs_r, d_r}^{(r)} \sum_{h_{r-1}=1}^{H_{r-1}} \bar{w}_{s_r h_{r-1}}^{(r-1)} \cdot \sigma' \left( \chi_{h_{r-1}}^{(r-1)}(n - d_r - k) \right) \cdot
 \end{aligned} \tag{2.52}$$

$$\begin{aligned}
 & \cdot \sum_{s_{r-1}=1}^{S_{r-1}} \sum_{d_{r-1}=1}^{\rho_2} \tilde{w}_{h_{r-1}s_{r-1}d_{r-1}}^{(r-1)} \sum_{h_{r-2}=1}^{H_{r-2}} \bar{w}_{s_{r-1}h_{r-2}}^{(r-2)} \sigma' \left( \chi_{h_{r-2}}^{(r-2)} (n - d_r - d_{r-1} - k) \right) \cdot \\
 & \cdot \dots \cdot \sum_{h_{m+1}=1}^{H_{m+1}} \bar{w}_{s_{m+2}h_{m+1}}^{(m+1)} \sigma' \left( \chi_{h_{m+1}}^{(m+1)} (n - d_r - d_{r-1} - \dots - d_{m+2} - k) \right) \cdot \\
 & \cdot \sum_{s_{m+1}=1}^{S_{m+1}} \sum_{d_{m+1}=1}^{\rho_{r-m}} \tilde{w}_{h_{m+1},s_{m+1},d_{m+1}}^{(m+1)} \sigma' \left( \chi_i^{(m)} (n - d_r - d_{r-1} - \dots - d_{m+1} - k) \right) \cdot \\
 & \cdot \bar{w}_{s_{m+1}i}^{(m)} \cdot \mu_i^{(m)} (n - d_r - d_{r-1} - \dots - d_{m+1} - j - k).
 \end{aligned} \tag{2.53}$$

Ostatnie równanie można zapisać w skróconej postaci:

$$\begin{aligned}
 & \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \hat{w}_{ij}^{(m)}} = \sigma' \left( \chi_h^{(r)}(n) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\varrho_1} \tilde{w}_{hs_r d_r}^{(r)} \cdot \\
 & \cdot \prod_{\gamma=1}^{r-m-1} \left\{ \sum_{h_{r-\gamma}=1}^{H_{r-\gamma}} \bar{w}_{s_{r-\gamma+1}h_{r-\gamma}}^{(r-\gamma)} \sigma' \left( \chi_{h_{r-\gamma}}^{(r-\gamma)} \left( n - \sum_{\nu=1}^{\gamma} d_{r-\nu+1} \right) \right) \sum_{s_{r-\gamma}=1}^{S_{r-\gamma}} \sum_{d_{r-\gamma}=1}^{\varrho_{\gamma+1}} \tilde{w}_{h_{r-\gamma}s_{r-\gamma}d_{r-\gamma}}^{(r-\gamma)} \right\} \cdot \\
 & \cdot \bar{w}_{s_{m+1}h}^{(m)} \cdot \sigma' \left( \chi_i^{(m)} (n - d_r - d_{r-1} - \dots - d_{m+1}) \right) \mu_i^{(m)} (n - d_r - d_{r-1} - \dots - d_{m+1} - j) + \\
 & + \sum_{k=1}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n-k)} \sigma' \left( \chi_h^{(r)}(n-k) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\varrho_1} \tilde{w}_{hs_r d_r}^{(r)} \cdot \prod_{\gamma=1}^{r-m-1} \left\{ \sum_{h_{r-\gamma}=1}^{H_{r-\gamma}} \bar{w}_{s_{r-\gamma+1}h_{r-\gamma}}^{(r-\gamma)} \cdot \right. \\
 & \cdot \sigma' \left( \chi_{h_{r-\gamma}}^{(r-\gamma)} \left( n - \sum_{\nu=1}^{\gamma} d_{r-\nu+1} - k \right) \right) \cdot \left. \sum_{s_{r-\gamma}=1}^{S_{r-\gamma}} \sum_{d_{r-\gamma}=1}^{\varrho_{\gamma+1}} \tilde{w}_{h_{r-\gamma}s_{r-\gamma}d_{r-\gamma}}^{(r-\gamma)} \right\} \bar{w}_{s_{m+1}i}^{(m)} \cdot \\
 & \cdot \sigma' \left( \chi_i^{(m)} (n - d_r - d_{r-1} - \dots - d_{m+1} - k) \right) \cdot \mu_i^{(m)} (n - d_r - d_{r-1} - \dots - d_{m+1} - j - k),
 \end{aligned} \tag{2.54}$$

lub jeszcze zwięźlej:

$$\begin{aligned}
 \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \widehat{w}_{ij}^{(m)}} &= \sum_{k=0}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n-k)} \sigma' \left( \chi_h^{(r)}(n-k) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\rho_1} \widetilde{w}_{hs_r d_r}^{(r)} \cdot \\
 &\cdot \prod_{\gamma=1}^{r-m-1} \left\{ \sum_{h_{r-\gamma}=1}^{H_{r-\gamma}} \overline{w}_{s_{r-\gamma+1} h_{r-\gamma}}^{(r-\gamma)} \sigma' \left( \chi_{h_{r-\gamma}}^{(r-\gamma)} \left( n - \sum_{\nu=1}^{\gamma} d_{r-\nu+1} - k \right) \right) \right\} \cdot \\
 &\cdot \sum_{s_{r-\gamma}=1}^{S_{r-\gamma}} \sum_{d_{r-\gamma}=1}^{\rho_{\gamma+1}} \widetilde{w}_{h_{r-\gamma} s_{r-\gamma} d_{r-\gamma}}^{(r-\gamma)} \left\{ \overline{w}_{s_{m+1} i}^{(m)} \cdot \sigma' \left( \chi_i^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - k) \right) \right\} \cdot \\
 &\cdot \mu_i^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - j - k).
 \end{aligned} \tag{2.55}$$

Regułę aktualizacji parametrów  $\overline{w}_{ij}^{(m)}$  ( $i = 1, 2, \dots, L_m$ ;  $j = 0, 1, \dots, H_m$ ,  $m = 1, 2, \dots, M$ ) wyprowadza się podobnie. Obliczenia różnią się dopiero od równania (2.51), a dokładniej, inne będą rozwinięcia pochodnych cząstkowych z  $\chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2})$  i  $\chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2} - k)$  względem wag  $\overline{w}_{ih}^{(m)}$ , czyli czynników  $\frac{\partial \chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2})}{\partial \overline{w}_{ih}^{(m)}}$  i  $\frac{\partial \chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2} - k)}{\partial \overline{w}_{ij}^{(m)}}$ :

$$\begin{aligned}
 \frac{\partial \chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2})}{\partial \overline{w}_{ij}^{(m)}} &= \\
 &= \sum_{s_{m+1}=1}^{S_{m+1}} \sum_{d_{m+1}=1}^{\varrho_{r-m}} \widetilde{w}_{h_{m+1} s_{m+1} d_{m+1}}^{(m+1)} \cdot \frac{\partial \overline{u}_{s_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+1})}{\partial \overline{w}_{ij}^{(m)}} = \\
 &= \sum_{s_{m+1}=1}^{S_{m+1}} \sum_{d_{m+1}=1}^{\varrho_{r-m}} \widetilde{w}_{h_{m+1} s_{m+1} d_{m+1}}^{(m+1)} \cdot \frac{\partial \overline{v}_{s_{m+1}}^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1})}{\partial \overline{w}_{ij}^{(m)}} = \\
 &= \sum_{d_{m+1}=1}^{\varrho_{r-m}} \widetilde{w}_{h_{m+1} i d_{m+1}}^{(m+1)} \cdot \mu_j^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1})
 \end{aligned} \tag{2.56}$$

oraz

$$\begin{aligned}
 & \frac{\partial \chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2} - k)}{\partial \bar{w}_{ij}^{(m)}} = \\
 & = \sum_{s_{m+1}=1}^{S_{m+1}} \sum_{d_{m+1}=1}^{\varrho_{r-m}} \tilde{w}_{h_{m+1}s_{m+1}d_{m+1}}^{(m+1)} \cdot \frac{\partial \bar{u}_{s_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+1} - k)}{\partial \bar{w}_{ij}^{(m)}} = \\
 & = \sum_{s_{m+1}=1}^{S_{m+1}} \sum_{d_{m+1}=1}^{\varrho_{r-m}} \tilde{w}_{h_{m+1}s_{m+1}d_{m+1}}^{(m+1)} \cdot \frac{\partial \bar{v}_{s_{m+1}}^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - k)}{\partial \bar{w}_{ij}^{(m)}} = \\
 & = \sum_{d_{m+1}=1}^{\varrho_{r-m}} \tilde{w}_{h_{m+1}id_{m+1}}^{(m+1)} \cdot \mu_j^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - k).
 \end{aligned} \tag{2.57}$$

Zbierając dwa powyższe wyprowadzenia razem, możemy zapisać odpowiednik skróconej formuły (2.54) dla parametrów wagowych  $\bar{w}_{ij}^{(m)}$ :

$$\begin{aligned}
 & \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \bar{w}_{ij}^{(m)}} = \sigma' \left( \chi_h^{(r)}(n) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\varrho_1} \tilde{w}_{hs_r d_r}^{(r)} \prod_{\gamma=1}^{r-m-2} \left\{ \sum_{h_{r-\gamma}=1}^{H_{r-\gamma}} \bar{w}_{s_{r-\gamma+1} h_{r-\gamma}}^{(r-\gamma)} \cdot \right. \\
 & \cdot \sigma' \left( \chi_{h_{r-\gamma}}^{(r-\gamma)} \left( n - \sum_{\nu=1}^{\gamma} d_{r-\nu+1} \right) \right) \sum_{s_{r-\gamma}=1}^{S_{r-\gamma}} \sum_{d_{r-\gamma}=1}^{\varrho_{\gamma+1}} \tilde{w}_{h_{r-\gamma} s_{r-\gamma} d_{r-\gamma}}^{(r-\gamma)} \left. \right\} \cdot \\
 & \cdot \sum_{h_{m+1}=1}^{H_{m+1}} \bar{w}_{s_{m+2} h_{m+1}}^{(m+1)} \sigma' \left( \chi_{h_{m+1}}^{(m+1)}(n - d_r - d_{r-1} - \dots - d_{m+2}) \right) \sum_{d_{m+1}=1}^{\varrho_{r-m}} \tilde{w}_{h_{m+1} id_{m+1}}^{(m+1)} \cdot \\
 & \cdot \mu_j^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1}) + \sum_{k=1}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n-k)} \sigma' \left( \chi_h^{(r)}(n-k) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\varrho_1} \tilde{w}_{hs_r d_r}^{(r)} \cdot \\
 & \cdot \prod_{\gamma=1}^{r-m-2} \left\{ \sum_{h_{r-\gamma}=1}^{H_{r-\gamma}} \bar{w}_{s_{r-\gamma+1} h_{r-\gamma}}^{(r-\gamma)} \sigma' \left( \chi_{h_{r-\gamma}}^{(r-\gamma)} \left( n - \sum_{\nu=1}^{\gamma} d_{r-\nu+1} - k \right) \right) \sum_{s_{r-\gamma}=1}^{S_{r-\gamma}} \sum_{d_{r-\gamma}=1}^{\varrho_{\gamma+1}} \tilde{w}_{h_{r-\gamma} s_{r-\gamma} d_{r-\gamma}}^{(r-\gamma)} \right\} \cdot \\
 & \cdot \sum_{d_{m+1}=1}^{\varrho_{r-m}} \tilde{w}_{h_{m+1} id_{m+1}}^{(m+1)} \cdot \mu_j^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - k), \tag{2.58}
 \end{aligned}$$



lub jeszcze zwiężlej:

$$\begin{aligned}
 \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \bar{w}_{ij}^{(m)}} &= \sum_{k=0}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n-k)} \sigma' \left( \chi_h^{(r)}(n-k) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\rho_1} \tilde{w}_{hs_r d_r}^{(r)} \cdot \\
 &\cdot \prod_{\gamma=1}^{r-m-2} \left\{ \sum_{h_{r-\gamma}=1}^{H_{r-\gamma}} \bar{w}_{s_{r-\gamma+1} h_{r-\gamma}}^{(r-\gamma)} \sigma' \left( \chi_{h_{r-\gamma}}^{(r-\gamma)} \left( n - \sum_{\nu=1}^{\gamma} d_{r-\nu+1} - k \right) \right) \sum_{s_{r-\gamma}=1}^{S_{r-\gamma}} \sum_{d_{r-\gamma}=1}^{\rho_{\gamma+1}} \tilde{w}_{h_{r-\gamma} s_{r-\gamma} d_{r-\gamma}}^{(r-\gamma)} \right\} \cdot \\
 &\cdot \sum_{d_{m+1}=1}^{\rho_{r-m}} \tilde{w}_{h_{m+1} d_{m+1}}^{(m+1)} \cdot \mu_j^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - k).
 \end{aligned} \tag{2.59}$$

Regułę aktualizacji parametrów  $\tilde{w}_{ijp}^{(m)}$  ( $i = 1, 2, \dots, H_m$ ;  $j = 0, 1, \dots, S_m$ ;  $p = 1, 2, \dots, D_u^{(m)}$ ;  $m = 1, 2, \dots, M$ ) również wyprowadzamy podobnie. Tym razem obliczenia różnią się od poprzednich począwszy od równań (2.56) i (2.57), a dokładniej, inne będą rozwinięcia pochodnych cząstkowych z

$\bar{v}_{s_{m+1}}^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1})$  i  $\bar{v}_{s_{m+1}}^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - k)$  względem wag  $\tilde{w}_{hsd}^{(m)}$ , czyli czynników  $\frac{\partial \bar{v}_{s_{m+1}}^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1})}{\partial \tilde{w}_{hsd}^{(m)}}$  i  $\frac{\partial \bar{v}_{s_{m+1}}^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - k)}{\partial \tilde{w}_{hsd}^{(m)}}$ :

$$\begin{aligned}
 \frac{\partial \bar{v}_{s_{m+1}}^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1})}{\partial \tilde{w}_{ijp}^{(m)}} &= \bar{w}_{s_{m+1}i}^{(m)} \cdot \sigma' \left( \chi_i^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1}) \right) \cdot \\
 &\cdot \bar{u}_j^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - p),
 \end{aligned} \tag{2.60}$$

oraz:

$$\begin{aligned}
 \frac{\partial \bar{v}_{s_{m+1}}^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - k)}{\partial \tilde{w}_{ijp}^{(m)}} &= \\
 &= \bar{w}_{s_{m+1}i}^{(m)} \cdot \sigma' \left( \chi_i^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - k) \right) \cdot \\
 &\cdot \bar{u}_j^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - p - k).
 \end{aligned} \tag{2.61}$$

Zbierając dwa powyższe wyprowadzenia razem, możemy zapisać odpowiednik skróconej formuły (2.59) dla uszeregowanych pochodnych cząstkowych względem parametrów wagowych  $\tilde{w}_{ijp}^{(m)}$ :

$$\begin{aligned}
 \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \tilde{w}_{ijp}^{(m)}} &= \sum_{k=0}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n-k)} \sigma' \left( \chi_h^{(r)}(n-k) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\rho_1} \tilde{w}_{hs_r d_r}^{(r)} \cdot \\
 &\cdot \prod_{\gamma=1}^{r-m-1} \left\{ \sum_{h_{r-\gamma}=1}^{H_{r-\gamma}} \bar{w}_{s_{r-\gamma+1} h_{r-\gamma}}^{(r-\gamma)} \sigma' \left( \chi_{h_{r-\gamma}}^{(r-\gamma)} \left( n - \sum_{\nu=1}^{\gamma} d_{r-\nu+1} - k \right) \right) \cdot \right. \\
 &\cdot \left. \sum_{s_{r-\gamma}=1}^{S_{r-\gamma}} \sum_{d_{r-\gamma}=1}^{\rho_{\gamma+1}} \tilde{w}_{h_{r-\gamma} s_{r-\gamma} d_{r-\gamma}}^{(r-\gamma)} \right\} \bar{w}_{s_{m+1} i}^{(m)} \cdot \sigma' \left( \chi_i^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - k) \right) \cdot \\
 &\cdot \bar{w}_j^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - p - k).
 \end{aligned} \tag{2.62}$$

Końcowe wzory na uszeregowane pochodne cząstkowe względem wag, tj. (2.55), (2.59), (2.62), sprawiają wrażenie bardzo złożonych obliczeniowo. Dlatego należy tu zaznaczyć, że zwykle wykonywanych jest tylko kilka kroków każdego sumowania (wartość parametru  $K$  rzadko przekracza 5), z uwagi na niewielkie wartości zmiennych  $\varrho_1, \varrho_2, \dots, \varrho_{r-m}$  oraz  $\rho_1, \rho_2, \dots, \rho_{r-m}$ , nawet dla dużej liczby elementów  $M$ . Ponadto we wzorach dla różnych wag powtarzają się często te same wyrażenia.

Wstawiając (2.55), (2.59) i (2.62) odpowiednio do (2.44), (2.46) i (2.45), wyniki tych podstawień umieszczając jako składowe w wektorze  $\nabla_{\mathbf{w}^{(m)}} \bar{v}_l^{(r)}$  we wzorze (2.43) i wraz z (2.47), (2.48), (2.47), (2.50) ostateczny rezultat wstawiając do formuły gradientowej (2.40), otrzymujemy podstawowe reguły uczenia złożonej sieci DRNN, czyli algorytm  $\Psi_G$  identyfikacji globalnej dynamicznego systemu złożonego o strukturze kaskadowej:

$$\tilde{w}_{ijp}^{(m)}(n+1) = \tilde{w}_{ijp}^{(m)}(n) + \eta^{(m)} \sum_{r=m}^M \beta_r \sum_{l=1}^{L_r} \varepsilon_l^{(r)}(n) \frac{\partial \bar{v}_l^{(r)}(n)}{\partial \tilde{w}_{ijp}^{(m)}}, \tag{2.63}$$

$$\hat{w}_{ij}^{(m)}(n+1) = \hat{w}_{ij}^{(m)}(n) + \eta^{(m)} \sum_{r=m}^M \beta_r \sum_{l=1}^{L_r} \varepsilon_l^{(r)}(n) \frac{\partial \bar{v}_l^{(r)}(n)}{\partial \hat{w}_{ij}^{(m)}}, \tag{2.64}$$

$$\bar{w}_{ij}^{(m)}(n+1) = \bar{w}_{ij}^{(m)}(n) + \eta^{(m)} \sum_{r=m}^M \beta_r \sum_{l=1}^{L_r} \varepsilon_l^{(r)}(n) \frac{\partial \bar{v}_l^{(r)}(n)}{\partial \bar{w}_{ij}^{(m)}}. \tag{2.65}$$

Wykonanie tych obliczeń dla  $m = 1, 2, \dots, M$  daje nam przybliżenie wektora parametrów  $\mathbf{w}_G^*$  globalnie optymalnego modelu neuronowego systemu złożonego. Po wstawieniu do powyższych reguł wzorów na pochodne cząstkowe, otrzymamy dokładną postać algorytmu BPTT uogólnionego na sieć złożoną DRNN o strukturze kaskadowej:

$$\begin{aligned}
 \tilde{w}_{ijp}^{(m)}(n+1) &= \tilde{w}_{ijp}^{(m)}(n) + \eta^{(m)} \beta_m \sum_{l=1}^{L_m} \varepsilon_l^{(m)}(n) \sum_{i=1}^{H_m} \bar{w}_{li}^{(m)} \left\{ \sigma' \left( \chi_i^{(m)}(n) \right) \bar{u}_j^{(m)}(n-p) + \right. \\
 &+ \sum_{k=1}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} \sigma' \left( \chi_i^{(m)}(n-k) \right) \bar{u}_j^{(m)}(n-p-k) \left. \right\} + \eta^{(m)} \sum_{r=m+1}^M \beta_r \cdot \\
 &\cdot \sum_{l=1}^{L_r} \varepsilon_l^{(r)}(n) \sum_{h=1}^{H_r} \bar{w}_{lh}^{(r)} \sum_{k=0}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n-k)} \sigma' \left( \chi_h^{(r)}(n-k) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\rho_1} \tilde{w}_{hs_r d_r}^{(r)} \cdot \\
 &\cdot \prod_{\gamma=1}^{r-m-1} \left\{ \sum_{h_{r-\gamma}=1}^{H_{r-\gamma}} \bar{w}_{s_{r-\gamma+1} h_{r-\gamma}}^{(r-\gamma)} \sigma' \left( \chi_{h_{r-\gamma}}^{(r-\gamma)} \left( n - \sum_{\nu=1}^{\gamma} d_{r-\nu+1} - k \right) \right) \right\} \cdot \\
 &\cdot \sum_{s_{r-\gamma}=1}^{S_{r-\gamma}} \sum_{d_{r-\gamma}=1}^{\rho_{\gamma+1}} \tilde{w}_{h_{r-\gamma} s_{r-\gamma} d_{r-\gamma}}^{(r-\gamma)} \left. \right\} \bar{w}_{s_{m+1} i}^{(m)} \cdot \\
 &\cdot \sigma' \left( \chi_i^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - k) \right) \cdot \\
 &\cdot \bar{u}_j^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - p - k),
 \end{aligned} \tag{2.66}$$

$$\begin{aligned}
 \hat{w}_{ij}^{(m)}(n+1) &= \hat{w}_{ij}^{(m)}(n) + \eta^{(m)} \beta_m \sum_{l=1}^{L_m} \varepsilon_l^{(m)}(n) \sum_{i=1}^{H_m} \bar{w}_{li}^{(m)} \left\{ \sigma' \left( \chi_i^{(m)}(n) \right) \mu_i^{(m)}(n-j) + \right. \\
 &+ \sum_{k=1}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} \sigma' \left( \chi_i^{(m)}(n-k) \right) \mu_i^{(m)}(n-j-k) \left. \right\} + \eta^{(m)} \sum_{r=m+1}^M \beta_r \cdot \\
 &\tag{2.67}
 \end{aligned}$$

$$\cdot \sum_{l=1}^{L_r} \varepsilon_l^{(r)}(n) \sum_{h=1}^{H_r} \bar{w}_{lh}^{(r)} \sum_{k=0}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n-k)} \sigma' \left( \chi_h^{(r)}(n-k) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\rho_1} \tilde{w}_{hs_r d_r}^{(r)}. \quad (2.68)$$

$$\cdot \prod_{\gamma=1}^{r-m-1} \left\{ \sum_{h_{r-\gamma}=1}^{H_{r-\gamma}} \bar{w}_{s_{r-\gamma+1} h_{r-\gamma}}^{(r-\gamma)} \sigma' \left( \chi_{h_{r-\gamma}}^{(r-\gamma)} \left( n - \sum_{\nu=1}^{\gamma} d_{r-\nu+1} - k \right) \right) \right\}. \quad (2.69)$$

$$\cdot \left. \sum_{s_{r-\gamma}=1}^{S_{r-\gamma}} \sum_{d_{r-\gamma}=1}^{\rho_{\gamma+1}} \tilde{w}_{h_{r-\gamma} s_{r-\gamma} d_{r-\gamma}}^{(r-\gamma)} \right\} \bar{w}_{s_{m+1} i}^{(m)} \cdot \sigma' \left( \chi_i^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - k) \right) \cdot \mu_i^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - j - k), \quad (2.70)$$

$$\bar{w}_{ij}^{(m)}(n+1) = \bar{w}_{ij}^{(m)}(n) + \eta^{(m)} \beta_m \varepsilon_i^{(m)}(n) \mu_j^{(m)}(n) + \eta^{(m)} \sum_{r=m+1}^M \beta_r \sum_{l=1}^{L_r} \varepsilon_l^{(r)}(n) \cdot$$

$$\cdot \sum_{k=0}^{K-(r-m)} \frac{\partial^+ \mu_h^{(r)}(n)}{\partial \mu_h^{(r)}(n-k)} \sigma' \left( \chi_h^{(r)}(n-k) \right) \sum_{s_r=1}^{S_r} \sum_{d_r=1}^{\rho_1} \tilde{w}_{hs_r d_r}^{(r)} \cdot \prod_{\gamma=1}^{r-m-2} \left\{ \sum_{h_{r-\gamma}=1}^{H_{r-\gamma}} \bar{w}_{s_{r-\gamma+1} h_{r-\gamma}}^{(r-\gamma)} \sigma' \left( \chi_{h_{r-\gamma}}^{(r-\gamma)} \left( n - \sum_{\nu=1}^{\gamma} d_{r-\nu+1} - k \right) \right) \right\} \cdot \left. \sum_{s_{r-\gamma}=1}^{S_{r-\gamma}} \sum_{d_{r-\gamma}=1}^{\rho_{\gamma+1}} \tilde{w}_{h_{r-\gamma} s_{r-\gamma} d_{r-\gamma}}^{(r-\gamma)} \right\} \sum_{d_{m+1}=1}^{\rho_{r-m}} \tilde{w}_{h_{m+1} i d_{m+1}}^{(m+1)} \cdot \mu_j^{(m)}(n - d_r - d_{r-1} - \dots - d_{m+1} - k), \quad (2.71)$$

gdzie:

$$\begin{aligned} \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} &= \sigma' \left( \chi_i^{(m)}(n) \right) \hat{w}_{ik}^{(m)} \cdot \mathbf{1} \left( k \leq D_\mu^{(m)} \right) + \\ &+ \sum_{\tau=1}^{\min\{D_\mu^{(m)}, k-1\}} \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k+\tau)} \sigma' \left( \chi_i^{(m)}(n-k+\tau) \right) \cdot \hat{w}_{i\tau}^{(m)}. \end{aligned} \quad (2.72)$$

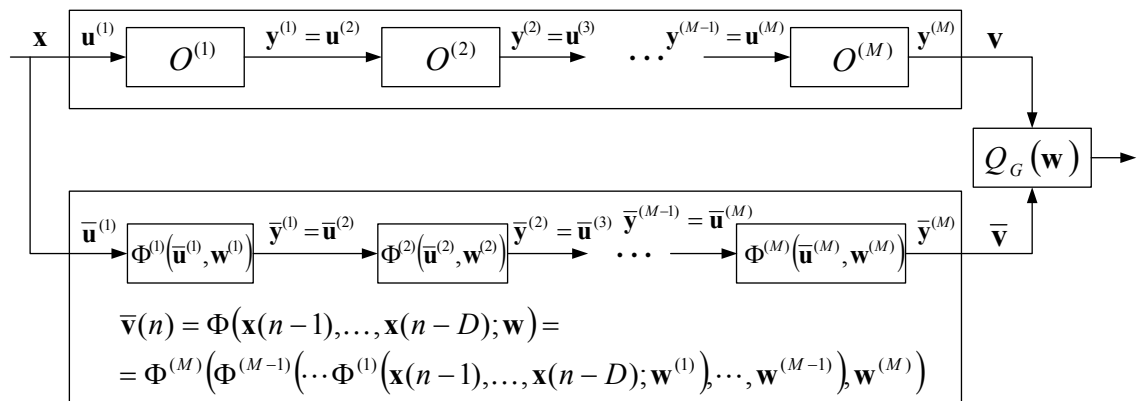
## 2.3 Model globalnie optymalny z uwzględnieniem jakości modeli lokalnych

Przedmiotem rozważań w tym rozdziale jest algorytm identyfikacji globalnej systemu złożonego, w którym uwzględnione są jakości modeli lokalnych. Jako model globalny systemu o strukturze kaskadowej przyjmujemy szczególny przypadek, zilustrowany na rys. 2.5. Wyróżnionymi wyjściami globalnymi są wyjścia ostatniego elementu w kaskadzie. Odpowiada to wyborowi wartości

$$\beta_1 = 0, \beta_2 = 0, \dots, \beta_{M-1} = 0, \beta_M = 1$$

w globalnym kryterium jakości identyfikacji  $Q_G(n, \mathbf{w})$ , (2.38), które przyjmuje wówczas postać:

$$Q_G(n, \mathbf{w}) = \frac{1}{2} \sum_{l=1}^{L_M} \left[ \varepsilon_l^{(M)}(n) \right]^2. \quad (2.73)$$



Rysunek 2.5: Model globalny – przypadek szczególny

Wyjście każdego modelu  $\Phi_m$  z parametrami  $\mathbf{w}^{(m)}$  można wyznaczać na dwa sposoby. W sposób lokalny, jako:

$$\bar{\mathbf{y}}^{(m)} = \Phi_m \left( \mathbf{u}^{(m)}, \mathbf{w}^{(m)} \right), \quad (2.74)$$

czyli wyjście modelu jest odpowiedzią na sygnały, jakie podano na wejścia obiektu lub jakie na tych wejściach zmierzono. Drugi sposób, globalny, polega na wyliczeniu wyjść tego samego modelu, z identycznymi wartościami parametrów, lecz przy użyciu jako sygnałów wejściowych wyjść modeli powiązanych elementów:

$$\bar{\mathbf{v}}^{(m)} = \Phi_m \left( \bar{\mathbf{u}}^{(m)}, \mathbf{w}^{(m)} \right). \quad (2.75)$$

W systemie o strukturze kaskadowej pomiary będą podawane jedynie na pierwszy model. Ogólnie, wartości składowych wektora  $\bar{\mathbf{u}}^{(m)}$  pochodzić będą z pomiarów tylko w przypadku gdy są one wejściami zewnętrznymi.

W tym rozdziale rozważymy konstrukcję algorytmów uczenia dających takie wartości parametrów  $\mathbf{w}^{(m)}$  ( $m = 1, 2, \dots, M$ ) modeli  $\Psi_m$ , dla których udaje się osiągnąć kompromis pomiędzy różnicami między  $\bar{\mathbf{y}}^{(m)}$  a  $\mathbf{y}^{(m)}$  oraz między  $\bar{\mathbf{v}}^{(m)}$  a  $\mathbf{v}^{(m)}$ . Zastosowane zostaną dwa podejścia: z użyciem syntetycznego wskaźnika jakości, z wykorzystaniem funkcji kary zewnętrznej i wewnętrznej nakładanej na rozwiązania prowadzące do niezadowolającej jakości modeli lokalnych.

### 2.3.1 Syntetyczny wskaźnik jakości

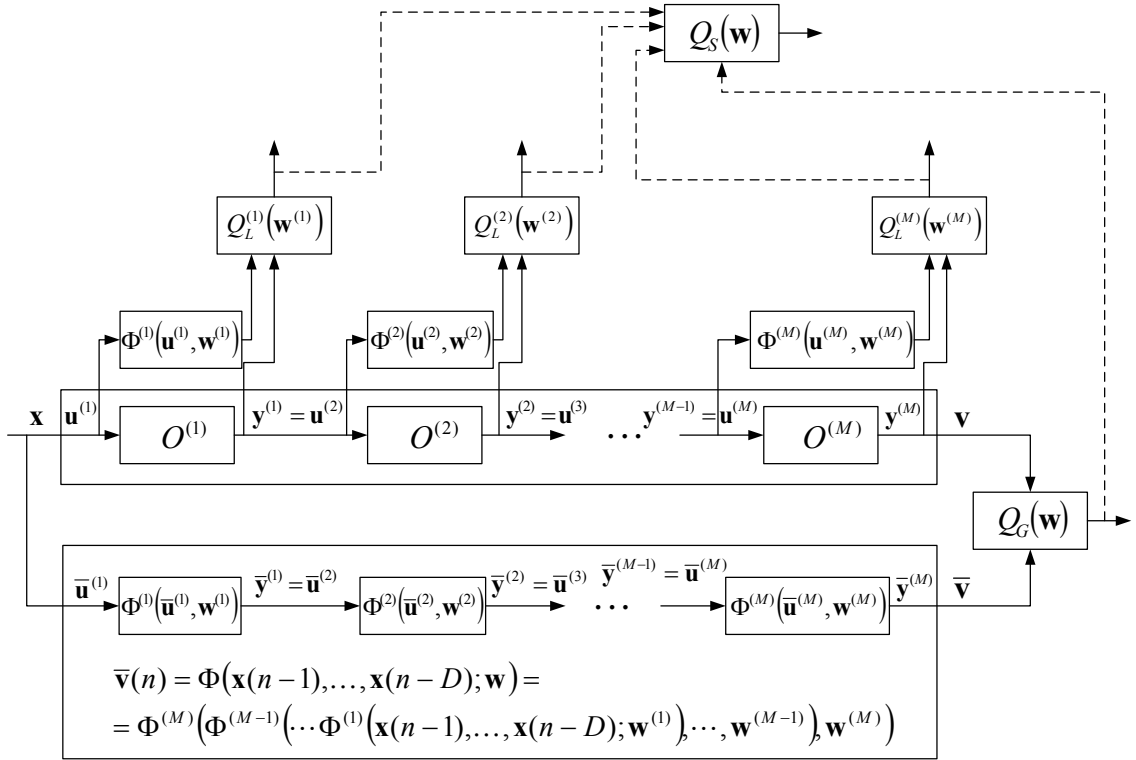
Istnieją różne metody poszukiwania rozwiązań problemów wielokryterialnych. Krótki przegląd podejść wielokryterialnych zawarto w pracy [88]. Tutaj posłużymy się metodą sprowadzenia problemu wielokryterialnego do jednokryterialnego przy użyciu syntetycznego wskaźnika jakości, [39]. Ideę oceny syntetycznej szczególnego przypadku ( $\beta_M = 1$ ) modelu globalnego o strukturze kaskadowej przedstawia rys. 2.6. Przypadek ogólny zilustrowano na rys. 2.7.

Jako wskaźnik syntetyczny zastosujemy kryterium (1.97) o postaci liniowej:

$$Q_S(n, \mathbf{w}) = \alpha_0 Q_G(n, \mathbf{w}) + \sum_{m=1}^M \alpha_m Q_L^{(m)} \left( n, \mathbf{w}^{(m)} \right), \quad (2.76)$$

gdzie  $\alpha_m \in [0, 1]$  ( $m = 1, 2, \dots, M$ ) są współczynnikami liczbowymi spełniającymi warunek

$$\sum_{m=0}^M \alpha_m = 1 \quad (2.77)$$



Rysunek 2.6: System złożony o strukturze kaskadowej ze szczególnym przypadkiem modelu globalnie optymalnego z uwzględnieniem jakości modeli lokalnych

i określającymi udział wartości lokalnych wskaźników  $(\alpha_1, \alpha_2, \dots, \alpha_M)$  oraz wskaźnika globalnego  $(\alpha_0)$  w wartości syntetycznego wskaźnika jakości identyfikacji. Wstawiając (2.5) i (2.73) do (2.78), możemy napisać:

$$Q_S(n, \mathbf{w}) = \frac{1}{2} \alpha_0 \sum_{l=1}^{L_M} [\varepsilon_l^{(M)}(n)]^2 + \frac{1}{2} \sum_{m=1}^M \alpha_m \sum_{l=1}^{L_m} [e_l^{(m)}(n)]^2, \quad (2.78)$$

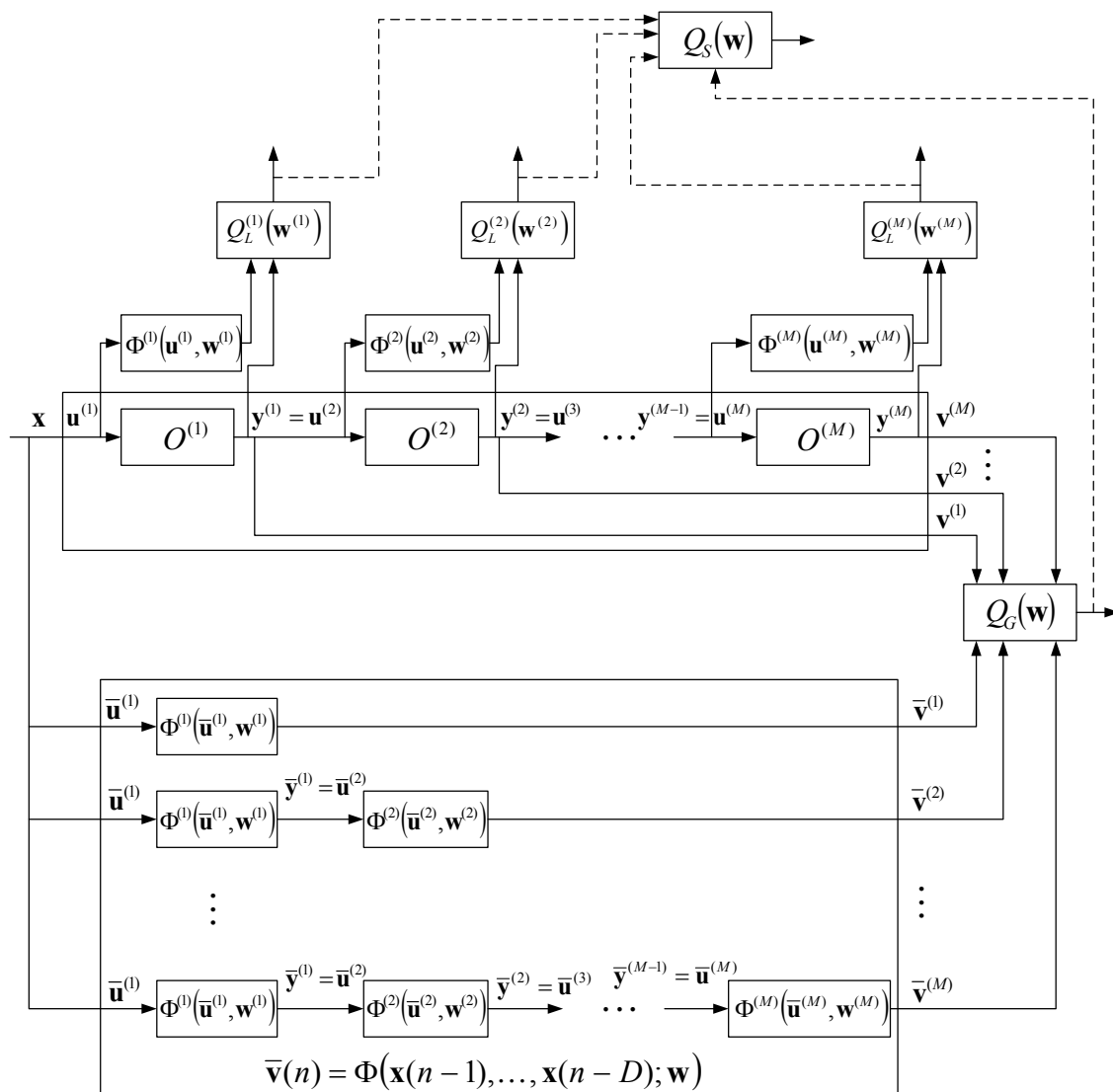
gdzie:

$$\varepsilon_l^{(M)}(n) = v_l^{(M)}(n) - \bar{v}_l^{(M)}(n), \quad e_l^{(m)}(n) = y_l^{(m)}(n) - \bar{y}_l^{(m)}(n),$$

a ostatecznie:

$$Q_S(n, \mathbf{w}) = \frac{1}{2} \alpha_0 \sum_{l=1}^{L_M} [v_l^{(M)}(n) - \bar{v}_l^{(M)}(n)]^2 + \frac{1}{2} \sum_{m=1}^M \alpha_m \sum_{l=1}^{L_m} [y_l^{(m)}(n) - \bar{y}_l^{(m)}(n)]^2. \quad (2.79)$$

W celu minimalizacji syntetycznego wskaźnika jakości  $Q_S(n, \mathbf{w})$  użyjemy metody gradientu prostego. Po zebraniu kolejnych pomiarów (2.4) i (2.22), aktualizujemy wartości parametrów wagowych  $m$ -tej  $(m = 1, 2, \dots, M)$  sieci według



Rysunek 2.7: System złożony o strukturze kaskadowej z modelem globalnie optymalnym z uwzględnieniem jakości modeli lokalnych

wzoru:

$$\mathbf{w}^{(m)}(n+1) = \mathbf{w}^{(m)}(n) - \eta^{(m)} \nabla_{\mathbf{w}^{(m)}} Q_S(n, \mathbf{w})|_{\mathbf{w}^{(m)} = \mathbf{w}^{(m)}(n)}. \quad (2.80)$$

Gradient z powyższej formuły ma ogólną postać:

$$\nabla_{\mathbf{w}^{(m)}} Q_S(n, \mathbf{w}) = -\alpha_0 \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \nabla_{\mathbf{w}^{(m)}} \bar{v}_l^{(M)}(n) - \alpha_m \sum_{l=1}^{L_m} e_l^{(m)}(n) \nabla_{\mathbf{w}^{(m)}} \bar{y}_l^{(m)}(n). \quad (2.81)$$



Pochodne cząstkowe  $\frac{\partial \bar{y}_l^{(m)}(n)}{\partial \bar{w}_{ij}^{(m)}}$ ,  $\frac{\partial \bar{y}_l^{(m)}(n)}{\partial \hat{w}_{ij}^{(m)}}$  i  $\frac{\partial \bar{y}_l^{(m)}(n)}{\partial \tilde{w}_{ijp}^{(m)}}$  będące składowymi gradientu  $\nabla_{\mathbf{w}^{(m)}} \bar{y}_l^{(m)}(n)$  zostały obliczone wcześniej, w równaniach (2.8), (2.11) i (2.9). Podobnie pochodne cząstkowe  $\frac{\partial \bar{v}_l^{(M)}(n)}{\partial \bar{w}_{ij}^{(m)}}$ ,  $\frac{\partial \bar{v}_l^{(M)}(n)}{\partial \hat{w}_{ij}^{(m)}}$  i  $\frac{\partial \bar{v}_l^{(M)}(n)}{\partial \tilde{w}_{ijp}^{(m)}}$  są szczególnym przypadkiem formuł (2.46), (2.44) i (2.45):

$$\frac{\partial \bar{v}_l^{(M)}(n)}{\partial \bar{w}_{ij}^{(m)}} = \sum_{h=1}^{H_M} \bar{w}_{ih}^{(M)} \frac{\partial^+ \mu_h^{(M)}(n)}{\partial \hat{w}_{ij}^{(m)}}, \quad (2.82)$$

$$\frac{\partial \bar{v}_l^{(M)}(n)}{\partial \tilde{w}_{ijp}^{(m)}} = \sum_{h=1}^{H_M} \bar{w}_{ih}^{(M)} \frac{\partial^+ \mu_h^{(M)}(n)}{\partial \tilde{w}_{ijp}^{(m)}}, \quad (2.83)$$

$$\frac{\partial \bar{v}_l^{(M)}(n)}{\partial \bar{w}_{ij}^{(m)}} = \sum_{h=1}^{H_M} \bar{w}_{lh}^{(M)} \frac{\partial^+ \mu_h^{(M)}(n)}{\partial \bar{w}_{ij}^{(m)}}. \quad (2.84)$$

Wstawiając do powyższych formuł wzory (2.55), (2.59) i (2.62) przy  $r = M$ , wyniki tych podstawień umieszczając jako składowe w wektorze  $\nabla_{\mathbf{w}^{(m)}} \bar{v}_l^{(M)}$  we wzorze (2.81) wraz z wyliczeniami (2.8), (2.11) i (2.9) i (2.13) umieszczonymi w wektorach  $\nabla_{\mathbf{w}^{(m)}} \bar{y}_l^{(m)}$  we wzorze (2.81), otrzymujemy – po podstawieniu do reguły (2.80) – algorytm  $\Psi_S$  identyfikacji globalnej z uwzględnieniem jakości modeli lokalnych dynamicznego systemu złożonego o strukturze kaskadowej, przybliżający wartości parametrów  $\mathbf{w}_S^*$  sieci globalnej DRNN minimalizujące syntetyczny wskaźnik jakości (2.76). Formuły dla  $m = 1, 2, \dots, M - 1$  są następujące:

$$\begin{aligned} \hat{w}_{ij}^{(m)}(n+1) &= \hat{w}_{ij}^{(m)}(n) + \eta^{(m)} \alpha_0 \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \sum_{h=1}^{H_M} \bar{w}_{lh}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_h^{(M)}(n)}{\partial \mu_h^{(M)}(n-k)} \\ &\cdot \sigma' \left( \chi_h^{(M)}(n-k) \right) \sum_{s_M=1}^{S_M} \sum_{d_M=1}^{\rho_1} \tilde{w}_{hs_M d_M}^{(M)} \prod_{\gamma=1}^{M-m-1} \left\{ \sum_{h_{M-\gamma}=1}^{H_{M-\gamma}} \bar{w}_{s_{M-\gamma+1} h_{M-\gamma}}^{(M-\gamma)} \right. \\ &\cdot \left. \sigma' \left( \chi_{h_{M-\gamma}}^{(M-\gamma)} \left( n - \sum_{\nu=1}^{\gamma} d_{M-\nu+1} - k \right) \right) \sum_{s_{M-\gamma}=1}^{S_{M-\gamma}} \sum_{d_{M-\gamma}=1}^{\rho_{\gamma+1}} \tilde{w}_{h_{M-\gamma} s_{M-\gamma} d_{M-\gamma}}^{(M-\gamma)} \right\} \\ &\cdot \bar{w}_{s_{m+1} i}^{(m)} \cdot \sigma' \left( \chi_i^{(m)}(n - d_M - d_{M-1} - \dots - d_{m+1} - k) \right). \end{aligned}$$

$$\begin{aligned}
 & \cdot \mu_i^{(m)}(n - d_M - d_{M-1} - \dots - d_{m+1} - j - k) + \eta^{(m)} \alpha_m \sum_{l=1}^{L_m} e_l^{(m)}(n) \cdot \\
 & \cdot \bar{w}_{ii}^{(m)} \sum_{k=0}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n - k)} \sigma' \left( \chi_i^{(m)}(n - k) \right) \mu_i^{(m)}(n - j - k), \\
 \tilde{w}_{ijp}^{(m)}(n + 1) = & \tilde{w}_{ijp}^{(m)}(n) + \eta^{(m)} \alpha_0 \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \sum_{h=1}^{H_M} \bar{w}_{lh}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_h^{(M)}(n)}{\partial \mu_h^{(M)}(n - k)} \cdot \\
 & \cdot \sigma' \left( \chi_h^{(M)}(n - k) \right) \sum_{s_M=1}^{S_M} \sum_{d_M=1}^{\rho_1} \tilde{w}_{hs_M d_M}^{(M)} \prod_{\gamma=1}^{M-m-1} \left\{ \sum_{h_{M-\gamma}=1}^{H_{M-\gamma}} \bar{w}_{s_{M-\gamma+1} h_{M-\gamma}}^{(M-\gamma)} \cdot \right. \\
 & \cdot \left. \sigma' \left( \chi_{h_{M-\gamma}}^{(M-\gamma)} \left( n - \sum_{\nu=1}^{\gamma} d_{M-\nu+1} - k \right) \right) \sum_{s_{M-\gamma}=1}^{S_{M-\gamma}} \sum_{d_{M-\gamma}=1}^{\rho_{\gamma+1}} \tilde{w}_{h_{M-\gamma} s_{M-\gamma} d_{M-\gamma}}^{(M-\gamma)} \right\} \cdot \\
 & \cdot \bar{w}_{s_{m+1} i}^{(m)} \cdot \sigma' \left( \chi_i^{(m)}(n - d_M - d_{M-1} - \dots - d_{m+1} - k) \right) \cdot \\
 & \cdot \bar{u}_j^{(m)}(n - d_M - d_{M-1} - \dots - d_{m+1} - p - k) + \eta^{(m)} \alpha_m \sum_{l=1}^{L_m} e_l^{(m)}(n) \cdot \\
 & \cdot \bar{w}_{ii}^{(m)} \sum_{k=0}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n - k)} \sigma' \left( \chi_i^{(m)}(n - k) \right) \bar{u}_j^{(m)}(n - p - k),
 \end{aligned} \tag{2.85}$$

(2.86)

$$\begin{aligned}
 \bar{w}_{ij}^{(m)}(n+1) &= \bar{w}_{ij}^{(m)} + \eta^{(m)} \alpha_0 \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \sum_{h=1}^{H_M} \bar{w}_{lh}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_h^{(M)}(n)}{\partial \mu_h^{(M)}(n-k)} \cdot \\
 &\cdot \sigma' \left( \chi_h^{(M)}(n-k) \right) \sum_{s_M=1}^{S_M} \sum_{d_M=1}^{\rho_1} \tilde{w}_{hs_M d_M}^{(M)} \prod_{\gamma=1}^{M-m-2} \left\{ \sum_{h_{M-\gamma}=1}^{H_{M-\gamma}} \bar{w}_{s_{M-\gamma+1} h_{M-\gamma}}^{(M-\gamma)} \cdot \right. \\
 &\cdot \left. \sigma' \left( \chi_{h_{M-\gamma}}^{(M-\gamma)} \left( n - \sum_{\nu=1}^{\gamma} d_{M-\nu+1} - k \right) \right) \sum_{s_{M-\gamma}=1}^{S_{M-\gamma}} \sum_{d_{M-\gamma}=1}^{\rho_{\gamma+1}} \tilde{w}_{h_{M-\gamma} s_{M-\gamma} d_{M-\gamma}}^{(M-\gamma)} \right\} \cdot \\
 &\cdot \sum_{d_{m+1}=1}^{\rho_{M-m}} \tilde{w}_{h_{m+1} i d_{m+1}}^{(m+1)} \cdot \mu_j^{(m)}(n - d_M - d_{M-1} - \dots - d_{m+1} - k) + \\
 &+ \eta^{(m)} \alpha_m e_i^{(m)}(n) \mu_j^{(m)}(n).
 \end{aligned} \tag{2.87}$$

Dla  $m = M$  obowiązują formuły:

$$\begin{aligned}
 \hat{w}_{ij}^{(M)}(n+1) &= \hat{w}_{ij}^{(M)}(n) + \eta^{(M)} \alpha_0 \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \cdot \bar{w}_{li}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_i^{(M)}(n)}{\partial \mu_i^{(M)}(n-k)} \cdot \\
 &\cdot \sigma' \left( \chi_i^{(M)}(n-k) \right) \mu_i^{(M)}(n-j-k) + \eta^{(M)} \alpha_M \sum_{l=1}^{L_M} e_l^{(M)}(n) \cdot \\
 &\cdot \bar{w}_{li}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_i^{(M)}(n)}{\partial \mu_i^{(M)}(n-k)} \sigma' \left( \chi_i^{(M)}(n-k) \right) \mu_i^{(M)}(n-j-k),
 \end{aligned}$$

co zapiszemy zwięźle jako:

$$\hat{w}_{ij}^{(M)}(n+1) = \hat{w}_{ij}^{(M)}(n) + \eta^{(M)} \sum_{l=1}^{L_M} \left( \alpha_0 \varepsilon_l^{(M)}(n) + \alpha_M e_l^{(M)}(n) \right) \bar{w}_{li}^{(M)}. \tag{2.88}$$

$$\cdot \sum_{k=0}^K \frac{\partial^+ \mu_i^{(M)}(n)}{\partial \mu_i^{(M)}(n-k)} \sigma' \left( \chi_i^{(M)}(n-k) \right) \mu_i^{(M)}(n-j-k),$$

$$\begin{aligned}
 \tilde{w}_{ijp}^{(M)}(n+1) &= \tilde{w}_{ijp}^{(M)}(n) + \eta^{(M)} \alpha_0 \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \cdot \bar{w}_{li}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_i^{(M)}(n)}{\partial \mu_i^{(M)}(n-k)} \cdot \\
 &\quad \cdot \sigma' \left( \chi_i^{(M)}(n-k) \right) \bar{u}_j^{(M)}(n-p-k) + \eta^{(M)} \alpha_M \sum_{l=1}^{L_M} e_l^{(M)}(n) \cdot \\
 &\quad \cdot \bar{w}_{li}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_i^{(M)}(n)}{\partial \mu_i^{(M)}(n-k)} \sigma' \left( \chi_i^{(M)}(n-k) \right) \bar{u}_j^{(M)}(n-p-k),
 \end{aligned} \tag{2.89}$$

co można zapisać krótko jako:

$$\tilde{w}_{ijp}^{(M)}(n+1) = \tilde{w}_{ijp}^{(M)}(n) + \eta^{(M)} \sum_{l=1}^{L_M} \left( \alpha_0 \varepsilon_l^{(M)}(n) + \alpha_M e_l^{(M)}(n) \right) \bar{w}_{li}^{(M)}. \tag{2.90}$$

$$\cdot \sum_{k=0}^K \frac{\partial^+ \mu_i^{(M)}(n)}{\partial \mu_i^{(M)}(n-k)} \sigma' \left( \chi_i^{(M)}(n-k) \right) \bar{u}_j^{(M)}(n-p-k),$$

$$\bar{w}_{ij}^{(M)}(n+1) = \bar{w}_{ij}^{(M)} + \eta^{(M)} \alpha_0 \varepsilon_i^{(M)}(n) \mu_j^{(M)}(n) + \eta^{(M)} \alpha_M e_i^{(M)}(n) \mu_j^{(M)}(n). \tag{2.91}$$

lub równoważnie:

$$\bar{w}_{ij}^{(M)}(n+1) = \bar{w}_{ij}^{(M)} + \eta^{(M)} \left( \alpha_0 \varepsilon_i^{(M)}(n) + \alpha_M e_i^{(M)}(n) \right) \mu_j^{(M)}(n). \tag{2.92}$$

Podobnie jak w poprzednim podpunkcie, należy tu podkreślić, że w powyższych wzorach, sprawiających wrażenie wymagających obliczeniowo, najczęściej wykonywanych jest tylko kilka kroków sumowań. Ponadto wiele wyników cząstkowych wykorzystywanych jest kilkakrotnie w różnych miejscach, więc staranna organizacja obliczeń pozwala na zmniejszenie złożoności algorytmu.

### 2.3.2 Zadawalające modele lokalne

Innym sposobem na uwzględnienie jakości modeli lokalnych podczas doboru parametrów modelu globalnego jest nałożenie ograniczeń na wartości parametrów modelu uzyskiwane w drodze minimalizacji globalnego wskaźnika jakości identyfikacji. W ogólnym przypadku, dla systemu złożonego o strukturze kaskadowej,

byłby to wskaźnik (2.38). W przypadku szczególnym modelu globalnego, przyjętym w poprzednim podpunkcie minimalizowany będzie wskaźnik (2.73):

$$Q_G(n, \mathbf{w}) = \frac{1}{2} \sum_{l=1}^{L_M} [\varepsilon_l^{(M)}(n)]^2, \quad (2.93)$$

lecz zarazem wymagane będzie spełnienie warunków (1.103):

$$Q_L^{(m)}(n, \mathbf{w}^{(m)}) \leq \lambda_m, \quad m = 1, 2, \dots, M, \quad (2.94)$$

gdzie  $\lambda_m$  jest najmniejszą dopuszczalną wartością lokalnego wskaźnika jakości  $Q_L^{(m)}$  dla modelu  $m$ -tego elementu. Oczywiście, nie może ona być mniejsza niż  $Q_L^{(m)}(\mathbf{w}_L^{(m)*})$ . Zgodnie z zasadami formułowania zadań optymalizacji, [9], [22], warunki (2.94) zapisujemy w postaci funkcji  $g_m$  ( $m = 1, 2, \dots, M$ ) ograniczeń:

$$g_m(n, \mathbf{w}^{(m)}) \triangleq Q_L^{(m)}(n, \mathbf{w}^{(m)}) - \lambda_m \leq 0. \quad (2.95)$$

W celu rozwiązania tak postawionego zadania modyfikujemy minimalizowaną funkcję kryterialną, dodając do niej tzw. *funkcję kary*. Nowe, zmodyfikowane kryterium jakości identyfikacji jest postaci:

$$Q_Z(n, \mathbf{w}, c) \triangleq Q_G(n, \mathbf{w}) + P(n, \mathbf{w}, c), \quad (2.96)$$

gdzie  $P(n, \mathbf{w}, c)$  – jest funkcją kary,  $c$  – to parametr liczbowy z przedziału  $(0, +\infty)$ . Wraz ze wzrostem wartości  $c$ , rośnie wartość funkcji kary. W ten sposób zadanie optymalizacji z ograniczeniami sprowadzone zostało do zadania optymalizacji bez ograniczeń, ze zmodyfikowaną funkcją celu (2.96). Przy odpowiednio dobranej postaci funkcji kary  $P$  i wartości jej parametru  $c$ , minimum funkcji kryterialnej  $Q_Z(n, \mathbf{w}, c)$  jest przybliżeniem minimum funkcji  $Q_G(n, \mathbf{w})$  przy ograniczeniach (2.94).

W zależności od postaci funkcji kary, numeryczny algorytm optymalizacji może prowadzić poszukiwania w całej przestrzeni rozwiązań lub może być ograniczony do obszaru rozwiązań dopuszczalnych. W pierwszym przypadku mówimy o *metodzie zewnętrznej funkcji kary*, w drugim o *metodzie wewnętrznej funkcji kary*, [42]. Funkcję  $P(n, \mathbf{w}, c)$  nazywać będziemy, w pierwszym przypadku – karą zewnętrzną, w drugim – karą wewnętrzną lub barierą.

**Funkcja kary zewnętrznej.** Kara zewnętrzna  $P_Z(n, \mathbf{w}, c)$  jest funkcją, która przyjmuje wartości niezerowe tylko poza obszarem rozwiązań dopuszczalnych. Ustalając, że wszystkie ograniczenia mają identyczne znaczenie, decydujemy się na poniższą postać funkcji kary, w której sumują się kary za przekroczenie poszczególnych ograniczeń:

$$P_Z(n, \mathbf{w}, c) = c \sum_{r=1}^M P_r(n, \mathbf{w}^{(r)}), \quad (2.97)$$

gdzie jako karę za niespełnienie  $r$ -tego ( $r = 1, 2, \dots, M$ ) ograniczenia przyjmujemy:

$$P_r(n, \mathbf{w}^{(r)}) = \max\{0, g_r(n, \mathbf{w}^{(r)})\}. \quad (2.98)$$

Zmodyfikowany wskaźnik jakości identyfikacji ma teraz postać:

$$Q_{ZZ}(n, \mathbf{w}) = \frac{1}{2} \sum_{l=1}^{L_M} [\varepsilon_l^{(M)}(n)]^2 + c \sum_{r=1}^M P_r(n, \mathbf{w}^{(r)}). \quad (2.99)$$

Zgodnie z metodą gradientu prostego, minimalizacja wskaźnika jakości  $Q_{ZZ}(n, \mathbf{w}, c)$  dokonywana jest na podstawie kolejnych danych pomiarowych (2.4), z wyróżnionymi w nich pomiarami (2.22), w drodze iteracyjnego aktualizowania wartości parametrów wagowych  $m$ -tej ( $m = 1, 2, \dots, M$ ) sieci według wzoru:

$$\mathbf{w}^{(m)}(n+1) = \mathbf{w}^{(m)}(n) - \eta^{(m)} \nabla_{\mathbf{w}^{(m)}} Q_{ZZ}(n, \mathbf{w})|_{\mathbf{w}^{(m)}=\mathbf{w}^{(m)}(n)}. \quad (2.100)$$

Gradient z funkcji kryterialnej (2.99) wyliczamy następująco:

$$\nabla_{\mathbf{w}^{(m)}} Q_{ZZ}(n, \mathbf{w}) = - \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \cdot \nabla_{\mathbf{w}^{(m)}} \bar{v}_l^{(M)}(n) + c \cdot \nabla_{\mathbf{w}^{(m)}} P_m(n, \mathbf{w}^{(m)}). \quad (2.101)$$

Funkcja  $P_m(n, \mathbf{w}^{(m)})$ , określona wzorem (2.98) jest nieróżniczkowalna w zerze, w związku z tym przyjmujemy przybliżenie:

$$\nabla_{\mathbf{w}^{(m)}} P_m(n, \mathbf{w}^{(m)}) = \begin{cases} 0 & \text{dla } g_m(n, \mathbf{w}^{(m)}) \leq 0 \\ \nabla_{\mathbf{w}^{(m)}} g_m(n, \mathbf{w}^{(m)}) & \text{dla } g_m(n, \mathbf{w}^{(m)}) > 0 \end{cases}, \quad (2.102)$$

a wstawiając wzór (2.95) i uwzględniając (2.7):

$$\nabla_{\mathbf{w}^{(m)}} g_m(n, \mathbf{w}^{(m)}) = \nabla_{\mathbf{w}^{(m)}} (Q_L^{(m)}(n, \mathbf{w}^{(m)}) - \lambda_m) = - \sum_{l=1}^{L_m} e_l^{(m)}(n) \nabla_{\mathbf{w}^{(m)}} \bar{y}_l^{(m)}(n), \quad (2.103)$$

dostajemy ostatecznie wyrażenie opisujące gradient z funkcji kryterialnej względem zestawu wag  $m$ -tej sieci:

$$\begin{aligned} \nabla_{\mathbf{w}^{(m)}} Q_{ZZ}(n, \mathbf{w}) = & - \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \cdot \nabla_{\mathbf{w}^{(m)}} \bar{v}_l^{(M)}(n) + \\ & - c \cdot \sum_{l=1}^{L_m} e_l^{(m)}(n) \cdot \nabla_{\mathbf{w}^{(m)}} \bar{y}_l^{(m)}(n) \cdot \mathbf{1} \left( g_m \left( n, \mathbf{w}^{(m)} \right) > 0 \right), \end{aligned} \quad (2.104)$$

gdzie:

$$\mathbf{1}(\kappa > k) = \begin{cases} 1 & \text{dla } \kappa > k \\ 0 & \text{w przeciwnym przypadku} \end{cases}. \quad (2.105)$$

Rozpisanie równania wektorowego (2.104) na poszczególne składowe, wstawienie lokalnych pochodnych cząstkowych  $\frac{\partial \bar{y}_l^{(m)}(n)}{\partial \bar{w}_{ij}^{(m)}}$ ,  $\frac{\partial \bar{y}_l^{(m)}(n)}{\partial \hat{w}_{ij}^{(m)}}$  i  $\frac{\partial \bar{y}_l^{(m)}(n)}{\partial \tilde{w}_{ijp}^{(m)}}$  z (2.8), (2.11) i (2.9) oraz globalnych pochodnych cząstkowych  $\frac{\partial \bar{v}_l^{(M)}(n)}{\partial \bar{w}_{ij}^{(m)}}$ ,  $\frac{\partial \bar{v}_l^{(M)}(n)}{\partial \hat{w}_{ij}^{(m)}}$  i  $\frac{\partial \bar{v}_l^{(M)}(n)}{\partial \tilde{w}_{ijp}^{(m)}}$  z (2.82), (2.83) i (2.84), daje nam reguły aktualizacji wag  $m$ -tej ( $m = 1, 2, \dots, M$ ) sieci DRNN:

$$\begin{aligned} \tilde{w}_{ijp}^{(m)}(n+1) = & \tilde{w}_{ijp}^{(m)}(n) + \eta^{(m)} \left( \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \frac{\partial \bar{v}_l^{(M)}(n)}{\partial \tilde{w}_{ijp}^{(m)}} + \right. \\ & \left. + c \cdot \sum_{l=1}^{L_m} e_l^{(m)}(n) \frac{\partial \bar{y}_l^{(m)}(n)}{\partial \tilde{w}_{ijp}^{(m)}} \cdot \mathbf{1} \left( g_m(n, \mathbf{w}^{(m)}) > 0 \right) \right), \end{aligned} \quad (2.106)$$

$$\begin{aligned} \hat{w}_{ij}^{(m)}(n+1) = & \hat{w}_{ij}^{(m)}(n) + \eta^{(m)} \left( \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \frac{\partial \bar{v}_l^{(M)}(n)}{\partial \hat{w}_{ij}^{(m)}} + \right. \\ & \left. + c \cdot \sum_{l=1}^{L_m} e_l^{(m)}(n) \frac{\partial \bar{y}_l^{(m)}(n)}{\partial \hat{w}_{ij}^{(m)}} \cdot \mathbf{1} \left( g_m(n, \mathbf{w}^{(m)}) > 0 \right) \right), \end{aligned} \quad (2.107)$$

$$\begin{aligned}
 \bar{w}_{ij}^{(m)}(n+1) &= \bar{w}_{ij}^{(m)}(n) + \eta^{(m)} \left( \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \frac{\partial \bar{v}_l^{(M)}(n)}{\partial \bar{w}_{ij}^{(m)}} + \right. \\
 &\quad \left. + c \cdot \sum_{l=1}^{L_m} e_l^{(m)}(n) \frac{\partial \bar{y}_l^{(m)}(n)}{\partial \bar{w}_{ij}^{(m)}} \cdot \mathbf{1} \left( g_m(n, \mathbf{w}^{(m)}) > 0 \right) \right). \tag{2.108}
 \end{aligned}$$

Dokładna postać reguł uczenia sieci  $m$ -tej ( $m = 1, 2, \dots, M-1$ ) jest więc następująca:

$$\begin{aligned}
 \tilde{w}_{ijp}^{(m)}(n+1) &= \tilde{w}_{ijp}^{(m)}(n) + \eta^{(m)} \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \sum_{h=1}^{H_M} \bar{w}_{lh}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_h^{(M)}(n)}{\partial \mu_h^{(M)}(n-k)} \cdot \\
 &\quad \cdot \sigma' \left( \chi_h^{(M)}(n-k) \right) \sum_{s_M=1}^{S_M} \sum_{d_M=1}^{\rho_1} \tilde{w}_{hs_M d_M}^{(M)} \prod_{\gamma=1}^{M-m-1} \left\{ \sum_{h_{M-\gamma}=1}^{H_{M-\gamma}} \bar{w}_{s_{M-\gamma+1} h_{M-\gamma}}^{(M-\gamma)} \cdot \right. \\
 &\quad \left. \cdot \sigma' \left( \chi_{h_{M-\gamma}}^{(M-\gamma)} \left( n - \sum_{\nu=1}^{\gamma} d_{M-\nu+1} - k \right) \right) \sum_{s_{M-\gamma}=1}^{S_{M-\gamma}} \sum_{d_{M-\gamma}=1}^{\rho_{\gamma+1}} \tilde{w}_{h_{M-\gamma} s_{M-\gamma} d_{M-\gamma}}^{(M-\gamma)} \right\} \cdot \\
 &\quad \cdot \bar{w}_{s_{m+1} i}^{(m)} \cdot \sigma' \left( \chi_i^{(m)}(n - d_M - d_{M-1} - \dots - d_{m+1} - k) \right) \cdot \\
 &\quad \cdot \bar{u}_j^{(m)}(n - d_M - d_{M-1} - \dots - d_{m+1} - p - k) + \\
 &\quad + \eta^{(m)} c \sum_{l=1}^{L_m} e_l^{(m)}(n) \cdot \bar{w}_{li}^{(m)} \sum_{k=0}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} \sigma' \left( \chi_i^{(m)}(n-k) \right) \cdot \tag{2.109} \\
 &\quad \cdot \bar{u}_j^{(m)}(n - p - k) \cdot \mathbf{1} \left( g_m(n, \mathbf{w}^{(m)}) > 0 \right),
 \end{aligned}$$



$$\begin{aligned}
 \widehat{w}_{ij}^{(m)}(n+1) &= \widehat{w}_{ij}^{(m)}(n) + \eta^{(m)} \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \sum_{h=1}^{H_M} \bar{w}_{lh}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_h^{(M)}(n)}{\partial \mu_h^{(M)}(n-k)} \\
 &\cdot \sigma' \left( \chi_h^{(M)}(n-k) \right) \sum_{s_M=1}^{S_M} \sum_{d_M=1}^{\rho_1} \tilde{w}_{hs_M d_M}^{(M)} \prod_{\gamma=1}^{M-m-1} \left\{ \sum_{h_{M-\gamma}=1}^{H_{M-\gamma}} \bar{w}_{s_{M-\gamma+1} h_{M-\gamma}}^{(M-\gamma)} \cdot \right. \\
 &\cdot \left. \sigma' \left( \chi_{h_{M-\gamma}}^{(M-\gamma)} \left( n - \sum_{\nu=1}^{\gamma} d_{M-\nu+1} - k \right) \right) \sum_{s_{M-\gamma}=1}^{S_{M-\gamma}} \sum_{d_{M-\gamma}=1}^{\rho_{\gamma+1}} \tilde{w}_{h_{M-\gamma} s_{M-\gamma} d_{M-\gamma}}^{(M-\gamma)} \right\} \\
 &\cdot \bar{w}_{s_{m+1} i}^{(m)} \cdot \sigma' \left( \chi_i^{(m)}(n - d_M - d_{M-1} - \dots - d_{m+1} - k) \right) \cdot \\
 &\cdot \mu_i^{(m)}(n - d_M - d_{M-1} - \dots - d_{m+1} - j - k) + \\
 &+ \eta^{(m)} c \cdot \sum_{l=1}^{L_m} e_l^{(m)}(n) \cdot \bar{w}_{li}^{(m)} \sum_{k=0}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} \sigma' \left( \chi_i^{(m)}(n-k) \right) \cdot
 \end{aligned} \tag{2.110}$$

$$\cdot \mu_i^{(m)}(n - j - k) \cdot \mathbf{1} \left( g_m(n, \mathbf{w}^{(m)}) > 0 \right),$$

$$\begin{aligned}
 \bar{w}_{ij}^{(m)}(n+1) &= \bar{w}_{ij}^{(m)}(n) + \eta^{(m)} \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \sum_{h=1}^{H_M} \bar{w}_{lh}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_h^{(M)}(n)}{\partial \mu_h^{(M)}(n-k)} \\
 &\cdot \sigma' \left( \chi_h^{(M)}(n-k) \right) \sum_{s_M=1}^{S_M} \sum_{d_M=1}^{\rho_1} \tilde{w}_{hs_M d_M}^{(M)} \prod_{\gamma=1}^{M-m-2} \left\{ \sum_{h_{M-\gamma}=1}^{H_{M-\gamma}} \bar{w}_{s_{M-\gamma+1} h_{M-\gamma}}^{(M-\gamma)} \cdot \right. \\
 &\cdot \left. \sigma' \left( \chi_{h_{M-\gamma}}^{(M-\gamma)} \left( n - \sum_{\nu=1}^{\gamma} d_{M-\nu+1} - k \right) \right) \sum_{s_{M-\gamma}=1}^{S_{M-\gamma}} \sum_{d_{M-\gamma}=1}^{\rho_{\gamma+1}} \tilde{w}_{h_{M-\gamma} s_{M-\gamma} d_{M-\gamma}}^{(M-\gamma)} \right\} \\
 &\cdot \sum_{d_{m+1}=1}^{\rho_{M-m}} \tilde{w}_{h_{m+1} i d_{m+1}}^{(m+1)} \cdot \mu_j^{(m)}(n - d_M - d_{M-1} - \dots - d_{m+1} - k) +
 \end{aligned} \tag{2.111}$$

$$+ \eta^{(m)} c \cdot \sum_{l=1}^{L_m} e_l^{(m)}(n) \cdot \mu_j^{(m)}(n) \cdot \mathbf{1} \left( g_m(n, \mathbf{w}^{(m)}) > 0 \right),$$

natomiast w uczeniu sieci  $M$ -tej obowiązują formuły:

$$\begin{aligned}
 \tilde{w}_{ijp}^{(M)}(n+1) = & \tilde{w}_{ijp}^{(M)}(n) + \eta^{(M)} \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \cdot \bar{w}_{li}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_i^{(M)}(n)}{\partial \mu_i^{(M)}(n-k)} \sigma' \left( \chi_i^{(M)}(n-k) \right) \cdot \\
 & \cdot \bar{w}_j^{(M)}(n-p-k) + \eta^{(M)} c \sum_{l=1}^{L_M} e_l^{(M)}(n) \cdot \bar{w}_{li}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_i^{(M)}(n)}{\partial \mu_i^{(M)}(n-k)} \cdot \\
 & \cdot \sigma' \left( \chi_i^{(M)}(n-k) \right) \cdot \bar{u}_j^{(M)}(n-p-k) \cdot \mathbf{1} \left( g_M(n, \mathbf{w}^{(M)}) > 0 \right),
 \end{aligned} \tag{2.112}$$

co zwięźlej można zapisać jako:

$$\begin{aligned}
 \tilde{w}_{ijp}^{(M)}(n+1) = & \tilde{w}_{ijp}^{(M)}(n) + \eta^{(M)} \sum_{l=1}^{L_M} \left[ \varepsilon_l^{(M)}(n) + c \cdot e_l^{(M)}(n) \cdot \mathbf{1} \left( g_M(n, \mathbf{w}^{(M)}) > 0 \right) \right] \cdot \\
 & \cdot \bar{w}_{li}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_i^{(M)}(n)}{\partial \mu_i^{(M)}(n-k)} \sigma' \left( \chi_i^{(M)}(n-k) \right) \cdot \bar{u}_j^{(M)}(n-p-k)
 \end{aligned} \tag{2.113}$$

Analogicznie dla pozostałych typów wag:

$$\begin{aligned}
 \hat{w}_{ij}^{(M)}(n+1) = & \hat{w}_{ij}^{(M)}(n) + \eta^{(M)} \sum_{l=1}^{L_M} \left[ \varepsilon_l^{(M)}(n) + c \cdot e_l^{(M)}(n) \cdot \mathbf{1} \left( g_M(n, \mathbf{w}^{(M)}) > 0 \right) \right] \cdot \\
 & \cdot \bar{w}_{li}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_i^{(M)}(n)}{\partial \mu_i^{(M)}(n-k)} \sigma' \left( \chi_i^{(M)}(n-k) \right) \mu_i^{(M)}(n-j-k),
 \end{aligned} \tag{2.114}$$

$$\bar{w}_{ij}^{(M)}(n+1) = \bar{w}_{ij}^{(M)}(n) + \eta^{(M)} \left[ \varepsilon_i^{(M)}(n) + c \cdot e_i^{(M)}(n) \cdot \mathbf{1} \left( g_M(n, \mathbf{w}^{(M)}) > 0 \right) \right] \cdot \mu_j^{(M)}(n), \tag{2.115}$$

Reguły uczenia (2.109), (2.110), (2.111) oraz (2.113), (2.114), (2.115) stanowią algorytm  $\Psi_{GL}$  identyfikacji globalnej z uwzględnieniem jakości modeli lokalnych dynamicznego systemu złożonego o strukturze kaskadowej, przybliżający wartości parametrów  $\mathbf{w}_{GL}^*$  sieci złożonej DRNN. Parametry  $\mathbf{w}_{GL}^*$  minimalizują, zmodyfikowany o funkcję kary zewnętrznej, wskaźnik jakości (2.96) lub – równoważnie –

globalny wskaźnik jakości, przypadek szczególny, (2.93) z ograniczeniami (2.94) na jakości modeli lokalnych.

**Funkcja kary wewnętrznej (bariera).** W metodzie wewnętrznej funkcji kary punkt poszukiwań nie może wyjść poza obszar rozwiązań dopuszczalnych, a kara nakładana jest za zbliżanie się do granicy tego obszaru. Kara wewnętrzna  $P_W(n, \mathbf{w}, c)$  jest funkcją, określoną tylko w obszarze rozwiązań dopuszczalnych, przy czym na granicach tego obszaru wartości funkcji  $P_W$  zmierzają do nieskończoności. Ustalając jak poprzednio, że wszystkie ograniczenia mają identyczne znaczenie, decydujemy się na poniższą postać funkcji kary wewnętrznej, w której sumują się kary za przekroczenie poszczególnych ograniczeń:

$$P_W(n, \mathbf{w}, c) = \frac{1}{c} \sum_{r=1}^M B_r(n, \mathbf{w}^{(r)}), \quad (2.116)$$

gdzie jako karę za niespełnienie  $r$ -tego ( $r = 1, 2, \dots, M$ ) ograniczenia przyjmujemy:

$$B_r(n, \mathbf{w}^{(r)}) = -\frac{1}{g_r(n, \mathbf{w}^{(r)})}. \quad (2.117)$$

Zmodyfikowany wskaźnik jakości identyfikacji ma tu postać:

$$Q_{ZW}(n, \mathbf{w}) = \frac{1}{2} \sum_{l=1}^{L_M} [\varepsilon_l^{(M)}(n)]^2 + \frac{1}{c} \sum_{r=1}^M B_r(n, \mathbf{w}^{(r)}). \quad (2.118)$$

Korzystając w każdym,  $n$ -tym, takcie z danych pomiarowych (2.4), z wyróżnionymi w nich pomiarami (2.22), minimalizujemy wskaźnik jakości  $Q_{ZW}(n, \mathbf{w}, c)$ , aktualizując wartości parametrów wagowych  $m$ -tej ( $m = 1, 2, \dots, M$ ) sieci według metody gradientu prostego (2.100). Gradient z funkcji kryterialnej (2.118) wyliczamy następująco:

$$\nabla_{\mathbf{w}^{(m)}} Q_{ZW}(n, \mathbf{w}) = -\sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \nabla_{\mathbf{w}^{(m)}} \bar{v}_l^{(M)}(n) + \frac{1}{c} \nabla_{\mathbf{w}^{(m)}} B_m(n, \mathbf{w}^{(m)}). \quad (2.119)$$

Tym razem nie ma problemu z różniczkowalnością funkcji  $B_m(n, \mathbf{w}^{(m)})$ , więc gradient z tej funkcji względem parametrów  $m$ -tej sieci jest równy:

$$\nabla_{\mathbf{w}^{(m)}} B_m(n, \mathbf{w}^{(m)}) = \frac{1}{[g_m(n, \mathbf{w}^{(m)})]^2} \nabla_{\mathbf{w}^{(m)}} g_m(n, \mathbf{w}^{(m)}), \quad (2.120)$$

a uwzględniając (2.95), otrzymujemy:

$$\nabla_{\mathbf{w}^{(m)}} B_m(n, \mathbf{w}^{(m)}) = -\frac{1}{[g_m(n, \mathbf{w}^{(m)})]^2} \sum_{l=1}^{L_m} e_l^{(m)}(n) \nabla_{\mathbf{w}^{(m)}} \bar{y}_l^{(m)}(n). \quad (2.121)$$

Ostatecznie, wyrażenie opisujące gradient z funkcji kryterialnej względem wag  $m$ -tej sieci przyjmuje postać:

$$\begin{aligned} \nabla_{\mathbf{w}^{(m)}} Q_{ZW}(n, \mathbf{w}) = & - \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \nabla_{\mathbf{w}^{(m)}} \bar{v}_l^{(M)}(n) + \\ & - \frac{1}{c [g_m(n, \mathbf{w}^{(m)})]^2} \sum_{l=1}^{L_m} e_l^{(m)}(n) \nabla_{\mathbf{w}^{(m)}} \bar{y}_l^{(m)}(n). \end{aligned} \quad (2.122)$$

Po rozpisaniu równania wektorowego (2.122) na poszczególne składowe, wstawieniu lokalnych pochodnych cząstkowych  $\frac{\partial \bar{y}_l^{(m)}(n)}{\partial \bar{w}_{ij}^{(m)}}$ ,  $\frac{\partial \bar{y}_l^{(m)}(n)}{\partial \hat{w}_{ij}^{(m)}}$  i  $\frac{\partial \bar{y}_l^{(m)}(n)}{\partial \tilde{w}_{ijp}^{(m)}}$  z (2.8), (2.11) i (2.9) oraz globalnych pochodnych cząstkowych  $\frac{\partial \bar{v}_l^{(M)}(n)}{\partial \bar{w}_{ij}^{(m)}}$ ,  $\frac{\partial \bar{v}_l^{(M)}(n)}{\partial \hat{w}_{ij}^{(m)}}$  i  $\frac{\partial \bar{v}_l^{(M)}(n)}{\partial \tilde{w}_{ijp}^{(m)}}$  z (2.82), (2.83) i (2.84), dostajemy reguły aktualizacji wag  $m$ -tej ( $m = 1, 2, \dots, M$ ) sieci DRNN:

$$\begin{aligned} \tilde{w}_{ijp}^{(m)}(n+1) = & \tilde{w}_{ijp}^{(m)}(n) + \eta^{(m)} \left( \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \frac{\partial \bar{v}_l^{(M)}(n)}{\partial \tilde{w}_{ijp}^{(m)}} + \right. \\ & \left. + \frac{1}{c [g_m(n, \mathbf{w}^{(m)})]^2} \sum_{l=1}^{L_m} e_l^{(m)}(n) \frac{\partial \bar{y}_l^{(m)}(n)}{\partial \tilde{w}_{ijp}^{(m)}} \right), \end{aligned} \quad (2.123)$$

$$\begin{aligned} \hat{w}_{ij}^{(m)}(n+1) = & \hat{w}_{ij}^{(m)}(n) + \eta^{(m)} \left( \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \frac{\partial \bar{v}_l^{(M)}(n)}{\partial \hat{w}_{ij}^{(m)}} + \right. \\ & \left. + \frac{1}{c [g_m(n, \mathbf{w}^{(m)})]^2} \sum_{l=1}^{L_m} e_l^{(m)}(n) \frac{\partial \bar{y}_l^{(m)}(n)}{\partial \hat{w}_{ij}^{(m)}} \right), \end{aligned} \quad (2.124)$$

$$\begin{aligned} \bar{w}_{ij}^{(m)}(n+1) = & \bar{w}_{ij}^{(m)}(n) + \eta^{(m)} \left( \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \frac{\partial \bar{v}_l^{(M)}(n)}{\partial \bar{w}_{ij}^{(m)}} + \right. \\ & \left. + \frac{1}{c [g_m(n, \mathbf{w}^{(m)})]^2} \sum_{l=1}^{L_m} e_l^{(m)}(n) \frac{\partial \bar{y}_l^{(m)}(n)}{\partial \bar{w}_{ij}^{(m)}} \right), \end{aligned} \quad (2.125)$$

Dokładna postać reguł uczenia sieci  $m$ -tej ( $m = 1, 2, \dots, M - 1$ ) jest więc następująca:

$$\begin{aligned}
 \tilde{w}_{ijp}^{(m)}(n+1) &= \tilde{w}_{ijp}^{(m)}(n) + \eta^{(m)} \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \sum_{h=1}^{H_M} \bar{w}_{lh}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_h^{(M)}(n)}{\partial \mu_h^{(M)}(n-k)} \cdot \\
 &\cdot \sigma' \left( \chi_h^{(M)}(n-k) \right) \sum_{s_M=1}^{S_M} \sum_{d_M=1}^{\rho_1} \tilde{w}_{hs_M d_M}^{(M)} \prod_{\gamma=1}^{M-m-1} \left\{ \sum_{h_{M-\gamma}=1}^{H_{M-\gamma}} \bar{w}_{s_{M-\gamma+1} h_{M-\gamma}}^{(M-\gamma)} \cdot \right. \\
 &\cdot \left. \sigma' \left( \chi_{h_{M-\gamma}}^{(M-\gamma)} \left( n - \sum_{\nu=1}^{\gamma} d_{M-\nu+1} - k \right) \right) \sum_{s_{M-\gamma}=1}^{S_{M-\gamma}} \sum_{d_{M-\gamma}=1}^{\rho_{\gamma+1}} \tilde{w}_{h_{M-\gamma} s_{M-\gamma} d_{M-\gamma}}^{(M-\gamma)} \right\} \cdot \\
 &\cdot \bar{w}_{s_{m+1} i}^{(m)} \cdot \sigma' \left( \chi_i^{(m)}(n - d_M - d_{M-1} - \dots - d_{m+1} - k) \right) \cdot \\
 &\cdot \bar{u}_j^{(m)}(n - d_M - d_{M-1} - \dots - d_{m+1} - p - k) + \\
 &+ \eta^{(m)} \frac{1}{c [g_m(n, \mathbf{w}^{(m)})]^2} \sum_{l=1}^{L_m} e_l^{(m)}(n) \cdot \bar{w}_{li}^{(m)} \sum_{k=0}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)}. \tag{2.126}
 \end{aligned}$$

$$\cdot \sigma' \left( \chi_i^{(m)}(n-k) \right) \bar{u}_j^{(m)}(n-p-k),$$

$$\begin{aligned}
 \hat{w}_{ij}^{(m)}(n+1) &= \hat{w}_{ij}^{(m)}(n) + \eta^{(m)} \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \sum_{h=1}^{H_M} \bar{w}_{lh}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_h^{(M)}(n)}{\partial \mu_h^{(M)}(n-k)} \cdot \\
 &\cdot \sigma' \left( \chi_h^{(M)}(n-k) \right) \sum_{s_M=1}^{S_M} \sum_{d_M=1}^{\rho_1} \tilde{w}_{hs_M d_M}^{(M)} \prod_{\gamma=1}^{M-m-1} \left\{ \sum_{h_{M-\gamma}=1}^{H_{M-\gamma}} \bar{w}_{s_{M-\gamma+1} h_{M-\gamma}}^{(M-\gamma)} \cdot \right. \\
 &\cdot \left. \sigma' \left( \chi_{h_{M-\gamma}}^{(M-\gamma)} \left( n - \sum_{\nu=1}^{\gamma} d_{M-\nu+1} - k \right) \right) \sum_{s_{M-\gamma}=1}^{S_{M-\gamma}} \sum_{d_{M-\gamma}=1}^{\rho_{\gamma+1}} \tilde{w}_{h_{M-\gamma} s_{M-\gamma} d_{M-\gamma}}^{(M-\gamma)} \right\} \cdot \\
 &\cdot \bar{w}_{s_{m+1} i}^{(m)} \cdot \sigma' \left( \chi_i^{(m)}(n - d_M - d_{M-1} - \dots - d_{m+1} - k) \right) \cdot \\
 &\cdot \mu_i^{(m)}(n - d_M - d_{M-1} - \dots - d_{m+1} - j - k) +
 \end{aligned}$$

$$\begin{aligned}
 & + \eta^{(m)} \frac{1}{c [g_m(n, \mathbf{w}^{(m)})]^2} \sum_{l=1}^{L_m} e_l^{(m)}(n) \cdot \bar{w}_{li}^{(m)} \sum_{k=0}^K \frac{\partial^+ \mu_i^{(m)}(n)}{\partial \mu_i^{(m)}(n-k)} \cdot \\
 & \cdot \sigma' \left( \chi_i^{(m)}(n-k) \right) \cdot \mu_i^{(m)}(n-j-k),
 \end{aligned} \tag{2.127}$$

$$\begin{aligned}
 \bar{w}_{ij}^{(m)}(n+1) & = \bar{w}_{ij}^{(m)}(n) + \eta^{(m)} \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \sum_{h=1}^{H_M} \bar{w}_{lh}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_h^{(M)}(n)}{\partial \mu_h^{(M)}(n-k)} \cdot \\
 & \cdot \sigma' \left( \chi_h^{(M)}(n-k) \right) \sum_{s_M=1}^{S_M} \sum_{d_M=1}^{\rho_1} \tilde{w}_{hs_M d_M}^{(M)} \prod_{\gamma=1}^{M-m-2} \left\{ \sum_{h_{M-\gamma}=1}^{H_{M-\gamma}} \bar{w}_{s_{M-\gamma+1} h_{M-\gamma}}^{(M-\gamma)} \cdot \right. \\
 & \cdot \left. \sigma' \left( \chi_{h_{M-\gamma}}^{(M-\gamma)} \left( n - \sum_{\nu=1}^{\gamma} d_{M-\nu+1} - k \right) \right) \sum_{s_{M-\gamma}=1}^{S_{M-\gamma}} \sum_{d_{M-\gamma}=1}^{\rho_{\gamma+1}} \tilde{w}_{h_{M-\gamma} s_{M-\gamma} d_{M-\gamma}}^{(M-\gamma)} \right\}.
 \end{aligned} \tag{2.128}$$

$$\begin{aligned}
 & \cdot \sum_{d_{m+1}=1}^{\rho_{M-m}} \tilde{w}_{h_{m+1} i d_{m+1}}^{(m+1)} \cdot \mu_j^{(m)}(n - d_M - d_{M-1} - \dots - d_{m+1} - k) + \\
 & + \eta^{(m)} \frac{1}{c [g_m(n, \mathbf{w}^{(m)})]^2} \sum_{l=1}^{L_m} e_l^{(m)}(n) \cdot \mu_j^{(m)}(n).
 \end{aligned} \tag{2.129}$$

Przy uczeniu sieci  $M$ -tej obowiązują formuły:

$$\begin{aligned}
 \tilde{w}_{ijp}^{(M)}(n+1) & = \tilde{w}_{ijp}^{(M)}(n) + \eta^{(M)} \sum_{l=1}^{L_M} \varepsilon_l^{(M)}(n) \cdot \bar{w}_{li}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_i^{(M)}(n)}{\partial \mu_i^{(M)}(n-k)} \cdot \\
 & \cdot \sigma' \left( \chi_i^{(M)}(n-k) \right) \cdot \bar{u}_j^{(M)}(n-p-k) + \eta^{(M)} \frac{1}{c [g_M(n, \mathbf{w}^{(M)})]^2} \cdot \\
 & \cdot \sum_{l=1}^{L_M} e_l^{(M)}(n) \cdot \bar{w}_{li}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_i^{(M)}(n)}{\partial \mu_i^{(M)}(n-k)} \sigma' \left( \chi_i^{(M)}(n-k) \right) \cdot \bar{u}_j^{(M)}(n-p-k),
 \end{aligned}$$

co zwięźle można zapisać jako:

$$\begin{aligned} \tilde{w}_{ijp}^{(M)}(n+1) &= \tilde{w}_{ijp}^{(M)}(n) + \eta^{(M)} \sum_{l=1}^{L_M} \left( \varepsilon_l^{(M)}(n) + \frac{1}{c [g_M(n, \mathbf{w}^{(M)})]^2} e_l^{(M)}(n) \right) \cdot \\ &\cdot \bar{w}_{li}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_i^{(M)}(n)}{\partial \mu_i^{(M)}(n-k)} \sigma'(\chi_i^{(M)}(n-k)) \bar{u}_j^{(M)}(n-p-k), \end{aligned} \quad (2.130)$$

Dla pozostałych typów wag  $M$ -tej sieci mamy analogicznie:

$$\begin{aligned} \hat{w}_{ij}^{(M)}(n+1) &= \hat{w}_{ij}^{(M)}(n) + \eta^{(M)} \sum_{l=1}^{L_M} \left( \varepsilon_l^{(M)}(n) + \frac{1}{c [g_M(n, \mathbf{w}^{(M)})]^2} e_l^{(M)}(n) \right) \cdot \\ &\cdot \bar{w}_{li}^{(M)} \sum_{k=0}^K \frac{\partial^+ \mu_i^{(M)}(n)}{\partial \mu_i^{(M)}(n-k)} \sigma'(\chi_i^{(M)}(n-k)) \mu_i^{(M)}(n-j-k), \end{aligned} \quad (2.131)$$

oraz

$$\bar{w}_{ij}^{(M)}(n+1) = \bar{w}_{ij}^{(M)}(n) + \eta^{(M)} \left( \varepsilon_i^{(M)}(n) + \frac{1}{c [g_M(n, \mathbf{w}^{(M)})]^2} e_i^{(M)}(n) \right) \mu_j^{(M)}. \quad (2.132)$$

Otrzymane reguły uczenia (2.126), (2.127), (2.129) oraz (2.130), (2.131), (2.132) stanowią algorytm  $\Psi_{GL}$  identyfikacji globalnej z uwzględnieniem jakości modeli lokalnych dynamicznego systemu złożonego o strukturze kaskadowej, przybliżający wartości parametrów  $\mathbf{w}_{GL}^*$  sieci złożonej DRNN. Parametry  $\mathbf{w}_{GL}^*$  minimalizują, zmodyfikowany o funkcję kary wewnętrznej, wskaźnik jakości (2.96) lub – równoważnie – globalny wskaźnik jakości, przypadek szczególny, (2.93) z ograniczeniami (2.94) na jakości modeli lokalnych.

## Rozdział 3

# Analiza zbieżności algorytmów uczenia sieci neuronowych w systemie złożonym

W trakcie uczenia sieci neuronowej z użyciem numerycznego algorytmu optymalizacji w kolejnych iteracjach generowany jest ciąg wartości parametrów sieci neuronowej. Pojawia się pytanie: czy i w jakich warunkach ten ciąg jest zbieżny? Należy zaznaczyć, że – z uwagi na nieliniowość i wielomodalność optymalizowanych funkcji celu oraz użycie metody gradientu prostego – można tu mówić jedynie o zbieżności lokalnej. Możemy też zadać równoważne pytanie: czy opracowane algorytmy uczenia są odwzorowaniami zwężającymi?

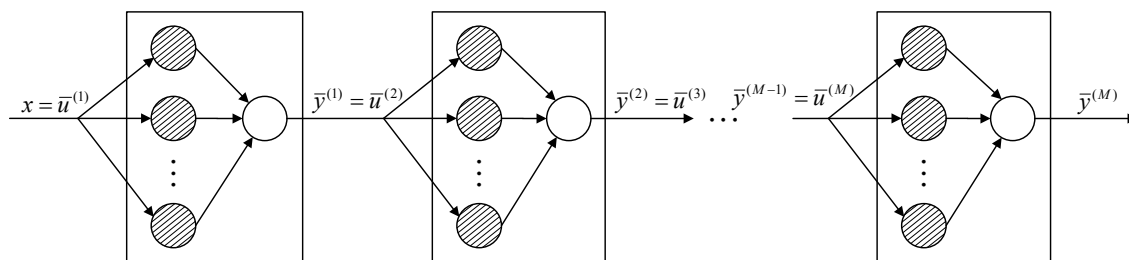
W literaturze znane są pewne podejścia do analizy zbieżności algorytmów uczenia pojedynczych statycznych i rekurencyjnych sieci neuronowych, [21], [58], [87], [49], [133]. Najczęściej stosowana jest druga (bezpośrednia) metoda Lapunowa. W pracy [82] zaproponowano dodanie parametrów wagowych do funkcji Lapunowa, których zadaniem jest zmniejszenie udziału starszych pomiarów. Oryginalne podejście do analizy zbieżności algorytmów uczenia sieci oraz stabilności samych sieci rekurencyjnych zaprezentowali Suykens, Vandewalle i De Moor w [116] i [117]. Zagadnienie uzyskiwania w procesie uczenia sieci rekurencyjnych o stabilnych charakterystykach podjęto w pracach [63] i [65]. Zastosowanie analizy Lapunowa do klasycznego modelu systemu o strukturze kaskadowej przedstawia artykuł [62]. Analiza zbieżności algorytmu uczenia RTRL sieci DRNN była przed-



miotem prac [69], [70], [77], [132].

Decydujące dla zbieżności opracowanych algorytmów uczenia są wartości współczynnika uczenia  $\eta$  będącego wielkością poprawki w metodzie gradientowej. Przez analizę zbieżności wartości algorytmu rozumiemy tu wyznaczenie warunku, określającego zbiór wartości współczynnika  $\eta$ , dla których ciąg wartości parametrów sieci neuronowej jest zbieżny. W przypadku uczenia na bieżąco interesuje nas zbiór wartości dla  $\eta(n)$  w bieżącym,  $n$ -tym taktie. Jednak aby nie komplikować oznaczeń, pomijając będziemy w zapisie  $\eta$  numer taktu.

W celu uproszczenia rozważań przyjmiemy, że system składa się z elementów typu SISO ( $L_m = S_m = 1$  dla  $m = 1, 2, \dots, M$ ). Sieć złożona modelująca ten system zilustrowana jest na rys. 3.1. Dodatkowo ustalimy dla każdej sieci tę samą wartość  $\eta$ . Analizę zbieżności przeprowadzimy korzystając z bezpośredniej metody Lapunowa, opisaną w [66].



Rysunek 3.1: Złożona sieć DRNN SISO

Kolejny podpunkt przedstawia znany rezultat analizy zbieżności dla pojedynczej sieci neuronowej. W podpunktach 3.2 – 3.5, zawierających oryginalne rezultaty pracy, przeprowadzono analizę zbieżności algorytmu uczenia złożonej sieci neuronowej przy różnych, wprowadzanych w poprzednich rozdziałach, wskaźnikach jakości.

### 3.1 Zbieżność algorytmu uczenia sieci lokalnej

Rozważmy algorytm (2.17) – (2.19) uczenia pojedynczej sieci ( $m$  ustalone), nie biorąc pod uwagę, że jest ona częścią sieci złożonej. Wybór funkcji Lapunowa dla tego przypadku jest oczywisty:

$$V_m(n) = \frac{1}{2} \left[ e^{(m)}(n) \right]^2. \quad (3.1)$$

Algorytm uczenia (każdy z przedstawionych pracy) jest zbieżny, jeżeli różnica pierwszego rzędu funkcji Lapunowa:

$$\Delta V_m(n) = V_m(n+1) - V_m(n), \quad (3.2)$$

czyli:

$$\Delta V_m(n) = \frac{1}{2} \left( [e^{(m)}(n+1)]^2 - [e^{(m)}(n)]^2 \right)$$

jest ujemna, [66]. Dążymy do tego, aby z nierówności

$$\Delta V_m(n) < 0 \quad (3.3)$$

otrzywać warunek na wartości  $\eta$ , zależny od wielkości dostępnych w  $n$ -tym takcie działania algorytmu uczenia. Celem dalszych przekształceń jest zastąpienie wielkości dotyczących taktu  $n+1$  przez wyrażenia zawierające wielkości tylko z taktu bieżącego, tj. o numerze  $n$ .

Oznaczmy:

$$\Delta e^{(m)}(n) = e^{(m)}(n+1) - e^{(m)}(n), \quad (3.4)$$

oraz rozwińmy funkcję błędu  $e^{(m)}(n+1)$  w szereg Taylora (do wyrazu pierwszego rzędu), [67]:

$$e^{(m)}(n+1) = e^{(m)}(n) + \Delta e^{(m)}(n) = e^{(m)}(n) + \left[ \nabla_{\mathbf{w}^{(m)}} e^{(m)}(n) \right]^T \Delta \mathbf{w}^{(m)},$$

gdzie  $\Delta \mathbf{w}^{(m)}$  oznacza wektor zmian parametrów wagowych sieci neuronowej  $\Phi_m$ :

$$\Delta \mathbf{w}^{(m)}(n) = \mathbf{w}^{(m)}(n+1) - \mathbf{w}^{(m)}(n), \quad (3.5)$$

który po uwzględnieniu (2.6) i (2.7) przy  $L_{m-1} = S_m = 1$  można zapisać w postaci:

$$\Delta \mathbf{w}^{(m)}(n) = \eta \cdot e^{(m)}(n) \left[ \nabla_{\mathbf{w}^{(m)}} \bar{y}^{(m)}(n) \right]. \quad (3.6)$$

Ponieważ  $\nabla_{\mathbf{w}^{(m)}} e^{(m)}(n) = -\nabla_{\mathbf{w}^{(m)}} \bar{y}^{(m)}(n)$ , mamy:

$$\Delta e^{(m)}(n) = \left[ \nabla_{\mathbf{w}^{(m)}} e^{(m)}(n) \right]^T \Delta \mathbf{w}^{(m)}(n) = - \left[ \nabla_{\mathbf{w}^{(m)}} \bar{y}^{(m)}(n) \right]^T \Delta \mathbf{w}^{(m)}(n). \quad (3.7)$$

Wstawiając (3.6) do (3.7), otrzymujemy wyrażenie na przyrost błędu:

$$\Delta e^{(m)}(n) = -\eta \cdot \zeta_m(n), \quad (3.8)$$

gdzie:

$$\zeta_m(n) = e^{(m)}(n) \left\| \nabla_{\mathbf{w}^{(m)}} \bar{y}^{(m)}(n) \right\|^2, \quad (3.9)$$

a składowe wektora gradientu  $\nabla_{\mathbf{w}^{(m)}}\bar{y}^{(m)}(n)$  określone są wzorami (2.8), (2.9) i (2.11).

Można teraz wyrazić różnicę pierwszego rzędu  $\Delta V_m(n)$  funkcji Lapunowa poprzez wielkości dostępne w takcie  $n$ -tym:

$$\begin{aligned}\Delta V_m(n) &= \frac{1}{2} \left[ e^{(m)}(n) + \Delta e^{(m)}(n) \right]^2 - \frac{1}{2} \left[ e^{(m)}(n) \right]^2 = \\ &= \frac{1}{2} \left[ e^{(m)}(n) \right]^2 + e^{(m)}(n)\Delta e^{(m)}(n) + \frac{1}{2} \left[ \Delta e^{(m)}(n) \right]^2 - \frac{1}{2} \left[ e^{(m)}(n) \right]^2.\end{aligned}$$

Otrzymujemy ogólne wyrażenie:

$$\Delta V_m(n) = \Delta e^{(m)}(n) \left( e^{(m)}(n) + \frac{1}{2} \Delta e^{(m)}(n) \right), \quad (3.10)$$

do którego wstawiamy (3.8):

$$\Delta V_m(n) = -\eta \zeta_m(n) \left( e^{(m)}(n) - \frac{1}{2} \eta \zeta_m(n) \right).$$

Warunek zbieżności (3.3) jest spełniony, jeżeli:

$$\zeta_m(n) \left( e^{(m)}(n) - \frac{1}{2} \eta \zeta_m(n) \right) > 0,$$

$$\zeta_m(n) e^{(m)}(n) > \frac{1}{2} \eta [\zeta_m(n)]^2,$$

$$\eta < \frac{2\zeta_m(n)e^{(m)}(n)}{[\zeta_m(n)]^2},$$

$$\eta < \frac{2e^{(m)}(n)}{\zeta_m(n)}.$$

Wstawienie (3.9) za  $\zeta_m(n)$  do ostatniej nierówności, daje w rezultacie warunek zbieżności algorytmu uczenia sieci lokalnej:

$$0 < \eta < \frac{2}{\left\| \nabla_{\mathbf{w}^{(m)}}\bar{y}^{(m)}(n) \right\|^2}. \quad (3.11)$$

Warto zwrócić uwagę, że jest to warunek bardzo ogólny, tzn. jego postać nie zależy od modelu, a jedynie od algorytmu. Wyrażenie  $\nabla_{\mathbf{w}^{(m)}}\bar{y}^{(m)}(n)$  może dotyczyć w zasadzie modelu o dowolnej postaci, w szczególności sieci DRNN. Ponieważ wartość wyrażenia  $\nabla_{\mathbf{w}^{(m)}}\bar{y}^{(m)}(n)$  musiała zostać wyznaczona w  $n$ -tym takcie na potrzeby samego algorytmu uczenia, koszt dodatkowych obliczeń związanych z otrzymaniem warunku (3.11) dla modelu lokalnego jest pomijalnie mały.

### 3.2 Zbieżność algorytmu uczenia sieci złożonej przy globalnym wskaźniku jakości

W niniejszym podpunkcie udowodnimy następujące twierdzenie:

**Twierdzenie 1.** *Algorytm uczenia sieci złożonej prowadzący do modelu globalnie optymalnego jest zbieżny wówczas, gdy spełniona jest nierówność:*

$$\sum_{\gamma=1}^m \sum_{r=\gamma}^M \beta_r \varepsilon^{(m)}(n) \varepsilon^{(r)}(n) \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(m)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(r)}(n) \right] > 0. \quad (3.12)$$

**Dowód** Stosując schemat postępowania przedstawiony w poprzednim podpunkcie, rozpoczynamy od wyboru funkcji Lapunowa o postaci:

$$V_m(n) = \frac{1}{2} \left[ \varepsilon^{(m)}(n) \right]^2, \quad (3.13)$$

a jej różnicę pierwszego rzędu wyznaczamy podobnie jak poprzednio:

$$\Delta V_m(n) = \frac{1}{2} \left( \left[ \varepsilon^{(m)}(n+1) \right]^2 - \left[ \varepsilon^{(m)}(n) \right]^2 \right).$$

Oznaczmy przyrost błędu  $m$ -tej sieci lokalnej przy uwzględnieniu, że wchodzi ona w skład sieci złożonej:

$$\Delta \varepsilon^{(m)}(n) = \varepsilon^{(m)}(n+1) - \varepsilon^{(m)}(n), \quad (3.14)$$

oraz rozwińmy funkcję błędu  $\varepsilon^{(m)}(n+1)$  w szereg Taylora:

$$\varepsilon^{(m)}(n+1) = \varepsilon^{(m)}(n) + \Delta \varepsilon^{(m)}(n) = \varepsilon^{(m)}(n) + \left[ \nabla_{\mathbf{w}} \varepsilon^{(m)}(n) \right]^T \Delta \mathbf{w}(n),$$

gdzie  $\Delta \mathbf{w}(n)$  jest wektorem zmian parametrów wagowych sieci złożonej, czyli wszystkich sieci  $\Phi_m$  ( $m = 1, 2, \dots, M$ ) typu SISO wchodzących w jej skład. Jego postać wynika z formuły opisującej algorytm uczenia (2.40) oraz z (2.41):

$$\Delta \mathbf{w}(n) = \mathbf{w}(n+1) - \mathbf{w}(n) = \begin{bmatrix} \Delta \mathbf{w}^{(1)}(n) \\ \Delta \mathbf{w}^{(2)}(n) \\ \vdots \\ \Delta \mathbf{w}^{(M)}(n) \end{bmatrix} = \eta \sum_{r=1}^M \beta_r \varepsilon^{(r)}(n) \begin{bmatrix} \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(r)}(n) \\ \nabla_{\mathbf{w}^{(2)}} \bar{v}^{(r)}(n) \\ \vdots \\ \nabla_{\mathbf{w}^{(M)}} \bar{v}^{(r)}(n) \end{bmatrix},$$

a po uwzględnieniu, że dla  $m > r$  zachodzi  $\nabla_{\mathbf{w}^{(m)}} \bar{v}^{(r)}(n) = 0$ , mamy:

$$\Delta \mathbf{w}(n) = \eta \begin{bmatrix} \sum_{r=1}^M \beta_r \varepsilon^{(r)}(n) \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(r)}(n) \right] \\ \sum_{r=2}^M \beta_r \varepsilon^{(r)}(n) \left[ \nabla_{\mathbf{w}^{(2)}} \bar{v}^{(r)}(n) \right] \\ \vdots \\ \sum_{r=M}^M \beta_r \varepsilon^{(r)}(n) \left[ \nabla_{\mathbf{w}^{(M)}} \bar{v}^{(r)}(n) \right] \end{bmatrix}. \quad (3.15)$$

Przyrost błędu popełnianego przez  $m$ -tą sieć jest następujący:

$$\Delta \varepsilon^{(m)}(n) = \left[ \nabla_{\mathbf{w}} \varepsilon^{(m)}(n) \right]^T \Delta \mathbf{w}(n) = - \left[ \nabla_{\mathbf{w}} \bar{v}^{(m)}(n) \right]^T \Delta \mathbf{w}(n),$$

co można rozpisać na iloczyn skalarny dwóch wektorów blokowych:

$$\Delta \varepsilon^{(m)}(n) = - \left[ \begin{array}{cccc} \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(m)}(n) \right]^T & \dots & \left[ \nabla_{\mathbf{w}^{(m)}} \bar{v}^{(m)}(n) \right]^T & 0 \dots 0 \end{array} \right] \begin{bmatrix} \Delta \mathbf{w}^{(1)}(n) \\ \vdots \\ \Delta \mathbf{w}^{(m)}(n) \\ \vdots \\ \Delta \mathbf{w}^{(M)}(n) \end{bmatrix}.$$

Część składników pierwszego wektora blokowego to wektory zerowe, stąd powyższy iloczyn skalarny sprowadza się do sumy iloczynów  $m$  początkowych wektorów:

$$\Delta \varepsilon^{(m)}(n) = - \sum_{\gamma=1}^m \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(m)}(n) \right]^T \Delta \mathbf{w}^{(\gamma)}(n), \quad (3.16)$$

gdzie  $\Delta \mathbf{w}^{(\gamma)}(n)$  jest elementem składowym wektora  $\Delta \mathbf{w}(n)$  o numerze  $\gamma$ :

$$\Delta \mathbf{w}^{(\gamma)}(n) = \eta \sum_{r=\gamma}^M \beta_r \varepsilon^{(r)}(n) \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(r)}(n) \right]. \quad (3.17)$$

Wstawiając (3.17) do (3.16), otrzymujemy wyrażenie na przyrost błędu:

$$\Delta \varepsilon^{(m)}(n) = -\eta \cdot \zeta_m(n), \quad (3.18)$$

gdzie:

$$\zeta_m(n) = \sum_{\gamma=1}^m \sum_{r=\gamma}^M \beta_r \varepsilon^{(r)}(n) \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(m)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(r)}(n) \right]. \quad (3.19)$$

Różnica pierwszego rzędu  $\Delta V_m(n)$  funkcji Lapunowa wyrażona przez wielkości dostępne w takcie  $n$ -tym przyjmuje postać:

$$\begin{aligned}\Delta V_m(n) &= \frac{1}{2} \left[ \varepsilon^{(m)}(n) + \Delta \varepsilon^{(m)}(n) \right]^2 - \frac{1}{2} \left[ \varepsilon^{(m)}(n) \right]^2 = \\ &= \frac{1}{2} \left[ \varepsilon^{(m)}(n) \right]^2 + \varepsilon^{(m)}(n) \Delta \varepsilon^{(m)}(n) + \frac{1}{2} \left[ \Delta \varepsilon^{(m)}(n) \right]^2 - \frac{1}{2} \left[ \varepsilon^{(m)}(n) \right]^2.\end{aligned}$$

Przekształcając powyższe, otrzymujemy ogólne wyrażenie:

$$\Delta V_m(n) = \Delta \varepsilon^{(m)}(n) \left( \varepsilon^{(m)}(n) + \frac{1}{2} \Delta \varepsilon^{(m)}(n) \right), \quad (3.20)$$

a po wstawieniu do niego (3.18):

$$\Delta V_m(n) = -\eta \zeta_m(n) \left( \varepsilon^{(m)}(n) - \frac{1}{2} \eta \zeta_m(n) \right).$$

Spełnienie warunku zbieżności (3.3) jest równoważne ze spełnieniem nierówności:

$$\zeta_m(n) \left( \varepsilon^{(m)}(n) - \frac{1}{2} \eta \zeta_m(n) \right) > 0,$$

$$\zeta_m(n) \varepsilon^{(m)}(n) > \frac{1}{2} \eta [\zeta_m(n)]^2,$$

$$\eta < \frac{2\zeta_m(n) \varepsilon^{(m)}(n)}{[\zeta_m(n)]^2},$$

$$\eta < \frac{2\varepsilon^{(m)}(n)}{\zeta_m(n)}.$$

Wstawienie (3.19) za  $\zeta_m(n)$  do ostatniej nierówności, daje w rezultacie warunek zbieżności algorytmu uczenia sieci globalnej przy globalnym wskaźniku jakości identyfikacji:

$$0 < \eta < \frac{2\varepsilon^{(m)}(n)}{\sum_{\gamma=1}^m \sum_{r=\gamma}^M \beta_r \varepsilon^{(r)}(n) \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(m)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(r)}(n) \right]}. \quad (3.21)$$

Zbadamy teraz, w jakich sytuacjach prawa strona przedziału (3.21) jest dodatnia, co oznacza, że przedział zbieżności w ogóle istnieje. W rozważanym przypadku stwierdzić można, że algorytm jest zbieżny (tzn. daje się określić przedział zbieżności dla współczynnika  $\eta$ ), jeżeli iloczyn licznika i mianownika w (3.21) jest liczbą dodatnią:

$$\sum_{\gamma=1}^m \sum_{r=\gamma}^M \beta_r \varepsilon^{(m)}(n) \varepsilon^{(r)}(n) \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(m)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(r)}(n) \right] > 0, \quad (3.22)$$

co należało pokazać. □

Podkreślmy, że – podobnie jak w poprzednim podpunkcie – warunek (3.21) ma charakter ogólny. Jest on słuszny dla dowolnego modelu dyskretnego. Wyznaczenie dokładnego przedziału dla wartości współczynnika  $\eta$  w zadaniu identyfikacji globalnej pociąga za sobą jednak pewien koszt obliczeniowy. Należałoby obliczyć wartości wszystkich  $\frac{M(M+1)}{2}$  wyrażeń typu  $[\nabla_{\mathbf{w}(\gamma)} \bar{v}^{(m)}(n)]^T [\nabla_{\mathbf{w}(\gamma)} \bar{v}^{(r)}(n)]$  w nierówności (3.21), a zatem złożoność tej procedury jest kwadratowa. Istnieją dużo szybsze, heurystyczne, metody doboru wartości współczynnika  $\eta$  zapewniającej zbieżność algorytmu uczenia, [97].

W celu łatwiejszej interpretacji twierdzenia 1, zilustrujemy jego działanie na prostym przykładzie modelu globalnego składającego się z szeregowego połączenia dwóch modeli SISO ( $M = 2$ ).

**Przykład dla  $M=2$ .** Dla pierwszego modelu (pierwszej sieci), tj dla  $m = 1$ , warunek zbieżności (3.22) przyjmuje postać:

$$\begin{aligned} & \sum_{r=1}^2 \beta_r \varepsilon^{(1)}(n) \varepsilon^{(r)}(n) \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(1)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(r)}(n) \right] = \\ & = \beta_1 \left[ \varepsilon^{(1)}(n) \right]^2 \left\| \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(1)}(n) \right\|^2 + \beta_2 \varepsilon^{(1)}(n) \varepsilon^{(2)}(n) \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(1)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(2)}(n) \right] > 0, \end{aligned}$$

natomiast dla modelu drugiego ( $m = 2$ ):

$$\begin{aligned} & \sum_{\gamma=1}^2 \sum_{r=\gamma}^2 \beta_r \varepsilon^{(2)}(n) \varepsilon^{(r)}(n) \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(2)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(r)}(n) \right] = \\ & = \beta_1 \varepsilon^{(1)}(n) \varepsilon^{(2)}(n) \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(1)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(2)}(n) \right] + \\ & + \beta_2 \left[ \varepsilon^{(2)}(n) \right]^2 \left\| \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(2)}(n) \right\|^2 + \beta_2 \left[ \varepsilon^{(2)}(n) \right]^2 \left\| \nabla_{\mathbf{w}^{(2)}} \bar{v}^{(2)}(n) \right\|^2 = \\ & = \beta_2 \left[ \varepsilon^{(2)}(n) \right]^2 \left( \left\| \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(2)}(n) \right\|^2 + \left\| \nabla_{\mathbf{w}^{(2)}} \bar{v}^{(2)}(n) \right\|^2 \right) + \\ & + \beta_1 \varepsilon^{(1)}(n) \varepsilon^{(2)}(n) \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(1)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(2)}(n) \right] > 0. \end{aligned}$$

Górnym ograniczeniem wartości współczynnika  $\eta$  jest mniejsza z liczb uzyskanych dla obu modeli:

$$\min \left\{ \beta_1 \left[ \varepsilon^{(1)}(n) \right]^2 \left\| \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(1)}(n) \right\|^2 + \beta_2 \varepsilon^{(1)}(n) \varepsilon^{(2)}(n) \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(1)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(2)}(n) \right], \right. \\ \left. \beta_2 \left[ \varepsilon^{(2)}(n) \right]^2 \left( \left\| \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(2)}(n) \right\|^2 + \left\| \nabla_{\mathbf{w}^{(2)}} \bar{v}^{(2)}(n) \right\|^2 \right) + \right. \\ \left. + \beta_1 \varepsilon^{(1)}(n) \varepsilon^{(2)}(n) \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(1)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(2)}(n) \right] \right\},$$

co oznacza, że algorytm uczenia jest zbieżny, jeżeli żadna z tych dwóch liczb nie będzie ujemna. Ponieważ każda z nich jest sumą dwóch składników, z których pierwszy jest zawsze dodatni, lecz drugi może przyjmować wartości ujemne, szerokość przedziału zbieżności zależy od wartości składników

$$\beta_m \varepsilon^{(1)}(n) \varepsilon^{(2)}(n) \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(1)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(2)}(n) \right], \quad (m = 1, 2).$$

Najkorzystniejsza jest największa szerokość przedziału zbieżności. Sytuacja taka występuje w dwóch skrajnych przypadkach. Pierwszy: iloczyn skalarny

$\left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(1)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(2)}(n) \right]$  ma znak dodatni i jednocześnie błędy  $\varepsilon^{(m)}(n)$ ,  $(m = 1, 2)$  są tego samego znaku. Drugi przypadek, gdy wspomniany iloczyn skalarny ma znak ujemny i jednocześnie błędy  $\varepsilon^{(1)}(n)$ ,  $\varepsilon^{(2)}(n)$  są przeciwnego znaku. Obie sytuacje dotyczą zgodności znaków gradientów wyjść  $\bar{v}^{(1)}(n)$ ,  $\bar{v}^{(2)}(n)$  po parametrach  $\mathbf{w}^{(1)}$  pierwszego modelu ze znakami błędów tych modeli. Streszczając te rozważania powiemy, że zmiana parametrów pierwszego modelu powoduje, iż zmiany błędów  $\varepsilon^{(m)}(n)$  obu modeli zachodzą w kierunkach zgodnych z ich *wrażliwościami*, rozumianymi tu jako pochodne cząstkowe z wyjść po parametrach. Innymi słowy: zmiana parametru pierwszego modelu powinna wpływać w ten sam sposób na zmianę błędów obu modeli. Każde odstępstwo od tej reguły zmniejsza wielkość przedziału zbieżności dla współczynnika  $\eta$ , aż do braku zbieżności, która to sytuacja zachodzi wówczas, gdy spełniona jest co najmniej jedna z nierówności:

$$\beta_1 \left[ \varepsilon^{(1)}(n) \right]^2 \left\| \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(1)}(n) \right\|^2 < -\beta_2 \varepsilon^{(1)}(n) \varepsilon^{(2)}(n) \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(1)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(2)}(n) \right],$$

$$\beta_2 \left[ \varepsilon^{(2)}(n) \right]^2 \left( \left\| \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(2)}(n) \right\|^2 + \left\| \nabla_{\mathbf{w}^{(2)}} \bar{v}^{(2)}(n) \right\|^2 \right) <$$

$$< -\beta_1 \varepsilon^{(1)}(n) \varepsilon^{(2)}(n) \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(1)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(2)}(n) \right].$$



Zilustrowaną tu, na przykładzie systemu dwuelementowego, zasadę zgodności kierunków zmian błędów ze znakami wrażliwości, wywieść można ze wzoru (3.22) dla dowolnego  $M$ .

### 3.3 Zbieżność przy syntetycznym wskaźniku jakości

W skład funkcji Lapunowa  $m$ -tej sieci w algorytmie uczenia sieci globalnej, minimalizującym syntetyczny wskaźnik jakości, wchodzi zarówno błąd  $\varepsilon^{(M)}(n)$  modelu globalnego jak i błąd  $e^{(m)}(n)$  modelu lokalnego:

$$V_m(n) = \frac{1}{2}\alpha_0 \left[ \varepsilon^{(M)}(n) \right]^2 + \frac{1}{2}\alpha_m \left[ e^{(m)}(n) \right]^2. \quad (3.23)$$

Różnica pierwszego rzędu ma postać:

$$\begin{aligned} \Delta V_m(n) = & \alpha_0 \left( \frac{1}{2} \left[ \varepsilon^{(M)}(n+1) \right]^2 - \frac{1}{2} \left[ \varepsilon^{(M)}(n) \right]^2 \right) + \\ & + \alpha_m \left( \frac{1}{2} \left[ e^{(m)}(n+1) \right]^2 - \frac{1}{2} \left[ e^{(m)}(n) \right]^2 \right), \end{aligned}$$

a po wstawieniu (3.10) i (3.20) mamy:

$$\begin{aligned} \Delta V_m(n) = & \alpha_0 \Delta \varepsilon^{(M)}(n) \left( \varepsilon^{(M)}(n) + \frac{1}{2} \Delta \varepsilon^{(M)}(n) \right) + \\ & + \alpha_m \Delta e^{(m)}(n) \left( e^{(m)}(n) + \frac{1}{2} \Delta e^{(m)}(n) \right). \end{aligned} \quad (3.24)$$

Przyrost błędu modelu globalnego ma postać jak w (3.16):

$$\Delta \varepsilon^{(M)}(n) = - \sum_{\gamma=1}^M \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(M)}(n) \right]^T \Delta \mathbf{w}^{(\gamma)}(n), \quad (3.25)$$

przy czym przyrost wartości wektora parametrów sieci o numerze  $\gamma$  jest równy:

$$\Delta \mathbf{w}^{(\gamma)}(n) = \eta \left( \alpha_0 \varepsilon^{(M)}(n) \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(M)}(n) \right] + \alpha_\gamma e^{(\gamma)}(n) \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{y}^{(\gamma)}(n) \right] \right). \quad (3.26)$$

Wstawiając (3.26) do (3.25), otrzymujemy wyrażenie na przyrost błędu:

$$\Delta \varepsilon^{(M)}(n) = -\eta \zeta_m(n), \quad (3.27)$$

gdzie

$$\begin{aligned} \zeta_m(n) = & \alpha_0 \varepsilon^{(M)}(n) \sum_{\gamma=1}^M \left\| \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(M)}(n) \right\|^2 + \\ & + \sum_{\gamma=1}^M \alpha_\gamma e^{(\gamma)}(n) \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(M)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{y}^{(\gamma)}(n) \right]. \end{aligned} \quad (3.28)$$

Podobnie, przyrost błędu  $m$ -tego modelu lokalnego ma postać jak w (3.7):

$$\Delta e^{(m)}(n) = - \left[ \nabla_{\mathbf{w}^{(m)}} \bar{y}^{(m)}(n) \right]^T \Delta \mathbf{w}^{(m)}(n),$$

skąd, po wstawieniu za  $\Delta \mathbf{w}^{(m)}(n)$  wzoru (3.26), otrzymujemy:

$$\Delta e^{(m)}(n) = -\eta \xi_m(n), \quad (3.29)$$

gdzie:

$$\begin{aligned} \xi_m(n) = & \alpha_m e^{(m)}(n) \left\| \nabla_{\mathbf{w}^{(m)}} \bar{y}^{(m)}(n) \right\|^2 + \\ & + \alpha_0 \varepsilon^{(M)}(n) \left[ \nabla_{\mathbf{w}^{(m)}} \bar{v}^{(M)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(m)}} \bar{y}^{(m)}(n) \right]. \end{aligned} \quad (3.30)$$

Wstawiając (3.27) i (3.29) do (3.24) możemy wyrazić różnicę pierwszego rzędu  $\Delta V_m(n)$  funkcji Lapunowa przez wielkości dostępne w  $n$ -tym takcie:

$$\Delta V_m(n) = -\eta \alpha_0 \zeta_m(n) \left( \varepsilon^{(M)}(n) - \frac{1}{2} \eta \zeta_m(n) \right) - \eta \alpha_m \xi_m(n) \left( e^{(m)}(n) - \frac{1}{2} \eta \xi_m(n) \right).$$

Warunek zbieżności

$$\Delta V_m(n) < 0$$

jest spełniony, gdy zachodzi nierówność:

$$\alpha_0 \zeta_m(n) \varepsilon^{(M)}(n) - \frac{1}{2} \eta \alpha_0 [\zeta_m(n)]^2 + \alpha_m \xi_m(n) e^{(m)}(n) - \frac{1}{2} \eta \alpha_m [\xi_m(n)]^2 > 0,$$

którą przekształcamy do prostszej postaci:

$$\alpha_0 \varepsilon^{(M)}(n) \zeta_m(n) + \alpha_m e^{(m)}(n) \xi_m(n) > \frac{1}{2} \eta \left( \alpha_0 [\zeta_m(n)]^2 + \alpha_m [\xi_m(n)]^2 \right).$$

W rezultacie otrzymujemy warunek zbieżności algorytmu uczenia sieci złożonej przy syntetycznym wskaźniku jakości identyfikacji:

$$0 < \eta < \frac{2 \left( \alpha_0 \varepsilon^{(M)}(n) \zeta_m(n) + \alpha_m e^{(m)}(n) \xi_m(n) \right)}{\alpha_0 [\zeta_m(n)]^2 + \alpha_m [\xi_m(n)]^2}, \quad (3.31)$$

natomiast przedział zbieżności jest określony dla dodatnich wartości ułamka z powyższej nierówności. Ponieważ mianownik jest zawsze nieujemny, wystarczy, że licznik będzie mieć znak dodatni:

$$\alpha_0 \varepsilon^{(M)}(n) \zeta_m(n) + \alpha_m e^{(m)}(n) \xi_m(n) > 0.$$

Po wstawieniu (3.28) i (3.30), otrzymujemy warunek istnienia przedziału zbieżności dla współczynnika  $\eta$ :

$$\begin{aligned}
 & \alpha_0^2 [\varepsilon^{(M)}(n)]^2 \sum_{\gamma=1}^M \left\| \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(M)}(n) \right\|^2 + \alpha_m^2 [e^{(m)}(n)]^2 \left\| \nabla_{\mathbf{w}^{(m)}} \bar{y}^{(m)}(n) \right\|^2 + \\
 & + \alpha_0 \sum_{\gamma=1}^M \alpha_\gamma \varepsilon^{(M)}(n) e^{(\gamma)}(n) \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(M)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{y}^{(\gamma)}(n) \right] + \\
 & + \alpha_m \alpha_0 \varepsilon^{(M)}(n) e^{(m)}(n) \left[ \nabla_{\mathbf{w}^{(m)}} \bar{v}^{(M)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(m)}} \bar{y}^{(m)}(n) \right] > 0,
 \end{aligned} \tag{3.32}$$

który ponownie zilustrujemy na prostym przykładzie dwuelementowego systemu złożonego SISO.

### Przykład dla M=2

Dla pierwszego modelu (pierwszej sieci), tj dla  $m = 1$ , warunek (3.32) przyjmuje postać:

$$\begin{aligned}
 & \alpha_0^2 [\varepsilon^{(2)}(n)]^2 \left( \left\| \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(2)}(n) \right\|^2 + \left\| \nabla_{\mathbf{w}^{(2)}} \bar{v}^{(2)}(n) \right\|^2 \right) + \alpha_1^2 [e^{(1)}(n)]^2 \left\| \nabla_{\mathbf{w}^{(1)}} \bar{y}^{(1)}(n) \right\|^2 + \\
 & + 2\alpha_0 \alpha_1 \varepsilon^{(2)}(n) e^{(1)}(n) \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(2)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(1)}} \bar{y}^{(1)}(n) \right] + \\
 & + \alpha_0 \alpha_2 \varepsilon^{(2)}(n) e^{(2)}(n) \left[ \nabla_{\mathbf{w}^{(2)}} \bar{v}^{(2)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(2)}} \bar{y}^{(2)}(n) \right] > 0,
 \end{aligned}$$

a dla drugiego modelu ( $m = 2$ ):

$$\begin{aligned}
 & \alpha_0^2 [\varepsilon^{(2)}(n)]^2 \left( \left\| \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(2)}(n) \right\|^2 + \left\| \nabla_{\mathbf{w}^{(2)}} \bar{v}^{(2)}(n) \right\|^2 \right) + \alpha_2^2 [e^{(2)}(n)]^2 \left\| \nabla_{\mathbf{w}^{(1)}} \bar{y}^{(2)}(n) \right\|^2 + \\
 & + \alpha_0 \alpha_1 \varepsilon^{(2)}(n) e^{(1)}(n) \left[ \nabla_{\mathbf{w}^{(1)}} \bar{v}^{(2)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(1)}} \bar{y}^{(2)}(n) \right] + \\
 & + 2\alpha_0 \alpha_2 \varepsilon^{(2)}(n) e^{(2)}(n) \left[ \nabla_{\mathbf{w}^{(2)}} \bar{v}^{(2)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(2)}} \bar{y}^{(2)}(n) \right] > 0
 \end{aligned}$$

Znaczenie tych nierówności jest podobne jak w poprzednim przykładzie. Tym razem przedział zbieżności jest największy, gdy występuje zgodność między znakami gradientów wyjścia  $\bar{v}^{(M)}(n)$  sieci złożonej traktowanej jak model globalny i wyjść  $\bar{y}^{(m)}(n)$  ( $m = 1, 2$ ) sieci lokalnych a znakami błędów na wyjściach tych sieci. Podkreślimy, że zgodność w szczególności dotyczy również wyjścia globalnego  $\bar{v}^{(M)}(n)$  oraz wyjścia lokalnego  $\bar{v}^{(M)}(n)$  tego samego elementu.

### 3.4 Zbieżność przy zewnętrznej funkcji kary

Dla  $m$ -tej sieci w algorytmie uczenia sieci złożonej z kryterium globalnym zmodyfikowanym o funkcję kary zewnętrznej wybieramy następującą funkcję Lapunowa:

$$V_m(n) = \frac{1}{2}\alpha_0 \left[ \varepsilon^{(M)}(n) \right]^2 + c \max\{0, g_m(n)\}. \quad (3.33)$$

Różnica pierwszego rzędu tej funkcji ma postać:

$$\Delta V_m(n) = \frac{1}{2} \left[ \varepsilon^{(M)}(n+1) \right]^2 - \frac{1}{2} \left[ \varepsilon^{(M)}(n) \right]^2 + c \max\{0, g_m(n+1)\} - c \max\{0, g_m(n)\},$$

a po wstawieniu (3.20) mamy:

$$\begin{aligned} \Delta V_m(n) = \Delta \varepsilon^{(M)}(n) \left( \varepsilon^{(M)}(n) + \frac{1}{2} \Delta \varepsilon^{(M)}(n) \right) + \\ + c (\max\{0, g_m(n+1)\} - \max\{0, g_m(n)\}), \end{aligned} \quad (3.34)$$

gdzie  $\Delta \varepsilon^{(M)}(n)$  dane jest wzorem (3.25), w którym czynnik  $\Delta \mathbf{w}^{(\gamma)}(n)$  uzyskujemy przez złożenie zależności (2.100), (2.101), (2.102) i (2.103), otrzymując:

$$\begin{aligned} \Delta \mathbf{w}^{(\gamma)}(n) = \eta \left( \varepsilon^{(M)}(n) \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(M)}(n) \right] + \right. \\ \left. + c \begin{cases} e^{(\gamma)}(n) \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{y}^{(\gamma)}(n) \right] & \text{dla } g_\gamma(n) > 0 \\ 0 & \text{dla } g_\gamma(n) \leq 0 \end{cases} \right). \end{aligned} \quad (3.35)$$

Przyrost błędu modelu globalnego jest równy:

$$\Delta \varepsilon^{(M)}(n) = -\eta \zeta_m(n), \quad (3.36)$$

gdzie:

$$\begin{aligned} \zeta_m(n) = \varepsilon^{(M)}(n) \sum_{\gamma=1}^M \left\| \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(M)}(n) \right\|^2 + \\ + c \sum_{\gamma=1}^M \begin{cases} e^{(\gamma)}(n) \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(M)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{y}^{(\gamma)}(n) \right] & \text{dla } g_\gamma(n) > 0 \\ 0 & \text{dla } g_\gamma(n) \leq 0 \end{cases}. \end{aligned} \quad (3.37)$$

W (3.34) występuje nieciągły składnik, który może przyjmować cztery wartości:

$$\begin{aligned} \max\{0, g_m(n+1)\} - \max\{0, g_m(n)\} = \\ = \begin{cases} 0 & \text{dla } g_m(n+1) \leq 0 \text{ i } g_m(n) \leq 0 \\ -g_m(n) & \text{dla } g_m(n+1) \leq 0 \text{ i } g_m(n) > 0 \\ g_m(n+1) & \text{dla } g_m(n+1) > 0 \text{ i } g_m(n) \leq 0 \\ g_m(n+1) - g_m(n) & \text{dla } g_m(n+1) > 0 \text{ i } g_m(n) > 0 \end{cases} \end{aligned}$$

Wystąpienie pierwszej sytuacji oznacza, że ograniczenie nałożone na jakość  $m$ -tej sieci było spełnione zarówno przed aktualizacją wag jak i po niej, w drugiej sytuacji ta sieć wejdzie w obszar rozwiązań dopuszczalnych po aktualizacji wag, trzecia sytuacja oznacza przekroczenie przez nią obszaru rozwiązań dopuszczalnych, a w czwartej sieć pozostaje zarówno przed jak i po aktualizacji w obszarze rozwiązań niedopuszczalnych.

W każdej z tych sytuacji otrzymamy nieco różniące się warunki zbieżności. Dalej rozparzymy je kolejno.

$$1) g_m(n+1) \leq 0 \text{ i } g_m(n) \leq 0$$

Z uwagi na to, że parametry  $m$ -tej sieci pozostają w obszarze rozwiązań dopuszczalnych, różnica pierwszego rzędu funkcji Lapunowa ma ogólną postać identyczną z (3.20) dla  $m = M$  po wstawieniu (3.36):

$$\Delta V_m(n) = -\eta \zeta_m(n) \left( \varepsilon^{(M)}(n) - \frac{1}{2} \eta \zeta_m(n) \right)$$

Po odpowiednich przekształceniach otrzymujemy warunek zbieżności algorytmu uczenia sieci złożonej przy wskaźniku jakości identyfikacji zmodyfikowanym o funkcję kary zewnętrznej:

$$0 < \eta < \frac{2\varepsilon^{(M)}(n)}{\zeta_m(n)}. \quad (3.38)$$

skąd wynika warunek istnienia przedziału zbieżności:

$$\varepsilon^{(M)}(n) \zeta_m(n) > 0$$

Po wstawieniu (3.37) otrzymujemy warunek istnienia przedziału zbieżności dla współczynnika  $\eta$ :

$$\begin{aligned} & \left[ \varepsilon^{(M)}(n) \right]^2 \sum_{\gamma=1}^M \left\| \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(M)}(n) \right\|^2 + \\ & + c \sum_{\gamma=1}^M \begin{cases} \varepsilon^{(M)}(n) e^{(\gamma)}(n) \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(M)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{y}^{(\gamma)}(n) \right] & \text{dla } g_\gamma(n) > 0 \\ 0 & \text{dla } g_\gamma(n) \leq 0 \end{cases} > 0 \end{aligned}$$

$$2) g_m(n+1) \leq 0 \text{ i } g_m(n) > 0$$

Tutaj różnica pierwszego rzędu funkcji Lapunowa ma dodatkowy składnik:

$$\Delta V_m(n) = \Delta \varepsilon^{(M)}(n) \left( \varepsilon^{(M)}(n) + \frac{1}{2} \Delta \varepsilon^{(M)}(n) \right) - cg_m(n),$$

a po wstawieniu (3.36):

$$\Delta V_m(n) = -\eta \zeta_m(n) \left( \varepsilon^{(M)}(n) - \frac{1}{2} \eta \zeta_m(n) \right) - cg_m(n).$$

Nałożenie na powyższą formułę warunku zbieżności

$$\Delta V_m(n) < 0$$

prowadzi do nierówności:

$$-\eta \varepsilon^{(M)}(n) \zeta_m(n) + \frac{1}{2} \eta^2 [\zeta_m(n)]^2 - cg_m(n) < 0,$$

która po przekształceniach przyjmuje postać:

$$\frac{1}{2} [\zeta_m(n)]^2 \eta^2 - \varepsilon^{(M)}(n) \zeta_m(n) \eta - cg_m(n) < 0.$$

Rozwiązanie równania kwadratowego z lewej strony powyższej nierówności i wstawienie (3.37) daje przedział zbieżności dla współczynnika  $\eta$ :

$$\eta \in \left( \frac{\varepsilon^{(M)}(n) - \sqrt{[\varepsilon^{(M)}(n)]^2 + 2cg_m(n)}}{\zeta_m(n)}, \frac{\varepsilon^{(M)}(n) + \sqrt{[\varepsilon^{(M)}(n)]^2 + 2cg_m(n)}}{\zeta_m(n)} \right).$$

Warto zwrócić uwagę, że – w odróżnieniu od poprzednio wyprowadzanych warunków – lewa strona przedziału może być różna od zera. Licznik wyrażenia opisującego prawą stronę przedziału zbieżności jest zawsze nieujemny, więc warunkiem istnienia przedziału zbieżności jest, aby  $\zeta_m(n)$  dane wzorem (3.37) przyjmowało wartości nieujemne.

$$3) g_m(n+1) > 0 \text{ i } g_m(n) \leq 0$$

Różnica pierwszego rzędu funkcji Lapunowa ma w tym przypadku ponownie dodatkowy składnik:

$$\Delta V_m(n) = \Delta \varepsilon^{(M)}(n) \left( \varepsilon^{(M)}(n) + \frac{1}{2} \Delta \varepsilon^{(M)}(n) \right) + cg_m(n+1),$$

który – w celu uzyskania zależności funkcji  $\Delta V_m(n)$  od wielkości z taktu  $n$  – można rozpisać jako:

$$g_m(n+1) = \frac{1}{2} [e^{(m)}(n+1)]^2 - \lambda_m = \frac{1}{2} [e^{(m)}(n) + \Delta e^{(m)}(n)]^2 - \lambda_m.$$

Na podstawie (3.7) i (3.35), zapisać możemy wyrażenie na przyrost błędu  $m$ -tej sieci:

$$\Delta e^{(m)}(n) = - \left[ \nabla_{\mathbf{w}^{(m)}} \bar{y}^{(m)}(n) \right]^T \Delta \mathbf{w}^{(m)}(n) = -\eta \xi_m(n),$$

gdzie:

$$\xi_m(n) = \varepsilon^{(M)}(n) \left[ \nabla_{\mathbf{w}^{(m)}} \bar{v}^{(M)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(m)}} \bar{y}^{(m)}(n) \right]. \quad (3.39)$$

Funkcję ograniczenia  $g_m(n+1)$  wyrażamy więc jako:

$$\begin{aligned} g_m(n+1) &= \frac{1}{2} \left[ e^{(m)}(n) \right]^2 + e^{(m)}(n) \Delta e^{(m)}(n) + \left[ \Delta e^{(m)}(n) \right]^2 - \lambda_m = \\ &= g_m(n) + \Delta e^{(m)}(n) \left( e^{(m)}(n) + \frac{1}{2} \Delta e^{(m)}(n) \right), \end{aligned}$$

a po wstawieniu do  $\Delta V_m(n)$  dostajemy:

$$\begin{aligned} \Delta V_m(n) &= -\eta \zeta_m(n) \left( \varepsilon^{(M)}(n) - \frac{1}{2} \eta \zeta_m(n) \right) + c g_m(n) - c \eta \xi_m(n) \left( e^{(m)}(n) - \frac{1}{2} \eta \xi_m(n) \right) = \\ &= \eta \varepsilon^{(M)}(n) \zeta_m(n) + \frac{1}{2} \eta^2 [\zeta_m(n)]^2 + c g_m(n) - \eta c e^{(m)}(n) \xi_m(n) + \frac{1}{2} \eta^2 c [\xi_m(n)]^2 \end{aligned}$$

Warunkiem zbieżności jest ujemny znak funkcji  $\Delta V_m(n)$ , co prowadzi do nierówności:

$$\frac{1}{2} \left( [\zeta_m(n)]^2 + c [\xi_m(n)]^2 \right) \eta^2 - \left( \varepsilon^{(M)}(n) \zeta_m(n) + c e^{(m)}(n) \xi_m(n) \right) \eta + c g_m(n) < 0.$$

Rozwiązanie równania kwadratowego z lewej strony powyższej nierówności daje przedział zbieżności dla współczynnika  $\eta$ :

$$\eta \in \left( \frac{\left( \varepsilon^{(M)}(n) \zeta_m(n) + c e^{(m)}(n) \xi_m(n) \right) - \sqrt{\left( \varepsilon^{(M)}(n) \zeta_m(n) + c e^{(m)}(n) \xi_m(n) \right)^2 - 2c g_m(n) \left( [\zeta_m(n)]^2 + c [\xi_m(n)]^2 \right)}}{[\zeta_m(n)]^2 + c [\xi_m(n)]^2}, \right),$$

$$\left. \begin{aligned} & \left( \varepsilon^{(M)}(n)\zeta_m(n) + ce^{(m)}(n)\xi_m(n) \right) + \\ & + \sqrt{\left( \varepsilon^{(M)}(n)\zeta_m(n) + ce^{(m)}(n)\xi_m(n) \right)^2 - 2cg_m(n)([\zeta_m(n)]^2 + c[\xi_m(n)]^2)} \\ & \hline & [\zeta_m(n)]^2 + c[\xi_m(n)]^2 \end{aligned} \right\}$$

gdzie  $\zeta_m(n)$  i  $\xi_m(n)$  są określone odpowiednio wzorami (3.37) i (3.39). Wyrażenie opisujące prawą stronę przedziału zbieżności ma dodatnie zarówno mianownik jak i licznik, zatem przedział ten istnieje.

$$4) g_m(n+1) > 0 \text{ i } g_m(n) > 0$$

W sytuacji, w której parametry sieci po aktualizacji nadal pozostają w obszarze rozwiązań niedopuszczalnych, różnica pierwszego rzędu funkcji Lapunowa ma postać:

$$\Delta V_m(n) = \Delta \varepsilon^{(M)}(n) \left( \varepsilon^{(M)}(n) + \frac{1}{2} \Delta \varepsilon^{(M)}(n) \right) + c(g_m(n+1) - g_m(n)),$$

gdzie:

$$\begin{aligned} g_m(n+1) - g_m(n) &= c \left( \frac{1}{2} [e^{(m)}(n+1)]^2 - \lambda_m - \frac{1}{2} [e^{(m)}(n)]^2 + \lambda_m \right) = \\ &= c \left( \frac{1}{2} [e^{(m)}(n) + \Delta e^{(m)}(n)]^2 - \frac{1}{2} [e^{(m)}(n)]^2 \right) = \\ &= c \cdot \Delta e^{(m)}(n) \left( e^{(m)}(n) + \frac{1}{2} \Delta e^{(m)}(n) \right), \end{aligned}$$

przy czym  $\Delta e^{(m)}(n)$  jest określone wzorem

$$\Delta e^{(m)}(n) = -\eta \xi_m(n), \quad (3.40)$$



gdzie:

$$\begin{aligned} \xi_m(n) = & \eta \left( \varepsilon^{(M)}(n) \left[ \nabla_{\mathbf{w}^{(m)}} \bar{v}^{(M)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(m)}} \bar{y}^{(m)}(n) \right] + \right. \\ & \left. + c \begin{cases} e^{(m)}(n) \left\| \nabla_{\mathbf{w}^{(m)}} \bar{y}^{(m)}(n) \right\|^2 & \text{dla } g_m(n) > 0 \\ 0 & \text{dla } g_m(n) \leq 0 \end{cases} \right). \end{aligned} \quad (3.41)$$

Po wstawieniu do  $\Delta V_m(n)$  (3.40) i (3.36) dostajemy:

$$\Delta V_m(n) = -\eta \zeta_m(n) \left( \varepsilon^{(M)}(n) - \frac{1}{2} \eta \zeta_m(n) \right) - c \eta \xi_m(n) \left( e^{(m)}(n) - \frac{1}{2} \eta \xi_m(n) \right),$$

a po przekształceniach i nałożeniu warunku zbieżności

$$\Delta V_m(n) < 0$$

otrzymujemy nierówność:

$$\varepsilon^{(M)}(n) \zeta_m(n) - \frac{1}{2} \eta [\zeta_m(n)]^2 + c e^{(m)}(n) \xi_m(n) - \frac{1}{2} \eta c [\xi_m(n)]^2 > 0.$$

Po przekształceniach dostajemy warunek zbieżności algorytmu uczenia:

$$0 < \eta < \frac{2 \left( \varepsilon^{(M)}(n) \zeta_m(n) + c e^{(m)}(n) \xi_m(n) \right)}{[\zeta_m(n)]^2 + c [\xi_m(n)]^2}.$$

Ponieważ w mianowniku występuje wyrażenie o dodatnim znaku, warunkiem istnienia przedziału zbieżności jest:

$$\varepsilon^{(M)}(n) \zeta_m(n) + c e^{(m)}(n) \xi_m(n) > 0,$$

a po wstawieniu za  $\zeta_m(n)$  i  $\xi_m(n)$  wzorów (3.37) i (3.41), dostajemy dokładną postać ostatniego warunku:

$$\begin{aligned} & \left[ \varepsilon^{(M)}(n) \right]^2 \sum_{\gamma=1}^M \left\| \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(M)}(n) \right\|^2 + \\ & + c \sum_{\gamma=1}^M \begin{cases} \varepsilon^{(M)}(n) e^{(\gamma)}(n) \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(M)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{y}^{(\gamma)}(n) \right] & \text{dla } g_\gamma(n) > 0 \\ 0 & \text{dla } g_\gamma(n) \leq 0 \end{cases} + \\ & + c \varepsilon^{(M)}(n) e^{(m)}(n) \left[ \nabla_{\mathbf{w}^{(m)}} \bar{v}^{(M)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(m)}} \bar{y}^{(m)}(n) \right] + \\ & + c^2 \begin{cases} \left[ e^{(m)}(n) \right]^2 \left\| \nabla_{\mathbf{w}^{(m)}} \bar{y}^{(m)}(n) \right\|^2 & \text{dla } g_m(n) > 0 \\ 0 & \text{dla } g_m(n) \leq 0 \end{cases} > 0. \end{aligned}$$

### 3.5 Zbieżność przy wewnętrznej funkcji kary

Funkcję Lapunowa dla  $m$ -tej sieci w algorytmie uczenia sieci globalnej z kryterium globalnym zmodyfikowanym o funkcję kary wewnętrznej wybieramy następującą:

$$V_m(n) = \frac{1}{2}\alpha_0 \left[ \varepsilon^{(M)}(n) \right]^2 - \frac{1}{cg_m(n)},$$

Różnica pierwszego rzędu jest postaci:

$$\Delta V_m(n) = \frac{1}{2} \left[ \varepsilon^{(M)}(n+1) \right]^2 - \frac{1}{2} \left[ \varepsilon^{(M)}(n) \right]^2 - \frac{1}{cg_m(n+1)} + \frac{1}{cg_m(n)},$$

a po odpowiednich przekształceniach mamy:

$$\Delta V_m(n) = \Delta \varepsilon^{(M)}(n) \left( \varepsilon^{(M)}(n) + \frac{1}{2} \Delta \varepsilon^{(M)}(n) \right) + \frac{g_m(n+1) - g_m(n)}{cg_m(n)g_m(n+1)}.$$

Wielkość aktualizacji wartości parametrów  $m$ -tej sieci wynika z algorytmu (2.100), (2.122)

$$\Delta \mathbf{w}^{(m)}(n) = \eta \cdot \left( \varepsilon^{(M)}(n) \left[ \nabla_{\mathbf{w}^{(m)}} \bar{v}^{(M)}(n) \right] + \frac{1}{c[g_m(n)]^2} e^{(m)}(n) \left[ \nabla_{\mathbf{w}^{(m)}} \bar{y}^{(m)}(n) \right] \right).$$

Wstawiając powyższe do ogólnego wzoru (3.16) dostajemy przyrost błędu sieci złożonej traktowanej jako model globalnie optymalny:

$$\Delta \varepsilon^{(M)}(n) = -\eta \zeta_m(n),$$

gdzie:

$$\begin{aligned} \zeta_m(n) = & \varepsilon^{(M)}(n) \sum_{\gamma=1}^M \left\| \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(M)}(n) \right\|^2 + \\ & + \frac{1}{c} \sum_{\gamma=1}^M \frac{1}{[g_\gamma(n)]^2} e^{(\gamma)}(n) \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{v}^{(M)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(\gamma)}} \bar{y}^{(\gamma)}(n) \right]. \end{aligned}$$

Analogicznie dla  $m$ -tej sieci mamy:

$$\Delta e^{(m)}(n) = -\eta \xi_m(n),$$

gdzie:

$$\begin{aligned} \xi_m(n) = & \varepsilon^{(M)}(n) \left[ \nabla_{\mathbf{w}^{(m)}} \bar{v}^{(M)}(n) \right]^T \left[ \nabla_{\mathbf{w}^{(m)}} \bar{y}^{(m)}(n) \right] + \\ & + \frac{1}{c[g_m(n)]^2} e^{(m)}(n) \left\| \nabla_{\mathbf{w}^{(m)}} \bar{y}^{(m)}(n) \right\|^2. \end{aligned}$$

Przy wprowadzonych oznaczeniach różnicę pierwszego rzędu funkcji Lapunowa wyrazimy jako:

$$\begin{aligned} \Delta V_m(n) = & -\eta \zeta_m(n) \left( \varepsilon^{(M)}(n) - \frac{1}{2} \eta \zeta_m(n) \right) + \\ & + \frac{\eta \xi_m(n) \left( e^{(m)}(n) - \frac{1}{2} \eta \xi_m(n) \right)}{c g_m(n) \left( g_m(n) - \eta \xi_m(n) \left[ e^{(m)}(n) - \frac{1}{2} \eta \xi_m(n) \right] \right)}. \end{aligned}$$

Po odpowiednich przekształceniach, warunek zbieżności  $\Delta V_m(n) < 0$  przybiera (dosyć złożoną) postać:

$$\frac{\zeta_m(n) \left( \varepsilon^{(M)}(n) - \frac{1}{2} \eta \zeta_m(n) \right) c g_m(n) \left( g_m(n) - \eta \xi_m(n) \left[ e^{(m)}(n) - \frac{1}{2} \eta \xi_m(n) \right] \right) + \eta \xi_m(n) \left( e^{(m)}(n) - \frac{1}{2} \eta \xi_m(n) \right)}{c g_m(n) \left( g_m(n) - \eta \xi_m(n) \left[ e^{(m)}(n) - \frac{1}{2} \eta \xi_m(n) \right] \right)} > 0.$$

Jest on spełniony, jeżeli iloczyn licznika i mianownika jest dodatni. Niestety, aby to stwierdzić, należałoby rozwiązać ze względu na  $\eta$  następujące równanie piątego stopnia:

$$\begin{aligned} & \eta^5 \left( -\frac{1}{8} c^2 [g_m(n)]^2 \varepsilon^{(M)}(n) [\zeta_m(n)]^2 [\xi_m(n)]^3 \right) + \\ & + \eta^4 \frac{1}{4} c g_m(n) [\xi_m(n)]^3 \left( c g_m(n) \varepsilon^{(M)}(n) e^{(m)}(n) [\zeta_m(n)]^2 + c g_m(n) \varepsilon^{(M)}(n) \zeta_m(n) + \right. \\ & \left. + c [g_m(n)]^3 \varepsilon^{(M)}(n) e^{(m)}(n) [\zeta_m(n)]^2 - 1 \right) + \\ & + \eta^3 \frac{1}{2} c g_m(n) \xi_m(n) \left( -\frac{1}{2} c [g_m(n)]^2 \varepsilon^{(M)}(n) [\zeta_m(n)]^2 [\xi_m(n)]^2 + \right. \\ & - \frac{1}{2} c g_m(n) \varepsilon^{(M)}(n) e^{(m)}(n) \zeta_m(n) [\xi_m(n)]^2 - c g_m(n) \varepsilon^{(M)}(n) [e^{(m)}(n)]^2 [\zeta_m(n) \xi_m(n)]^2 + \\ & - c g_m(n) \varepsilon^{(M)}(n) e^{(m)}(n) \zeta_m(n) \xi_m(n) - \frac{1}{2} c [g_m(n)]^2 \varepsilon^{(M)}(n) e^{(m)}(n) [\zeta_m(n)]^2 + \\ & \left. + e^{(m)}(n) \xi_m(n) \right) + \\ & + \eta^2 c g_m(n) \xi_m(n) \left( \frac{1}{2} c [g_m(n)]^2 \varepsilon^{(M)}(n) \zeta_m(n) \xi_m(n) - \frac{1}{2} g_m(n) \xi_m(n) + \right. \\ & + c g_m(n) \varepsilon^{(M)}(n) [e^{(m)}(n)]^2 \zeta_m(n) \xi_m(n) + \frac{1}{2} c [g_m(n)]^2 \varepsilon^{(M)}(n) e^{(m)}(n) [\zeta_m(n)]^2 + \\ & \left. - [e^{(m)}(n)]^2 \xi_m(n) + \frac{1}{2} c [g_m(n)]^2 \varepsilon^{(M)}(n) \zeta_m(n) \right) + \end{aligned}$$

$$\begin{aligned}
& + \eta c [g_m(n)]^2 \left( -c g_m(n) \varepsilon^{(M)}(n) e^{(m)}(n) \zeta_m(n) \xi_m(n) - \frac{1}{2} c [g_m(n)]^2 \varepsilon^{(M)}(n) [\zeta_m(n)]^2 + \right. \\
& + \frac{1}{2} c g_m(n) \varepsilon^{(M)}(n) e^{(m)}(n) [\zeta_m(n)]^2 \xi_m(n) + e^{(m)}(n) \xi_m(n) + \\
& \left. - c g_m(n) \varepsilon^{(M)}(n) e^{(m)}(n) \zeta_m(n) \xi_m(n) \right) + \\
& + c^2 [g_m(n)]^4 \varepsilon^{(M)}(n) \zeta_m(n) > 0,
\end{aligned}$$

więc wartość analityczna uzyskanego w tym punkcie rezultatu, którego nie będziemy dalej kontynuować, uznać musimy za niewielką.

Wyniki przedstawione w punktach 3.3, 3.4 i 3.5 można przedstawić w formie twierdzeń, jak to zrobiono w punkcie 3.2.

Podsumowując rozważania i wyniki zawarte w niniejszym rozdziale, stwierdzamy, że identyfikacja systemów złożonych generuje problemy, których nie było w identyfikacji pojedynczych obiektów wejściowo-wyjściowych. Przedstawione warunki zbieżności dla sieci złożonej różnią się zasadniczo od warunku dla pojedynczej sieci. Główna różnica polega na tym, że warunek zbieżności pojedynczej sieci ma postać zależną od pozostałych sieci wchodzących w skład sieci złożonej. Próba polepszenia jakości wybranej sieci może skutkować pogorszeniem jakości sieci będących kolejnymi w kaskadzie, jeśli ich wrażliwości (czyli pochodne cząstkowe po parametrach) nie są ze sobą "zgodne", tzn. kierunek zmian parametrów nie odpowiada kierunkom zmian błędów.

Wprawdzie rezultaty te osiągnięto opierając algorytm uczenia o prostą metodę gradientową i zakładając dla uproszczenia system SISO, jednak spodziewać się można, że wykorzystanie innych metod optymalizacji i elementów wielowymiarowych doprowadziłoby do rezultatów o podobnym charakterze. Należy również podkreślić, że wyniki przedstawione w niniejszym podpunkcie mają zastosowanie do dowolnej klasy modeli, nie tylko sieci DRNN.

Otwartym problemem pozostaje dobór wartości współczynników  $\eta$  nie tylko dopuszczalnych, ale zarazem optymalnych, tj. odznaczających się największą szybkością zbieżności, o ile tylko rozwiązanie tego problemu istnieje i jest jednoznaczne.

## Rozdział 4

# Badania symulacyjne

### 4.1 Plan i zakres badań

W ramach niniejszego rozdziału przeprowadzono szereg badań symulacyjnych z użyciem opracowanych i przedstawionych w rozdziale 2 algorytmów identyfikacji systemów złożonych. Oprócz zilustrowania poprawności zaproponowanych algorytmów, eksperymenty numeryczne miały głównie na celu zbadanie wpływu:

- wyboru rodzaju algorytmu na jakość modelowania przy podejściu globalnym i lokalnym;
- liczby rozwinięć modelu wstecz w czasie  $K$  na wartości wskaźników jakości.

Badania obejmowały także:

- zbieżność parametrów modelu;
- porównanie poszczególnych algorytmów;
- porównanie jakości obu modeli: liniowego i sieci neuronowej.

W dalszej części rozdziału stosować będziemy następujące skróty dla oznaczenia poszczególnych rodzajów modeli:

- ML – model lokalnie optymalny;
- MG – model globalnie optymalny;
- MS – model ze wskaźnikiem syntetycznym;

- MZZ – lokalny model zadowalający – kara zewnętrzna;
- MZW – lokalny model zadowalający – kara wewnętrzna.

Badania przeprowadzono dla dwóch przykładów systemów złożonych o strukturze kaskadowej:

- złożonego procesu liniowego;
- reaktora chemicznego z ciągłym przepływem składników (proces nieliniowy).

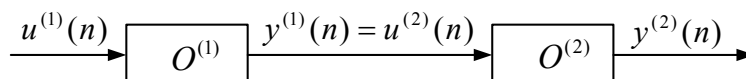
Dalej przedstawimy warunki, w jakich przeprowadzono eksperymenty numeryczne.

**Złożony proces liniowy.** Pierwszym przykładem obiektu identyfikacji będzie system złożony z dwóch jednowymiarowych liniowych elementów, opisanych równaniami różnicowymi:

$$\bar{y}^{(1)}(n) = 0.2y^{(1)}(n-1) + 0.3y^{(1)}(n-2) + 0.5u^{(1)}(n-1) - 0.2u^{(1)}(n-2) + 0.3u^{(1)}(n-3), \quad (4.1)$$

$$\bar{y}^{(2)}(n) = 0.5y^{(2)}(n-1) + 0.1y^{(2)}(n-2) - 0.1y^{(1)}(n-1) + 0.4y^{(1)}(n-2) + 0.3y^{(1)}(n-3). \quad (4.2)$$

Strukturę systemu przedstawia rysunek 4.1.



Rysunek 4.1: Liniowy system złożony

Dane pomiarowe otrzymano poprzez podanie na wejście elementu  $O^{(1)}$  przebiegu  $\{u^{(1)}(n)\}_{n=1}^{190}$ , wygenerowanego w postaci dyskretnego sygnału losowego o długości 190 taktów. Każda wartość  $u^{(1)}(n)$  ( $n = 1, 2, \dots, 190$ ) została wylosowana niezależnie z rozkładem równomiernym na przedziale  $[-1, 1]$ . Po podaniu tego sygnału na wejście pierwszego elementu wyznaczono przebieg  $\{y^{(1)}(n)\}_{n=1}^{190}$  na jego wyjściu. Następnie przebieg wyjściowy pierwszego elementu podano na wejście drugiego elementu, uzyskując przebieg  $\{y^{(2)}(n)\}_{n=1}^{190}$ . Dane testujące wygenerowano w ten sam sposób, w oparciu o inny wylosowany przebieg  $\{u^{(1)}(n)\}_{n=1}^{190}$ .

Przyjmujemy model powyższego systemu o tej samej strukturze, liczbie wejść i wyjść oraz tego samego rzędu. Modele elementów należą do tej samej klasy co

elementy systemu złożonego, jest to zatem sytuacja, w której obiekt leży w klasie modeli, [125].

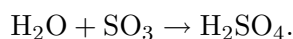
Początkowe wartości parametrów modelu wygenerowano losowo z rozkładem równomiernym, przy czym dla parametrów sprzężeń zwrotnych  $\hat{w}_d^{(m)}$  ( $m = 1, 2$ ;  $d = 1, 2$ ) był to przedział  $[-0.01, 0.01]$ , a dla parametrów związanych z wejściami  $\tilde{w}_d^{(m)}$  ( $m = 1, 2$ ;  $d = 1, 2, 3$ ) przedział  $[-1, 1]$ .

Wartości współczynników w kryterium globalnym (2.25) przyjęto  $\beta_1 = \beta_2 = 0.5$ , natomiast w algorytmie z syntetycznym wskaźnikiem jakości (2.76) ustalono  $\alpha_0 = 0.5$ ,  $\alpha_1 = \alpha_2 = 0.25$ . W algorytmach z funkcją kary ustalono ograniczenia na jakość modeli lokalnych równe  $\lambda_1 = 0.12$ ,  $\lambda_2 = 1$ , przy  $c = 10$ .

Następnie wykorzystano algorytmy opracowane w rozdziale drugim do doboru parametrów modeli ML, MG, MS, MZZ, MZW, wykonując 190 iteracji w trybie identyfikacji na bieżąco. Wszystkie algorytmy uruchamiane były niezależnie, z identycznymi początkowymi wartościami parametrów modelu. Działanie algorytmu dla modelu MZW można podzielić na dwa etapy. W pierwszym dokonuje się optymalizacji wskaźników lokalnych  $Q_L^{(m)}$ ,  $m = 1, 2$ , aby znaleźć się w obszarze rozwiązań dopuszczalnych. Dopiero w drugim etapie optymalizuje się wskaźnik globalny powiększony o funkcję kary wewnętrznej. Suma liczb iteracji obu etapów jest równa 190. Zadanie identyfikacji wykonano czterokrotnie, dla różnych liczb rozwinięć modelu w czasie  $K = 0, 2, 4, 8$ , za każdym razem z tego samego początkowego zestawu parametrów. Dla wartości  $K = 0$  uogólniony algorytm BPTT sprowadza się do klasycznego algorytmu wstecznej propagacji błędów (BP). Wartość współczynnika uczenia  $\eta$  ustalono na 0.3.

Jako optymalne wartości parametrów modeli przyjmujemy te uzyskane po 190-tym takcie identyfikacji. Dla tych wartości oceniano jakość modelu lokalnego (ML), modelu globalnego (MG), modelu globalnego ze wskaźnikiem syntetycznym (MS), modeli globalnych ze wskaźnikami zmodyfikowanymi o funkcje kar: zewnętrznej (MZZ) i wewnętrznej (MZW). Ocenę przeprowadzano przy użyciu sygnału testującego.

**Reaktor chemiczny.** Drugi przykład systemu złożonego został zaczerpnięty z literatury, [98]. Jest to reaktor chemiczny z ciągłym przepływem składników reakcji podawanych na wejście reaktora. Składnikami są woda ( $H_2O$ ) i tlenek siarki ( $SO_3$ ), natomiast w wyniku reakcji powstaje kwas siarkowy ( $H_2SO_4$ ):

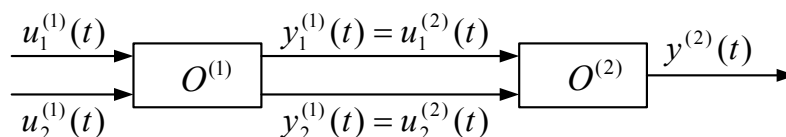


Zakłada się, że temperatura reaktora jest utrzymywana na stałym poziomie. Opis procesu zachodzącego w reaktorze, podany w [98], można wyrazić w postaci następującego układu równań różniczkowych:

$$O^{(1)} : \begin{cases} \frac{dy_1^{(1)}(t)}{dt} = -\frac{Q}{V}y_1^{(1)}(t) - ky_1^{(1)}(t)y_2^{(1)}(t) + \frac{Q}{V}u_1^{(1)}(t) \\ \frac{dy_2^{(1)}(t)}{dt} = -\frac{Q}{V}y_2^{(1)}(t) - ky_1^{(1)}(t)y_2^{(1)}(t) + \frac{Q}{V}u_2^{(1)}(t) \end{cases}, \quad (4.3)$$

$$O^{(2)} : \begin{cases} \frac{dy^{(2)}(t)}{dt} = -\frac{Q}{V}y^{(2)}(t) + ky_1^{(1)}(t)y_2^{(1)}(t) \end{cases}$$

w którym:  $u_1^{(1)}, y_1^{(1)}$  – stężenia  $\text{H}_2\text{O}$  [mol/l],  $u_2^{(1)}, y_2^{(1)}$  – stężenia  $\text{SO}_3$  [mol/l],  $y^{(2)}$  – stężenie  $\text{H}_2\text{SO}_4$  [mol/l],  $k$  – stała reakcji chemicznej [l/mol·min],  $\frac{Q}{V}$  – przepływ w reaktorze (zasilanie i wydalenie) [l/min]. Strukturę systemu przedstawia rys. 4.2.



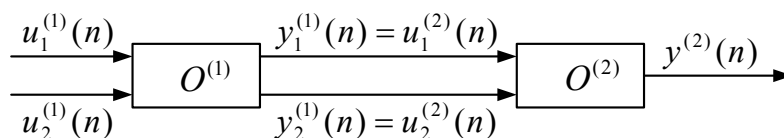
Rysunek 4.2: Reaktor chemiczny

Na podstawie powyższego opisu wykonano symulator w Simulinku. Przyjęto następujące wartości parametrów obiektu:  $k = 0.3$  [l/mol·min],  $\frac{Q}{V} = 0.05$  [l/min]. Warunki początkowe ustalono na  $y_1^{(1)}(0) = 0.5$  [mol/l],  $y_2^{(1)}(0) = 0$  [mol/l],  $y^{(2)}(0) = 0$  [mol/l]. Celem uzyskania danych uczących, wejścia elementu  $O^{(1)}$  pobudzono serią impulsów. Obiekt obserwowano w sposób dyskretny (z całego przebiegu czasowego zdjęto 200 równomiernie rozłożonych próbek). Dla wejścia  $u_1^{(1)}$  były to impulsy o amplitudzie 0.2, która to wartość była utrzymywana przez czas odpowiadający 14 taktom pomiarowym, a przez czas równoważny kolejnym 21 taktom sygnał wejściowy był zerowy. Cały cykl powtarzany był aż do zakończenia symulacji, przy czym moment jego rozpoczęcia odpowiadał 6-temu taktowi pomiarowemu. Dla wejścia  $u_2^{(1)}$  impulsy miały amplitudę 0.3, która to wartość była utrzymywana przez czas odpowiadający 15 taktom, a przez czas równoważny kolejnym 40 taktom sygnał wejściowy był zerowy. Cykl rozpoczął się od początku



symulacji. Ilustracją opisanych sygnałów wejściowych  $\{u_1^{(1)}(n)\}_{n=1}^{200}, \{u_2^{(1)}(n)\}_{n=1}^{200}$  jest rys. 4.20. Po podaniu tych sygnałów na wejście pierwszego elementu zaobserwowano wyjścia  $\{y_1^{(1)}(n)\}_{n=1}^{200}, \{y_2^{(1)}(n)\}_{n=1}^{200}$  pierwszego elementu oraz wyjście  $\{y^{(2)}(n)\}_{n=1}^{200}$  elementu drugiego. Dane testujące wygenerowano analogicznie, w oparciu o nieco inne przebiegi  $\{u_1^{(1)}(n)\}_{n=1}^{200}, \{u_2^{(1)}(n)\}_{n=1}^{200}$  (patrz rys. 4.23).

Zaproponowano model reaktora w postaci sieci złożonej o strukturze identycznej z modelowanym systemem (rys. 4.3). Jako model pierwszego elementu przyjęto sieć DRNN składającą się z 6 neuronów w warstwie ukrytej ( $H_1 = 6$ ) o rzędach  $D_\mu^{(1)} = D_u^{(1)} = 6$  oraz 2 neuronów w warstwie wyjściowej ( $L_1 = 2$ ). Jako model drugiego elementu przyjęto sieć DRNN składającą się z 4 neuronów w warstwie ukrytej ( $H_2 = 4$ ) o rzędach  $D_\mu^{(1)} = D_u^{(1)} = 6$  oraz 1 neuronu w warstwie wyjściowej ( $L_2 = 1$ ).



Rysunek 4.3: Model reaktora chemicznego

Początkowe wartości parametrów sieci wygenerowano losowo z rozkładem równomiernym, przy czym – podobnie jak w modelu liniowym – dla parametrów sprzężeń zwrotnych był to przedział  $[-0.01, 0.01]$ , a dla parametrów związanych z wejściami i z wyjściami przedział  $[-1, 1]$ .

Wartości współczynników w kryterium globalnym (2.25) przyjęto ponownie jako  $\beta_1 = \beta_2 = 0.5$ , natomiast w algorytmie z syntetycznym wskaźnikiem jakości (2.76) ustalono, jak poprzednio,  $\alpha_0 = 0.5$ ,  $\alpha_1 = \alpha_2 = 0.25$ . W algorytmach z funkcją kary ograniczenia na jakość modeli lokalnych dobrano jako  $\lambda_1 = 0.07$ ,  $\lambda_2 = 0.01$ , przy  $c = 20$ .

Następnie wykorzystano algorytmy opracowane w rozdziale drugim do doboru parametrów modeli ML, MG, MS, MZZ, MZW, wykonując 6 cykli uczenia po 200 iteracji w trybie identyfikacji na bieżąco. W pierwszym cyklu użyto wylosowanych początkowych wartości parametrów sieci. W kolejnych cyklach początkowymi wartościami parametrów były wartości końcowe uzyskane w cyklu poprzednim. Wszystkie algorytmy uruchamiane były niezależnie, z identycznymi początkowymi (dla pierwszego cyklu) wartościami parametrów modelu. Zadanie identyfikacji wykonywano czterokrotnie, dla różnych liczb rozwinięć modelu w

czasie  $K = 0, 2, 4, 8$ , za każdym razem startując z tego samego początkowego zestawu parametrów. Wartość współczynnika uczenia  $\eta$  ustalono na 0.05.

Jako optymalne wartości parametrów modeli przyjmujemy te uzyskane po 200-nym takcie 6-tego cyklu. Dla tych wartości oceniano jakość modelu lokalnego (ML), modelu globalnego (MG), modelu globalnego ze wskaźnikiem syntetycznym (MS), modeli globalnych ze wskaźnikami zmodyfikowanymi o funkcje kar: zewnętrznej (MZZ) i wewnętrznej (MZW). Ocenę przeprowadzano przy użyciu sygnałów testujących.

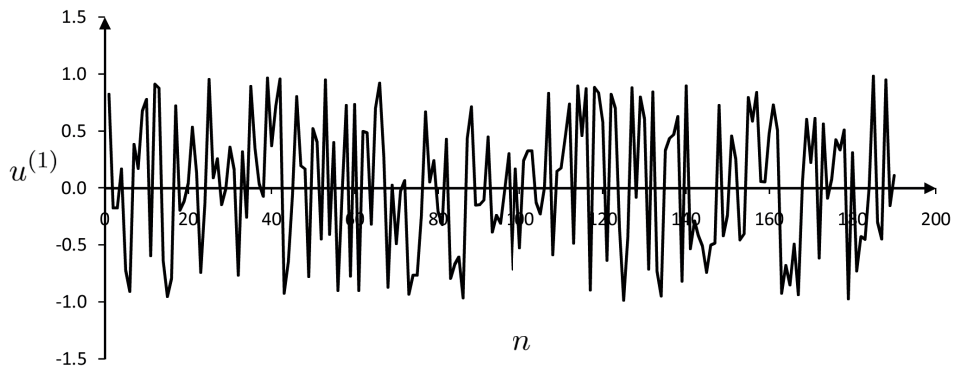
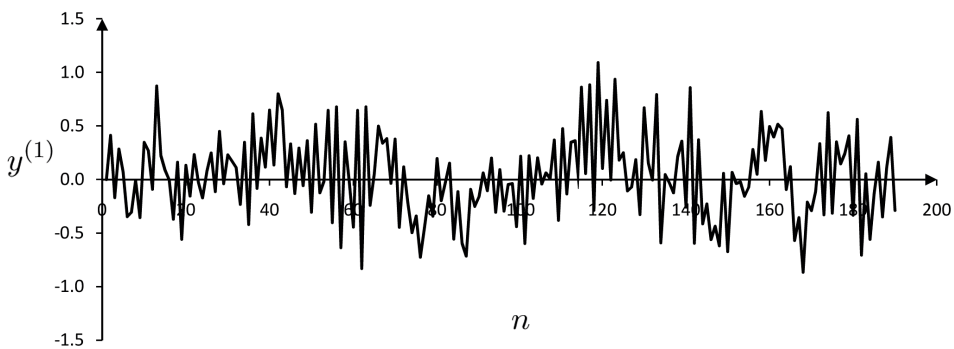
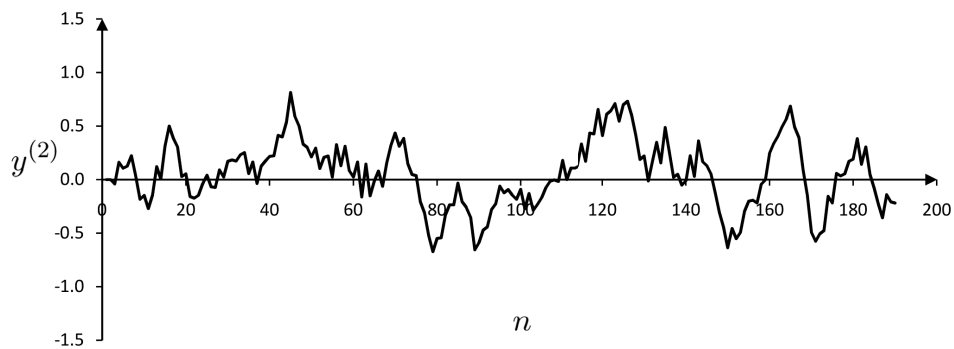
W celu porównania różnych modeli, zastosowano unormowany wskaźnik jakości [90]:

$$\text{REL RMSE} = \sqrt{\frac{\sum_{n=1}^N (\omega(n) - \bar{\omega}(n))^2}{\sum_{n=1}^N (\omega(n) - \omega_{\text{sr}})^2}}, \quad (4.4)$$

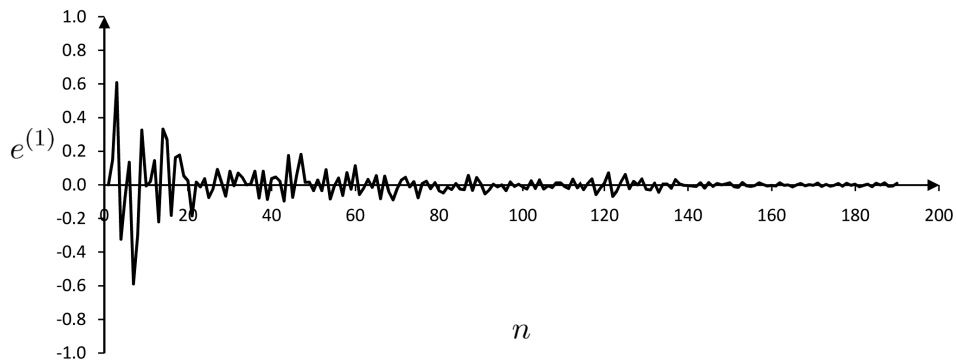
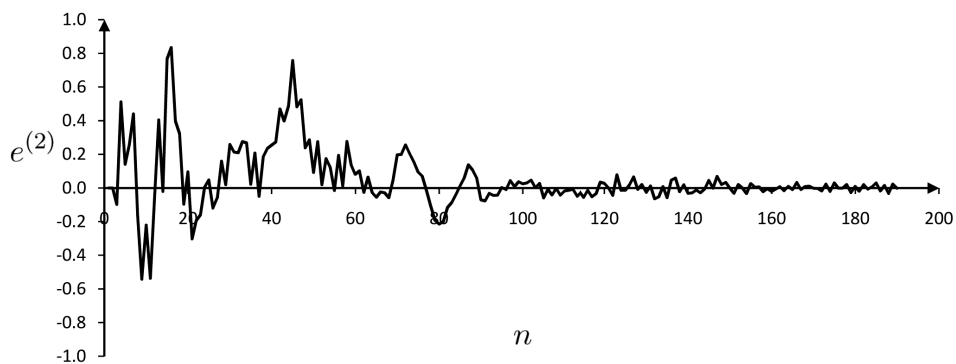
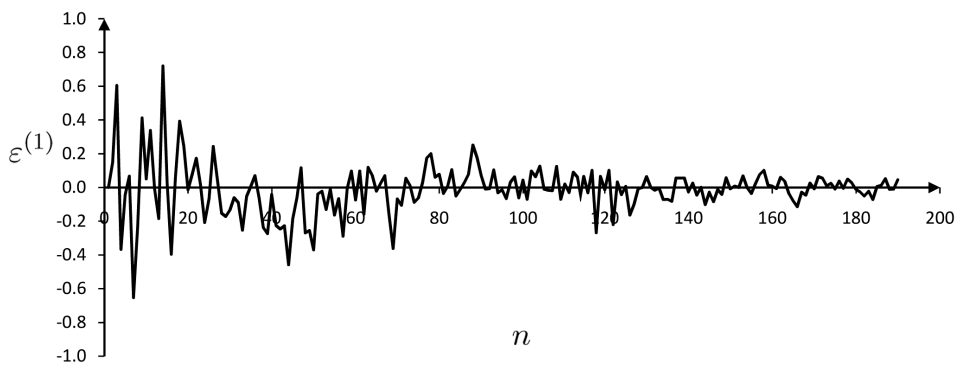
gdzie  $\omega$  – wyjście obiektu,  $\bar{\omega}$  – wyjście modelu,  $\omega_{\text{sr}}$  – średnia wartość wyjścia obiektu po wszystkich  $N$ . Pokazuje on jakość modelu niezależnie od długości serii pomiarowej i zakresu sygnałów. W miejsce  $\omega(n)$  i  $\bar{\omega}(n)$  należy podstawić odpowiednia wyjścia. I tak w przypadku modeli ML wstawimy  $y_l^{(m)}$  i  $\bar{y}_l^{(m)}$  dla  $l = 1, 2, \dots, L_m$  i ustalonego  $m$ , w przypadku modelu MG wstawimy  $v_l^{(m)}$  i  $\bar{v}_l^{(m)}$  dla  $l = 1, 2, \dots, L_m$  i  $m = 1, 2, \dots, M$ . W ocenach modeli MS, MZZ i MZW pojawią się oba wymienione rodzaje wyjść, a przy sumowaniu i wyliczaniu średniej traktowane są jak jeden ciąg liczbowy.

## 4.2 Wyniki badań symulacyjnych – system liniowy

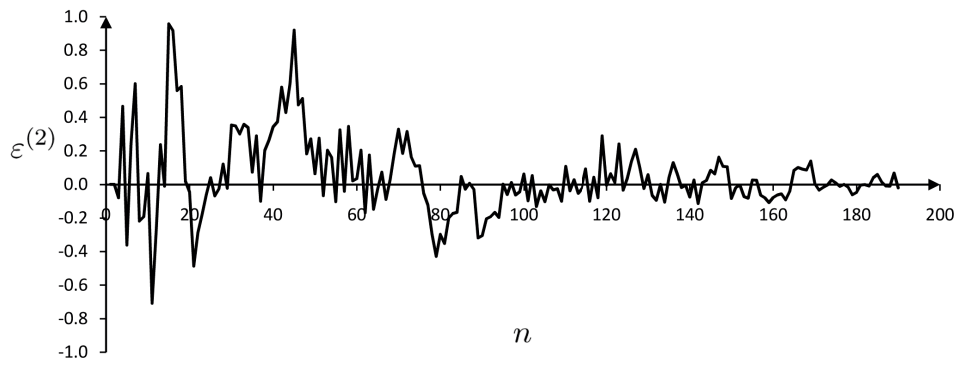
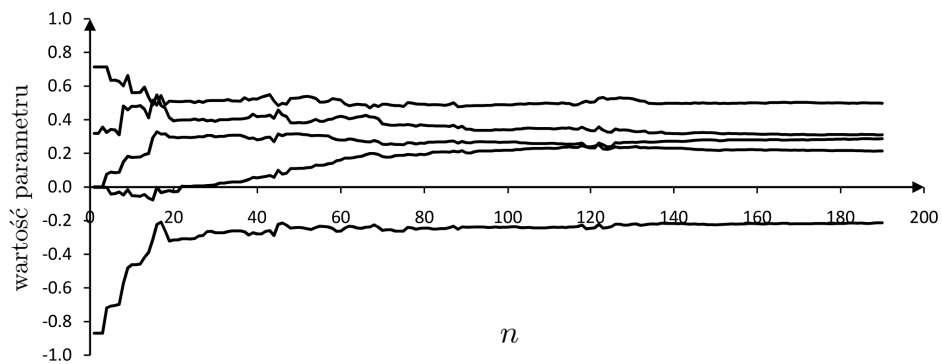
Wykresy 4.4 – 4.18 przedstawiają wybrane przebiegi z systemu złożonego oraz z jego modeli lokalnie optymalnych i z globalnie optymalnego. Wykresy 4.4 – 4.6 dotyczą danych uczących, a konkretnie wejścia pierwszego elementu (wejście zewnętrzne) oraz wyjść obu elementów. Przebiegi 4.7 – 4.14 ilustrują proces identyfikacji.

Rysunek 4.4: Sygnał wejściowy  $u^{(1)}$  pierwszego elementuRysunek 4.5: Sygnał wyjściowy  $y^{(1)} = u^{(2)}$  pierwszego elementuRysunek 4.6: Sygnał wyjściowy  $y^{(2)}$  drugiego elementu

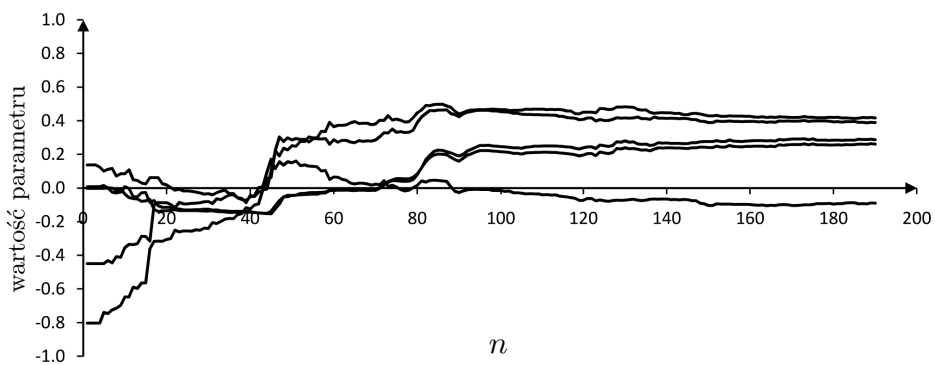
Para wykresów (4.7 i 4.8) przedstawia przebieg błędów w trakcie uczenia na wyjściach modeli poszczególnych elementów modelu lokalnego. Z kolei para (4.9 i 4.10) zawiera przebieg błędów na wyjściach poszczególnych elementów modelu globalnego.

Rysunek 4.7: Uczenie – błąd  $e^{(1)}$  na wyjściu modelu pierwszego elementuRysunek 4.8: Uczenie – błąd  $e^{(2)}$  na wyjściu modelu drugiego elementuRysunek 4.9: Uczenie – błąd  $\varepsilon^{(1)}$  na wyjściu modelu pierwszego elementu

Kolejne wykresy przedstawiają przebieg zmian wartości parametrów modeli lokalnych i modelu globalnego. Celowo nie rozróżniano na nich poszczególnych parametrów, gdyż przebiegi te służą jedynie do ilustracji zbieżności algorytmów.

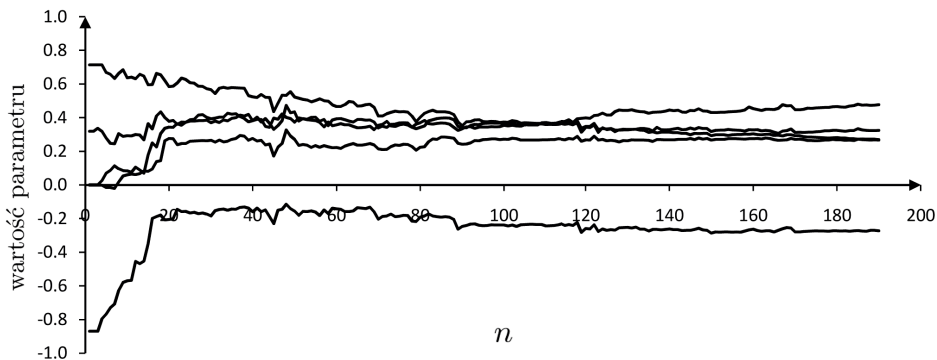
Rysunek 4.10: Uczenie – błąd  $\varepsilon^{(2)}$  na wyjściu modelu drugiego elementu

Rysunek 4.11: Uczenie – przebieg zmian parametrów modelu lokalnego pierwszego elementu

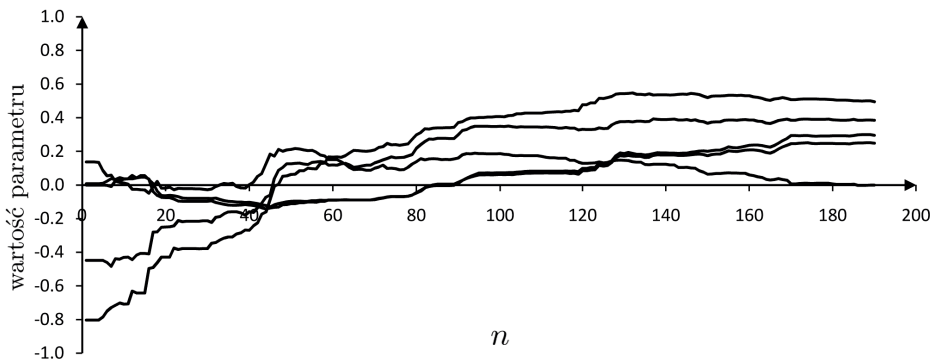


Rysunek 4.12: Uczenie – przebieg zmian parametrów modelu lokalnego drugiego elementu

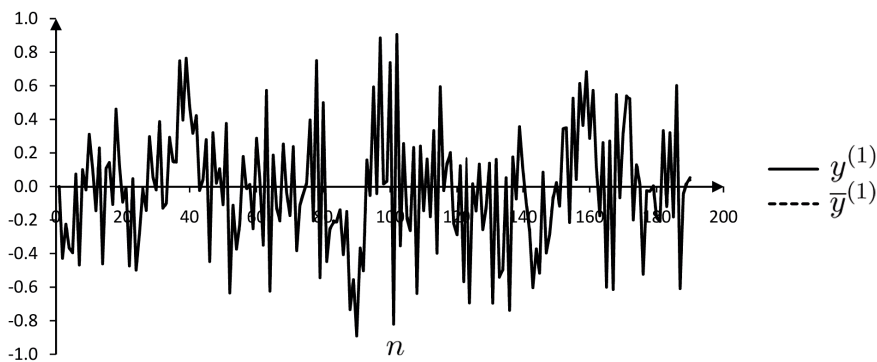
Przebiegi z rysunków 4.15 – 4.18 pokazują jakość działania modeli po złożeniu, na sygnałach testujących.



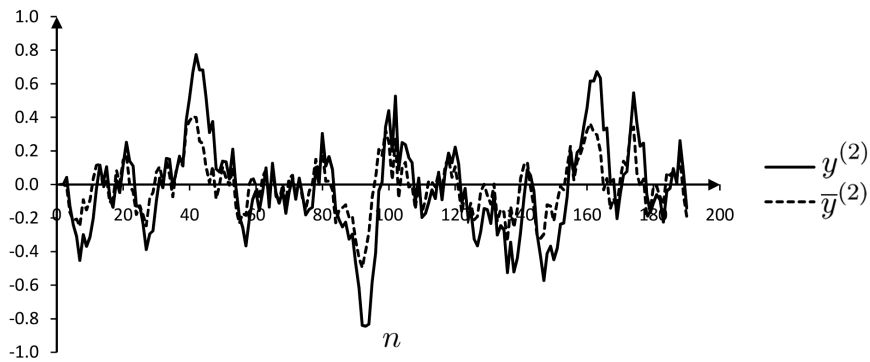
Rysunek 4.13: Uczenie – przebieg zmian parametrów modelu pierwszego elementu w modelu globalnym



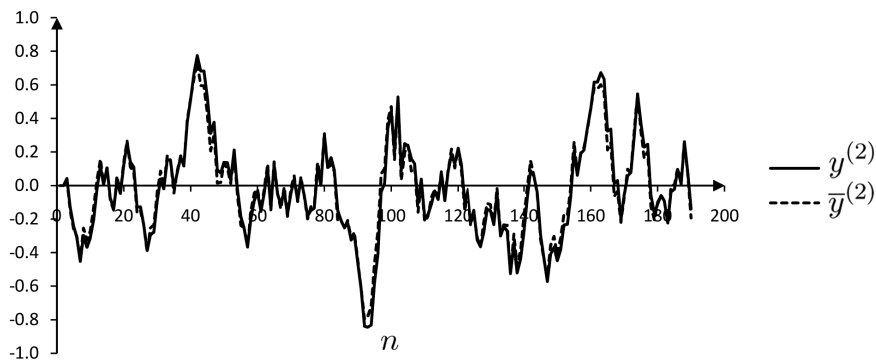
Rysunek 4.14: Uczenie – przebieg zmian parametrów modelu drugiego elementu w modelu globalnym



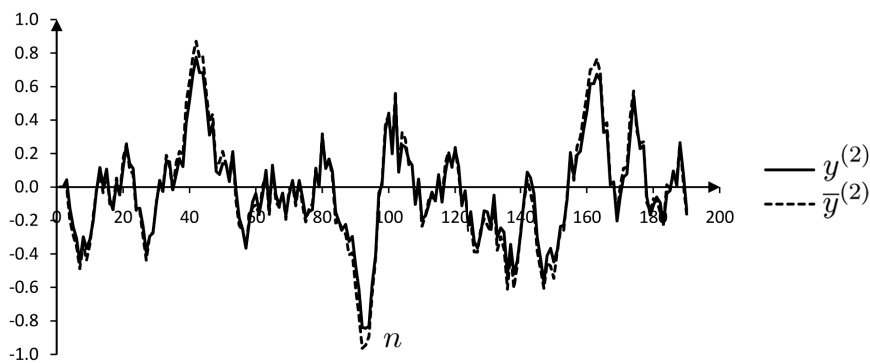
Rysunek 4.15: Testowanie modelu lokalnego po złożeniu – wyjście pierwszego elementu



Rysunek 4.16: Testowanie modelu lokalnego po złożeniu – wyjście drugiego elementu



Rysunek 4.17: Testowanie modelu globalnego po złożeniu – wyjście pierwszego elementu



Rysunek 4.18: Testowanie modelu globalnego po złożeniu – wyjście drugiego elementu

Tabela 4.1 zawiera wartości wskaźników jakości  $Q_L^{(1)}$ ,  $Q_L^{(2)}$  (modele ML),  $Q_G$  (model MG),  $Q_S$  (model MS),  $Q_{ZZ}$  (model MZZ),  $Q_{ZW}$  (model MZW), wyliczo-

nych dla danych testujących uzyskanych kolejno dla  $K = 0, 2, 4, 8$ . Są to wartości wskaźników, które w trakcie uczenia były faktycznie optymalizowane. Natomiast kolejne tabele, od 4.2 do 4.5, zawierają wskaźniki lokalne i wskaźnik globalny dla danych testowych przy kolejnych wartościach  $K = 0, 2, 4, 8$ , uzyskanych po złożeniu modeli. Pozwalają one uzyskać wgląd w jakość działania lokalnie optymalnego i globalnie optymalnego modelu systemu złożonego oraz porównywać poszczególne modele. Lewa strona tabel zawiera nieunormowane wartości wskaźników  $Q_L^{(1)}$  i  $Q_L^{(2)}$ , będące sumą kwadratów błędów na wyjściach modeli poszczególnych elementów, oraz  $\bar{Q}_G$  będący sumą kwadratów błędów z wyjść wszystkich elementów. Wskaźniki QL1, QL2 i QG są ich znormalizowanymi odpowiednikami, wyliczonymi ze wzoru (4.4) na błąd REL RMSE.

Rysunek 4.19 jest obrazem tabeli 4.4, która będzie podstawą do porównania algorytmów.

	$Q_L^{(1)}$	$Q_L^{(2)}$	$Q_G$	$Q_S$	$Q_{ZZ}$	$Q_{ZW}$
$K = 0$	2.5465	1.9343	5.1444	7.6816	14.6828	—
$K = 2$	1.0024	8.6264	2.3493	4.0466	6.1319	—
$K = 4$	0.0140	0.1052	0.6881	1.8057	1.7555	4.3531
$K = 8$	0.0140	0.0949	0.6216	1.7196	1.6869	3.7483

Tabela 4.1: Wartości wskaźników faktycznie optymalizowanych w trakcie uczenia

model	$Q_L^{(1)}$	$Q_L^{(2)}$	$\bar{Q}_G$	model	QL1	QL2	QG
ML	2.5465	1.9343	13.3669	ML	0.0912	0.1305	0.3118
MG	3.4764	4.4125	10.2888	MG	0.1245	0.2977	0.2400
MS	3.2893	3.8330	11.8021	MS	0.1178	0.2586	0.2753
MZZ	2.9654	2.5272	20.6205	MZZ	0.1062	0.1705	0.4810
MZW	2.5465	1.9343	13.3669	MZW	0.0912	0.1305	0.3118

Tabela 4.2: Wskaźniki lokalne i wskaźnik globalny dla danych testowych przy  $K = 0$



model	$Q_L^{(1)}$	$Q_L^{(2)}$	$\bar{Q}_G$	model	QL1	QL2	QG
ML	1.0024	8.6264	9.3114	ML	0.0359	0.5820	0.2172
MG	1.4827	2.4827	4.6986	MG	0.0531	0.1675	0.1096
MS	1.1560	1.9209	6.5548	MS	0.0414	0.1296	0.1529
MZZ	1.0499	1.3577	9.6886	MZZ	0.0376	0.0916	0.2260
MZW	0.9019	1.0939	7.2836	MZW	0.0323	0.0738	0.1699

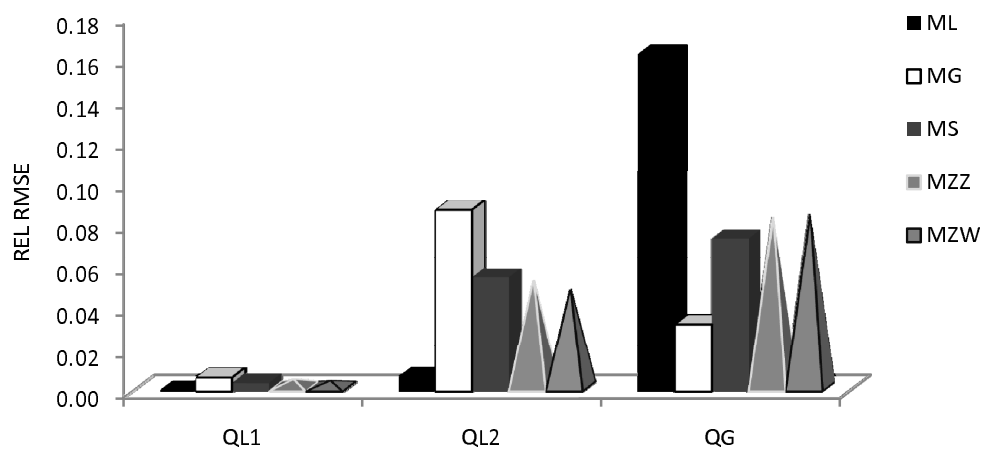
Tabela 4.3: Wskaźniki lokalne i wskaźnik globalny dla danych testowych przy  $K = 2$ 

model	$Q_L^{(1)}$	$Q_L^{(2)}$	$\bar{Q}_G$	model	QL1	QL2	QG
ML	0.0140	0.1052	6.9621	ML	0.0005	0.0071	0.1624
MG	0.1787	1.2910	1.3761	MG	0.0064	0.0871	0.0321
MS	0.1089	0.8034	3.1552	MS	0.0039	0.0542	0.0736
MZZ	0.0949	0.7604	3.5111	MZZ	0.0034	0.0513	0.0819
MZW	0.0754	0.6952	3.5668	MZW	0.0027	0.0469	0.0832

Tabela 4.4: Wskaźniki lokalne i wskaźnik globalny dla danych testowych przy  $K = 4$ 

model	$Q_L^{(1)}$	$Q_L^{(2)}$	$\bar{Q}_G$	model	QL1	QL2	QG
ML	0.0140	0.0949	6.7563	ML	0.0005	0.0064	0.1576
MG	0.1703	1.2732	1.2432	MG	0.0061	0.0859	0.0290
MS	0.0894	0.7959	2.9966	MS	0.0032	0.0537	0.0699
MZZ	0.0866	0.7278	3.3739	MZZ	0.0031	0.0491	0.0787
MZW	0.0614	0.6388	3.5282	MZW	0.0022	0.0431	0.0823

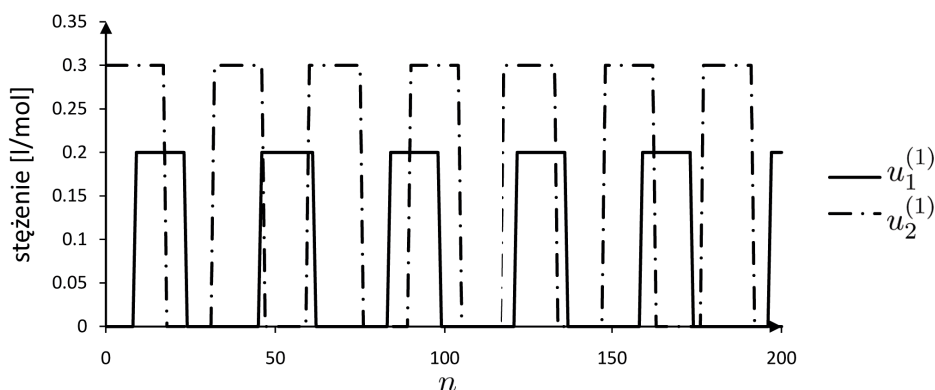
Tabela 4.5: Wskaźniki lokalne i wskaźnik globalny dla danych testowych przy  $K = 8$



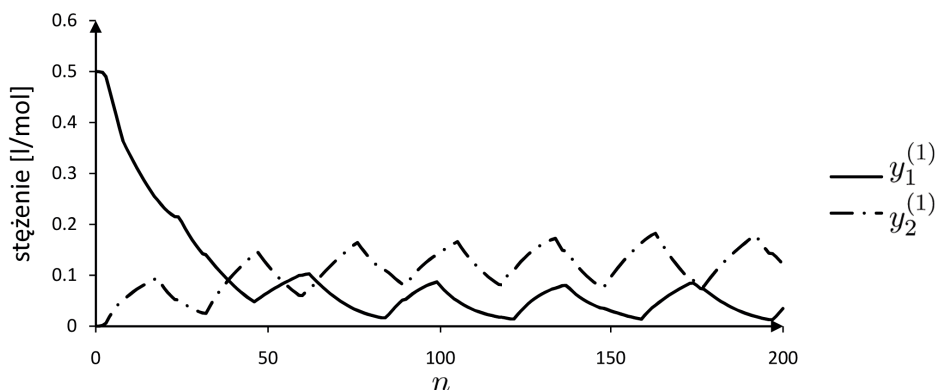
Rysunek 4.19: Porównanie algorytmów (wskaźniki REL RMSE z tab. 4.4)

### 4.3 Wyniki badań symulacyjnych – reaktor chemiczny

Wykresy 4.20 – 4.31 przedstawiają wybrane przebiegi z modelowanego systemu złożonego oraz z jego modeli lokalnie optymalnych i z globalnie optymalnego. Wykresy 4.20 – 4.22 dotyczą danych uczących, a wykresy 4.23 – 4.25 danych użytych do testowania. Przebiegi 4.25 – 4.31 ilustrują rezultaty procesu uczenia sieci neuronowych.



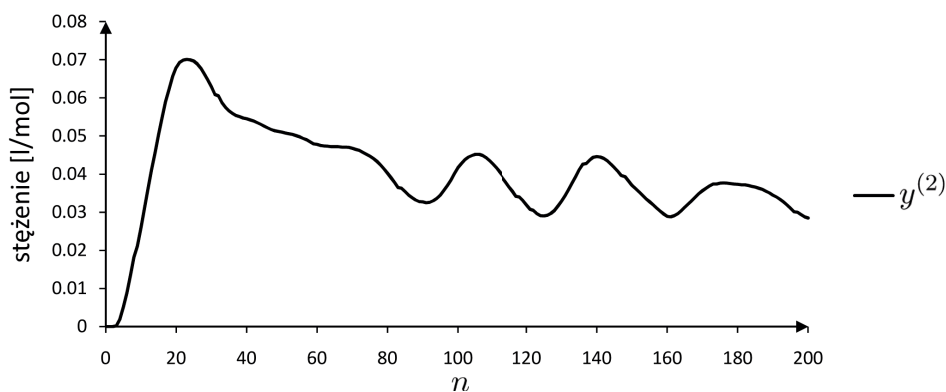
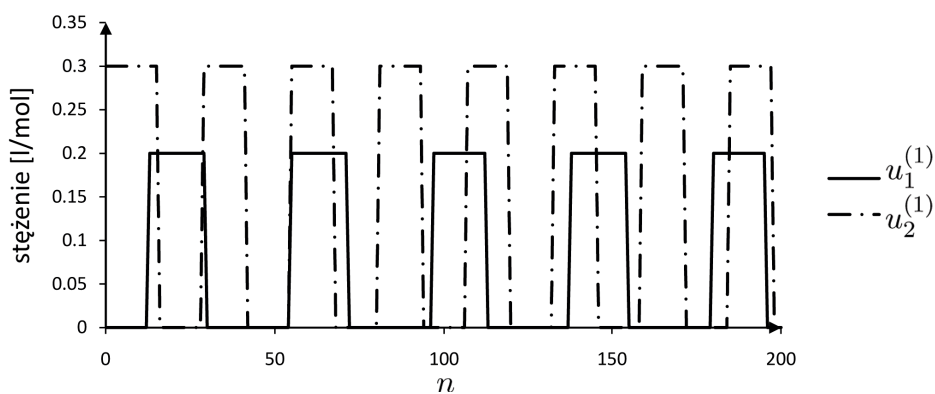
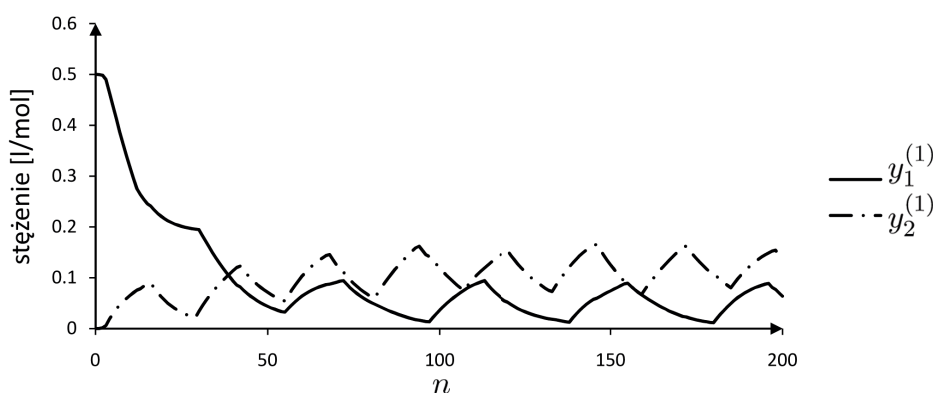
Rysunek 4.20: Dane uczące – sygnały wejściowe  $u_1^{(1)}$  i  $u_2^{(1)}$  pierwszego elementu



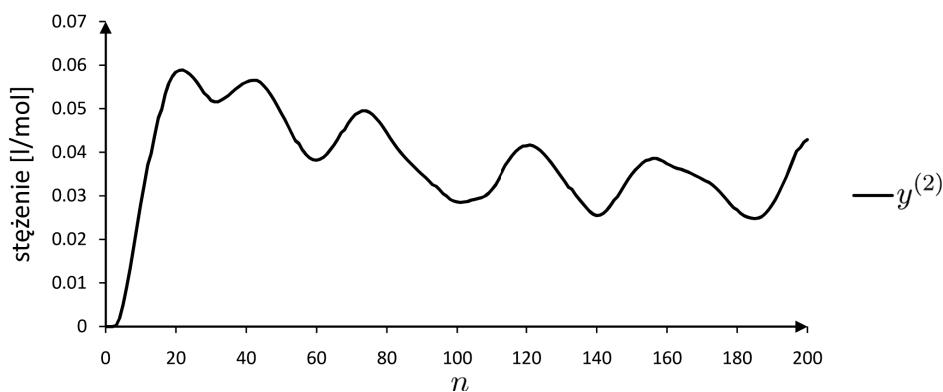
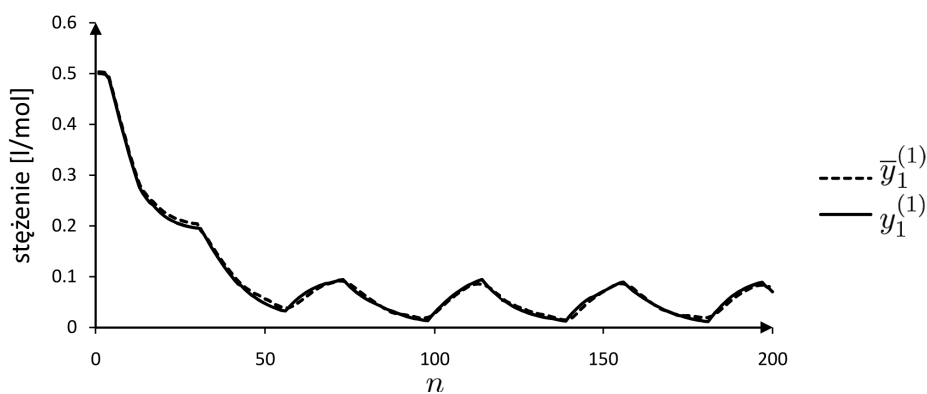
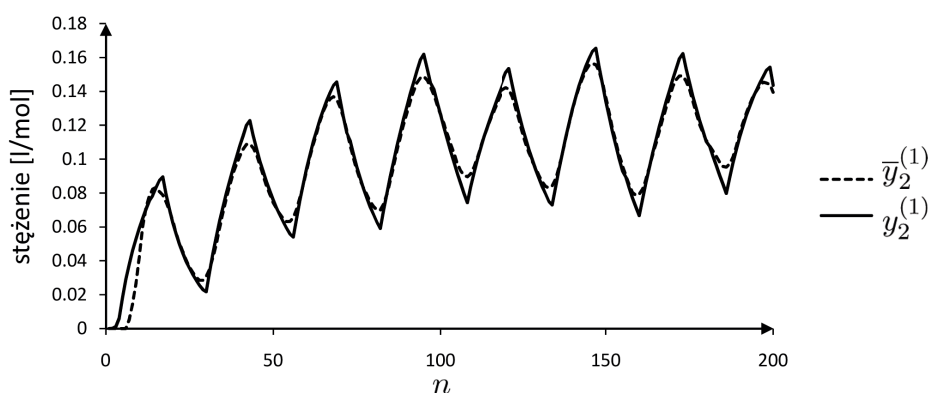
Rysunek 4.21: Dane uczące – sygnały wyjściowe  $y_1^{(1)} = u_1^{(2)}$  i  $y_2^{(1)} = u_2^{(2)}$  drugiego elementu

Wykresy 4.26 – 4.31 pokazują jakość działania sieci neuronowych po złożeniu, na sygnałach testujących. Wykresy 4.26 – 4.28 dotyczą sieci lokalnie optymalnej, a wykresy 4.29 – 4.31 sieci globalnie optymalnej.

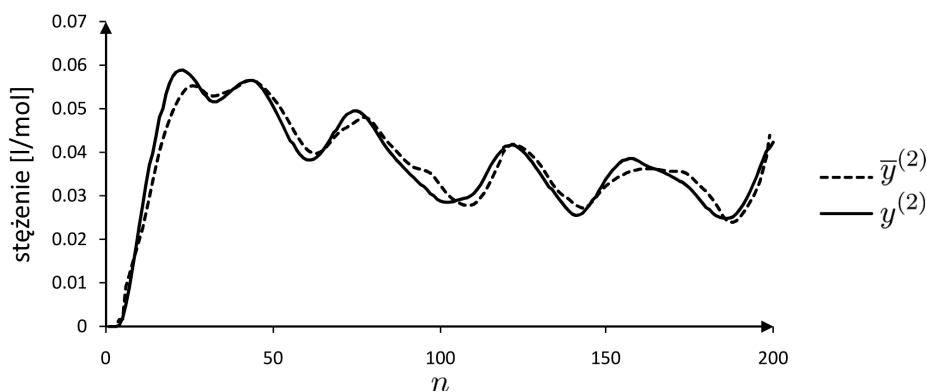
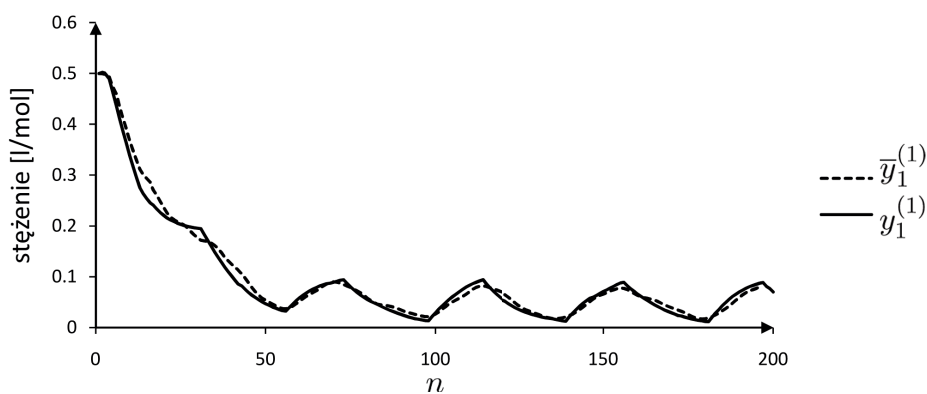
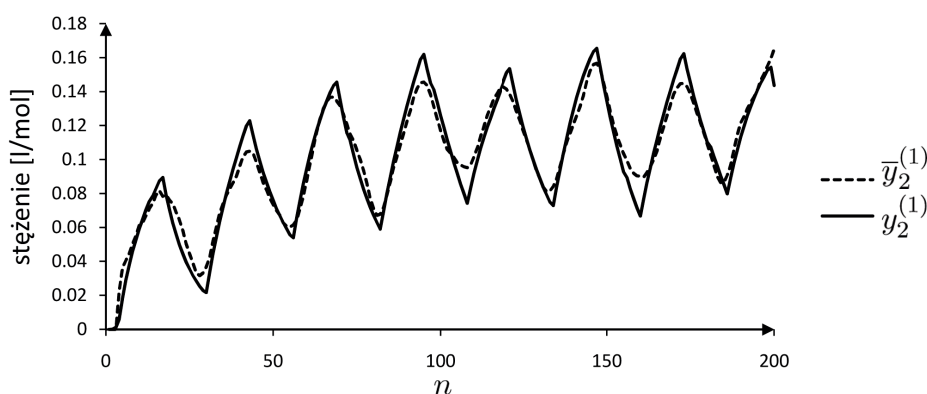
Tabela 4.6 zawiera wartości wskaźników jakości  $Q_L^{(1)}$ ,  $Q_L^{(2)}$  (modele ML),  $Q_G$

Rysunek 4.22: Dane uczące – sygnał wyjściowy  $y^{(2)}$  drugiego elementuRysunek 4.23: Dane testujące – sygnały wejściowe  $u_1^{(1)}$  i  $u_2^{(1)}$  pierwszego elementuRysunek 4.24: Dane testujące – sygnały wyjściowe  $y_1^{(1)} = u_1^{(2)}$  i  $y_2^{(1)} = u_2^{(2)}$  drugiego elementu

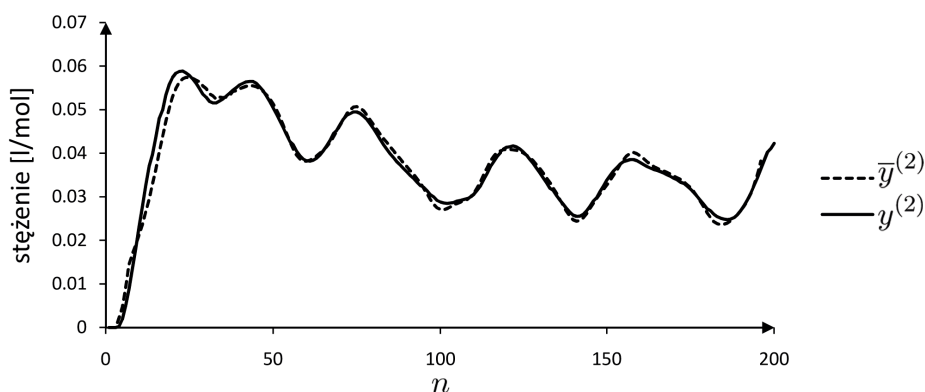
(model MG),  $Q_S$  (model MS),  $Q_{ZZ}$  (model MZZ),  $Q_{ZW}$  (model MZW), wyliczonych dla danych testujących uzyskanych kolejno dla  $K = 0, 2, 4, 8$ . Są to wartości

Rysunek 4.25: Dane testujące – sygnał wyjściowy  $y^{(2)}$  drugiego elementuRysunek 4.26: Testowanie sieci złożonej lokalnie optymalnej – wyjście  $\bar{y}_1^{(1)}$ Rysunek 4.27: Testowanie sieci złożonej lokalnie optymalnej – wyjście  $\bar{y}_2^{(1)}$ 

wskaźników, które w trakcie uczenia sieci były faktycznie optymalizowane. Natomiast kolejne tabele, od 4.7 do 4.10, zawierają wskaźniki lokalne i wskaźnik globalny dla danych testowych przy kolejnych wartościach  $K = 0, 2, 4, 8$ , uzyska-

Rysunek 4.28: Testowanie sieci złożonej lokalnie optymalnej – wyjście  $\bar{y}^{(2)}$ Rysunek 4.29: Testowanie sieci złożonej globalnie optymalnej – wyjście  $\bar{y}_1^{(1)}$ Rysunek 4.30: Testowanie sieci złożonej globalnie optymalnej – wyjście  $\bar{y}_2^{(1)}$ 

nych po złożeniu sieci w model systemu złożonego. Pozwalają one uzyskać wgląd w jakość działania sieci złożonej lokalnie optymalnej i globalnie optymalnej oraz porównywać poszczególne modele w postaci sieci neuronowych. Lewa strona tabel

Rysunek 4.31: Testowanie sieci złożonej globalnie optymalnej – wyjście  $\bar{y}^{(2)}$ 

zawiera nieunormowane wartości wskaźników  $Q_L^{(1)}$  i  $Q_L^{(2)}$ , będące sumą kwadratów błędów na wyjściach sieci modelujących poszczególne elementy, oraz  $\bar{Q}_G$  będący sumą kwadratów błędów z wyjść wszystkich elementów. Wskaźniki QL1, QL2 i QG są ich znormalizowanymi odpowiednikami, wyliczonymi ze wzoru (4.4) na błąd REL RMSE.

Rysunek 4.32 jest obrazem tabeli 4.9, która będzie podstawą do porównania algorytmów.

	$Q_L^{(1)}$	$Q_L^{(2)}$	$Q_G$	$Q_S$	$Q_{ZZ}$	$Q_{ZW}$
$K = 0$	0.3848	0.0075	0.5410	0.6673	0.9819	—
$K = 2$	0.1670	0.0039	0.2138	0.3892	0.6485	—
$K = 4$	0.0212	0.0015	0.0214	0.2223	0.2464	8.3183
$K = 8$	0.0166	0.0014	0.0205	0.2161	0.2381	8.0716

Tabela 4.6: Wartości wskaźników faktycznie optymalizowanych w trakcie uczenia

model	$Q_L^{(1)}$	$Q_L^{(2)}$	$\bar{Q}_G$	model	QL1	QL2	QG
ML	0.3848	0.0075	1.2112	ML	0.1671	0.2731	0.4237
MG	0.4679	0.0081	1.0820	MG	0.2032	0.2959	0.3785
MS	0.4520	0.0075	1.1049	MS	0.1963	0.2728	0.3865
MZZ	0.4325	0.0076	1.2390	MZZ	0.1878	0.2784	0.4334
MZW	0.3848	0.0075	1.2112	MZW	0.1671	0.2731	0.4237

Tabela 4.7: Wskaźniki lokalne i wskaźnik globalny dla danych testowych przy  $K = 0$ 

model	$Q_L^{(1)}$	$Q_L^{(2)}$	$\bar{Q}_G$	model	QL1	QL2	QG
ML	0.1670	0.0039	0.9113	ML	0.0725	0.1411	0.3188
MG	0.2457	0.0074	0.4277	MG	0.1067	0.2697	0.1496
MS	0.1907	0.0050	0.6807	MS	0.0828	0.1829	0.2381
MZZ	0.3339	0.0049	0.7693	MZZ	0.1450	0.1773	0.26914
MZW	0.1494	0.0045	0.7598	MZW	0.0649	0.1629	0.2658

Tabela 4.8: Wskaźniki lokalne i wskaźnik globalny dla danych testowych przy  $K = 2$ 

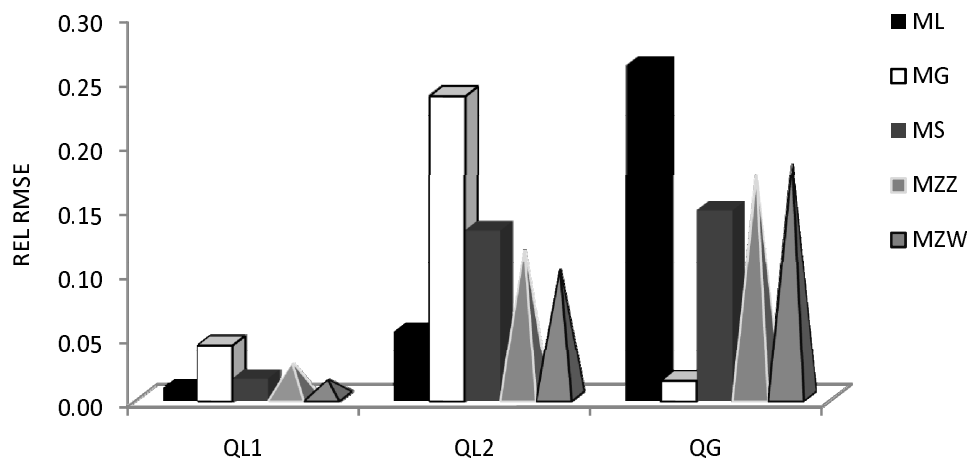
model	$Q_L^{(1)}$	$Q_L^{(2)}$	$\bar{Q}_G$	model	QL1	QL2	QG
ML	0.0212	0.0015	0.7467	ML	0.0092	0.0538	0.2612
MG	0.0999	0.0065	0.0429	MG	0.0434	0.2380	0.0150
MS	0.0380	0.0037	0.4239	MS	0.0165	0.1331	0.1483
MZZ	0.0571	0.0031	0.4928	MZZ	0.0248	0.1147	0.1724
MZW	0.0286	0.0027	0.5177	MZW	0.0124	0.0986	0.1811

Tabela 4.9: Wskaźniki lokalne i wskaźnik globalny dla danych testowych przy  $K = 4$



model	$Q_L^{(1)}$	$Q_L^{(2)}$	$\bar{Q}_G$	model	QL1	QL2	QG
ML	0.0166	0.0014	0.7012	ML	0.0072	0.0517	0.2453
MG	0.0944	0.0058	0.0412	MG	0.0410	0.2115	0.0144
MS	0.0320	0.0036	0.4145	MS	0.0139	0.1308	0.1450
MZZ	0.0472	0.0030	0.4763	MZZ	0.0205	0.1094	0.1666
MZW	0.0274	0.0025	0.5137	MZW	0.0119	0.0901	0.1797

Tabela 4.10: Wskaźniki lokalne i wskaźnik globalny dla danych testowych przy  $K = 8$



Rysunek 4.32: Porównanie algorytmów (wskaźniki REL RMSE z tab. 4.9)

## 4.4 Omówienie wyników

Omówienie wyników eksperymentów numerycznych rozpoczniemy od wpływu liczby rozwinięć  $K$  na wartości wskaźników jakości. Analizując wartości z wszystkich tabel widzimy, że wraz ze wzrostem  $K$  maleją wartości wszystkich wskaźników, przy czym różnica między  $K = 4$  a  $K = 8$  jest już niewielka. Z tabel 4.1 i 4.6 widać też, że brak jest wyliczeń wartości wskaźnika  $Q_{ZW}$  dla  $K = 0$  i dla  $K = 2$ . Okazuje się, że przy niewielkiej liczbie rozwinięć sieci wstecz w czasie algorytm optymalizujący wskaźnik globalny z funkcją kary zewnętrznej nie zdążył zakończyć pierwszego etapu działania, tj. nie osiągnął obszaru rozwiązań dopuszczalnych, dla którego jakości modeli lokalnych są zadowalające w sensie przyjętych w symulacji wartości  $\lambda_1$  i  $\lambda_2$ . W przeprowadzonych badaniach za wystarczającą liczbę rozwinięć modeli wstecz w czasie dla obu rozważanych systemów złożonych należy uznać  $K = 4$ .

Kolejną obserwacją jest występowanie ogólnie mniejszych wartości wskaźników jakości dla modeli pierwszego elementu niż dla modeli drugiego. Dotyczy to zarówno systemu liniowego jak i reaktora chemicznego. Można to wytłumaczyć dodatkowym udziałem błędów modelu elementu  $O^{(1)}$  w błędach popełnianych przez model elementu  $O^{(2)}$ . Innym wyjaśnieniem, dotyczącym zwłaszcza systemu liniowego, jest fakt, że element  $O^{(2)}$  dostaje "uboższy" sygnał identyfikujący. Sygnał  $u^{(2)} = y^{(1)}$  jest odfiltrowanym przebiegiem wejścia  $u^{(1)}$  pierwszego elementu, ponieważ element liniowy o przyjętych w symulacji parametrach działa jak filtr. Stąd dobór prawidłowych wartości parametrów modelu drugiego elementu wymagał większej liczby iteracji, co widać dobrze na wykresach 4.11 – 4.14, ilustrujących zbieżność parametrów modeli liniowych.

Zestawienie wartości wskaźników REL RMSE dla systemu liniowego i dla reaktora chemicznego pokazuje, że dla tego pierwszego udało się uzyskać dokładniejsze modele. Powodem tego stanu rzeczy jest jednak to, że w przypadku liniowym model był identyczny jak obiekt (z dokładnością do parametrów), liczba parametrów modeli była niewielka oraz sygnał identyfikujący był losowy. Natomiast w przypadku reaktora chemicznego klasa modelu (zdeterminowana przez strukturę sieci neuronowej) różniła się od charakterystyki obiektu, liczba parametrów wagowych sieci była stosunkowo duża a przyjęty sygnał identyfikujący charakteryzował się mniejszą zmiennością. Przytoczone wyniki pokazują więc ważność zagadnień doboru struktury modelu oraz doboru sygnału identyfikującego.

Najważniejsze rezultaty przedstawiają rysunki 4.19 i 4.32, obrazujące wartości unormowanych wskaźników REL RMSE dla wszystkich modeli: ML, MG, MS, MZZ, MZW, ocenianych na poziomach lokalnych (QL1 i QL2) i na poziomie globalnym (QG). Rezultaty mają podobny charakter w obu przypadkach: systemu liniowego i reaktora chemicznego.

Rozpocznijmy od omówienia jakości modelu lokalnie optymalnego, a więc takiego, który powstał ze złożenia modeli utworzonych w oderwaniu od całości, jaką jest system złożony. Najmniejszą wartość wskaźników QL1 i QL2 uzyskiwały modele ML odpowiednio pierwszego i drugiego elementu. Największe wartości tych wskaźników dawał model MG. Wartości pośrednie uzyskiwały modele MS, MZZ, MZW. Należy tu zaznaczyć, że podobieństwo między jakościami trzech ostatnich modeli spowodowane jest odpowiednim wyborem ograniczeń na jakość modeli lokalnych w modelach MZZ i MZW. Można byłoby nałożyć ostrzejsze wymagania, a wówczas podobieństwo byłoby mniejsze. Nadal jednak jakość modeli MZZ i MZW na poziomie lokalnym nie powinna być mniejsza niż modelu MG, ani wyższa od modelu ML.

Wskaźnik QG daje inne – globalne – spojrzenie na otrzymane modele. Tym razem modelem o najniższej jakości jest model ML, a najwyższą jakością charakteryzuje się model MG. Modele MS, MZZ, MZW uzyskiwały pośrednie, zbliżone do siebie wartości wskaźnika QG. Widać więc wyraźną różnicę między modelami i prowadzącymi do nich algorytmami. Model lokalnie optymalny, mimo, że ma najmniejsze wartości wskaźników lokalnych, daje duże wartości wskaźnika globalnego QG. Połączenie modeli, które są optymalne dla poszczególnych elementów nie prowadzi na ogół do optymalnego modelu całości. I analogicznie, model globalnie optymalny, pomimo małych wartości wskaźnika globalnego, składa się z modeli, które – rozpatrywane niezależnie – dają duże wartości wskaźników lokalnych. W przypadku liniowym różnice te widoczne są na wykresach 4.7 i 4.8 oraz 4.9 i 4.10, z których widać, że model globalnie optymalny charakteryzował się większymi błędami na wyjściach modeli poszczególnych elementów oraz uczył się dłużej (porównanie przebiegów 4.11 i 4.12 do 4.13 i 4.14), jednak po złożeniu okazywał się lepszym modelem całości (porównanie wykresu 4.16 z 4.18). W przypadku reaktora chemicznego różnicę tę widać najwyraźniej przy porównywaniu wykresu 4.28 z 4.31. Modele MS, MZZ, MZW pozwalają połączyć cechy obu podejść: lokalnego i globalnego, co przedstawiono na rysunkach 4.19 i 4.32.

Można by zadać pytanie: które z przedstawionych podejść jest najlepsze, tzn.

które prowadzi do najlepszego modelu? Przeprowadzone badania symulacyjne skłaniają do następującej odpowiedzi: to zależy od celu tworzenia modelu. Jeżeli, przykładowo model ma być dalej użyty do projektowania algorytmów sterowania lokalnego, najlepszym wyborem będzie skorzystanie z modelu ML. Gdyby natomiast w oparciu o model opracowywano algorytmy sterowania nadrzędnego, należałoby zdecydować się na model MG. Wreszcie, jeśli oceniana będzie jakość sterowania na obu poziomach: lokalnym i globalnym, dobrym rozwiązaniem jest jeden z modeli: MS, MZZ, MZW. Wybór konkretnego podyktowany jest rodzajem wymagań nałożonych na jakość modeli poszczególnych elementów. Jeżeli podano górne ograniczenie na jakość, najwygodniej jest zastosować model MZZ lub MZW. W innym przypadku dogodniejsze może być ustalenie odpowiednich współczynników wagowych dla modelu MS.

## Rozdział 5

# Podsumowanie

W pracy zaproponowano i zbadano algorytmy identyfikacji dynamicznych systemów złożonych przy użyciu rekurencyjnych sieci neuronowych z uwzględnieniem:

- lokalnych ocen modeli;
- globalnej oceny modelu;
- oceny globalnej z uwzględnieniem ocen modeli lokalnych.

Realizując cel pracy, osiągnięto wyniki stanowiące oryginalny wkład autora w dziedzinie identyfikacji systemów i uczenia sieci neuronowych. Przeprowadzone badania pozwoliły wskazać na kierunki dalszych prac.

### 5.1 Oryginalny wkład pracy w dziedzinie identyfikacji systemów oraz sieci neuronowych

Jako elementy nowości w niniejszej pracy należy przede wszystkim wymienić: sformułowania zadań identyfikacji dynamicznych systemów złożonych z wykorzystaniem sieci rekurencyjnych, opracowanie nowych algorytmów uczenia sieci oraz analizę zbieżności algorytmów uczenia sieci złożonej. Dalej omówimy krótko każdy z wymienionych elementów.

**Sformułowania problemów.** Z uwagi na różne możliwości określenia wskaźnika jakości, sformułowano zadania uczenia rekurencyjnej sieci złożonej, będącej:

- modelem lokalnie optymalnym (złożenie modeli lokalnie optymalnych);
- modelem globalnie optymalnym;
- modelem globalnie optymalnym, w którym uwzględniono jakości modeli lokalnych.

W ostatnim przypadku – wielokryterialnym – jakość sieci lokalnych podczas optymalizacji globalnego wskaźnika jakości uwzględniono na dwa sposoby:

- przez utworzenie syntetycznego wskaźnika jakości (w pracy przyjęto kombinację liniową wskaźników lokalnych i wskaźnika globalnego);
- przez minimalizację wskaźnika globalnego w taki sposób, aby jakość modeli lokalnych była zadowalająca. Zadowalające wartości wskaźników lokalnych zapewniono na dwa sposoby:
  - poprzez dodanie do wskaźnika globalnego funkcji kary zewnętrznej;
  - poprzez dodanie do wskaźnika globalnego funkcji kary wewnętrznej.

**Nowe algorytmy.** Dla systemów o strukturze kaskadowej opracowano algorytmy uczenia sieci bazując na metodzie optymalizacji numerycznej gradientu prostego. W celu wyznaczenia wartości gradientów uogólniono na sieć złożoną znaną metodę wstecznej propagacji błędów w czasie (BPTT), posługując się koncepcją uszeregowanych pochodnych cząstkowych.

**Analiza zbieżności.** Zaproponowano wykorzystanie bezpośredniej metody Lapunowa do analizy zbieżności algorytmu uczenia sieci złożonej. Dla przypadku lokalnego, globalnego, ze wskaźnikiem syntetycznym i z zadowalającymi modelami lokalnymi zaproponowano odpowiednie postaci funkcji Lapunowa. Przedstawiono warunki zbieżności algorytmów uczenia sieci rekurencyjnych opartych o metodę gradientu prostego. Uzyskane rezultaty są na tyle ogólne, że stosują się do dowolnej klasy modeli, nie tylko do sieci neuronowych. Wadą przedstawionych warunków zbieżności jest duży koszt obliczeniowy związany z ich dokładnym określaniem w trakcie uczenia na bieżąco.

## 5.2 Plan dalszych badań

Przedmiotem pracy było opracowanie metod modelowania dynamicznych systemów złożonych. W dalszej kolejności można by wykorzystać otrzymane algorytmy w sterowaniu systemami złożonymi. Opracowane techniki identyfikacji pozwoliłyby użytkownikowi wybierać różne metody modelowania, w zależności od wymagań. Podejścia "klasyczne" pozwalają na uzyskanie modeli lokalnie optymalnych. Spośród metod przedstawionych w niniejszej pracy można wybrać takie, które dają model globalnie optymalny, lub też takie, które stanowią połączenie obu tych podejść.

Podstawą do konstrukcji algorytmów identyfikacji była metoda optymalizacji numerycznej gradientu prostego. Można jednak użyć metod efektywniejszych (jak np. metoda Levenberga-Marquardta lub gradientów sprzężonych) i mniej podatnych na utykanie w optimach lokalnych, (takich jak algorytm genetyczny lub symulowane wyżarzanie). W toku dalszych prac więcej uwagi można byłoby poświęcić problemowi doboru parametrów początkowych oraz metodom adaptacyjnego doboru współczynnika uczenia. Nie mniej istotnym czynnikiem wpływającym na prędkość uczenia i na jakość otrzymanego rozwiązania jest dobór serii identyfikującej. Interesujący wydaje się problem automatycznego doboru struktury sieci złożonej. Ważnym zagadnieniem jest przeprowadzenie analizy zbieżności algorytmu uczenia sieci złożonej w obecności zakłóceń pomiarowych. W pracy nie czyniono w sposób jawny żadnych założeń odnośnie obecności zakłóceń w danych pomiarowych.

Każde z wymienionych zagadnień nadaje się na osobne opracowanie.

## 5.3 Uwagi końcowe

W zakończeniu pracy podsumujemy zwięźle jej treść i rezultaty.

Podstawą dla sformułowania problemów identyfikacji systemów złożonych jest możliwość wyróżnienia w systemie elementów składowych oraz wskazania powiązań między nimi. W pracy przedstawiono dwa przykłady: proces produkcji aspiryny, [125] oraz użyty w badaniach symulacyjnych reaktor chemiczny, [98], jednak w literaturze znanych jest więcej przykładów, [24], [118], [51]. To skłania do różnych podejść do modelowania systemów złożonych. Po pierwsze, można utworzyć model każdego elementu z osobna, bez uwzględniania powiązań. Złożenie uzyskanych w ten sposób modeli prowadzi do modelu lokalnie optymalnego.

Ale można też spojrzeć na system złożony jako całość, uwzględniając połączenia między elementami i oceniać wyróżnione ze względu na cel tworzenia modelu wyjścia. To drugie podejście prowadzi do modelu globalnie optymalnego. Możliwe jest również połączenie obu tych podejść, co z kolei prowadzi do modelu globalnie optymalnego, w którym uwzględniono jakość modeli lokalnych.

Celem pracy było opracowanie algorytmów identyfikacji dynamicznych systemów złożonych przy użyciu rekurencyjnych sieci neuronowych z uwzględnieniem:

- lokalnej oceny modelu;
- globalnej oceny modelu;
- oceny globalnej z uwzględnieniem ocen modeli lokalnych.

W związku z realizacją tematu przyjęto pewne założenia oraz wykonano szereg zadań, które dalej pokrótce streścimy.

Założono znajomość struktury modelowanego systemu złożonego. Przyjęto, że dostępne są dyskretne serie pomiarowe wejść i wyjść wszystkich elementów, a co za tym idzie, modele elementów również są dyskretne. Jako klasy modeli elementów systemu złożonego wykorzystano diagonalne rekurencyjne sieci neuronowe (DRNN), składające się z warstwy neuronów rekurencyjnych, tj. ze sprzężeniami zwrotnymi, oraz warstwy neuronów statycznych. Model systemu złożonego uzyskiwany jest przez złożenie poszczególnych sieci neuronowych zgodnie ze strukturą systemu. Otrzymaną sieć określono nazwą *sieć złożona*.

W pracy ograniczono się do systemów o strukturze kaskadowej, przy czym uzasadniono, że nie jest to duże zawężenie gdyż wiele struktur systemów złożonych daje się sprowadzić do kaskadowej. Dla każdego elementu proponowano rekurencyjną sieć neuronową jako model. Po złożeniu wszystkich sieci otrzymuje się sieć złożoną, dla której wyprowadzono algorytmy uczenia pozwalające uzyskać:

- sieć złożoną lokalnie optymalną;
- sieć złożoną globalnie optymalną;
- sieć złożoną, w której połączono cechy obu wyżej wymienionych.

Sformułowania problemów identyfikacji miały charakter ogólny, natomiast konkretne algorytmy uzyskano dla systemów o strukturze kaskadowej. Algorytmy



uczenia sieci skonstruowano w oparciu o metodę optymalizacji gradientu prostego. W celu wyznaczenia wartości gradientów uogólniono na sieć złożoną znaną metodę wstecznej propagacji błędów w czasie (BPTT), posługując się koncepcją uszeregowanych pochodnych cząstkowych.

Tworzenie modelu lokalnie optymalnego sprowadziło się do opracowania algorytmu lokalnego, powtarzanego dla każdej sieci z osobna. Przeniesiono to bezpośrednio metody znane z literatury. Istotny problem pojawił się dla sieci złożonej globalnie optymalnej. W sformułowaniu zadania wprowadzono nowy wskaźnik jakości, w którym uwzględniono błędy na wyróżnionych wyjściach wraz ze współczynnikami liczbowymi określającymi ich wagę. Wymagało to rozwinięcia algorytmu uczenia sieci. Modyfikacja wprowadzona w pracy wiązała się z koniecznością uwzględnienia w algorytmie nie tylko połączeń między warstwami sieci, ale również połączeń między samymi sieciami wchodzącymi w skład sieci złożonej.

Zaproponowano następnie algorytmy, które pozwalają połączyć oba podejścia: lokalne i globalne. Korzystając z podejścia wielokryterialnego, sformułowano zadanie identyfikacji z syntetycznym wskaźnikiem jakości, w którym jakość modeli lokalnych w modelu globalnym uwzględniana jest poprzez odpowiednie współczynniki wagowe w kryterium jakości identyfikacji. Innym sposobem uwzględnienia jakości modeli lokalnych było zapewnienie zadowalających wartości wskaźników lokalnych. W sformułowaniu problemu wymagania przyjęły postać ograniczeń, a opracowane algorytmy optymalizowały globalny wskaźnik jakości zmodyfikowany o

- funkcję kary zewnętrznej;
- funkcję kary wewnętrznej.

Dla opracowanych algorytmów uczenia przeprowadzono analizę zbieżności w oparciu o bezpośrednią metodę Lapunowa. W rezultacie analizy otrzymano warunki, które musi spełniać kluczowy dla algorytmu uczenia sieci współczynnik uczenia  $\eta$ .

Przeprowadzono badania symulacyjne z wykorzystaniem modeli:

- lokalnie optymalnych;
- globalnie optymalnych;
- globalnie optymalnych z uwzględnieniem modeli lokalnych poprzez:

- syntetyczny wskaźnik jakości;
- funkcję kary zewnętrznej i wewnętrznej.

Badania potwierdziły skuteczność algorytmów oraz pozwoliły na porównanie różnych podejść do identyfikacji systemów złożonych. Implementacji opracowanych algorytmów dokonano w środowisku Matlab, Simulink oraz w programie Microsoft Excel. Posłużono się dwoma przykładami systemów złożonych:

- złożonym procesem liniowym;
- reaktorem chemicznym z ciągłym przepływem składników, którego symulator zrealizowano w środowisku Simulink.

Na przykładzie procesu liniowego pokazano w przejrzysty sposób przebiegi błędów w trakcie uczenia i testowania modeli, przebiegi zmian parametrów w trakcie uczenia i ich zbieżność. Przykład modelowania reaktora chemicznego pokazał zdolność modelowania nieliniowego, dynamicznego systemu złożonego przy użyciu rekurencyjnej sieci złożonej oraz wskazał na możliwość praktycznego zastosowania opracowanych algorytmów. Oceniona została jakość modeli przed i po złożeniu. Porównanie algorytmów uczenia pokazało, że każdy z nich może okazać się przydatny, a przy wyborze konkretnego algorytmu powinno się kierować celem tworzenia modelu, a dokładniej wymaganiami, jakie w konkretnym zadaniu powinien spełniać model na poziomie globalnym i na poziomach lokalnych.

Część rezultatów prezentowana była w pracach [29], [30], [31], [33], [34], [35], [36], [37]. Prezentowana praca poszerza aktualny stan wiedzy i zakres dostępnych technik zarówno w dziedzinie identyfikacji dynamicznych systemów złożonych, jak i w dziedzinie uczenia sieci neuronowych. Rezultaty pracy są przydatne do opracowywania komputerowych systemów sterowania systemami złożonymi.

**Niniejsza praca jest finansowana ze środków na naukę w latach 2007/2008 jako projekt badawczy nr N N516 3946 33.**

# Bibliografia

- [1] Åstrom K.J., Wittenmark B., *Adaptive Control*. Addison-Wesley, New York, USA, 1989. [cytowanie na str. 21]
- [2] Atiya A.F., Parlos A.G., *New Results on Recurrent Network Training: Unifying the Algorithms and Accelerating Convergence*. IEEE Trans. on Neural Networks, Vol. 11, No. 3, May 2000. 697–709. [cytowanie na str. 8]
- [3] Abonyi J., Babuška R., *Local and Global Identification and Interpretation of Parameters in Takagi-Sugeno Fuzzy Models*. Proc. of IEEE Conf. Fuzzy Systems, San Antonio, TX, 2000. 835–840. [cytowanie na str. 6]
- [4] Babuška R., Verbruggen H., *Neuro-fuzzy methods for Nonlinear System Identification*. Annual Reviews in Control, Vol. 27, Pergamon, 2003. 73–85. [cytowanie na str. 4]
- [5] Babuška R., Johansen T.A., *Multiobjective Identification of Takagi-Sugeno Fuzzy Models*. IEEE Trans. on Fuzzy Systems, Vol. 11, No. 6, December 2004. 847–860. [cytowanie na str. 6]
- [6] Babuška R., Lendek Zs., De Schutter B., *Stability of Cascaded Takagi-Sugeno Fuzzy Systems*. Proc. of IEEE Conf. Fuzzy Systems, July 2007. 1–6. [cytowanie na str. 6]
- [7] Baratti R., Cannas B., Fanni A., Pilo F., *Automated Recurrent Neural Network Design to Model the Dynamics of Complex Systems*. Journal of Neural Computing & Applications, Vol. 9, No. 3, October 2000. 190–201. [cytowanie na str. 6]
- [8] Baretto G.A., Araujo A.F.R., *Identification and Control of Dynamical Systems Using the Self-Organizing Map*. IEEE Trans. on Neural Networks, Vol. 15, No. 5, September 2004. 1244–1259. [cytowanie na str. 8]
- [9] Bazaraa M.S., Sherali H.D., Shetty C.M., *Nonlinear Programming – Theory and Algorithms*. Wiley-Interscience, A John Wiley & Sons Inc., Hoboken, New Jersey, 2006. [cytowanie na str. 21, 89]

- [10] Beaufays F., Wan E.A., *Relating Real-Time BackPropagation and BackPropagation Through Time: An Application of Flow Graph Interreciprocity*. Neural Comput., Vol. 6, 1994. 296–305. [cytowanie na str. 34]
- [11] Behera L., Kumar S., Patnaik A., *On Adaptive Learning Rate That Guarantees Convergence in Feedforward Networks*. IEEE Trans. on Neural Networks, Vol. 17, No. 5, 2006. 1116–1125. [cytowanie na str. 24]
- [12] Billings S.A., *Identification of Nonlinear Systems – A Survey*. IEEE Proceedings, Vol. 127, No. 6, 1980. 272–285. [cytowanie na str. 3]
- [13] Bishop C.M., *Neural Networks for Pattern Recognition*. Oxford University Press Inc., New York, 2005. [cytowanie na str. 3]
- [14] Bubnicki Z., *Modeling Complex Systems*. Springer-Verlag, New York, 2004. [cytowanie na str. 7]
- [15] Bottura C.P., Torrico A.F.C., *Decentralized Multivariate Identification of Interconnected Systems by a Stochastic Subspace Method*. Proc. of 41<sup>st</sup> IEEE Conf. on Decision and Control, Las Vegas, Nevada, USA, December 2002. 1–6. [cytowanie na str. 6]
- [16] Bubnicki Z., *Identyfikacja obiektów sterowania*. PWN, Warszawa, 1974. [cytowanie na str. 4, 5]
- [17] Bubnicki Z., *Global modeling and identification of network systems*. Proc. of 18<sup>th</sup> International Conference on Systems Engineering, Wright State University, Dayton, USA, 1984. [cytowanie na str. 2]
- [18] Bubnicki Z., *Teoria i algorytmy sterowania*. PWN, Warszawa, 2005. [cytowanie na str. 45]
- [19] Campolucci P., Uncini A., Piazza F., Bhaskar D.R., *On-line Learning Algorithms for Locally Recurrent Neural Networks*. IEEE Trans. on Neural Networks, Vol. 10, No. 2, March 1999. 253–271. [cytowanie na str. 21]
- [20] Chen S., Billings S.A., *Neural Networks for Nonlinear Dynamic System Modelling and Identification*. IEEE Trans. on Neural Networks, Vol. 56, No. 2, 1992. 319–346. [cytowanie na str. 3]
- [21] Chen F.C., Khalil H.K., *Adaptive Control of Nonlinear Systems Using Neural Networks*. International Journal of Control, Vol. 55, 1992. 1299–1317. [cytowanie na str. 100]
- [22] Chong E.K.P., Żak S.H., *An Introduction to Optimization*. Wiley-Interscience series in discrete mathematics and optimization, John Wiley & Sons, Inc., 2001. [cytowanie na str. 24, 89]

- [23] Cybenko G., *Approximation by superposition of sigmoidal function*. Mathematics of Control Signals and Systems, Vol. 2, No. 4, 1989. 303–314. [cytowanie na str. 3]
- [24] Drałus G., Świątek J., *Sieci neuronowe w modelowaniu złożonego obiektu chemicznego*. XIV Krajowa Konferencja Automatyki, Zielona Góra, 2002. 825–830. [cytowanie na str. 6, 147]
- [25] Drałus G., Świątek J., *Global modeling of complex systems by neural networks*. Proc. of 7<sup>th</sup> International Symposium on Artificial Life and Robotics, Oita, Japan, 2002. 618–621. [cytowanie na str. 6]
- [26] Drałus G., Świątek J., *A Penalty Function to Obtain Satisfactory Local Models of Complex Systems*. Advances in Soft Computing; Neural Network and Soft Computing; Leszek Rutkowski, Janusz Kacprzyk Editors, Physica-Verlag, Heidelberg, New York, 2003. 167–172. [cytowanie na str. 6]
- [27] Drałus G., *Modeling of Dynamic Nonlinear Complex Systems Using Neural Networks*. Proc. of 15<sup>th</sup> International Conference on Systems Science, Wrocław, Vol. 3, 2004. 87–96. [cytowanie na str. 6]
- [28] Drapała J., Świątek J., *Identyfikacja dynamicznych systemów złożonych z wykorzystaniem sieci neuronowych*. VI Krajowa Konferencja Inżynieria Wiedzy i Systemy Ekspertowe, Vol. 1, Wrocław, 2006. 311–320. [cytowanie na str. 5]
- [29] Drapała J., Świątek J., *Modeling of Dynamic Complex Systems by Neural Networks*. Proc. of 18<sup>th</sup> International Conference on Systems Engineering ICSE, Coventry University, UK, September 2006. 109–112. [cytowanie na str. 150]
- [30] Drapała J., Świątek J., *Analysis of Optimization Methods in Neural Network Learning Algorithm for Modeling Complex Systems*. Information systems architecture and technology ISAT 2006, Wrocław: Oficyna Wydaw. PWroc. (Biblioteka Informatyki Szkół Wyższych), Wrzesień 2006. 49–58. [cytowanie na str. 150]
- [31] Drapała J., Świątek J., *Complexity Analysis of Algorithm of Recurrent Multi-layer Perceptron Learning for Global Modeling of Complex Systems*. Information systems architecture and technology ISAT 2007, Decision Making Models, Eds Jerzy Świątek [i in.], Wrocław: Oficyna Wydaw. PWroc., Wrzesień 2007. 217–223. [cytowanie na str. 150]
- [32] Drapała J., Świątek J., *Global and Local Approach to Complex Systems Modeling using Dynamic Neural Networks – Analogy with Multiagent Systems*. 11th International Conference KES 2007, Vietri sul Mare, Italy, Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence, Springer-Verlag Berlin Heidelberg, Vol. 4693, 2007. 279–286. [cytowanie na str. 5]

- [33] Drapała J., Świątek J., *Algorithm of Recurrent Multilayer Perceptrons Learning for Global Modeling of Complex Systems*. Proc. of 16<sup>th</sup> International Conference on Systems Science ICSS 2007, Wrocław, September 2007. 351–358. [cytowanie na str. 150]
- [34] Drapała J., *On Convergence of Algorithm of Neural Networks Learning for Global Modeling of Complex Systems*. Proc. of 16<sup>th</sup> International Conference on Systems Science ICSS 2007, Wrocław, September 2007. 359–368. [cytowanie na str. 150]
- [35] Drapała J., Świątek J., *Backpropagation Trough Time for Learning of Interconnected Neural Networks – Identification of Complex Systems*. Proc. of 19<sup>th</sup> International Conference on Systems Engineering ICSEng, Ed. by Henry Selvaraj and Mariusz Rawski, Los Alamitos [i in.] : IEEE Computer Society [Press], Las Vegas, Nevada, USA, August 2008. 165–170. [cytowanie na str. 150]
- [36] Drapała J., Świątek J., Brzostowski K., *Stable Learning Algorithm of Global Neural Network for Identification of Dynamic Complex Systems*. Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence, Vol. 5097, 2008. 17–27. [cytowanie na str. 150]
- [37] Drapała J., Świątek J., *Dynamiczne sieci neuronowe jako globalnie optymalny model systemu złożonego - zbieżność algorytmu uczenia*. Sterowanie i automatyzacja: aktualne problemy i ich rozwiązania. Pod red. Krzysztofa Malinowskiego, Leszka Rutkowskiego, Akademicka Oficyna Wydawnicza EXIT, Szczyrk, Maj 2008. 155–164. [cytowanie na str. 150]
- [38] Duch W., Korbicz J., Rutkowski L., Tadeusiewicz R. – red., *Biocybernetyka i Inżynieria Biomedyczna 2000. Sieci Neuronowe, Tom 6*. Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2000. [cytowanie na str. 3, 4]
- [39] Ehrgott M., *Multicriteria Optimization*. Springer-Verlag, Berlin-Heidelberg, 2005. [cytowanie na str. 82]
- [40] Eykhoff P., *Identyfikacja w układach dynamicznych*. PWN, Warszawa, 1980. [cytowanie na str. 19]
- [41] Findeisen W., *Wielopoziomowe układy sterowania*. PWN, Warszawa, 1974. [cytowanie na str. 7, 44]
- [42] Findeisen W., Szymanowski J., Wierzbicki A., *Teoria i metody obliczeniowe optymalizacji*. PWN, Warszawa, 1974. [cytowanie na str. 21, 89]
- [43] Findeisen W., *Struktury sterowania dla złożonych systemów*. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 1997. [cytowanie na str. 7]

- [44] Fresewinkel T., *Modelling of complex systems by sequential identification of subsystems*. Regelungstechnik, Vol. 32, No. 2, West Germany, February 1984. 51–55. [cytowanie na str. 6]
- [45] Fajarewicz K., *On Construction of Adjoint Systems for Hybrid Continuous-Discrete-Time Systems*. XVI Krajowa Konferencja Automatyki, Recent Advances in Control and Automation, Eds. Malinowski K., Rutkowski L., Akademicka Oficyna Wydawnicza EXIT, Szczyrk, Maj 2008. 202–211. [cytowanie na str. 8]
- [46] Goodwin G.C., Payne R.L., *Dynamic System Identification: Experiment Design and Data Analysis*. Academic Press Inc., New York, 1977. [cytowanie na str. 21]
- [47] Górecki H., *Optymalizacja i sterowanie systemów dynamicznych*. Uczelniane Wydawnictwa Naukowo-Dydaktyczne Akademii Górniczo-Hutniczej w Krakowie, Kraków, 2006. [cytowanie na str. 3]
- [48] Gręblicki W., Pawlak M., *Cascade Non-linear System Identification by a Nonparametric Method*. International Journal of System Science, Vol. 25, No. 1, 1994. 129–153. [cytowanie na str. 6]
- [49] Gupta M.M., Jin L., Homma N., *Static and Dynamic Neural Networks – from fundamentals to advanced theory*. IEEE Press, Wiley-Interscience, A John Wiley & Sons Inc., Hoboken, New Jersey, 2003. [cytowanie na str. 7, 100]
- [50] Gutenbaum J., *Modelowanie Matematyczne Systemów*. Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2003. [cytowanie na str. 1]
- [51] Gutenbaum J., *Identyfikacja sterowanych systemów o złożonej strukturze*. Prace Naukowe Instytutu Cybernetyki Technicznej Politechniki Wrocławskiej, Seria: Monografie, Nr 22, Wrocław, 1993. [cytowanie na str. 5, 147]
- [52] Hasiewicz Z., Mzyk G., *Combined parametric-nonparametric identification of Hammerstein systems*. IEEE Trans. on Automatic Control International Journal of System Science, Vol. 49, No. 8, August 2004. 1370–1375. [cytowanie na str. 6]
- [53] Haykin S., *Neural Networks – A Comprehensive Foundation*. Pearson, Prentice Hall, 1999. [cytowanie na str. 3]
- [54] Hornik K., Stinchcomb M., White H., *Multilayer feedforward networks are universal approximators*. Neural Networks, Vol. 2, 1989. 359–366. [cytowanie na str. 3]
- [55] Hovakimyan N., Lavretsky E., Bong-Jun Y., Calise A.J., *Coordinated Decentralized Adaptive Output Feedback Control of Interconnected Systems*. IEEE Trans. on Neural Networks, Vol. 16, No. 1, January 2005. 185–194. [cytowanie na str. 6]
- [56] Ikonen E., Najim K., *Advanced Process Identification and Control*. CRC Press LLC, 2002. [cytowanie na str. 22, 27, 29]

- [57] Ioannou P.A., Pitsillides A. – Editors., *Modeling and Control of Complex Systems*. Control Engineering Series, CRC Press, Francis & Taylor Group, 2008. [cytowanie na str. 7]
- [58] Jagannathan S., Lewis F.L., *Multilayer Discrete-Time Neural Net Controller With Guaranteed Performance*. IEEE Trans. on Automatic Control, Vol. 7, 1996. 107–130. [cytowanie na str. 21, 100]
- [59] Jagannathan S., Lewis F.L., *Identification of Nonlinear Dynamical Systems using Multilayered Neural Networks*. Automatica, Vol. 32, No. 12, 1996. 1707–1712. [cytowanie na str. 23]
- [60] Janczak A., *Identification of Nonlinear Systems using Neural Networks and Polynomial models – A Block-Oriented Approach*. Lecture Notes in Control and Information Sciences, Springer, Berlin-Heidelberg, New York, 2005. [cytowanie na str. 4, 29, 31]
- [61] Janiszowski K., *Identyfikacja modeli parametrycznych w przykładach*. Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2002. [cytowanie na str. 23]
- [62] Jankovic M., Sepulchre R., Kokotovic P.V., *Constructive Lyapunov Stabilization of Nonlinear Cascade Systems*. IEEE Trans. on Automatic Control, Vol. 41, No. 12, December 1996. 1723–1735. [cytowanie na str. 100]
- [63] Jin L., Nikiforuk P.N., Gupta M.M., *Absolute Stability Conditions for Discrete-Time Recurrent Neural Networks*. IEEE Trans. on Neural Networks, Vol. 5, November 1994. 954–964. [cytowanie na str. 100]
- [64] Jin L., Gupta M.M., Nikiforuk P.N., *Universal Approximation Using Dynamic Recurrent Neural Networks: Discrete-Time Version*. IEEE International Conference on Neural Networks, Perth, WA, Australia, 1995. 403–408. [cytowanie na str. 3]
- [65] Jin L., Gupta M.M., *Stable Dynamic Backpropagation Learning in Recurrent Neural Networks*. IEEE Trans. on Neural Networks, Vol. 10, No. 6, November 1999. 1321–1334. [cytowanie na str. 100]
- [66] Kaczorek T., Dzieliński A., Dąbrowski W., Łopatka R., *Podstawy Teorii Sterowania*. WNT, Warszawa, 2006. [cytowanie na str. 101, 102]
- [67] Kincaid D., Cheney W., *Numerical Analysis*. Brooks/Cole Publishing Company, Pacific Grove, California, USA, 1991. [cytowanie na str. 102]
- [68] Korbicz J., Obuchowicz A., Uciński D., *Sztuczne sieci neuronowe: podstawy i zastosowanie*. Wydawnictwo PLJ, Warszawa, 1994. [cytowanie na str. 3]



- [69] Ku C.C., Lee K.Y., *Diagonal Recurrent Neural Network-Based Control: Convergence and Stability*. Proc of the American Control Conference, Baltimore, Maryland, USA, June 1994. 3340–3345. [cytowanie na str. 8, 101]
- [70] Ku C.C., Lee K.Y., *Diagonal Recurrent Neural Networks for Dynamic Systems Control*. IEEE Trans. on Neural Networks, Vol. 6, No. 1, January 1995. 144–155. [cytowanie na str. 8, 15, 34, 101]
- [71] Kuczyński L., *Technologia leków*. WNT, Warszawa, 1971. [cytowanie na str. 38]
- [72] Kukolij D., *Identification of Complex Systems Based on Neural and Takagi-Sugeno Fuzzy Model*. IEEE Trans. on Systems, Man and Cybernetics – Part B: Cybernetics, Vol. 34, No. 1, February 2004. 272–282. [cytowanie na str. 6]
- [73] Kulikowski R., *Sterowanie w wielkich systemach*. WNT, Warszawa, 1970. [cytowanie na str. 5]
- [74] Kuschewski J.G., Hui S., Żak S.H., *Application of Feedforward Neural Networks to Dynamical System Identification and Control*. IEEE Trans. on Control Systems Technology, Vol. 1, No. 1, 1993. 37–49. [cytowanie na str. 23]
- [75] Kwaśnicka H., *Sieci neuronowe w zastosowaniach :praca zbiorowa pod redakcją Urszuli Markowskiej-Kaczmar, Haliny Kwaśnickiej*. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2005. [cytowanie na str. 8, 27]
- [76] Li L.K., *Approximation Theory and Recurrent Networks*. Proc. of International Joint Conference on Neural Networks, Vol. 2, 1992. 266–271. [cytowanie na str. 8, 14]
- [77] Liang X., *Comments On 'Diagonal Recurrent Neural Networks For Dynamic Systems Control' – Reproof of Theorems 2 and 4*. IEEE Trans. on Neural Networks, Vol. 86, No. 2, May 1997. 811–812. [cytowanie na str. 15, 101]
- [78] Ljung L., *Issue in System Identification*. IEEE Trans. on Control Systems, Vol. 11, January 1991. 25–29. [cytowanie na str. 7]
- [79] Ljung L., Glad T., *Modelling of Dynamic Systems*. Englewood Cliffs: Prentice Hall PTR, 1994. [cytowanie na str. 2]
- [80] Ljung L., *System Identification: Theory for the User*. Upper Saddle River, Prentice Hall PTR, 1998. [cytowanie na str. 19]
- [81] Luyben W.L., *Process Modeling, Simulation, and Control for Chemical Engineers*. McGraw-Hill Publishing Company, 1999. [cytowanie na str. 40, 47]
- [82] Man Z., Wu H.R., Liu S., Yu X. *A New Adaptive Backpropagation Algorithm Based on Lyapunov Stability Theory for Neural Networks*. IEEE Trans. on Neural Networks, Vol. 17, No. 6, November 2006. 1580–1591. [cytowanie na str. 100]

- [83] Mańczak K., *Technika planowania eksperymentu*. WNT, Warszawa, 1976. [cytowanie na str. 21]
- [84] Mańczak K., *Metody identyfikacji wielowymiarowych obiektów sterowania*. WNT, Warszawa, 1979. [cytowanie na str. 19]
- [85] Mańczak K., Nahorski Z., *Komputerowa identyfikacja obiektów dynamicznych*. PWN, Warszawa, 1983. [cytowanie na str. 19]
- [86] Mandic P.D., Chambers J.A., *A Normalised Real Time Recurrent Learning Algorithm*. Elsevier, Signal Processing, Vol. 80, 2000. 1909–1916. [cytowanie na str. 31]
- [87] Mandic P.D., Chambers J.A., *Recurrent Neural Networks for Prediction - learning algorithms, architectures and stability*. A Volume in Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications and Control, John Wiley & Sons Ltd., 2001. [cytowanie na str. 7, 8, 9, 14, 29, 31, 100]
- [88] Marler R.T., Arora J.S., *Survey of Multi-objective Optimization Methods for Engineering*. Journal of Structural and Multidisciplinary Optimization, Springer, Berlin / Heidelberg, Vol. 26, No. 6, April 2004. 369–395. [cytowanie na str. 82]
- [89] Masini R., Padovani E., Ricotti M.E., Zio E., *Dynamic simulation of a steam generator by neural networks*. Nuclear Engineering and Design, Elsevier, 1999. 197–213. [cytowanie na str. 6]
- [90] Masters T., *Sieci neuronowe w praktyce: programowanie w języku C++*. WNT, Warszawa, 1996. [cytowanie na str. 126]
- [91] Narendra K.S., Parthasarathy K., *Identification and Control of Dynamical Systems using Neural Networks*. IEEE Trans. on Neural Networks, Vol. 1, 1990. 4–27. [cytowanie na str. 8]
- [92] Narendra K.S., Parthasarathy K., *Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks*. IEEE Trans. on Neural Networks, Vol. 2, March 1991. 252–261. [cytowanie na str. 8]
- [93] Narendra K.S., *Neural Networks for Control Theory and Practice*. Proceedings of the IEEE, Vol. 84, No. 10, October 1996. 1385–1406. [cytowanie na str. 8]
- [94] Nelles O., *Nonlinear System Identification – From Classical Approaches to Neural Networks and Fuzzy Models*. Springer-Verlag Berlin Heidelberg, 2001. [cytowanie na str. 3, 8, 20, 29]
- [95] Nerrand O., Roussel-Ragot P., Dreyfus G., Personnaz L., Marcos S., *Neural Networks and Non-linear Adaptive Filtering: Unifying Concepts and New Algorithms*. Neural Computation, Vol. 5, No. 2, 1993. 165–197. [cytowanie na str. 8]

- [96] Nocedal J., Wright S.J., *Numerical Optimization*. Springer Science+Business Media, LLC, 2006. [cytowanie na str. 21]
- [97] Osowski S., *Sieci neuronowe w ujęciu algorytmicznym*. WNT, Warszawa, 1997. [cytowanie na str. 4, 8, 27, 107]
- [98] Osowski S., *Modelowanie i symulacja układów i procesów dynamicznych*. Oficyna Wydawnicza Politechniki Warszawskiej, 2007. [cytowanie na str. 123, 124, 147]
- [99] Parlos A.G., Chong K.T., Atiya A.F., *Application of the Recurrent Multilayer Perceptron in Modeling Complex Process Dynamics*. IEEE Trans. on Neural Networks, Vol. 5, No. 2, 1994. 255–266. [cytowanie na str. 6]
- [100] Patan K., *Stability Analysis and the Stabilization of a Class of Discrete-Time Dynamic Neural Networks*. IEEE Trans. on Neural Networks, Vol. 18, No. 3, May 1994. 660–673. [cytowanie na str. 8]
- [101] Patan K., Patan M., *Selection of Training Sequences for Locally Recurrent Neural Network*. XVI Krajowa Konferencja Automatyki, Recent Advances in Control and Automation, Eds. Malinowski K., Rutkowski L., Akademicka Oficyna Wydawnicza EXIT, Szczyrk, Maj 2008. 252–262. [cytowanie na str. 8]
- [102] Pearlmutter B.A., *Gradient Calculations for Dynamic Recurrent Neural Networks: A Survey*. IEEE Trans. on Neural Networks, Vol. 6, No. 5, September 1995. 660–673. [cytowanie na str. 8]
- [103] Piche S.W., *Steepest Descent Algorithms for Neural Network Controllers and Filters*. IEEE Trans. on Neural Networks, Vol. 5, No. 2, March 1994. 198–212. [cytowanie na str. 8]
- [104] Pilgrim K.M., *Combinations of Complex Dynamical Systems*. Lecture Notes in Mathematics, Springer-Verlag Berlin Heidelberg, 2003. 197–213. [cytowanie na str. 7]
- [105] Pingan Z., Renhou L., *A New Approach to Fuzzy Identification for Complex Systems*. Proc. of 5<sup>th</sup> IEEE International Conference on Fuzzy Systems, Vol. 2, September 1996. 1308–1313. [cytowanie na str. 6]
- [106] Polycarpou M.M., Ioannou P.A., *Identification and Control of Nonlinear Systems using Neural Network Models: Design and Stability Analysis*. Technical Report 91-09-01, Dept. Elec. Eng. – Systems, University of Southern California, 1991. [cytowanie na str. 8]
- [107] Polycarpou M.M., Ioannou P.A., *Neural Networks as On-Line Approximators of Nonlinear Systems*. Proc. of 31<sup>st</sup> Conf. on Decision and Control, Tuscon, Arizona, USA, December 1992. 7–12. [cytowanie na str. 3]

- [108] Rumelhart D.C., Hinton G.E., Williams R., *Learning Internal Representation by Error Propagation*. Nature, Vol. 323, 1986. 533–536. [cytowanie na str. 7]
- [109] Rutkowska D., Piliński M., Rutkowski L., *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte*. Wydawnictwo Naukowe PWN, Warszawa, 1993. [cytowanie na str. 3]
- [110] Rutkowski L., *Filtry adaptacyjne i adaptacyjne przetwarzanie sygnałów*. WNT, Warszawa, 1994. [cytowanie na str. 8]
- [111] Seidler J., Badach A., Molisz W., *Metody rozwiązywania zadań optymalizacji*. WNT, Warszawa, 1980. [cytowanie na str. 23]
- [112] Snyman J.A., *Practical Mathematical Optimization – An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*. Springer Science+Business Media, Inc., USA, 2005. [cytowanie na str. 24]
- [113] Söderström T., Stoica P., *Identyfikacja systemów*. Wydawnictwo Naukowe PWN, Warszawa, 1997. [cytowanie na str. 19]
- [114] Spooner J.T., Maggiore M., Ordóñez R., Passino K.M., *Stable Adaptive Control and Estimation for Nonlinear Systems: Neural and Fuzzy Approximation Technique*. John Wiley & Sons, Inc., 2002. [cytowanie na str. 3]
- [115] Srinivasan B., Prasad U.R., Rao N.J., *Back Propagation Through Adjoint for the Identification of Nonlinear Dynamic Systems Using Recurrent Neural Models*. IEEE Trans. on Neural Networks, Vol. 5, No. 2, March 1994. 213–228. [cytowanie na str. 8]
- [116] Suykens J.A.K., Vandewalle J., De Moor B.L.R., *NLq Theory: Checking and Imposing Stability of Recurrent Neural Networks for Nonlinear Modeling*. IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol. 45, No. 11, November 1997. 2682–2691. [cytowanie na str. 100]
- [117] Suykens J.A.K., De Moor B.L.R., Vandewalle J., *Robust Local Stability of Multilayer Recurrent Neural Networks*. IEEE Trans. on Neural Networks, Vol. 11, No. 1, January 2000. 222–229. [cytowanie na str. 100]
- [118] Świątek J., *Model matematyczny i identyfikacja procesów hemodynamicznych w układzie krążenia*. praca doktorska, promotor prof. zw. Dr. Hab. Zdzisław Bubnicki, Politechnika Wrocławska, Wrocław, 1979. [cytowanie na str. 5, 147]
- [119] Świątek J., *Identyfikacja dwustopniowa oraz jej zastosowania techniczne i biomedyczne*. Wydawnictwo Politechniki Wrocławskiej, Wrocław, 1987. [cytowanie na str. 2, 5]

- [120] Świątek J., Drałus G., *A modified back propagation algorithm for modeling static complex systems using neural networks*. Proc. of 5<sup>th</sup> International Conference on Neural Networks and Soft Computing, Zakopane, 2000. 463–468. [cytowanie na str. 6]
- [121] Świątek J., *Global and local modeling of complex input output systems*. Proc. of 16<sup>th</sup> International Conference on Systems Engineering, Coventry University, England, September 2003. 669–671. [cytowanie na str. 3, 37, 38]
- [122] Świątek J., *Globally optimal model of complex systems – identification algorithm for cascade structure*. Proc. of 15<sup>th</sup> International Conference on Systems Science, Wrocław, Vol.1, 2004. 367–376. [cytowanie na str. 5, 6, 47]
- [123] Świątek J., *Identification of complexes of operations system with limited measurement possibilities*. Proc. of 18<sup>th</sup> International Conference on Systems Engineering ICSEng, Las Vegas, USA, August 2005. 124–129. [cytowanie na str. 2]
- [124] Świątek J., *Identyfikacja systemów złożonych*. III Konferencja Naukowa Informatyka w Ekonomii, Zamojskie Studia i Materiały – Informatyka, Zamość, 2006. 25–33. [cytowanie na str. 3, 42, 43, 45, 46]
- [125] Świątek J., *Wybrane zagadnienia identyfikacji statycznych systemów złożonych*. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2009. [cytowanie na str. 123, 147]
- [126] Tadeusiewicz R., *Sieci Neuronowe*. Akademicka Oficyna Wydawnicza RM, Warszawa, 1993. [cytowanie na str. 3, 11, 27]
- [127] Tsoi A.C., Back A.D., *Locally Recurrent Globally Feedforward Networks: A Critical Review of Architectures*. IEEE Trans. on Neural Networks, Vol. 5, No. 2, March 1994. 229–238. [cytowanie na str. 8]
- [128] Venkatesh S.R., Dahleh M.A., *System Identification of Complex Systems: Problem Formulation and Results*. Proc. of 36<sup>th</sup> IEEE Conference on Decision and Control, Vol. 3, December 1997. 2441–2446. [cytowanie na str. 5]
- [129] Venkatesh S.R., Dahleh M.A., *On System Identification of Complex Systems from Finite Data*. IEEE Trans. on Automatic Control, Vol. 46, No. 2, February 2001. 235–257. [cytowanie na str. 5]
- [130] Walter E., Pronzato L., *Identification of Parametric Models from Experimental Data*. Springer-Verlag, Berlin, 1997. [cytowanie na str. 2, 19]
- [131] Wan E., Beaufays F., *Network Reciprocity: a Simple Approach to Derive Gradient Algorithms for Arbitrary Neural Network Structure*. ICNN, San Diego, 1994. [cytowanie na str. 34]

- [132] Wang P., Youfeng L., Feng S., Wei W., *Convergence of Diagonal Recurrent Neural Networks' Learning*. Proc. of 4<sup>th</sup> World Congress on Intelligent Control and Automation, Shanghai, China, June 2002. 2365–2369. [cytowanie na str. 15, 101]
- [133] Wen Y., *Nonlinear System Identification Using Discrete-Time Recurrent Neural Networks With Stable Learning Algorithms*. Elsevier, International Journal of Information Sciences, Vol. 158, No. 2, January 2004. 131–147. [cytowanie na str. 100]
- [134] Werbos P.J., *Neural Networks for Control and System Identification*. Proc. of IEEE International Conference on Decision and Control, Vol. 1, December 1989. 260–265. [cytowanie na str. 8]
- [135] Werbos P.J., *Backpropagation Through Time: What it does and how to do it*. Proc. IEEE, Vol. 78, October 1990. 1550–1560. [cytowanie na str. 8, 31]
- [136] Widrow B., Stearns S.D., *Adaptive Signal Processing*. Prentice-Hall PTR, Upper Saddle River, New Jersey, 1985. [cytowanie na str. 21]
- [137] Williams R.J., Zipser D., *A Learning Algorithm for Continually Running Fully Recurrent Neural Networks*. Neural Computations, Vol. 1, No. 3, 1989. 270–280. [cytowanie na str. 7, 8, 30]
- [138] Wu W., Feng G., Li Z., Xu Y., *Deterministic Convergence of an Online Gradient Method for BP Neural Networks*. IEEE Trans. on Neural Networks, Vol. 16, No. 3, May 2005. 533–540. [cytowanie na str. 24]
- [139] Yang W.Y., Cao W., Morris J., Chung T., *Applied Numerical Methods using MATLAB*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2005. [cytowanie na str. 21]
- [140] Yu D.L., Chang T.K., *Adaptation of Diagonal Recurrent Neural Network*. Neural Computations & Applications, Vol. 14, 2005. 189–197. [cytowanie na str. 8, 15]
- [141] Yu W., Poznyak A.S., Li X., *Multilayer Dynamic Neural Networks for Non-linear System On-line Identification*. International Journal of Control, Vol. 74, No. 18, December 2001. 1858–1864. [cytowanie na str. 21]
- [142] Zeng-Guang H., Gupta M.M., Nikiforuk P.N., Min T., Cheng L., *A Recurrent Neural Network for Hierarchical Control of Interconnected Dynamic Systems*. IEEE Trans. on Neural Networks, Vol. 18, No. 2, March 2007. 466–481. [cytowanie na str. 6]
- [143] Żurada J., Barski M., Jędruch W., *Sztuczne sieci neuronowe: podstawy teorii i zastosowania*. Wydawnictwo Naukowe PWN, Warszawa, 1996. [cytowanie na str. 4, 8]

# Spis symboli i skrótów

Symbol/skrót	Opis
$n$	numer taktu
$u$	sygnał wejściowy
$y$	wyjście obiektu
$\bar{y}$	wyjście modelu
$\mu$	wyjście neuronu
$e$	błąd modelu lokalnego
$\varepsilon$	błąd modelu globalnego
$S$	liczba wejść
$L$	liczba wyjść
$H$	liczba neuronów warstwy ukrytej
$J$	liczba warstw statycznej sieci wielowarstwowej
$w$	parametr modelu/sieci neuronowej
$\bar{w}$	parametr związany z wyjściem modelu
$\tilde{w}$	parametr związany z wejściem modelu
$\hat{w}$	parametr sprzężenia zwrotnego
$\chi$	wyjście części liniowej neuronu
$D, D_{mu}, D_u$	rzędy modelu
$Q$	kryterium/wskaźnik jakości
$\sigma$	funkcja sigmoidalna
$\Phi$	model
$O$	obiekt/element
$\Psi$	algorytm identyfikacji/uczenia
$\eta$	współczynnik uczenia
$z^{-d}$	operator opóźnienia
$\delta$	błąd propagowany wstecz
$\partial^+$	uszeregowana pochodna cząstkowa

Symbol/skrót	Opis
$\nabla$	gradient
$\mathcal{R}$	zbiór liczb rzeczywistych
$M$	liczba elementów systemu złożonego
$x$	wejście zewnętrzne systemu złożonego
$v$	wyróżnione wyjście systemu złożonego
$Q_L$	lokalny wskaźnik jakości
$Q_G$	globalny wskaźnik jakości
$Q_S$	syntetyczny wskaźnik jakości
$Q_Z$	zastępczy wskaźnik jakości
$Q_{ZZ}$	zastępczy wskaźnik jakości zmodyfikowany o funkcję kary zewnętrznej
$Q_{ZW}$	zastępczy wskaźnik jakości zmodyfikowany o funkcję kary wewnętrznej
ML	model lokalnie optymalny
MG	model globalnie optymalny
MS	model ze wskaźnikiem syntetycznym
MZZ	lokalny model zadowalający – kara zewnętrzna
MZW	lokalny model zadowalający – kara wewnętrzna
REL RMSE	unormowany błąd średniokwadratowy
$\beta, \alpha, \lambda$	współczynniki liczbowe, za pośrednictwem których nakładane są wymagania na jakość modeli lokalnych i na jakość modelu globalnego
$g$	funkcja ograniczeń
$P$	funkcja kary zewnętrznej
$B$	funkcja kary wewnętrznej
$c$	parametr związany z funkcjami kar
$V$	funkcja Lapunowa
$\ \cdot\ $	norma euklidesowa
BP	<i>BackPropagation</i>
BPTT	<i>BackPropagation Through Time</i>
DRNN	<i>Dynamic Recurrent Neural Network</i>
LRGF	<i>Locally Recurrent, Globally Feedforward</i>
NARX	<i>Nonlinear AutoRegressive Exogenous</i>
RTRL	<i>Real Time Recurrent Learning</i>
SISO	<i>Single Input Single Output</i>

Wytłuszczone symbole odnoszą się do wektorów i macierzy.



# Spis rysunków

1.1	Liczba artykułów nt. rekurencyjnych sieci neuronowych w czasopiśmie <i>IEEE Trans. on Neural Networks</i> w latach 1988–2008(badania własne)	7
1.2	Stacyjny neuron	10
1.3	Wykres funkcji sigmoidalnej (1.3) i jej pochodnej	10
1.4	Stacyjna sieć wielowarstwowa	12
1.5	Rekurencyjny neuron dla $D_u = D_\mu = 2$	15
1.6	Rekurencyjna sieć DRNN	16
1.7	Ogólny schemat identyfikacji	19
1.8	System identyfikacji	22
1.9	Stacyjna sieć wielowarstwowa modelująca obiekt dynamiczny	24
1.10	Model liniowy z dynamiką wewnętrzną dla $D_u = 3$ i $D_\mu = 2$	28
1.11	Model liniowy rozwinięty wstecz w czasie	32
1.12	Schemat procesu produkcji aspiryny	39
1.13	Przykład systemu złożonego – proces produkcji aspiryny	40
1.14	Przykład systemu złożonego z modelami lokalnymi	43
1.15	Przykład systemu złożonego z modelem globalnym	45
1.16	System złożony o strukturze kaskadowej	47
1.17	Przykład systemu złożonego dającego się sprowadzić do systemu o strukturze kaskadowej	47
2.1	System złożony o strukturze kaskadowej z modelami lokalnymi	51
2.2	System złożony o strukturze kaskadowej z modelem globalnym	57
2.3	Model złożony systemu liniowego rozwinięty wstecz w czasie dla $K = 2$	59
2.4	Złożona sieć DRNN modelująca system o strukturze kaskadowej	64
2.5	Model globalny – przypadek szczególny	81

2.6	System złożony o strukturze kaskadowej ze szczególnym przypadkiem modelu globalnie optymalnego z uwzględnieniem jakości modeli lokalnych . . . . .	83
2.7	System złożony o strukturze kaskadowej z modelem globalnie optymalnym z uwzględnieniem jakości modeli lokalnych . . . . .	84
3.1	Złożona sieć DRNN SISO . . . . .	101
4.1	Liniowy system złożony . . . . .	122
4.2	Reaktor chemiczny . . . . .	124
4.3	Model reaktora chemicznego . . . . .	125
4.4	Sygnal wejściowy $u^{(1)}$ pierwszego elementu . . . . .	127
4.5	Sygnal wyjściowy $y^{(1)} = u^{(2)}$ pierwszego elementu . . . . .	127
4.6	Sygnal wyjściowy $y^{(2)}$ drugiego elementu . . . . .	127
4.7	Uczenie – błąd $e^{(1)}$ na wyjściu modelu pierwszego elementu . . . . .	128
4.8	Uczenie – błąd $e^{(2)}$ na wyjściu modelu drugiego elementu . . . . .	128
4.9	Uczenie – błąd $\varepsilon^{(1)}$ na wyjściu modelu pierwszego elementu . . . . .	128
4.10	Uczenie – błąd $\varepsilon^{(2)}$ na wyjściu modelu drugiego elementu . . . . .	129
4.11	Uczenie – przebieg zmian parametrów modelu lokalnego pierwszego elementu . . . . .	129
4.12	Uczenie – przebieg zmian parametrów modelu lokalnego drugiego elementu . . . . .	129
4.13	Uczenie – przebieg zmian parametrów modelu pierwszego elementu w modelu globalnym . . . . .	130
4.14	Uczenie – przebieg zmian parametrów modelu drugiego elementu w modelu globalnym . . . . .	130
4.15	Testowanie modelu lokalnego po złożeniu – wyjście pierwszego elementu	130
4.16	Testowanie modelu lokalnego po złożeniu – wyjście drugiego elementu	131
4.17	Testowanie modelu globalnego po złożeniu – wyjście pierwszego elementu . . . . .	131
4.18	Testowanie modelu globalnego po złożeniu – wyjście drugiego elementu	131
4.19	Porównanie algorytmów (wskaźniki REL RMSE z tab. 4.4) . . . . .	134
4.20	Dane uczące – sygnały wejściowe $u_1^{(1)}$ i $u_2^{(1)}$ pierwszego elementu . . .	135
4.21	Dane uczące – sygnały wyjściowe $y_1^{(1)} = u_1^{(2)}$ i $y_2^{(1)} = u_2^{(2)}$ drugiego elementu . . . . .	135
4.22	Dane uczące – sygnał wyjściowy $y^{(2)}$ drugiego elementu . . . . .	136

4.23	Dane testujące – sygnały wejściowe $u_1^{(1)}$ i $u_2^{(1)}$ pierwszego elementu . . .	136
4.24	Dane testujące – sygnały wyjściowe $y_1^{(1)} = u_1^{(2)}$ i $y_2^{(1)} = u_2^{(2)}$ drugiego elementu . . . . .	136
4.25	Dane testujące – sygnał wyjściowy $y^{(2)}$ drugiego elementu . . . . .	137
4.26	Testowanie sieci złożonej lokalnie optymalnej – wyjście $\bar{y}_1^{(1)}$ . . . . .	137
4.27	Testowanie sieci złożonej lokalnie optymalnej – wyjście $\bar{y}_2^{(1)}$ . . . . .	137
4.28	Testowanie sieci złożonej lokalnie optymalnej – wyjście $\bar{y}^{(2)}$ . . . . .	138
4.29	Testowanie sieci złożonej globalnie optymalnej – wyjście $\bar{y}_1^{(1)}$ . . . . .	138
4.30	Testowanie sieci złożonej globalnie optymalnej – wyjście $\bar{y}_2^{(1)}$ . . . . .	138
4.31	Testowanie sieci złożonej globalnie optymalnej – wyjście $\bar{y}^{(2)}$ . . . . .	139
4.32	Porównanie algorytmów (wskaźniki REL RMSE z tab. 4.9) . . . . .	141

# Spis tabel

- 4.1 Wartości wskaźników faktycznie optymalizowanych w trakcie uczenia . 132
- 4.2 Wskaźniki lokalne i wskaźnik globalny dla danych testowych przy  $K = 0132$
- 4.3 Wskaźniki lokalne i wskaźnik globalny dla danych testowych przy  $K = 2133$
- 4.4 Wskaźniki lokalne i wskaźnik globalny dla danych testowych przy  $K = 4133$
- 4.5 Wskaźniki lokalne i wskaźnik globalny dla danych testowych przy  $K = 8133$
- 4.6 Wartości wskaźników faktycznie optymalizowanych w trakcie uczenia . 139
- 4.7 Wskaźniki lokalne i wskaźnik globalny dla danych testowych przy  $K = 0140$
- 4.8 Wskaźniki lokalne i wskaźnik globalny dla danych testowych przy  $K = 2140$
- 4.9 Wskaźniki lokalne i wskaźnik globalny dla danych testowych przy  $K = 4140$
- 4.10 Wskaźniki lokalne i wskaźnik globalny dla danych testowych przy  $K = 8141$

# Skorowidz

- adaptacja, 6
- algorytm identyfikacji, 94, 98
- algorytm adaptacji, 8
- algorytm identyfikacji, 20, 21, 33, 44, 46, 48, 49, 60, 82, 104
- algorytm numeryczny, 21, 30, 100
- algorytm sterowania, 2
- algorytm uczenia, 6, 8, 21, 24, 26, 29, 34, 36, 48, 53, 56, 102
- algorytmy identyfikacji, 42
- analiza systemowa, 38
- analiza zbieżności, 100, 101
- aproksymacja, 7, 31
- azotyn amonu, 6
  
- błąd modelowania, 4
- bariera, 89
- BP, 27, 33
- BPTT, 29, 31, 33, 36, 56, 79
  
- charakterystyka dynamiczna, 7
- człon opóźniający, 27
  
- dane pomiarowe, 4, 19–21, 24, 27, 34, 47, 49, 89, 94, 122
- diagnostyka, 2
- diagonalna rekurencyjna sieć neuronowa, 8
- DRNN, 8, 14, 17, 18, 22, 34, 36, 47, 48, 50, 56, 63, 78, 79, 94, 99, 103
- dwupoziomowy system sterowania, 44
- dynamiczny system złożony, 6, 7, 47–49, 78, 85, 94, 99
  
- eksperyment, 1, 19
- eksperyment aktywny, 19
- element wejściowo-wyjściowy, 37
  
- falki, 3
- filtr, 142
- filtr adaptacyjny, 8
- filtr o nieskończonej odpowiedzi impulsowej, 8
- funkcja kary, 82, 88, 89, 94, 99
- funkcja Lapunowa, 101, 104, 108, 111, 118
- funkcja liniowa, 45
- funkcja przejścia, 4
- funkcja wektorowa, 13, 18
- funkcja wielomodalna, 21, 24
  
- globalne kryterium jakości, 43
- globalne pochodne cząstkowe, 63, 66, 90, 95
- globalny wskaźnik jakości, 3, 43, 46, 58, 64, 82, 88, 94, 99
- gradient, 8, 24, 28, 33, 34, 53, 60, 64, 83, 90, 95, 103
- graf obliczeń, 11
  
- identyfikacja, 1, 8, 18, 21
- identyfikacja dynamicznych systemów złożonych, 49
- identyfikacja globalna, 2, 5, 6, 44, 57, 62, 63, 78, 81, 85, 94, 98, 106
- identyfikacja lokalna, 42, 53
- identyfikacja na bieżąco, 21, 41, 43, 53
- identyfikacja parametryczna, 6
- identyfikacja systemów dynamicznych, 7

- identyfikacja systemu złożonego, 42, 44, 120
- identyfikacja wielostopniowa, 2
- iloczyn Hadamarda, 18
- interpolacja, 4
- kara wewnętrzna, 82, 89, 94, 99, 118
- kara zewnętrzna, 82, 89, 94, 111, 113
- klasa modelu, 19–21
- komputerowy system sterowania, 49
- koordynator, 7
- kryterium jakości, 19–21, 34, 41, 53, 88
- linia opóźniająca, 14, 23
- lokalne kryterium jakości, 41, 42
- lokalne pochodne cząstkowe, 63, 65, 90, 95
- lokalnie optymalne parametry, 42
- lokalny układ regulacji, 44
- lokalny wskaźnik jakości, 2, 82, 88
- LRGF, 8
- macierz zerojedynkowa, 38
- maszyna Boltzmana, 8
- metoda aproksymacji stochastycznej, 8
- metoda gradientu prostego, 24, 28, 34, 47, 48, 53, 60, 64, 83, 89, 100
- metoda Lapunowa, 100
- metoda numeryczna, 48
- metoda wewnętrznej funkcji kary, 89
- metoda wstecznej propagacji błędów, 27
- metoda wstecznej propagacji błędów w czasie, 29, 33
- metoda zewnętrznej funkcji kary, 89
- miara, 42, 44, 53
- minimalizacja, 46
- model dyskretny, 19, 106
- model globalnie optymalny, 3, 5, 44, 46, 49, 56, 78
- model globalnie optymalny z uwzględnieniem jakości modeli lokalnych, 5, 49
- model globalny, 22, 45, 46, 56, 63, 81, 88, 120
- model klasyczny, 6, 7, 21, 27, 58
- model liniowy, 6, 21, 27, 58
- model lokalnie optymalny, 3, 5, 42, 44, 49, 50, 56
- model lokalny, 45, 63, 81, 88, 94, 99
- model neuronowy, 4, 11, 14, 15, 18, 22, 24, 34, 51, 65, 78
- model nieliniowy, 24
- model rozmyto-neuronowy, 3
- model rozmyty, 3, 6
- model rozwinięty wstecz w czasie, 31, 33, 59
- model statyczny, 24
- model systemu złożonego, 45, 46, 48
- model Takagi-Sugeno, 6
- model wielomianowy, 3
- model z dynamiką wewnętrzną, 14, 20, 22, 27, 31, 41, 60
- model z dynamiką zewnętrzną, 20, 22, 23
- modele lokalne, 46
- modelowanie globalne, 5
- modelowanie lokalne, 5
- modelowanie matematyczne, 1
- NARX, 8, 14
- neuron, 7, 11
- neuron liniowy, 14
- neuron rekurencyjny, 14
- neuron statyczny, 9
- norma, 20
- norma Euklidesowa, 20
- numeryczna metoda optymalizacji, 4, 24
- numeryczny algorytm identyfikacji, 21, 30
- obiekt dynamiczny, 3, 5, 6, 21–23, 27, 30, 34, 48
- obiekt identyfikacji, 21
- obiekt nieliniowy, 3, 6
- obiekt SISO, 27, 30, 58
- obiekt stacjonarny, 21, 24
- obiekt wejściowo-wyjściowy, 3, 5, 18
- ocena globalna, 3, 6
- ocena lokalna, 3, 6
- ocena wielokryterialna, 6, 45
- odwrócone wahadła, 6
- odwzorowanie wektorowe, 9, 11, 13, 16, 17
- odwzorowanie zwięzające, 100
- ograniczenia, 88, 89, 94
- operator opóźnienia, 14, 23

- optymalizacja, 2, 47, 88  
 optymalizacja z ograniczeniami, 89
- parametr swobodny, 10, 14, 16, 51  
 parametry modelu, 7, 20, 21, 27, 29, 38, 88  
 parametry sieci, 24, 34, 47, 90, 95, 100  
 persistence of excitation, 21  
 pochodne cząstkowe, 24, 29, 31, 32, 34, 36,  
 53–55, 60, 65, 77, 84  
 predykcja, 2  
 proces hemodynamiczny, 5  
 programowanie dynamiczne, 6
- równanie różnicowe, 9, 14, 15, 27, 122  
 równanie różniczkowe, 124  
 reaktor chemiczny, 122, 123  
 reguła delta, 27  
 rekurencyjna sieć neuronowa, 5–8  
 rekurencyjne uczenie w czasie rzeczywistym,  
 29  
 rozwiązanie analityczne, 6  
 rozwiązanie dopuszczalne, 89  
 RTRL, 29, 33, 34  
 rząd neuronu, 34
- schemat produkcji aspiryny, 38  
 sekwencja ucząca, 8  
 seria identyfikująca, 19, 21, 44  
 seria pomiarowa, 20, 21, 41, 42  
 sieć BAM, 8  
 sieć Elmana, 8  
 sieć globalna, 6, 47, 52, 63, 78, 79, 94, 99,  
 101, 104, 108, 113  
 sieć Hopfielda, 8  
 sieć jednokierunkowa, 6  
 sieć Jordana, 8  
 sieć lokalna, 52, 103, 104  
 sieć lokalnie rekurencyjna, globalnie jedno-  
 kierunkowa, 8  
 sieć neuronowa, 3, 6, 13, 21, 23, 47, 49  
 sieć neuronowo-rozmyta, 6  
 sieć rekurencyjna, 4, 16, 22, 27, 47–49  
 sieć samoorganizująca się, 8  
 sieć statyczna, 13, 16, 21, 22, 26, 48
- sieć wielowarstwowa, 5, 6, 11, 13, 16, 21, 22,  
 26  
 sigmoida, 9  
 SOM, 8  
 sprzężenie zwrotne, 4, 5, 8, 30, 31, 47, 66  
 stabilność, 30  
 sterowanie, 5–8  
 sterowanie adaptacyjne, 8  
 struktura dwupoziomowa, 7  
 struktura hierarchiczna, 7  
 struktura kaskadowa, 4–6, 47, 49, 50, 56, 63,  
 78, 81, 82, 85, 88, 94, 99  
 struktura systemu złożonego, 3, 37, 38, 40  
 symulacja, 2  
 syntetyczny wskaźnik jakości, 45, 46, 49, 82  
 system dynamiczny, 8  
 system Hammersteina, 6  
 system identyfikacji, 21  
 system nieliniowy, 6  
 system statyczny, 6  
 system sterowania nadrzędnego, 44  
 system wejściowo-wyjściowy, 7  
 system Wienera, 6  
 system złożony, 2, 5, 22, 36, 37, 39–41, 47  
 szereg Taylora, 102, 104  
 szeregi Fouriera, 3
- uczenie, 3, 8, 21, 47, 49, 91, 96  
 układ krążenia, 5  
 uszeregowane pochodne cząstkowe, 31, 33,  
 35, 36, 49, 54, 55, 60, 61, 77  
 uszeregowany układ równań, 31
- wagi neuronu, 9  
 wagi sieci, 8, 17, 24, 52, 64, 66  
 warstwa sieci, 7, 11, 13  
 warstwa ukryta, 11, 14, 17, 26, 34, 51, 52  
 warstwa wyjściowa, 11, 14, 17, 25, 34, 36,  
 51–53, 55  
 warunek Lipschitza, 9  
 warunek zbieżności, 103, 106, 110, 115, 117,  
 119, 120  
 wejścia zewnętrzne, 37, 39, 45, 57, 82  
 wielopoziomowy system sterowania, 2

wrażliwość, 108, 120  
wskaźnik globalny, 46, 106  
wskaźnik jakości, 2, 19, 20, 24, 27, 34, 89,  
94, 99, 101  
wskaźnik kwadratowy, 24, 27, 34  
wskaźnik syntetyczny, 46, 108, 110  
współczynnik uczenia, 4, 24, 30, 53, 123  
wybór optymalnego modelu, 4, 6  
wyjścia globalne, 37–39, 45, 57, 58, 81  
  
zbieżność, 5, 24, 48, 49, 106  
zbieżność lokalna, 21, 100  
zmodyfikowany wskaźnik jakości, 89, 94