



AGH

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

DZIEDZINA NAUK INŻYNIERYJNO-TECHNICZNYCH

DYSCYPLINA INŻYNIERIA MECHANICZNA

ROZPRAWA DOKTORSKA

Klasyfikacja odpowiedzi układu dyskretnego oraz klasyfikacja odpowiedzi układu ciągłego na losowe serie impulsów o małej intensywności — podejście oparte na danych eksperymentalnych i głębokich sieciach konwolucyjnych typu VGG16/VGG19

Autor: mgr inż. Marek Sulewski

Promotor rozprawy: dr hab. inż. Agnieszka Ozga, prof. AGH
Promotor pomocniczy: -

Praca wykonana: Wydział Inżynierii Mechanicznej i Robotyki

Kraków, 2025

Z całego serca dziękuję mojej Rodzinie za nieustanne wsparcie podczas powstawania tej rozprawy. Dziękuję za cierpliwość, wyrozumiałość i wiarę w powodzenie, zwłaszcza gdy obowiązki naukowe wymagały dodatkowego czasu i wysiłku. Ten sukces jest wspólny — dziękuję.

Abstract

The aim of the conducted research was the classification of the responses of a discrete system and a continuous system to a random series of impulses of known magnitude η_i . Deep convolutional networks were employed for classification, analyzing data obtained experimentally. The dissertation combines issues from the fields of vibration theory, mathematical modeling of random phenomena, stochastic mechanics, time series analysis, statistics, electromechanical analogies, and deep neural networks. In the first part of the thesis, the research context is presented along with a review of the literature, covering both the problem of system responses to random series of impulses and deep learning methods applied in signal analysis. Subsequently, a mathematical model of a damped oscillator excited by a random series of impulses was introduced. The equation of motion was formulated, and the random variables defining the impulse series were described. Experiments were carried out on the discrete system. Using the NI USB 6251 card, excitations in the form of triangular impulses were generated, approximating the Dirac delta. Data acquisition of the discrete system responses was performed, along with a procedure for extracting single responses from time series. Parameter estimation revealed that accurate reproduction of the responses is possible only for small impulse magnitudes, while further analysis showed significant variability of motion parameters (damping, frequency, amplitude, phase, and offset) across different impulse distributions. The process of data set preparation was then described, classes were defined, and the results of training and testing for the discrete system were presented. The experiments were limited to two impulse magnitudes, which allowed the inclusion of rare events in which a subsequent impulse appeared before the response to the previous one had fully decayed. Detecting such events made it possible to assess whether the course of the process changed. The results demonstrated that, for the discrete system, the VGG16 and VGG19 networks achieved very high accuracy, exceeding 99.9%. Further studies addressed the continuous system, modeled as an FR-4 composite beam with a piezoelectric actuator. In this part, ten different impulse magnitudes exciting the system vibrations were considered. The experimental setup, the method of generating images, and the classification procedure were described. Several distributions were chosen for training the classifier, while the testing phase involved distributions not present in the training and validation sets. The classification results for the continuous system showed that the VGG16 network maintained high effectiveness and stability (93.51% in training and 95.42% in testing), whereas VGG19, despite slightly better training accuracy (96.27%), exhibited clear difficulties in generalizing to new cases, achieving only 88.86% in testing. The greatest challenges concerned classes with overlapping features. Therefore, a new classifier was trained on an extended training and validation set. This resulted in an improvement in training accuracy (96.70%) and also enhanced test performance 93.14%. The conducted research revealed the limitations of classical stochastic models. In practice,

the responses of both the discrete and continuous systems exhibited variability between trials, dependence on the probability of impulse occurrence, as well as nonlinear damping behavior at higher excitation levels. These phenomena indicate that constructing an mathematical model would require incorporating complex stochastic and nonlinear effects. The application of artificial intelligence methods in stochastic mechanics provides a more effective approach, enabling the classification of system responses based on empirical data without the necessity of constructing a complex mathematical model. The proposed approach proved effective in classifying the responses of both discrete and continuous systems to random impulse series. The results of the dissertation demonstrate potential practical applications in technical diagnostics, structural health monitoring, and the study of random phenomena in dynamic systems.

Streszczenie

Celem przeprowadzonych badań była klasyfikacja odpowiedzi układu dyskretnego oraz odpowiedzi układu ciągłego na losową serię impulsów o znanej wielkości η_i . Do klasyfikacji wykorzystano głębokie sieci konwolucyjne, które analizowały dane uzyskane w drodze eksperymentu. Praca łączy zagadnienia z zakresu teorii drgań, modelowania matematycznego zjawisk losowych, mechaniki stochastycznej, analizy szeregów czasowych, statystyki, analogii elektromechanicznych oraz głębokich sieci neuronowych. W pierwszej części rozprawy przedstawiono kontekst badań oraz dokonano przeglądu literatury obejmujący zarówno problematykę odpowiedzi na losowe serie impulsów, jak i metody głębokiego uczenia, które używane są w analizie sygnałów. Następnie przedstawiono model matematyczny oscylatora tłumionego, którego drgania wymuszane są losową serią impulsów. Zapisano równanie ruchu i opisano zmienne losowe definiujące losową serię impulsów. Zrealizowano eksperymenty dla układu dyskretnego. Przy zastosowaniu karty NI USB 6251 wygenerowano wymuszenia w postaci impulsów trójkątnych, będących przybliżeniem delty Diraca. Przeprowadzono akwizycję odpowiedzi układu dyskretnego oraz procedurę wyodrębniania pojedynczych odpowiedzi z szeregów czasowych. Estymacja parametrów wykazała, że odwzorowanie odpowiedzi jest możliwe jedynie dla małych wielkości impulsów, natomiast dalsza analiza ujawniła istotną zmienność parametrów ruchu (tłumienia, częstości, amplitudy, fazy oraz składowej stałej) dla różnych rozkładów impulsów. Kolejno opisano proces przygotowania zbioru danych, zdefiniowano klasy oraz przedstawiono wyniki treningu i testów dla układu dyskretnego. Eksperymenty ograniczono do dwóch wartości impulsów, co pozwoliło uwzględnić w badaniach zdarzenia, w których kolejny impuls pojawia się zanim odpowiedź na poprzedni wygaśnie. Wykrywanie tego rodzaju zdarzeń umożliwia ocenę, czy przebieg procesu ulega zmianie. Uzyskane wyniki badań wykazały, że dla układu dyskretnego sieci VGG16 i VGG19 osiągnęły bardzo wysoką dokładność, przekraczającą 99,9%. Przedstawiono również badania układu ciągłego, którym była belka kompozytowa FR-4 z aktuatorem piezoelektrycznym. W analizach uwzględniono dziesięć różnych wartości impulsów wzbudzających drgania układu. Opisano stanowisko badawcze, sposób generowania obrazów oraz przeprowadzono klasyfikację odpowiedzi. Do uczenia klasyfikatora wybrano kilka rozkładów, a do testów klasyfikatora wykorzystano rozkłady, które nie były w zbiorze treningowym i walidacyjnym. Jak pokazują wyniki klasyfikacji dla układu ciągłego, sieć VGG16 zachowała wysoką skuteczność i stabilność (93.51% w treningu i 95.42% w testach), natomiast VGG19, mimo nieco lepszej dokładności treningowej (96.27%), wyraźnie miała problemy z generalizacją nowych przypadków uzyskując w testach jedynie 88.86%. Największe trudności dotyczyły klas o cechach podobnych. Z tego względu podjęto decyzję o wytrenowaniu nowego klasyfikatora na rozszerzonym zbiorze treningowym i walidacyjnym. Uzyskano poprawę w dokładności treningowej (96.70%) oraz poprawił się wynik w testach sieci, gdzie

uzyskano wynik 93.14%.

Przeprowadzone badania ujawniły ograniczenia klasycznych modeli stochastycznych. W praktyce odpowiedzi badanego układu dyskretnego oraz odpowiedzi układu ciągłego wykazywały zmienność między próbami, zależność od prawdopodobieństwa wystąpienia impulsów, a także nieliniowy charakter tłumienia przy wyższych wartościach wymuszeń. Zjawiska te wskazują, że budowa modelu matematycznego wymagałaby uwzględnienia złożonych efektów stochastycznych i nieliniowych. Zastosowanie metod sztucznej inteligencji w mechanice stochastycznej stanowi skuteczne podejście, umożliwiające klasyfikację odpowiedzi układu w oparciu o dane empiryczne, bez konieczności konstruowania złożonego modelu matematycznego. Opracowane podejście okazało się skuteczne w klasyfikacji odpowiedzi układu dyskretnego oraz układu ciągłego na losową serię impulsów. Wyniki rozprawy mają potencjał praktycznego zastosowania w diagnostyce technicznej, monitorowaniu stanu konstrukcji oraz w badaniach nad zjawiskami losowymi w układach dynamicznych.

Spis treści

Spis ważniejszych oznaczeń	4
1 Wstęp	6
2 Cel i zakres pracy	11
2.1 Cel pracy	11
2.2 Zakres pracy	12
2.3 Dodatkowe założenia	13
2.4 Uwarunkowania doboru parametrów w eksperymentach	13
3 Model matematyczny drgań oscylatora pod wpływem losowej serii impulsów	15
3.1 Równanie ruchu drgań harmonicznym tłumionych wymuszonych losową serią impulsów	15
3.2 Opis zmiennych losowych występujących w funkcji $F(t)$	16
3.3 Odpowiedź oscylatora na wymuszenie losową serią impulsów	17
4 Eksperymentalna realizacja modelu drgań oscylatora pod wpływem losowej serii impulsów z wykorzystaniem układu RLC	19
4.1 Analogie elektromechaniczne	19
4.2 Realizacja doświadczenia z układem RLC	20
4.3 Realizacja delty Diraca	24
4.4 Wyodrębnienie pojedynczych odpowiedzi z zarejestrowanego sygnału	25
5 Estymacja parametrów układu	28
5.1 Wyznaczenie wartości średniej dla odpowiedzi układu RLC na zadane wymuszenie	28
5.2 Wyznaczenie czasu propagacji	29
5.3 Zgrubna identyfikacja parametrów	30
5.4 Precyzyjna identyfikacja parametrów	32
5.5 Szacowanie parametrów równania dla wymuszenia 200 mV	33
5.6 Szacowanie parametrów równania dla wartości wymuszeń 100 mV i 300 mV	37
6 Architektura VGG16 i VGG19	41
6.1 Schemat działania sieci	43
6.2 Opis warstw modelu	44
6.2.1 Sieć VGG16	45

6.2.2	Sieć VGG19	46
6.3	Ocena skuteczności klasyfikacji	47
6.4	Obraz wejściowy i wstępne przetwarzanie przykładowego obrazu	48
6.4.1	Mapy cech dla wczesnych i głębokich warstw	48
6.4.2	Przepływ obrazu w sieci	50
6.4.3	Grad-CAM – wizualizacja istotności	51
6.4.4	Wyniki klasyfikacji Top-5	52
7	Rozpoznanie odpowiedzi układu na pojedynczy impuls oraz odpowiedzi układu na dwa impulsy następujące po sobie dla $\Delta t < 200\mu s$	55
7.1	Opis analizowanych przypadków	55
7.2	Przygotowanie zbioru danych	56
7.2.1	Definicja klas mieszanych (AA, BB, AB/BA)	59
7.3	Definicja bazy danych do sieci	63
7.4	Generowanie klas	68
7.5	Przygotowanie klasyfikatora do treningu	72
7.6	Trening klasyfikatora z wykorzystaniem danych dla układu RLC	73
7.6.1	Sieć VGG16	74
7.6.2	Sieć VGG19	76
7.6.3	Porównanie wyników VGG16 i VGG19	78
7.7	Skuteczności klasyfikatora na próbach spoza zbioru walidacyjnego	78
7.7.1	Wyniki dla sieci VGG16	79
7.7.2	Wyniki dla sieci VGG19	80
7.7.3	Podsumowanie wyników testów klasyfikatora	81
8	Klasyfikacja odpowiedzi belki kompozytowej FR-4 z piezoelektrykiem Macro Fiber Composite™ na losową serię impulsów	83
8.1	Stanowisko pomiarowe	85
8.2	Opis stanowiska badawczego	85
8.3	Przygotowanie zbioru danych	87
8.3.1	Definicja klas i prób pomiarowych	91
8.3.2	Definicja bazy danych dla sieci	92
8.3.3	Generowanie obrazów dla sieci z danych eksperymentalnych	93
8.4	Sieć VGG16 – proces uczenia oraz klasyfikacji danych spoza zbioru walidacyjnego	96
8.5	Sieć VGG19 – proces uczenia oraz klasyfikacji danych spoza zbioru walidacyjnego	99
8.6	Sieć VGG19 – proces uczenia na większym zbiorze treningowym i walidacyjnym oraz porównanie klasyfikacji na danych spoza zbioru walidacyjnego	101
9	Podsumowanie i wnioski końcowe	105
9.1	Najważniejsze wnioski	106
9.2	Kierunki dalszych badań	107

Spis rysunków	115
Spis tabel	118

Spis ważniejszych oznaczeń

Modelowanie zjawisk losowych

η_i – wielkość i -tego impulsu

λ – intensywność występowania impulsów (1/s)

$p(\eta_i)$ – prawdopodobieństwo wystąpienia impulsu (-)

$F(t)$ – ciąg impulsów o losowej wielkości η_i w chwilach losowych t_i

t_i^- – chwila bezpośrednio przed wystąpieniem impulsu

t_i^+ – chwila bezpośrednio po wystąpieniu impulsu

δ – delta Diraca

$t(\tau)$ – funkcja gęstości prawdopodobieństwa dla przedziałów czasowych między kolejnymi impulsami

$A(t), B(t)$ – funkcje pomocnicze

$x(t)$ – odpowiedź układu na wymuszenie

Badania eksperymentalne

$f(t_i)$ – wartość sygnału z eksperymentu w chwili t_i

$\hat{f}(t_i)$ – wartość sygnału uzyskana na podstawie modelu z wyestymowanymi parametrami w chwili t_i

f_s – częstotliwość próbkowania sygnału (Hz)

$a = \eta/c$ – amplituda (-)

b – współczynnik tłumienia (-)

ω – częstość drgań tłumionych (rad/s)

Φ – przesunięcie fazowe odpowiedzi (rad)

C – składowa stała sygnału (V)

V – błąd względny estymacji punktowej (-)

RP – względna precyzja oszacowania (-)

$\hat{s}$ – odchylenie standardowe z próby

$\bar{m}$ – średnia arytmetyczna z próby

n – liczebność próby (-)

$u_{(1-\alpha/2)}$ – kwantyl rozkładu normalnego dla $\alpha = 0.05$ (-)

E_{abs} – błąd bezwzględny (-)

E_{rel} – błąd względny (-)

MSE – błąd średniokwadratowy (-)

E_{max} – maksymalny błąd (-)

Rozdział 1

Wstęp

Historia modelowania zjawisk fizycznych to stopniowe przejście od opisów jakościowych do sformalizowanych, jednoznacznych ujęć matematycznych. Fundamenty pod ten proces położyły prace klasyków: od uporządkowania rachunku i teorii równań różniczkowych przez Cauchy’ego [32], przez wkład Poissona w mechanikę teoretyczną i metody estymacji [10], po wizję Kelvina, który świadomie budował analogie między elektrycznością, ciepłem i mechaniką ośrodków ciągłych, dążąc do jednego języka opisu tych zjawisk [66, 38].

Jednym z istotnych narzędzi, które umożliwiły modelowanie szybko zmieniających się w czasie zjawisk stała się delta Diraca, wprowadzona do mechaniki kwantowej w dziele *The Principles of Quantum Mechanics* [12]. Na gruncie inżynierii i matematyki stosowanej delta Diraca została szeroko adoptowana do opisu wymuszeń o charakterze impulsowym, zarówno w układach elektrycznych [84, 18, 7, 61], jak i mechanicznych [53, 54, 20, 45, 6, 72, 48, 15]. Współczesne opracowania wskazują połączenia delty Diraca z jej pochodnymi — od ujęć formalnych [47] po przypadki graniczne w zastosowaniach inżynierskich [8, 61]. Przykłady konstrukcji modeli z koncentracją masy/siły w mechanice belek i płyt, czy uogólnionych warunków brzegowych, ilustrują np. prace dotyczące belek Timoshenki z masami skupionymi [29]. Z kolei Chatterjee i in. [7] pokazali praktyczne aspekty wyznaczania odpowiedzi impulsowej przy losowym wzbudzeniu, a Rivera i in. [51] zwrócili uwagę na korzyści płynące z „wygładzania” funkcji Heaviside’a i delty w równaniach przewodnictwa ciepła.

Wymuszenie impulsowe można traktować jako strumień zdarzeń pojawiających się w losowych chwilach czasu. Takie ujęcie pozwala opisać sytuacje, w których układ determinuje krótkotrwałe zdarzenia. Istotny jest tutaj sposób losowania chwil pojawienia się impulsów (np. proces Poissona o pewnej intensywności). Ten schemat stoi u podstaw prac Roberta o odpowiedzi układów liniowych na losowe impulsy [53, 52] i jest rozwijany w monografii Roberta i Spanosa [54]. W układach liniowych odpowiedź na pojedynczy impuls jest w pełni zdeterminowana przez odpowiedź układu, a odpowiedź na serię impulsów powstaje przez ich „składanie” w czasie. W praktyce oznacza to, że statystyki odpowiedzi (średnia, wariancja, funkcja autokorelacji) można wyrazić bezpośrednio przez własności strumienia impulsów i charakterystykę dynamiczną układu. Jednym z najbardziej pouczających zjawisk w tym kontekście jest *rezonans*. Jeżeli impulsy pojawiają się losowo, ale układ jest tłumiony, to część energii pobudzenia kumuluje się w pobliżu częstości

własnej i prowadzi do wzrostu wariancji odpowiedzi. Roberts [52] oraz Roberts i Spanos [54] wykazali, że zjawisko to jest trwałe.

Założenie całkowitej niezależności impulsów bywa zbyt silne. W wielu sytuacjach zdarzenia mają skłonność do „nakładania się” (wysoka częstość następujących po sobie uderzeń), do repulsji (zwiększone odstępy) lub wykazują okresowe wzorce. W takich przypadkach statystyki odpowiedzi zależą nie tylko od intensywności i rozkładu amplitud. Badania Iwankiewicza i Sobaczyka [23] wprost uwzględniają ten efekt, pokazując jak korelacje w szeregu impulsów modyfikują widoczne wzbudzenia i odpowiedzi badanego ośrodka. Iwankiewicz [20] zaproponował rozpatrywanie wymuszenia niepunktowego o rozkładzie przestrzennym zmiennym od zdarzenia do zdarzenia. W konsekwencji odpowiedzi wymaga opisu modalnego, a wzbudzone mody zależą od geometrii impulsu i jego lokalizacji. Mazur-Śniady i Śniady [42] pokazali, że strumienie impulsów o dowolnych kształtach w czasie i przestrzeni lepiej oddają rzeczywistość, np. w konstrukcjach mostowych, cienkościennych czy wibroakustyce.

W przypadku nieliniowości zanika prosta addytywność odpowiedzi na poszczególne impulsy. Zmienia to charakter statystyk i może prowadzić do efektów niedostępnych w modelach liniowych (np. do asymetrycznych rozkładów odpowiedzi. Zaproponowany przez Iwankiewicza i Nielsena [22] formalizm całkowania po ścieżkach dla wymuszeń impulsowych generowanych przez procesy odnowy umożliwia śledzenie ewolucji gęstości prawdopodobieństwa oraz wyznaczanie charakterystyk drugiego rzędu. Komplementarne ujęcie stanowią metody aproksymacyjne [68, 67], które dają praktyczne recepty na przybliżenie układu nieliniowego przez „najlepiej dopasowany” układ liniowy w sensie statystycznym. Istotny wkład w badania zjawisk nieliniowych z wymuszeniem impulsowym wniosły prace Tylikowskiego i Marowskiego [76] oraz Skalmierskiego i Tylikowskiego [64], w których pokazano, że rozkład odstępów czasowych między impulsami (np. Poissona) oraz parametry nieliniowości kształtują statystykę odpowiedzi i warunki osiągnięcia stanu stacjonarnego.

Warto również podkreślić znaczenie metod obliczeniowych i eksperymentalnych: Iwankiewicz [21] przedstawia narzędzia symulacyjne do badania losowych serii impulsów i ich odpowiedzi. Innym przykładem jest praca, w której na podstawie rejestrowanych odpowiedzi wysokoczęstotliwościowych wnioskuje się o rozkładzie i intensywności impulsów wzbudzających badany układ [28].

Wyniki pokazują spójny obraz dla układów liniowych wzbudzanych seriami impulsów: odpowiedź statystyczna jest przewidywalna, a zjawiska takie jak rezonans średniokwadratowy mają jednoznaczną interpretację fizyczną. W przypadku układów nieliniowych potrzebne są narzędzia specjalne, jak całkowanie czy liniaryzacja równoważna, które pozwalają zachować informację o pełnym rozkładzie zmiennych, co pokazano w pracach [22, 68, 67]. Ten dorobek stanowi bezpośrednią podstawę metod identyfikacji rozkładów impulsów, rozwijanych w kolejnych rozdziałach oraz w pracach naszego zespołu badawczego. Odpowiedź układu jest pojęciem w teorii sygnałów i systemów dynamicznych. Opisuje on reakcję układu na wymuszenie o wysokiej amplitudzie lecz o krótkim czasie trwania. W literaturze utrwalone są zarówno klasyczne metody identyfikacji odpowiedzi [17], jak i modele wibroakustyczne [35]. Jak pokazali Zorica i in. i Haška i in. [84, 18], modyfikacja operatorów różniczkowych wpływa na szybkość wygasania oraz ogólny kształt odpowiedzi.

Odpowiedź układu dyskretnego na losową serię impulsów może zostać wykorzystana do pozyskiwania energii: Sapiński [56] proponuje elektromagnetyczny akumulator energii sprzężony z tłumikiem MR, w którym strojenie rezonansowe i przebieg przejściowy determinują efektywność. Z kolei Zhou i in. [83] wykazali, że właściwy dobór parametrów obwodu RLC sprzężonego z konstrukcją pozwala uzyskać pożądane właściwości tłumiące. Martino [40] pokazuje, jak analizę odpowiedzi dynamicznych — wywołanych m.in. impulsami — wykorzystać do estymacji parametrów praktycznych urządzeń (np. wzbudników elektrodynamicznych) i projektowania testów. Ponadto odpowiedź układu ujawnia sposób, w jaki układ „rozumie” zdarzenie nagle: jak szybko reaguje, jak długo pamięta zakłócenie i w których pasmach energii najłatwiej przepuszcza lub tłumi.

W ostatnich kilkunastu latach powstała propozycja badań, w której bazując na pomiarach drgań wyznacza się rozkład impulsów wymuszających. Jabłoński i Ozga [26] [28, 24] zaproponowali i stopniowo rozwijali podejście do wyznaczania parametrów odpowiedzi oscylatora i układów dyskretnych, w tym w warunkach pomiaru w skończonych oknach czasowych. Kolejne publikacje [46] analizują odpowiednio wpływ poziomu i rozkładu amplitud impulsów na jakość estymacji oraz dobór parametrów modelu oscylatora. Monografia Ozgi [45], a także wcześniejsze prace [25] pokazują związek między odpowiedzią układu, a statystykami impulsów.

Równolegle do metod identyfikacji parametrów układu rozwijają się narzędzia statystyczne i metody uczenia maszynowego stosowane do analizy szeregu czasowego. Z perspektywy teorii rozpoznawania wzorców i estymacji parametrów fundamentalne znaczenie mają opracowania podręcznikowe prace Bishopa [5] oraz przeglądy dotyczące głębokiego uczenia autorstwa LeCun i in. i Goodfellow i in. [36, 16]. W części zastosowań — takich jak rozróżnianie rozkładów amplitud czy intensywności impulsów — dobrze sprawdzają się klasyczne lasy losowe [6]. Ozga oraz Sulewski [72] przedstawili, że odpowiednio przygotowane cechy odpowiedzi umożliwiają klasyfikatorowi lasów losowych dokładny opis ruchu oscylatora wymuszanego losową serią impulsów. Z kolei w kolejnej pracy [48] pokazali potencjał metod nienadzorowanych w analizie drgań, a w publikacji [15] zespół badawczy przedstawił klasyfikację rozkładów wartości impulsów na bazie cech opisujących odpowiedzi na te impulsy. Wraz z popularyzacją głębokich sieci konwolucyjnych (CNN) — VGG [63] — upowszechniły się narzędzia automatycznej ekstrakcji cech wykorzystywane do semantycznego opisu obrazów. Coraz częściej adaptuje się je do sygnałów czy spektrogramów. W tym kontekście współczesne metody wizualizacji i interpretacji [82] dostarczają informacji jak kolejne warstwy sieci dokonują klasyfikacji.

Sztuczne sieci neuronowe stanowią rodzinę modeli uczących się reprezentacji danych poprzez kaskadę prostych przekształceń nieliniowych. Klasyczny neuron oblicza ważoną kombinację sygnałów wejściowych, dodaje składową progową i przepuszcza wynik przez funkcję aktywacji, a neurony ułożone w warstwy tworzą układ, w którym z poziomu na poziom rośnie złożoność opisywanych wzorców. Uczenie polega na dopasowaniu parametrów na podstawie zbioru przykładów wejście–wyjście tak, aby minimalizować błąd predykcji, przy czym informacja o błędzie propaguje się wstecz przez kolejne przekształcenia. W przeglądach i podręcznikach podkreśla się, że właśnie kompozycja wielu prostych kroków pozwala budować silnie nieliniowe funkcje decyzyjne, a jednocześnie zachować skalowalność obliczeń [36, 16, 5]. W zadaniach o strukturze siatki, takich jak analiza obrazów naturalnym wyborem są sieci konwolucyjne. Ich warstwy

operują na lokalnych wycinkach danych, współdzieląc wagi filtrów w przestrzeni i dzięki temu ucząc się translacyjnie niezmienniczych detektorów cech. Z praktycznego punktu widzenia pierwsze warstwy wykrywają proste wzorce, jak krawędzie i tekstury, a kolejne składają je w coraz bardziej semantyczne cechy definiujące przypadek. Istotne etapy rozwoju CNN to demonstracja skuteczności głębokich sieci na ImageNet¹ [33], wprowadzenie prostych, regularnych bloków z filtrami o małych rozmiarach w VGG [63] oraz liczne prace nad rozumieniem i wizualizacją reprezentacji wewnętrznych [82]. W literaturze zwraca się uwagę, że regularna architektura, stałe rozmiary filtrów i powtarzalne motywy konwolucja–aktywacja–zagęszczanie ułatwiają zarówno strojenie, jak i analizę modeli, a przy tym sprzyjają systematycznemu pogłębianiu bez projektowania złożonych połączeń.

Wydajność i stabilność trenowania głębokich modeli w dużej mierze zależą od doboru funkcji aktywacji. Zmiana funkcji aktywacji na prostą dodatnio–jednostajną nieliniowość typu ReLU przyspieszyła uczenie i poprawiła wyniki w wizji komputerowej [39, 19]. Kolejne prace eksplorowały własności auto–normalizujące czy poszukiwały aktywacji optymalnych dla danej głębokości i inicjalizacji, podkreślając związek między dynamiką sygnału i gradientu a zjawiskami takimi jak zanikanie czy eksplozja [31, 50, 16, 5].

Seci konwolucyjne trenowane na bardzo dużych zasobach danych pełnią dziś rolę uniwersalnych ekstraktorów cech. W praktyce często przejmują wagi z modeli wstępnie wyuczonych na ImageNet i adaptuje je do nowych zadań poprzez zamrożenie części warstw, przebudowę warstwy klasyfikacyjnej i krótkie strojenie parametrów. Takie podejście skraca czas uczenia, poprawia generalizację przy ograniczonej liczbie przykładów oraz stabilizuje wczesne etapy optymalizacji [55, 63, 16]. Wraz ze wzrostem głębokości sieci rośnie ryzyko nadmiernego dopasowania, zwłaszcza gdy dysponuje się małym zbiorem danych. Przeglądy metod oceny klasyfikatorów porządkują metryki oparte na macierzy pomyłek i pokazują, jak czytać skuteczność w warunkach nie zrównoważonych klas oraz jak interpretować powiązania między czułością a precyzją [69, 74, 16]. Choć CNN powstały z myślą o obrazach naturalnych, ich mechanizm lokalnego dopasowania filtrów nadaje się do analizy szerokiej klasy danych, w tym szeregów czasowych i map cech z danych pochodzących z doświadczenia. W praktyce często konwertuje się sygnał na obraz, zachowując struktury lokalne i globalne, a następnie stosuje architektury obrazowe. Takie podejście pozwala korzystać z dojrzałych modeli i narzędzi, a jednocześnie odsłania bogatsze wzorce [36, 16]. W niniejszej rozprawie ten kierunek wykorzystuje się do klasyfikacji odpowiedzi układu ciągłego oraz odpowiedzi układu dyskretnego na losowe serie impulsów.

Spośród wielu konkurencyjnych rozwiązań wybrano architektury VGG, kierując się ich przewidywalnym zachowaniem. W zastosowaniach z ograniczoną ilością danych VGG16 bywa kompromisem między mocą, a kosztami obliczeniowymi, natomiast VGG19, dzięki większej liczbie warstw, umożliwia nieco dokładniejszy wybór cech, co również wpływa na proces klasyfikacji.

Przegląd literatury podkreśla trzy aspekty, które są istotne dla zastosowań sieci konwolucyjnych w tej rozprawie. Po pierwsze, głębokie sieci konwolucyjne, rozwijane od AlexNet po VGG, tworzą uniwersalny mechanizm wyodrębnienia złożonych struktur z danych o różnym charakterze [33, 63, 36]. Po drugie, wła-

¹ImageNet to duża baza danych obrazów, zawierająca ponad 14 milionów ręcznie oznaczonych zdjęć, zorganizowanych według hierarchii WordNet. Jest wykorzystywana jako standardowy zestaw danych do oceny algorytmów rozpoznawania obiektów i klasyfikacji obrazów.

ściwy dobór warstw aktywacji stabilizuje uczenie w warunkach ograniczonych danych, pozwalając skupić proces na dopasowaniu ostatnich [55, 19]. Po trzecie, ocena skuteczności powinna być prowadzona w oparciu o metryki adekwatne do rozkładu klas, z pełnym wykorzystaniem informacji z macierzy pomyłek [69, 74]. Rozdziały przedstawiają szczegóły przygotowania danych, konfiguracji modeli VGG16/VGG19 oraz wyniki eksperymentów porównawczych, z odniesieniem do zakresu analizy obrazów skupiającej się na rozpoznawaniu odpowiedzi układu na wymuszenie impulsowe w postaci serii impulsów o małej intensywności.

Rozdział 2

Cel i zakres pracy

2.1 Cel pracy

Celem rozprawy jest opracowanie metody klasyfikacji odpowiedzi układu dyskretnego oraz odpowiedzi układu ciągłego na losową serię impulsów zlokalizowanych w jednym punkcie.

Aby osiągnąć cel główny, zdefiniowano cele szczegółowe, które pozwalają na stopniową realizację poszczególnych etapów badań:

1. **Przedstawienie modelu układu dyskretnego wzbudzonego losową serią impulsów.** Celem jest przedstawienie równania ruchu, definicja zmiennych losowych.
2. **Układ dyskretny jako układ RLC.** Celem jest przedstawienie analogii pomiędzy elementami elektrycznymi i mechanicznymi [30, 9], takimi jak indukcyjność–masa, pojemność–siła czy rezystancja–tłumienie. Umożliwi to traktowanie układu RLC jako modelu referencyjnego dla zjawisk mechanicznych.
3. **Szacowanie parametrów układu na podstawie danych eksperymentalnych.** Celem jest identyfikacja parametrów modelu w oparciu o eksperymentalny pomiar odpowiedzi układu RLC wzbudzonego losową serią impulsów oraz analizy tych odpowiedzi, których wynik uzasadnia konieczność wykorzystania głębokich sieci konwolucyjnych.
4. **Zastosowanie sieci konwolucyjnych typu VGG16/VGG19 do klasyfikacji odpowiedzi układu dyskretnego oraz odpowiedzi układu ciągłego na losową serię impulsów** W pracy wykorzystane zostaną wybrane metody uczenia maszynowego, ze szczególnym uwzględnieniem głębokich sieci neuronowych. Szczegółowo analizowana będzie architektura VGG16 oraz VGG19, która posłuży do klasyfikacji odpowiedzi układu na losową serię impulsów.
5. **Analiza odpowiedzi układu na wymuszenia przy różnych konfiguracjach impulsów.** Szczegółowo badane będą wpływy takich parametrów jak wielkość impulsu η_i oraz prawdopodobieństwo jego wystąpienia $p(\eta_i)$ przy stałej wartości intensywności λ . Analiza ta pozwoli określić zależność pomiędzy charakterystyką wymuszenia a kształtem odpowiedzi układu.

2.2 Zakres pracy

Zakres rozprawy obejmuje zarówno analizę teoretyczną, jak i eksperymentalną, a także zastosowanie metod sztucznej inteligencji w klasyfikacji odpowiedzi układu dyskretnego oraz odpowiedzi układu dyskretnego. Weryfikacji podlega teza, że odpowiednio skonstruowana reprezentacja danych oraz sieci konwolucyjne (VGG16/VGG19) umożliwiają skuteczną klasyfikację odpowiedzi układu na losową serię wymuszeń.

Struktura pracy została podzielona na dziewięć rozdziałów. Poniżej przedstawiono opis poszczególnych części.

Rozdział 1 – Wstęp. Pierwszy rozdział stanowi ogólny przegląd literatury wraz ze wskazaniem obecnych osiągnięć w dziedzinie analizy odpowiedzi na losową serię impulsów. Wprowadzono również odniesienie do metod uczenia maszynowego i sieci neuronowych.

Rozdział 2 – Cel i zakres pracy. W tym rozdziale określono główny cel badań, którym jest klasyfikacja odpowiedzi układu RLC oraz odpowiedzi belki kompozytowej na losowe impulsy z wykorzystaniem architektur VGG16 i VGG19. Sformułowano cel główny oraz cele szczegółowe pracy, a także dodatkowe założenia badawcze.

Rozdział 3 – Model matematyczny drgań oscylatora pod wpływem losowej serii impulsów. Rozdział ma charakter analityczny. W podrozdziale 3.1 przedstawiono równanie ruchu drgań tłumionych wymuszonych losową serią impulsów. Podrozdział 3.2 omawia opis zmiennych losowych występujących w funkcji $F(t)$, modelujących charakter wymuszeń. W podrozdziale 3.3 zaprezentowano odpowiedź oscylatora na takie wymuszenie, co stanowi teoretyczną podstawę dalszych badań.

Rozdział 4 – Eksperymentalna realizacja modelu drgań z wykorzystaniem układu RLC. W podrozdziale 4.1 opisano analogie elektromechaniczne, umożliwiające odwzorowanie drgań mechanicznych w układzie elektrycznym. Podrozdział 4.2 dotyczy realizacji doświadczenia z układem RLC. Podrozdział 4.3 omawia przybliżoną realizację delty Diraca, a 4.4 – metodę wyodrębniania pojedynczych odpowiedzi z zarejestrowanego sygnału.

Rozdział 5 – Estymacja parametrów układu dyskretnego. W tej części pracy skoncentrowano się na identyfikacji parametrów układu RLC. Omówiono wyznaczenie wartości średniej odpowiedzi, czasu propagacji oraz procedury zgrubnej i precyzyjnej identyfikacji parametrów. W podrozdziałach 5.5 i 5.6 przedstawiono szacowanie parametrów dla różnych wartości wymuszeń (100, 200 i 300 mV).

Rozdział 6 – Architektura VGG16 i VGG19. Rozdział ten omawia architektury sieci konwolucyjnych zastosowanych w klasyfikacji. Zaprezentowano schemat działania sieci, opis warstw obu modeli oraz analizę map cech. Przedstawiono też wstępne przetwarzanie obrazów, przepływ informacji przez sieć i interpretację wyników metodą Grad-CAM. Ostatnia część dotyczy wyników klasyfikacji typu Top-5.

Rozdział 7 – Rozpoznanie odpowiedzi układu na pojedynczy impuls oraz odpowiedzi układu na dwa impulsy następujące po sobie dla $\Delta t < 200 \mu s$. Rozdział pracy dotyczy klasyfikacji odpowiedzi układu dyskretnego na pojedyncze impulsy oraz na dwa impulsy następujące po sobie dla $\Delta t < 200 \mu s$. Omówiono definicję analizowanych przypadków i klas mieszanych, przygotowanie zbioru danych i bazy treningowej, a także proces etykietowania. Przedstawiono trening klasyfikatorów opartych na VGG16 i VGG19, ich porównanie oraz ocenę skuteczności na próbach spoza zbioru testowego.

Rozdział 8 – Klasyfikacja odpowiedzi belki kompozytowej FR-4 z piezoelektrykiem MFC. Rozdział przedstawia badania układu ciągłego i jego odpowiedzi na losową serię impulsów. Zaprezentowano stanowisko pomiarowe i opis schematu doświadczenia, sposób przygotowania zbioru danych i generowania obrazów wejściowych dla sieci. Następnie przedstawiono proces uczenia i klasyfikacji z wykorzystaniem sieci VGG16 oraz VGG19, w tym również dla rozszerzonego zbioru danych.

Rozdział 9 – Podsumowanie i wnioski końcowe. Ostatni rozdział podsumowuje realizację celów pracy, przedstawia najważniejsze wnioski oraz wskazuje kierunki dalszych badań. Rozdział ten łączy wszystkie wyniki i ukazuje potencjał dalszego rozwoju zaprezentowanej metody.

2.3 Dodatkowe założenia

W pracy przyjęto kilka założeń, które determinują sposób analizy problemu oraz interpretację uzyskanych wyników. Ich celem jest zawężenie pola badawczego i umożliwienie skoncentrowania się na aspektach klasyfikacji odpowiedzi układu RLC oraz belki kompozytowej.

- **Rozważane impulsy mają znaną wielkość impulsu η_i , prawdopodobieństwo wystąpienia tego impulsu $p(\eta_i)$ oraz znaną intensywność λ .**
- **Układ modelowany jest jako liniowy.** Przyjęto, że badany układ dyskretny jest układem liniowym, co oznacza spełnianie zasad superpozycji oraz proporcjonalności odpowiedzi do wymuszenia.
- **Impulsy wymuszające są traktowane jako sygnały zbliżone do delt Diraca.** W praktyce impuls generowany w układzie eksperymentalnym ma skończoną szerokość czasową i amplitudę.
- **Odpowiedzi są analizowane w warunkach braku zakłóceń zewnętrznych.** W analizie zakłada się, że rejestrowane odpowiedzi układu nie są obciążone istotnymi zakłóceniami wynikającymi z czynników zewnętrznych, takich jak szумы elektromagnetyczne, niestabilności źródła zasilania czy zakłócenia mechaniczne w otoczeniu układu pomiarowego. W praktyce szумы pomiarowe zostały zminimalizowane poprzez odpowiednią konfigurację aparatury i filtrowanie sygnału.
- **Badania układu ciągłego ograniczone do odpowiedzi na pojedyncze impulsy.** Badany układ poddany jest losowej serii impulsów zlokalizowanych w jednym punkcie. Belka jest utwierdzona jednostronnie. W klasyfikacji są brane pod uwagę przypadki, w których kolejna odpowiedź pojawia się po tym jak poprzednia ustała.

2.4 Uwarunkowania doboru parametrów w eksperymentach

Ze względu na stochastyczny charakter analizowanego problemu konieczne było przeprowadzenie dużej liczby prób doświadczalnych dla różnych rozkładów wielkości impulsów η_i . Czas trwania pojedynczego eksperymentu musiał zostać ograniczony do kilku minut, co wynikało zarówno z wymagań praktycznych, jak i ograniczeń technicznych związanych z rozmiarem plików zapisywanych przy częstotliwości próbkowania równej 1 MHz.

Istotnym założeniem było również to, aby w ramach każdej próby zarejestrować wystarczającą liczbę zdarzeń w postaci odpowiedzi układu na wymuszenia impulsowe. Tylko w ten sposób możliwe było zbudowanie reprezentatywnego zbioru danych, obejmującego część uczącą, walidacyjną oraz testową, niezbędnych w procesie klasyfikacji przy wykorzystaniu modeli głębokich sieci konwolucyjnych.

Powyższe uwarunkowania, w połączeniu z przyjętymi parametrami układu dyskretnego i parametrami układu ciągłego wykorzystanego w eksperymentach zdeterminowały parametry eksperymentów. Dotyczyły one przede wszystkim intensywności λ występowania impulsów, a także czasu trwania pomiaru i doboru wartości wielkości impulsu η_i . W ten sposób zapewniono zarówno powtarzalność procedury badawczej, jak i możliwość uzyskania danych o odpowiedniej jakości, koniecznej do dalszej analizy.

Rozdział 3

Model matematyczny drgań oscylatora pod wpływem losowej serii impulsów

W rozdziale przedstawiono model matematyczny opisujący zachowanie oscylatora z tłumieniem poddanego działaniu losowej serii impulsów. Rozdział rozpoczyna się od sformułowania równania ruchu. Następnie omówiono charakter impulsowego wymuszenia z wykorzystaniem funkcji delty Diraca $\delta(t)$. Kolejno wprowadzono opis zmiennych losowych oraz przedstawiono rozwiązanie układu.

3.1 Równanie ruchu drgań harmonicznym tłumionych wymuszonych losową serią impulsów

Drgania oscylatora [25, 45] opisane są równaniem (3.1):

$$\frac{d^2x}{dt^2} + 2b\frac{dx}{dt} + a^2x = F(t) \quad (3.1)$$

W tej pracy $F(t)$ stanowi ciąg impulsów o losowej wielkości impulsów η_i , które występują w losowych chwilach czasu t_i . Tego rodzaju funkcja znajduje zastosowanie w modelowaniu procesów stochastycznych [46, 1, 60, 13]. Zapis tej funkcji przedstawiono zgodnie z równaniem (3.2):

$$F(t) = \sum_{t_i < t} \eta_i \delta_{t_i} \quad (3.2)$$

gdzie:

δ_{t_i} – delta Diraca w chwili wystąpienia impulsu t_i [12].

Funkcja (3.2) umożliwia analizę ruchu układów pod wpływem sił o losowym charakterze [3]. Delta Diraca, oznaczana symbolem δ_{t_i} nie jest funkcją w tradycyjnym znaczeniu, lecz uogólnionym pojęciem – dystrybucją [25]. Głównym zadaniem delty Diraca jest modelowanie idealnie krótkiego impulsu o nieskończonej dużej amplitudzie. W modelu matematycznym odpowiedź układu na pojedynczy impuls odpowiada wzbudzeniu swobodnych drgań, wynikającemu z gwałtownego wzrostu prędkości początkowej przy zerowym przemieszczeniu początkowym [30, 14, 49, 70]. Poniżej przedstawiono matematyczną definicję delty Diraca (3.3).

$$\int_{-\infty}^{+\infty} \delta(t) dt = 1 \quad (3.3)$$

Ze względu na naturę $\delta(t)$, w literaturze [43, 77] definiuje się sekwencje funkcji zwyczajnych, które zbliżają się do delty Diraca. Przykładem takiego przybliżenia jest funkcja Gaussa przedstawiona przez zależność (3.4).

$$\delta_\epsilon(t) = \frac{1}{\sqrt{\pi\epsilon}} e^{-\frac{t^2}{\epsilon}} \quad (3.4)$$

gdzie:

$\epsilon > 0$ jest parametrem, który w granicy $\epsilon \rightarrow 0$ powoduje, że $\delta_\epsilon(t)$ skupia się w punkcie $t = 0$, zachowując warunek (3.5).

$$\lim_{\epsilon \rightarrow 0} \int_{-\infty}^{+\infty} \delta_\epsilon(t) dt = 1 \quad (3.5)$$

Innymi popularnymi przybliżeniami funkcji delty Diraca są sekwencje funkcji prostokątnych o bardzo małej szerokości i wysokiej amplitudzie [12].

3.2 Opis zmiennych losowych występujących w funkcji $F(t)$

W pracy wprowadzono następujące zależności dotyczące losowego czasu pojawienia się impulsu t_i oraz losowej wielkości impulsu η_i . Przedziały czasowe między [27, 26, 25, 24, 45], określone jako $\tau_i = t_i - t_{i-1}$, są niezależnymi, ciągłymi zmiennymi losowymi o rozkładzie wykładniczym. Ich funkcja gęstości prawdopodobieństwa opisana jest równaniem (3.6):

$$t(\tau) = \begin{cases} \lambda e^{-\lambda\tau}, & \text{dla } \tau \geq 0 \\ 0, & \text{dla } \tau < 0 \end{cases} \quad (3.6)$$

gdzie:

λ to intensywność występowania impulsów.

Generowanie impulsów w doświadczeniu wymaga określenia czasu ich występowania, który wyznacza się na podstawie rozkładu wykładniczego. Losowanie zmiennej random odbywa się z rozkładu jednostajnego w przedziale $(0, 1]$ zgodnie z zależnością (3.7):

$$\tau_i = -\frac{\ln(1 - \text{random})}{\lambda} \quad (3.7)$$

Wielkości impulsów η_i modelowane są jako niezależne, dyskretne zmienne losowe o skończonej wartości oczekiwanej [25]. Ich wartości losowane są na podstawie narzuconych prawdopodobieństw p_i zgodnie z równaniem (3.8):

$$\eta_i = \begin{cases} \eta_1 & \text{gdy } \text{random} \leq p_1, \\ \eta_2 & \text{gdy } p_1 < \text{random} \leq p_1 + p_2, \\ \vdots & \\ \eta_n & \text{gdy } \sum_{j=1}^{n-1} p_j < \text{random} \leq 1 \end{cases} \quad (3.8)$$

gdzie:

$\sum_{j=1}^n p_j = 1$, a zmienna *random* tak jak w równaniu 3.7 oznacza losowanie z rozkładu jednostajnego w przedziale $(0, 1]$.

Przedziały czasowe pomiędzy impulsami oraz wielkości impulsów η_i są wzajemnie niezależnymi zmiennymi losowymi.

Opisane powyżej zmienne losowe zostaną wykorzystane w eksperymentach przeprowadzonych dla układu dyskretnego (układu RLC) oraz układu ciągłego (belki kompozytowej).

3.3 Odpowiedź oscylatora na wymuszenie losową serią impulsów

Problem ten można sprowadzić do analizy rozwiązania równania różniczkowego drugiego rzędu przy założeniu zerowych warunków początkowych. Zakładając, że dla $t \leq t_1$ układ znajduje się w stanie spoczynku, co oznacza brak jakiegokolwiek wymuszenia to $x(t) = 0$, $\dot{x}(0) = 0$.

Zgodnie z równaniem odpowiedzi (3.9) oscylatora tłumionego na zadane wymuszenie impulsowe dla $(t - t_i)$ wynosi:

$$g(t - t_i) = \frac{\eta_i}{c} \sin[c(t - t_i)] e^{-b(t-t_i)} \quad (3.9)$$

gdzie:

$$c = \sqrt{a^2 - b^2}$$

Dla $t_1 < t \leq t_2$ układ reaguje na pierwszy impuls i odpowiada wzbudzeniu swobodnych drgań, wynikającemu z gwałtownego wzrostu prędkości początkowej przy zerowym przemieszczeniu początkowym [30]. Zgodnie z zasadą superpozycji [45], całkowita odpowiedź układu na wymuszenie (3.1 - 3.2) losową serią impulsów jest sumą odpowiedzi wywołanych przez każdy z impulsów osobno przy założeniu zerowych warunków początkowych dla każdego z rozwiązań.

Dla $t_2 < t \leq t_3$ odpowiedź układu jest sumą odpowiedzi na dwa kolejne impulsy zapoczątkowane w chwilach t_1 oraz t_2 .

Tym samym, dla ogólnego przedziału $t_i < t \leq t_{i+1}$, odpowiedź układu stanowi sumę drgań rozpoczętych w chwilach $t_1, t_2, \dots, t_i$. Zależność tę opisuje równanie (3.10).

$$x(t) = \begin{cases} 0 & \text{dla } t \leq t_1, \\ \frac{\eta_1}{c} e^{-b(t-t_1)} \sin(c(t-t_1)) & \text{dla } t_1 < t \leq t_2, \\ \frac{\eta_1}{c} e^{-b(t-t_1)} \sin(c(t-t_1)) + \\ \quad \frac{\eta_2}{c} e^{-b(t-t_2)} \sin(c(t-t_2)) & \text{dla } t_2 < t \leq t_3, \\ \dots & \\ \frac{\eta_1}{c} e^{-b(t-t_1)} \sin(c(t-t_1)) + \\ \quad \frac{\eta_2}{c} e^{-b(t-t_2)} \sin(c(t-t_2)) + \\ \quad \dots + \\ \quad \frac{\eta_i}{c} e^{-b(t-t_i)} \sin(c(t-t_i)) & \text{dla } t_i < t \leq t_{i+1} \end{cases} \quad (3.10)$$

Wykorzystując tożsamości trygonometryczne opisane równaniem (3.11),

$$\sin [c(t - t_i)] = \sin(ct) \cos(ct_i) - \cos(ct) \sin(ct_i) \quad (3.11)$$

wyznaczając funkcje pomocnicze (3.12) i (3.13) przedstawione w poniższych równaniach:

$$A(t) = \sum_{t_i < t} \frac{\eta_i}{c} \cos(ct_i) \quad (3.12)$$

oraz

$$B(t) = \sum_{t_i < t} \frac{\eta_i}{c} \sin(ct_i) \quad (3.13)$$

równanie odpowiedzi oscylatora tłumionego na zadane wymuszenie przedstawia się w postaci (3.14):

$$x(t) = \sin(ct) \sum_{t_i < t} \frac{\eta_i}{c} \cos(ct_i) e^{-b(t-t_i)} - \cos(ct) \sum_{t_i < t} \frac{\eta_i}{c} \sin(ct_i) e^{-b(t-t_i)} \quad (3.14)$$

Pomiędzy wystąpieniami impulsów, funkcje pomocnicze $A(t)$ (3.12) oraz $B(t)$ (3.13) maleją wykładniczo, lecz w przypadku, gdy $t = t_i$, następuje poniższa aktualizacja:

$$A(t_i^+) = A(t_i^-) e^{-b(t_i-t_{i-1})} + \frac{\eta_i}{c} \cos(ct_i) \quad (3.15)$$

oraz

$$B(t_i^+) = B(t_i^-) e^{-b(t_i-t_{i-1})} + \frac{\eta_i}{c} \sin(ct_i) \quad (3.16)$$

gdzie:

t_i^- oraz t_i^+ oznaczają chwile bezpośrednio przed i po wystąpieniu impulsu.

Odpowiedź układu $x(t)$ opisana równaniem jest wyznaczana poprzez przechowywanie i ciągłą aktualizację dwóch zmiennych pomocniczych funkcji $A(t)$ oraz $B(t)$, zgodnie z równaniami (3.15) oraz (3.16).

Rozwiązanie układu równań [45] (3.1) - (3.2) można przedstawić za pomocą równania (3.17).

$$x(t) = \frac{\eta_i}{c} \sum_{0 < t_i < t} \eta_i e^{-b(t-t_i)} \sin(c(t - t_i)) \quad (3.17)$$

Przedstawiony powyżej matematyczny model drgań oscylatora oraz opis zmiennych występujących w funkcji $x(t)$ umożliwiają analizę układów wzbudzanych losową serią impulsów.

Ze względu na złożoność realizacji takiego eksperymentu przy pomocy układu mechanicznego zdecydowano się na zastosowanie układu RLC[2]. Układ RLC, dzięki swoim właściwościom, pozwala na odwzorowanie zjawisk drgań i tłumienia za pomocą odpowiednio dobranych parametrów elektrycznych.

Rozdział 4

Eksperymentalna realizacja modelu drgań oscylatora pod wpływem losowej serii impulsów z wykorzystaniem układu RLC

W celu przeprowadzenia eksperymentów zastosowano układ elektryczny, zamiast mechanicznego. Wynika to z trudności w realizacji losowej serii impulsów w oscylatorze mechanicznym. Dodatkowo użycie układu RLC umożliwia jednoczesną rejestrację wymuszenia oraz odpowiedzi na wymuszenie. Dzięki analogiom elektromechanicznym możliwe jest odwzorowanie zjawisk charakterystycznych dla układów mechanicznych przy użyciu odpowiednio dobranych elementów elektrycznych.

W rozdziale omówiono podstawowe zależności pomiędzy wielkościami mechanicznymi a elektrycznymi, przedstawiono sposób realizacji doświadczenia w oparciu o układ RLC oraz opisano metodykę przetwarzania zarejestrowanych sygnałów w celu wyodrębnienia pojedynczych odpowiedzi układu na poszczególne impulsy.

4.1 Analogie elektromechaniczne

Obwód RLC jest przykładem układu dynamicznego, którego odpowiedź zależy od wartości rezystancji, indukcyjności oraz pojemności [30, 9]. Składa się on z rezystora (R), cewki indukcyjnej (L), kondensatora (C) oraz źródła napięcia, które wymusza odpowiedź układu. W przeprowadzonych badaniach jako źródło wymuszenia zastosowano przybliżoną realizację delta Diraca (rysunek 4.4).

W przeciwieństwie do idealnego obwodu LC, w którym oscylacje mogą trwać w nieskończoność (przy założeniu braku strat), w obwodzie RLC energia systematycznie maleje wskutek rezystancji, co prowadzi do stopniowego zaniku drgań. Zjawisko to jest analogiczne do układu mechanicznego z tłumieniem, w których opory ruchu powodują wygaszanie oscylacji.

Po przyłożeniu wymuszenia następuje przepływ prądu przez cewkę indukcyjną, generując pole magnetyczne i indukując siły, które mogą być interpretowane w sposób analogiczny do sił w układach mechanicznych. Oscylacje w układzie RLC zależą nie tylko od wartości indukcyjności i pojemności, ale także od wielkości rezystancji, określającej stopień tłumienia układu.

W literaturze wyróżnia się dwie podstawowe analogie elektromechaniczne [30, 9]. W pierwszej z nich napięcie elektryczne stanowi odpowiednik siły mechanicznej, a natężenie prądu odpowiada prędkości ruchu elementu mechanicznego. W tym ujęciu masa jest odwzorowywana przez indukcyjność, tłumienie przez rezystancję, a sprężystość przez odwrotność pojemności kondensatora.

Druga analogia traktuje natężenie prądu jako odpowiednik siły mechanicznej, a napięcie jako odpowiednik prędkości. Masa odpowiada pojemności, tłumienie — odwrotności rezystancji, a sprężystość — odwrotności indukcyjności. Podsumowanie obu tych analogii przedstawiono w tabeli 4.1.

Tabela 4.1: Zależności elektromechaniczne i ich analogie w układach elektrycznych

Układ mechaniczny	I elektryczny układ analogii	II elektryczny układ analogii
Siła $P(t)$	Napięcie $U(t)$	Natężenie prądu $I(t)$
Przemieszczenie x	Ładunek q	Strumień magnetyczny Ψ
Prędkość v	Natężenie prądu $I(t)$	Napięcie $U(t)$
Masa m	Indukcyjność cewki L	Pojemność kondensatora C
Współczynnik tłumienia b	Oporność R	Przewodność $\frac{1}{R}$
Sztywność k	Odwrotność pojemności $\frac{1}{C}$	Odwrotność indukcyjności $\frac{1}{L}$

W analizowanym obwodzie elektrycznym można zatem zmieniać parametry obwodu poprzez dobór bądź podmianę komponentów elektronicznych, co pozwala badać zachowanie rozpatrywanego modelu. W prowadzonych badaniach zastosowano **pierwszą analogię elektromechaniczną**, w której *napięcie* pełni rolę odpowiednika *siły mechanicznej*, a *natężenie prądu* odpowiada *prędkości*. W tym ujęciu *rezystancja* modeluje *tłumienie*, *indukcyjność* odpowiada *masie*, a *pojemność kondensatora* jest odwrotnością *sztywności*. Taki wybór analogii elektromechanicznej pozwala w sposób bezpośredni analizować wpływ poszczególnych parametrów na zachowanie układu, wykorzystując właściwości układów RLC.

W kontekście rejestracji odpowiedzi układu, uwagę poświęca się zmianie napięcia na kondensatorze, które – choć formalnie nie jest bezpośrednim odpowiednikiem przemieszczenia – pozostaje z nim ściśle powiązane. Napięcie to wynika ze zgromadzonego ładunku elektrycznego, a ten jest proporcjonalny do przemieszczenia w analizowanej analogii, zgodnie z zależnością (4.1).

$$U_C(t) = \frac{q(t)}{C}, \quad \text{gdzie} \quad q(t) = \int I(t) dt. \quad (4.1)$$

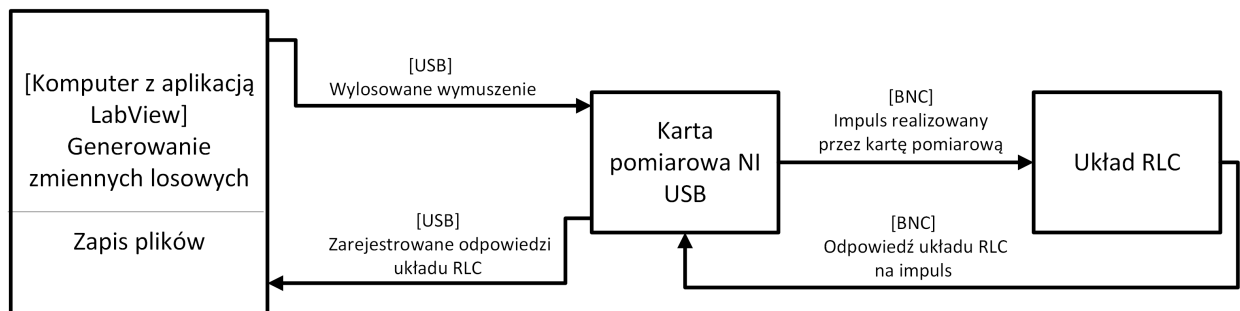
W konsekwencji, napięcie na kondensatorze $U_C(t)$ może być traktowane jako obserwowalna wielkość reprezentująca zachowanie układu mechanicznego w czasie, umożliwiając analizę jego dynamicznej odpowiedzi na wymuszenia impulsowe.

4.2 Realizacja doświadczenia z układem RLC

Przy realizacji doświadczenia zastosowano urządzenie pomiarowe National Instruments (NI) oraz oprogramowanie LabView, które umożliwiły akwizycję danych oraz kontrolę eksperymentów. Elementem zestawu pomiarowego była karta pomiarowa NI USB 6251 [44], która została użyta do realizacji wymuszenia

w układzie RLC oraz akwizycji danych. Karta ta należy do serii M i jest przeznaczona do różnorodnych aplikacji naukowych oraz inżynierskich. Oferuje ona 16 analogowych kanałów wejściowych, które mogą być skonfigurowane w trybie różnicowym lub jednoosiowym, a także 24 cyfrowe kanały wejściowe oraz 2 analogowe kanały wyjściowe. Wysoka rozdzielczość przetwornika analogowo-cyfrowego (ADC) wynosząca 16 bitów oraz maksymalna częstotliwość próbkowania 1,25 MS/s dla jednego kanału gwarantują dokładność pomiarów. Dodatkowo, karta posiada wbudowany bufor pamięci o pojemności 4095 próbek, co umożliwia efektywne zarządzanie danymi.

Do przeprowadzenia badania wykorzystano LabVIEW 2022 Q3. Wspomniany program, jako platforma graficzna, umożliwia tworzenie interfejsów użytkownika oraz aplikacji do analizy danych bez konieczności pisania kodu w tradycyjnych językach programowania. Dodatkowo wspiera integrację z językiem Python, co pozwala na użycie skryptów w aplikacjach LabVIEW, zwiększając tym samym elastyczność programowania. Użycie zaawansowanego oprogramowania oraz precyzyjnych urządzeń pomiarowych umożliwiło uzyskanie wyników, które stanowią podstawę dalszych analiz i wniosków.



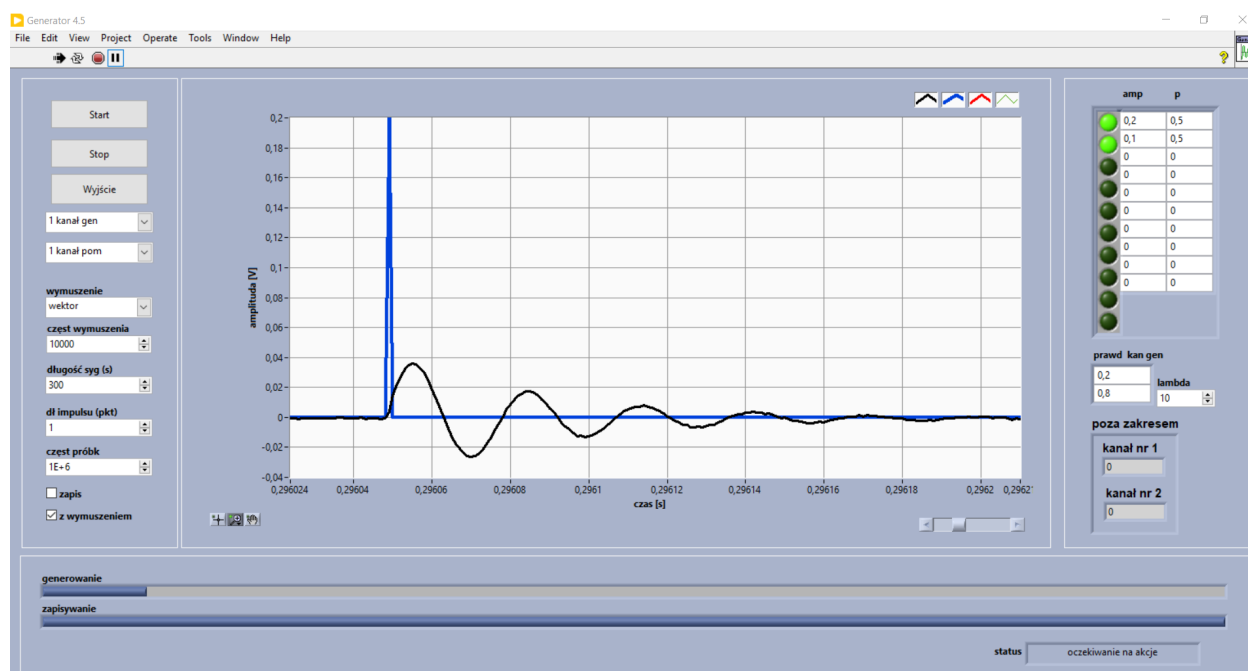
Rysunek 4.1: Uproszczony schemat doświadczenia z układem RLC.

Na rysunku 4.1 przedstawiono uproszczony schemat doświadczenia, którego celem jest analiza odpowiedzi układu RLC na losową serię impulsów. W komputerze została uruchomiona aplikacja stworzona w środowisku LabVIEW (rysunek 4.2), której zadaniem jest wygenerowanie losowej serii impulsów. Wylosowane wielkości impulsu oraz losowy czas ich wystąpienia są przekazywane do karty pomiarowej poprzez interfejs USB.

Karta pomiarowa pełni w układzie dwie funkcje. Pierwszą z nich jest realizacja wymuszenia wygenerowanego przez komputer w postaci impulsu napięciowego, przesyłanego do badanego układu RLC poprzez złącze BNC. Drugą funkcją karty jest rejestracja odpowiedzi układu RLC na zadane wymuszenie oraz przesłanie zarejestrowanego sygnału do komputera za pośrednictwem interfejsu USB.

Badany układ RLC odbiera impuls napięciowy jako sygnał wymuszający, natomiast karta pomiarowa równolegle rejestruje jego odpowiedź (rysunek 4.3). Sygnał odpowiedzi przesyłany jest z powrotem do karty pomiarowej przewodem BNC, co umożliwia jego dalszą analizę i zapis.

Dane zarejestrowane przez kartę pomiarową zapisywane są w pliku tekstowym w formacie tabelarycznym. Plik ten zawiera dwie kolumny: w pierwszej znajdują się wartości odpowiedzi układu RLC na wymuszenie, natomiast w drugiej — wartości sygnału wymuszającego, wygenerowanego (wylosowanego) przez algorytm.



Rysunek 4.2: Widok aplikacji LabVIEW do przeprowadzania doświadczeń. Odpowiedź układu RLC na jeden z impulsów z losowej serii impulsów.

W tabeli 4.2 przedstawiono zestawienie podstawowych parametrów definiowanych w badaniu. W tej próbie wykorzystano dziesięć wartości napięcia wymuszającego drgania oscylatora, o amplitudach rosnących od 100 mV do 1000 mV w krokach co 100 mV. Losowanie wartości impulsów odbywało się zgodnie z równaniem (3.8) przy czym każda z wartości $U_1, U_2, \dots, U_{10}$ (odpowiadających odpowiednio $\eta_1, \eta_2, \dots, \eta_{10}$) posiadały jednakowe prawdopodobieństwo wystąpienia równe 0.1.

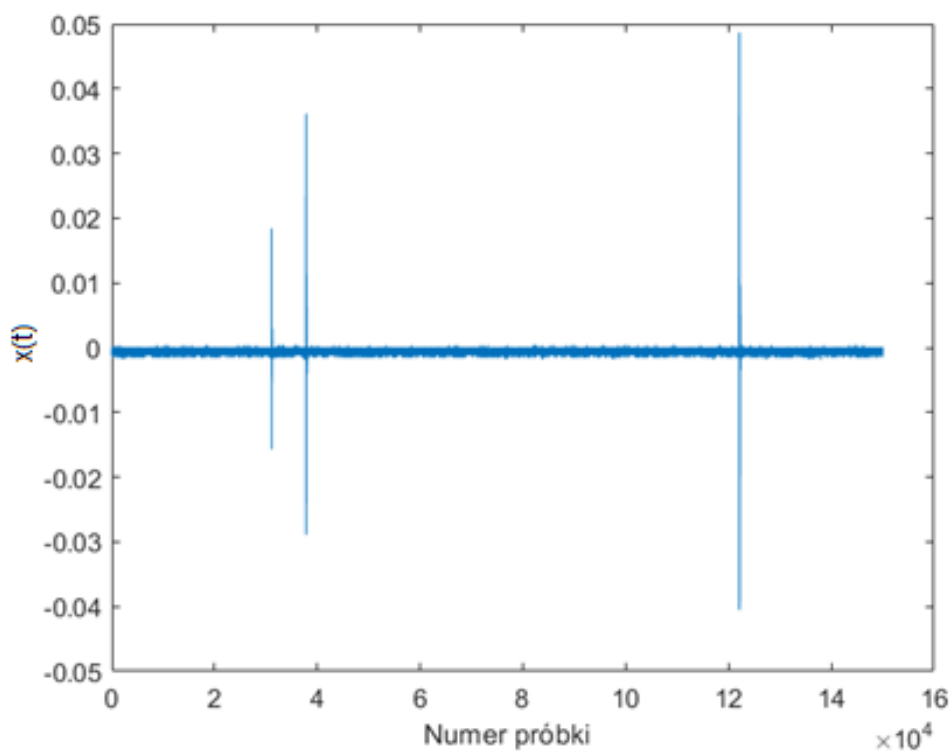
Następnym parametrem był czas pomiaru. Jest to całkowity czas trwania jednego eksperymentu, który wynosił 300 sekund, co pozwoliło na zebranie wystarczającej ilości danych niezbędnych do analizy. Należy tu wspomnieć, iż czas zapisu danych do późniejszej analizy wynosił dwukrotną wartość czasu pomiaru.

Wartość intensywności wzbudzenia λ została ustalona na poziomie 10. Losowanie czasu pomiędzy impulsami odbywa się zgodnie z równaniem (3.7).

W celu zapewnienia wysokiej precyzji pomiarów zastosowano częstotliwość próbkowania równą 1 MHz. Tak duża wartość pozwoliła na uchwycenie dynamicznych zmian w sygnale odpowiedzi układu. Niestety, taka częstotliwość pomiaru wpływała również na rozmiar plików wyjściowych. Fakt ten wymusił zastosowanie macierzy dyskowych o wysokiej pojemności.

Tabela 4.2: Parametry symulacji wykorzystane w doświadczeniu

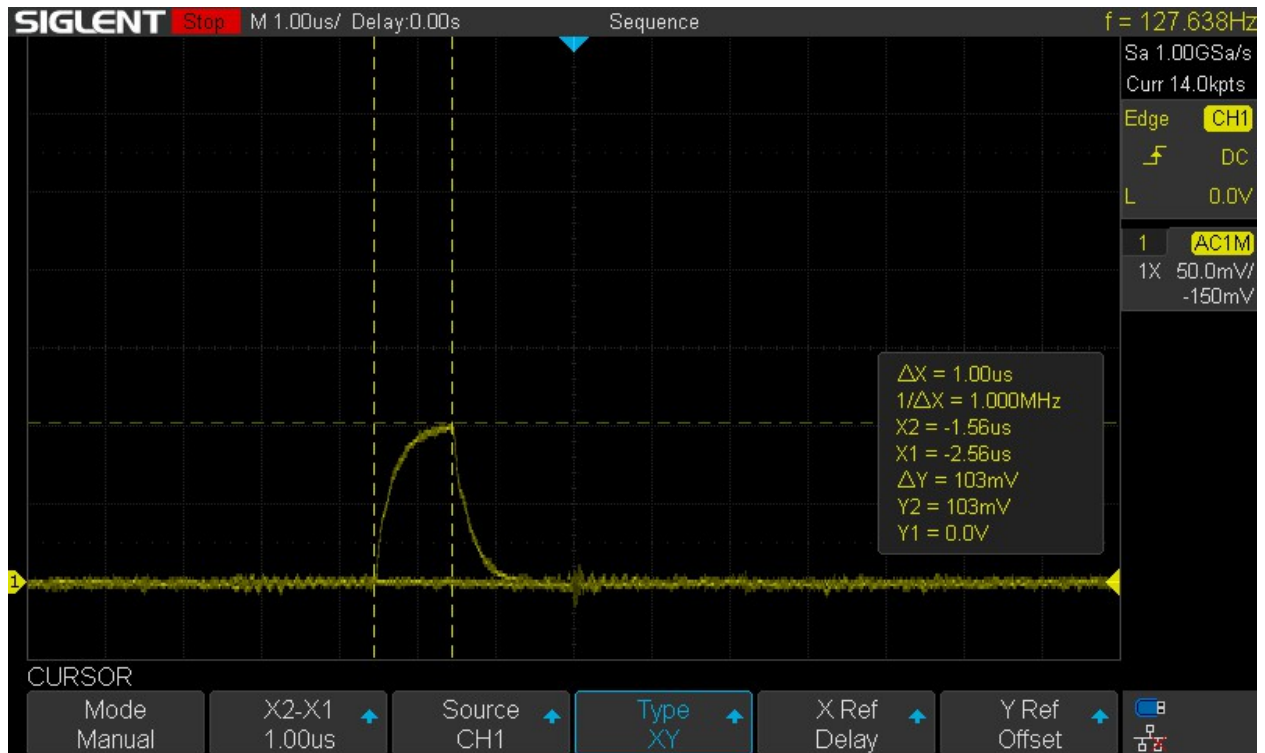
Parametr	Wartość
$U_1 (\eta_1)$	100 mV
$U_2 (\eta_2)$	200 mV
$U_3 (\eta_3)$	300 mV
$U_4 (\eta_4)$	400 mV
$U_5 (\eta_5)$	500 mV
$U_6 (\eta_6)$	600 mV
$U_7 (\eta_7)$	700 mV
$U_8 (\eta_8)$	800 mV
$U_9 (\eta_9)$	900 mV
$U_{10} (\eta_{10})$	1 000 mV
$p(\eta_i)$	0.1 (dla każdego $i = 1, \dots, 10$)
t_{pomiar}	300 s
Intensywność λ	10
Częstotliwość próbkowania	1 MHz



Rysunek 4.3: Fragment zarejestrowanych danych z doświadczenia zdefiniowanego w tabeli 4.2.

4.3 Realizacja delty Diraca

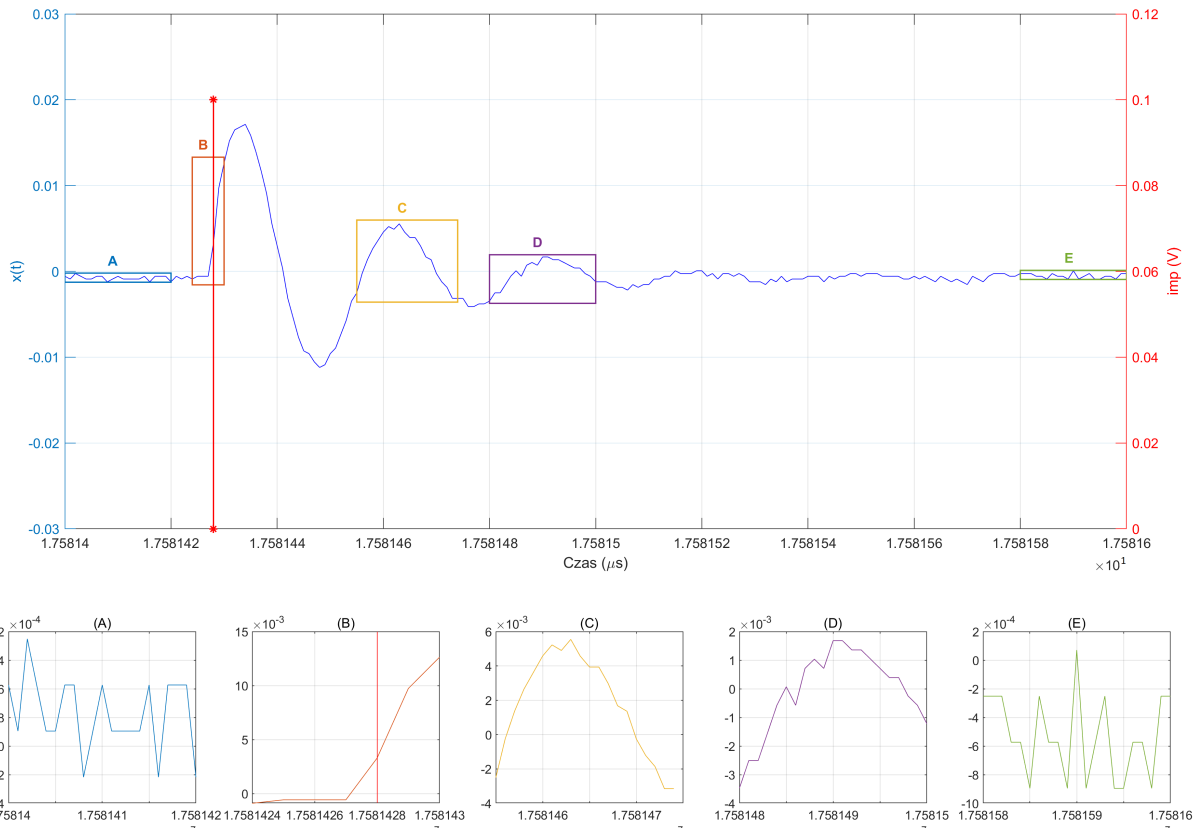
Delta Diraca nie może być bezpośrednio zrealizowana w warunkach fizycznych [75, 64, 78], ponieważ jest jedynie narzędziem matematycznym. W przypadku tej pracy, pojedynczy impuls realizowany jest jako wymuszenie o kształcie trójkąta przy użyciu karty pomiarowej NI USB 6251 (rysunek 4.4).



Rysunek 4.4: Realizacja delty Diraca przy użyciu karty pomiarowej NI USB 6251.

Należy zwrócić uwagę na funkcję opisującą narastanie impulsu do poziomu 100 mV oraz późniejszy spadek amplitudy. Po osiągnięciu maksymalnej wartości obserwuje się charakterystyczne wykładnicze zanikanie. Paraboliczny wzrost oraz wykładniczy spadek sygnału nie wykazują symetrii. Ważnym elementem analizy jest również szum tła, który występuje zarówno przed impulsem, jak i po jego wygaszeniu.

Na rysunku 4.5 przedstawiono fragment przebiegu czasowego odpowiedzi układu RLC na pojedynczy impuls. Impuls pojawia się w chwili $17,58 \mu\text{s}$. Wspomniany powyżej szum występujący przed momentem wzbudzenia oraz po wygaszeniu drgań zarejestrowany jest przez kartę (rysunek 4.5 (A) i (E)). Szum ten nie pozostaje bez wpływu na przebieg odpowiedzi. Wprowadza on stałą składową różną od zera, która równocześnie wpływa na wartość początkową odpowiedzi układu. Ponadto można zauważyć wpływ czasu narastania i zaniku impulsu na odpowiedź (rysunek 4.5 (B)). Dodatkowo są widoczne zakłócenia jako drobne fluktuacje amplitudy w obrębie samej krzywej reprezentującej odpowiedź (rysunek 4.5 (C) i (D)). W efekcie zarejestrowany sygnał nie ma postaci idealnie gładkiej funkcji tłumionych drgań. Widoczne na wykresie zaburzenia odpowiedzi mogą utrudniać dalszą analizę, a w szczególności w przypadku estymacji parametrów tej krzywej i klasyfikacji odpowiedzi układu.



Rysunek 4.5: Odpowiedź układu na pojedynczy impuls zarejestrowana przy użyciu karty pomiarowej NI USB 6251.

4.4 Wyodrębnienie pojedynczych odpowiedzi z zarejestrowanego sygnału

W trakcie prac nad analizą danych pochodzących z eksperymentu zaszła potrzeba opracowania narzędzia umożliwiającego przetworzenie dużych plików wyjściowych zawierających odpowiedzi układu na wymuszenia. W tym celu przygotowano program w środowisku MATLAB pozwalający na wyodrębnienie odpowiedzi na wzbudzenia. Składa się on z kilku etapów: inicjalizacji, wczytywania danych, wyodrębnienia pojedynczych odpowiedzi układu oraz zapisu wyników.

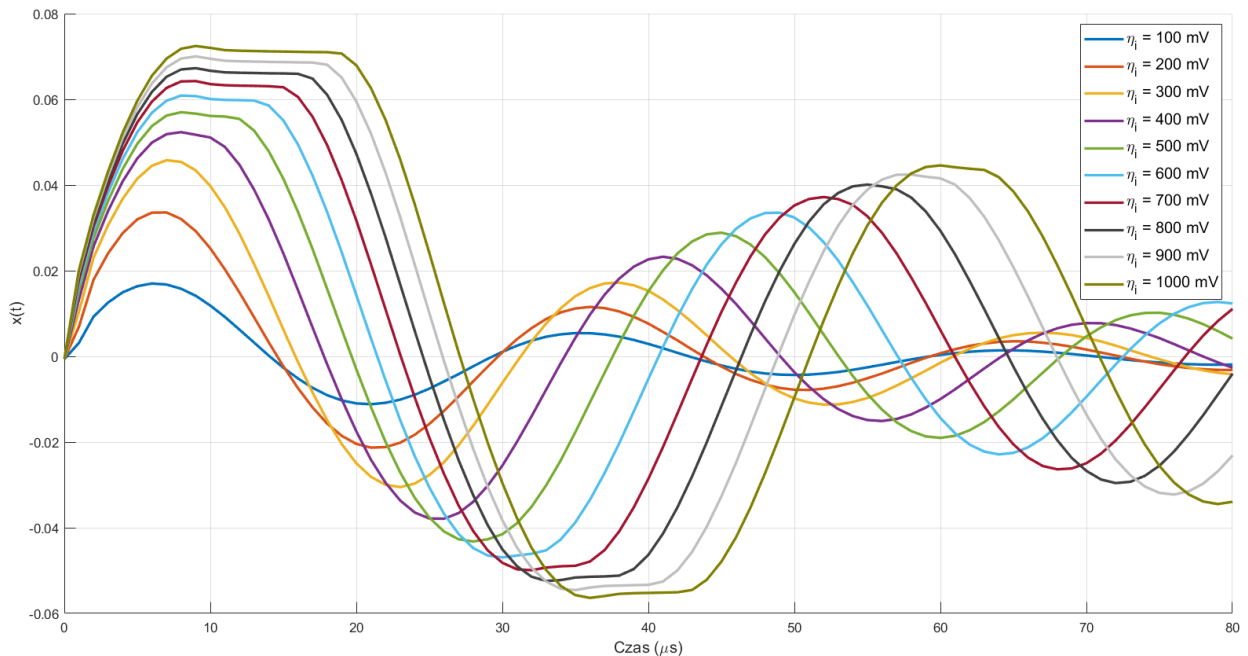
Pierwszym krokiem jest zdefiniowanie katalogów wejściowego i wyjściowego. Program sprawdza, czy katalog przeznaczony do zapisu wyników istnieje — jeśli nie, zostaje on utworzony automatycznie. Następnie generowana jest lista wszystkich plików tekstowych w katalogu wejściowym, które zostaną przetworzone w kolejnych krokach.

Dla każdego pliku z listy dane są wczytywane z wykorzystaniem pętli. Plik wejściowy zawiera dwie kolumny, których wartości są przypisywane do zmiennych. Pierwsza kolumna odpowiada zarejestrowanej odpowiedzi układu, która przechowywana jest w zmiennej X_1 , a druga odpowiada wielkości wymuszenia η_i , przechowywanej w zmiennej imp .

Po wczytaniu danych następuje identyfikacja wartości wymuszenia η_i — są to wartości większe od zera w wektorze imp . Dla każdej z wartości η_i tworzona jest osobna macierz, do której przypisywane są odpowiedzi układu związane z daną wielkością wymuszenia.

Program przeszukuje wektor imp w celu identyfikacji pozycji impulsów. Jeżeli między dwoma kolejnymi impulsami znajduje się co najmniej 2000 próbek, to dany impuls traktowany jest jako osobne zdarzenie. Z weryfikacji wynika, że wartość ta pozwala uniknąć sytuacji, kiedy kolejne impulsy pojawiają się zanim odpowiedzi na poprzednie impulsy wygasną. Następnie z sygnału X_1 przenoszony jest fragment o długości 500 próbek począwszy od próbki sygnału poprzedzającej wystąpienie impulsu. Taka ilość próbek zawiera pojedynczą odpowiedź układu RLC oraz zapewnia, że każda odpowiedź w zakresie od 100 mV do 1 000 mV będzie zachowana bez utraty informacji.

Po zakończeniu przetwarzania danego pliku macierze zawierające wyodrębnione fragmenty odpowiedzi są zapisywane jako osobne pliki tekstowe. Nazwy plików zawierają nazwę oryginalnego pliku wejściowego oraz wartość przypisanego impulsu. Każdy wiersz w pliku wyjściowym reprezentuje jedną odpowiedź układu na wymuszenie o zadanej wartości η_i , gdzie $i = 1, 2, \dots, 10$. Następnie wyznaczono wartości średnie dla każdego η_i



Rysunek 4.6: Wartości średnie odpowiedzi układu RLC na wymuszenia od 100 mV do 1 000 mV.

Badania eksploracyjne wskazują, że dla wymuszeń o amplitudach od 400 mV 1 000 mV zauważalne jest zmniejszenie amplitudy w postaci wypłaszczenia odpowiedzi układu w początkowej fazie, tj. w przedziale czasowym od $8 \mu\text{s}$ do $20 \mu\text{s}$. Zjawisko to zostało przedstawiono w postaci powiększenia fragmentu przebiegu na rysunku 4.6, który przedstawia średnie odpowiedzi układu RLC dla wymuszeń z zakresu od 100 mV 1 000 mV.

Należy zauważyć, że poszczególne krzywe, otrzymane w wyniku wyznaczenia średnich odpowiedzi na wymuszenia, przecinają oś X w różnych chwilach czasu. Oznacza to, że model nie odwzorował eksperymentu dla wszystkich wartości wymuszeń, a dla każdej średniej odpowiedzi uzyskano inną częstotliwość drgań tłumionych ω oraz inny współczynnik tłumienia b . W konsekwencji otrzymujemy kilka układów drgających o różnych parametrach.

W przypadku idealnych, liniowych układów RLC zasada superpozycji umożliwia sumowanie niezależnych odpowiedzi na każde z wymuszeń. Jednak w tym przypadku nie jest ona uzasadniona, gdy kolejny impuls pojawia się, zanim drgania wywołane poprzednim wymuszeniem ulegną całkowitemu wygaszeniu. Badanie tych odpowiedzi układu RLC jest szczególnie trudne w warunkach eksperymentalnych. Odległość pomiędzy impulsami jest zmienną losową ciągłą, co oznacza istnienie wielu możliwych konfiguracji, w których kilka impulsów następuje po sobie w krótkich odstępach czasu, zanim odpowiedzi na wcześniejsze wymuszenia całkowicie zanikną. Gdy odległość pomiędzy impulsami jest krótsza niż czas zaniku drgań, energia dostarczona przez kolejny impuls dodaje się do energii pozostałej w układzie po poprzednim wymuszeniu. Powoduje to zmianę chwilowych wartości prądu i napięcia w elementach układu, a tym samym modyfikację parametrów drgań, takich jak częstotliwość drgań tłumionych i współczynnik tłumienia.

Dla parametrów przyjętych w niniejszej pracy zdarzenie to występuje bardzo rzadko, dlatego w analizowanym zakresie amplitud od 100 mV do 300 mV stosowanie superpozycji jest rozwiązaniem przybliżonym gdyż nie występuje wypłaszczenie się odpowiedzi układu.

Z badań eksploracyjnych wynika (rysunek 4.6), że najmniejsze różnice pomiędzy parametrami układu są dla wymuszeń 100 mV, 200 mV i 300 mV. Te wartości wymuszenia będą wykorzystane do przeprowadzenia estymacji parametrów modelu przy założeniu prawdopodobieństwa wystąpienia impulsu na poziomie odpowiednio 0.33, 0.34, 0.33.

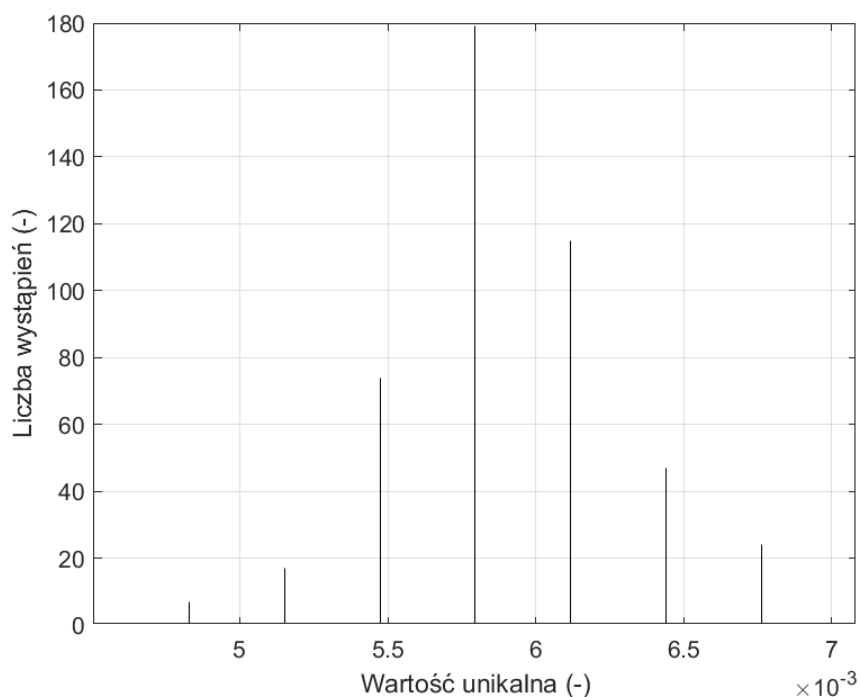
Rozdział 5

Estymacja parametrów układu

Poniższy opis przedstawia przebieg identyfikacji układu przeprowadzonej za pomocą algorytmu opracowanego w środowisku MATLAB. Algorytm realizuje szereg operacji takich jak wybór wartości początkowych oraz estymację wykonaną w celu szacowania parametrów oscylatora.

5.1 Wyznaczenie wartości średniej dla odpowiedzi układu RLC na zadane wymuszenie

Ze względu na obecność szumu zmieniającego wartość początkową, a tym samym zmieniającą się amplitudę odpowiedzi, zdecydowano się wykonać rozkład empiryczny wartości próbki sygnału poprzedzającej wystąpienie odpowiedzi na wymuszenie (rysunek 5.1).

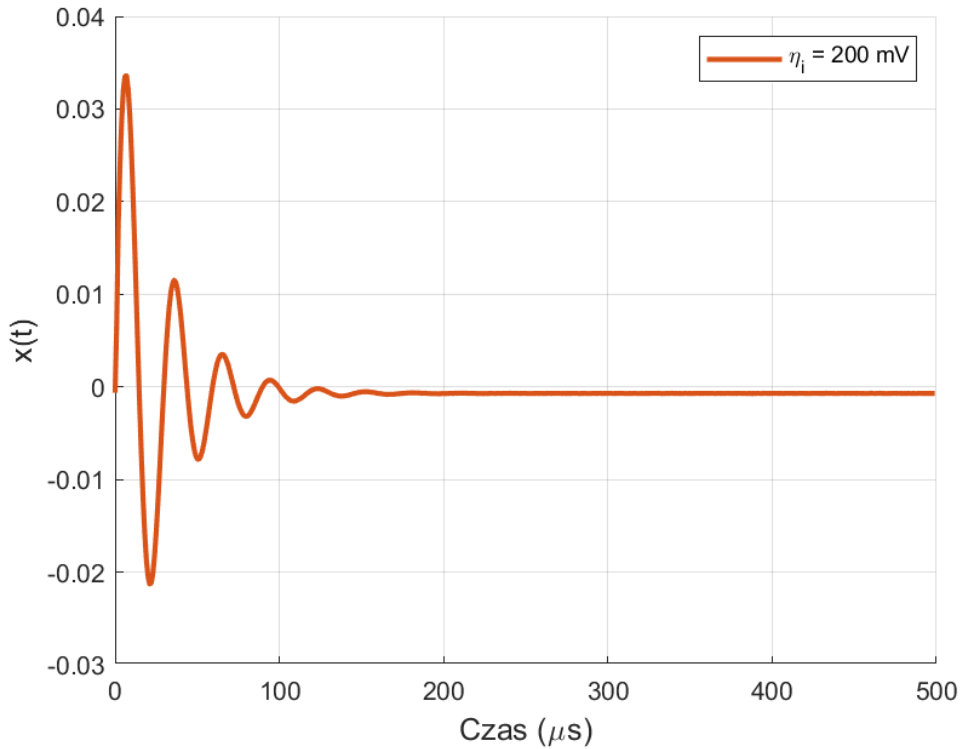


Rysunek 5.1: Wykres słupkowy dla wartości początkowej o wielkości impulsu 200 mV.

W celu graficznej reprezentacji rozkładu, generowany jest wykres, który przedstawia częstość występowania poszczególnych wartości. Oś X odpowiada unikalnym wartościom, a oś Y liczbie ich wystąpień. Do dalszej analizy wybierane są najczęściej pojawiające się wartości sygnału szumu.

W następnym kroku wyznaczono wartości średnie odpowiedzi układu RLC na wymuszenia 100 mV, 200 mV oraz 300 mV.

Na rysunku 5.2 pokazano wartość średnią z odpowiedzi układu, która będzie wykorzystana do przybliżenia wartości charakterystycznych równania opisującego odpowiedź układu RLC.



Rysunek 5.2: Wartość średnia dla wymuszenia o wartości 200 mV.

5.2 Wyznaczenie czasu propagacji

W celu wyznaczenia długości propagacji sygnału przeprowadzono estymację punktową oraz przedziałową wartości średniej dla każdej chwili w przedziale $t_i \in [0, 250] \mu s$ z krokiem $1 \mu s$, co odpowiada częstotliwości próbkowania $f_s = 1 \text{ MHz}$. W badaniach uwzględniono łącznie 251 estymatorów wartości średniej. Analiza ta jest niezbędna ze względu na drobne fluktuacje odpowiedzi w obrębie samej krzywej reprezentującej odpowiedź (rysunek 4.5 (C) i (D)) na wymuszenie.

Do oceny jakości estymacji wykorzystano błąd względny estymacji punktowej— wyrażony współczynnikiem zmienności, zgodnie z równaniem (5.1),

$$V = \frac{\hat{s} \cdot 100\%}{\bar{m}\sqrt{n}} \quad (5.1)$$

gdzie:

$\hat{s}$ – odchylenie standardowe z próby,

$\bar{m}$ – średnia arytmetyczna z próby,

n – liczebność próby.

Dodatkowo wyznaczono względną precyzję oszacowania dla estymacji przedziałowej — zgodnie z równaniem (5.2).

$$RP = \frac{u_{(1-\alpha/2)} \cdot \hat{s} \cdot 100\%}{\bar{m}\sqrt{n}} \quad (5.2)$$

gdzie:

$\hat{s}$ – odchylenie standardowe z próby,

$\bar{m}$ – średnia wartość z próby,

n – liczebność próby,

$u_{(1-\alpha/2)}$ – wartość odwrotnej dystrybuanty rozkładu normalnego dla poziomu istotności $\alpha = 0.05$.

W pracy przyjęto, że estymator uznaje się za niewiarygodny, jeśli wartość błędu względnego przekracza 15% lub jeśli względna precyzja oszacowania przekracza 10%. Na tej podstawie wyznaczono długość propagacji sygnału, przedstawioną w tabeli 5.1.

Tabela 5.1: Czas propagacji sygnału w zależności od wartości wymuszenia

Wymuszenie (mV)	Długość sygnału (μs)
100	59
200	89
300	121

Długość próbki sygnału przyjętej do dalszego procesu identyfikacji parametrów została określona przez zmienną `dlugosc_impulsu`, a jej wartość została wyznaczona na podstawie powyższej analizy wiarygodności estymatorów.

5.3 Zgrubna identyfikacja parametrów

W algorytmie przybliżania wartości charakterystycznych równania zastosowano podejście bazujące na dopasowaniu funkcji do eksperymentalnie uzyskanych danych. Analizie poddano zdefiniowaną wcześniej średnią odpowiedź układu RLC na losowe wymuszenie impulsem o wielkości 200 mV, dla której zastosowano metodę najmniejszych kwadratów (`lsqcurvefit`) w celu znalezienia najlepiej dopasowanych parametrów opisujących zachowanie układu.

Celem tej funkcji jest minimalizacja sumy kwadratów różnic między wartościami rzeczywistymi, a wartościami przewidywanymi przez model. Funkcja działa poprzez iteracyjny proces optymalizacji, w którym parametry modelu są stopniowo modyfikowane w celu minimalizacji tych różnic [41]. Funkcja `lsqcurvefit` wykorzystuje algorytmy optymalizacyjne, takie jak metoda Gaussa–Newtona, metoda Levenberga–Marquardta czy metoda *trust-region-reflective*, aby znaleźć najlepsze dopasowanie.

Do opisu rzeczywistego układu RLC przyjęto parametry stałe, określające jego fizyczne właściwości:

- Indukcyjność: $L = 22.7 \times 10^{-6}$ H,
- Pojemność: $C = 940 \times 10^{-9}$ F,
- Przedział czasu bezpośrednio związany z czasem propagacji sygnału:

$$t = 1 \times 10^{-6} \text{ s do } (\text{dlugosc_impulsu} - 1) \times 10^{-6} \text{ s, krok } 10^{-6} \text{ s.}$$

Ze względu na pojawienie się składowej stałej (różnej od zera) wynikającej z szumu przed i po zarejestrowanej odpowiedzi należało odpowiednio dostosować model matematyczny (równanie 3.17). Dodatkowo ze względu na niezerowe warunki początkowe należało uwzględnić przesunięcie fazowe φ .

Model opisujący drgania w układzie RLC można wyrazić za pomocą równania:

$$x(t) = ae^{-bt} \sin(\omega t + \varphi) + C \quad (5.3)$$

gdzie:

a – amplituda drgań, gdzie $a = \frac{\eta}{c}$,

b – współczynnik tłumienia,

ω – częstość drgań tłumionych,

φ – przesunięcie fazowe wynikająca z niezerowych warunków początkowych,

C – składowa stała wynikająca z szumu,

t – czas.

Wartości a , b , ω , φ , C są parametrami podlegającymi dopasowaniu. Początkowe wartości parametrów przyjęte w analizie odpowiedzi na wymuszenia o wartości 200 mV to odpowiednio:

$$a = 0.0432,$$

$$b = 20\,030,$$

$$\omega = 215\,550,$$

$$\varphi = 0.01548,$$

$$C = -0.000679.$$

Dopasowanie realizowane jest za pomocą funkcji ze środowiska MatLAB:

$$\text{params}(1,:) = \text{lsqcurvefit}(\text{fun}, \text{wartosc_początkowa}, t, \text{wartosc_średnia})$$

gdzie:

params – wektor zawierający wartości dopasowanych parametrów a , b , ω , φ , C ,

fun – funkcja modelu (równanie 5.3),

$\text{wartosc_początkowa}$ – początkowe wartości parametrów, t – wektor czasu,

wartosc_średnia – średnia odpowiedź układu RLC na wymuszenie.

Po dopasowaniu, uzyskane parametry modelu umożliwiają określenie współczynnika tłumienia oraz częstości drgań tłumionych układu. Porównanie tych wartości z wartościami wynikającymi z teorii dla układu RLC pozwala zweryfikować poprawność przeprowadzonej identyfikacji, a tym samym wstępnie

oszacować zakres danych początkowych dla właściwego przybliżenia danych. Niestety zgrubna identyfikacja wskazywała na błędy związane z relacjami pomiędzy tłumieniem b , a częstością drgań tłumionych ω , stąd przystąpiono do precyzyjnej identyfikacji parametrów.

5.4 Precyzyjna identyfikacja parametrów

Pierwszy etap przybliżania wartości charakterystycznych modelu umożliwił zawężenie zakresu poszukiwanych wartości, co znacznie poprawiło działanie całego algorytmu. Przed wykonaniem głównej pętli służącej do identyfikacji wartości równania definiowane są parametry pojemności (C) oraz indukcyjności (L) oraz zakres częstości drgań tłumionych, w której będzie dokonana próba dopasowania parametrów modelu.

Model opisujący drgania oscylatora został sprowadzony do postaci funkcji dopasowania (5.3) ze wskazaniem, które wielkości będą identyfikowane w głównej pętli optymalizacji. W tym przypadku przyjęto, że dla otrzymanej wartości ω ustawiono zakres przeszukiwania indeksów od 208 000 do 219 000 z krokiem 0.1.

W głównej pętli programu dla każdej wartości z wcześniej zdefiniowanego zakresu ω obliczana jest wartość tłumienia zgodnie z równaniem (5.4).

$$b = \sqrt{\frac{1}{LC} - \omega^2} \quad (5.4)$$

W tym miejscu następuje dostrajanie parametrów modelu za pomocą wcześniej wspomnianej funkcji `lsqcurvefit`, a dane z dopasowania przechowywane są we wcześniej zadeklarowanej zmiennej.

Identyfikacja parametrów modelu to tylko jeden z etapów analizy danych i nie kończy się on na uzyskaniu konkretnych wartości parametrów modelu, które zapisywane są w oddzielnym pliku tekstowym. Niezbędnym krokiem jest ocena jakości dopasowania, która weryfikuje, jak dobrze model odwzorowuje dane eksperymentalne. W tym celu wyznaczono wartości błędu względnego (E_{rel}), błędu bezwzględnego (E_{abs}), błędu średniokwadratowego (MSE) oraz błąd maksymalny (E_{max}). Dla każdej iteracji ze zdefiniowanego zakresu częstości drgań tłumionych wyznaczane są wartości błędów i są również przechowywane w tej samej macierzy, w której znajdują się współczynniki funkcji dopasowania.

Błąd względny (5.5) jest miarą stosunku błędu bezwzględnego do wartości pochodzącej z eksperymentu, wyrażonego jako procent lub ułamek. W badaniach nad układami RLC, wyznaczanie błędu względnego jest szczególnie użyteczne, gdy dane mają zróżnicowane wartości, ponieważ umożliwia porównywanie błędów w różnych skalach. Błąd względny pozwala zrozumieć, jak bardzo (procentowo) odpowiedź układu odbiega od wartości rzeczywistej.

$$E_{rel} = \frac{1}{n} \sum_{i=1}^n \frac{|f(t_i) - \hat{f}(t_i)|}{|f(t_i)|} \quad (5.5)$$

gdzie $f(t_i)$ oznacza wartość sygnału pochodzącego z eksperymentu w chwili t_i , natomiast $\hat{f}(t_i)$ odpowiada wartości uzyskanej na podstawie modelu z wyestymowanymi parametrami. Parametr n określa liczbę dyskretnych punktów czasowych t_i , w których dokonano porównania wartości sygnału pochodzącego z eksperymentu i wartości modelowych, a $|\cdot|$ oznacza wartość bezwzględną.

Błąd bezwzględny (5.6) definiuje się jako wartość bezwzględną różnicy pomiędzy wartościami pochodzącymi z eksperymentu a prognozowanymi. W kontekście badania odpowiedzi układu RLC, wyznaczenie błędu bezwzględnego umożliwia ilościową ocenę wielkości odchylenia dopasowania funkcji modelowej względem wartości rzeczywistych.

$$E_{abs} = \frac{1}{n} \sum_{i=1}^n |f(t_i) - \hat{f}(t_i)| \quad (5.6)$$

Błąd średniokwadratowy (MSE) (5.7) jest jedną z najczęściej stosowanych miar jakości dopasowania modelu. Definiuje się go jako średnią arytmetyczną kwadratów błędów, czyli różnic pomiędzy wartościami pochodzącymi z eksperymentu a prognozowanymi. Ze względu na podnoszenie różnic do kwadratu, metryka ta silniej karze duże odchylenia, co pozwala na ich lepsze uwzględnienie w procesie oceny jakości modelu.

$$MSE = \frac{1}{n} \sum_{i=1}^n (f(t_i) - \hat{f}(t_i))^2 \quad (5.7)$$

Błąd maksymalny (5.8) definiuje się jako największe odchylenie prognozy od wartości pochodzącej z eksperymentu w całym zbiorze danych. Metryka ta, jest użyteczna w analizie odpowiedzi układu RLC, gdyż pozwala na identyfikację skrajnych błędów, które mogą wskazywać na problemy z modelem.

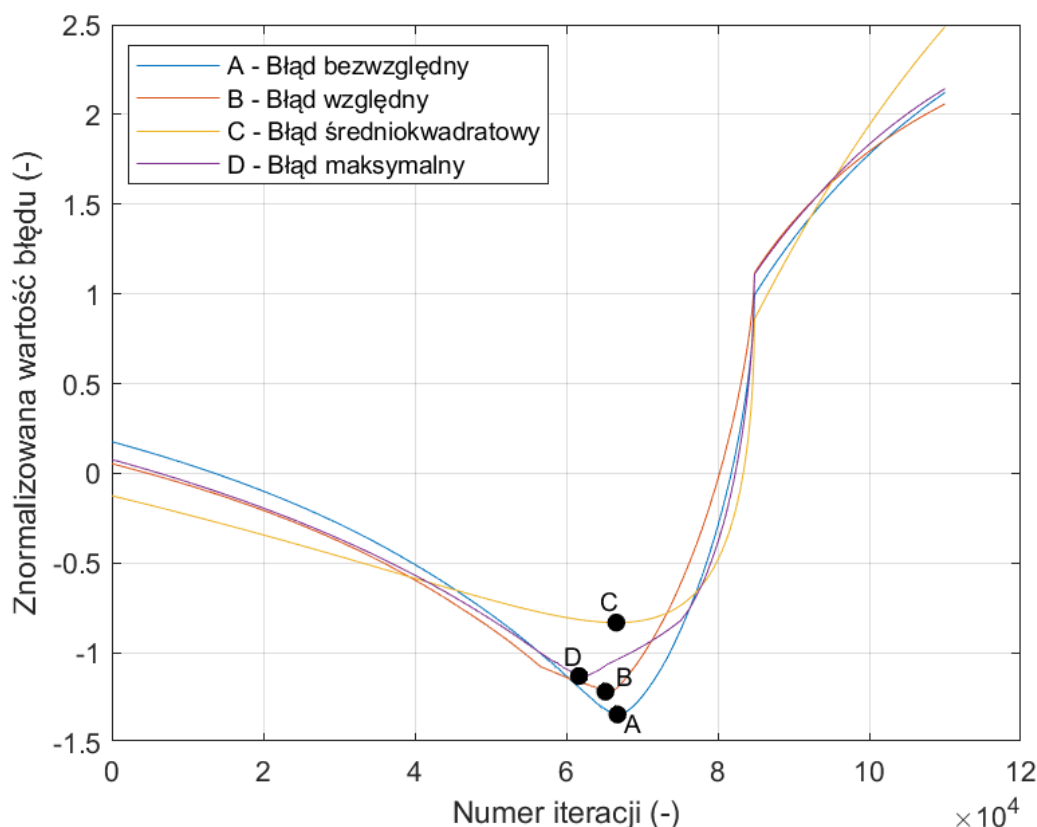
$$E_{max} = \max_i |f(t_i) - \hat{f}(t_i)| \quad (5.8)$$

Obliczenie wskazanych błędów pozwala wyznaczyć wartości minimalne dla każdego z nich i tym samym zbadać współczynniki modelu.

5.5 Szacowanie parametrów równania dla wymuszenia 200 mV

Chcąc dokonać porównania różnych miar błędów, konieczne było zastosowanie normalizacji wartości błędu. Normalizacja jest procesem przekształcania danych w taki sposób, aby znajdowały się one w określonym przedziale, umożliwiając ich porównanie mimo różnic skali [34]. W przypadku analizy błędów stosuje się najczęściej normalizację *min – max* lub normalizację względem średniej i odchylenia standardowego. Dzięki temu wszystkie wartości błędów zostają sprowadzone do tego samego przedziału, co pozwala na ich bezpośrednie zestawienie. Bez normalizacji różnice w wartościach poszczególnych błędów mogłyby prowadzić do mylących wniosków dotyczących jakości szacowania parametrów modelu.

Rysunek 5.3. przedstawia przebieg zmian wartości czterech różnych błędów – błędu bezwzględnego, względnego, średniokwadratowego oraz maksymalnego – w procesie szacowania parametrów modelu układu RLC. Wartości tych błędów (minima lokalne) powinny przecinać się w tym samym punkcie w przypadku idealnego dopasowania. Analizując wykres, można zauważyć, że każda z prezentowanych funkcji przyjmuje różne wartości minimalne w innym punkcie iteracji, co oznacza, że szukanie parametrów w oparciu o różne kryteria błędu może prowadzić do odmiennych wyników. Wartości błędów początkowo maleją, a następnie osiągają lokalne minima, po czym ponownie rosną. Na rysunku 5.3 zaznaczono wartości minimalne dla każdej z funkcji błędów.



Rysunek 5.3: Wartości błędów znormalizowanych w poszczególnych iteracjach dla wymuszenia 200 mV.

Błąd bezwzględny (A) osiąga swoje minimum w 66 764 iteracji. Jest to punkt, w którym wartości parametrów modelu są najbardziej zbliżone do eksperymentalnych wartości, przy minimalnej sumie bezwzględnych różnic pomiędzy wartościami z eksperymentu, a wartościami pochodzącymi z estymacji.

Błąd względny (B) przyjmuje minimum w nieco wcześniejszym punkcie niż błąd bezwzględny. Może to oznaczać, że względna dokładność przyjętych wartości modelowych stabilizuje się szybciej niż ich dokładność absolutna.

Błąd średniokwadratowy (C) wykazuje minimum w zbliżonym przedziale co błąd bezwzględny, ale z nieznacznie przesuniętym punktem minimalnym. Błąd ten jest bardziej wrażliwy na wartości odstające, co może wyjaśniać różnice w położeniu minimów.

Odległość maksymalna (D) osiąga swoje minimum jako pierwsza spośród analizowanych funkcji błędu. Oznacza to, że największe pojedyncze odchylenie od wartości rzeczywistej zmniejsza się najszybciej.

W tabeli 5.2 przedstawiono wartości trzech parametrów modelu układu RLC: tłumienia, częstości drgań tłumionych (ω) oraz wartości (a), wyznaczone dla pokazanych na rysunku 5.3 minimów lokalnych czterech różnych funkcji błędu: błędu bezwzględnego (A), błędu względnego (B), błędu średniokwadratowego (C) oraz maksymalnej odległości (D).

Tłumienie jest parametrem określającym szybkość wygaszania drgań w układzie RLC. Uzyskane wartości mieszczą się w przedziale **27 907,607 – 31 577,143**. Najniższa wartość tłumienia (**27 907,607**) została

Tabela 5.2: Wartości tłumienia, częstotliwości oraz amplitudy wyznaczonych dla minimów lokalnych dla wymuszenia 200 mV

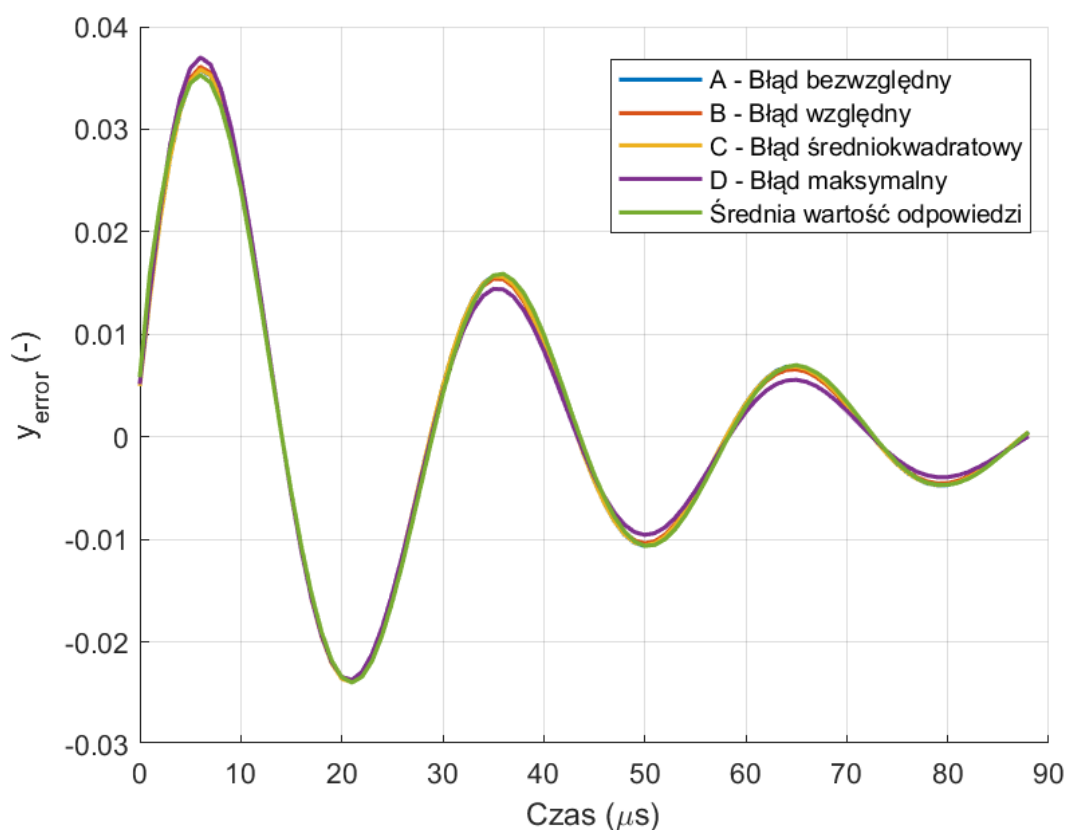
Funkcja błędu	Tłumienie b	ω	Amplituda a
A - Błąd bezwzględny	27 907.607	214 676.300	0.043005
B - Błąd względny	28 754.542	214 564.500	0.043648
C - Błąd średniokwadratowy	28 027.348	214 660.700	0.043096
D - Błąd maksymalny	31 577.143	214 167.300	0.045699

uzyskana przy minimalizacji błędu bezwzględnego (A). Natomiast najwyższa wartość (**31 577,143**) odpowiada minimalizacji maksymalnej odległości (D).

Wartości tłumienia dla błędu względnego (B) i średniokwadratowego (C) wynoszą odpowiednio **28 754,542** i **28 027,348**, co oznacza, że metody te dają wyniki pośrednie i są zbliżone.

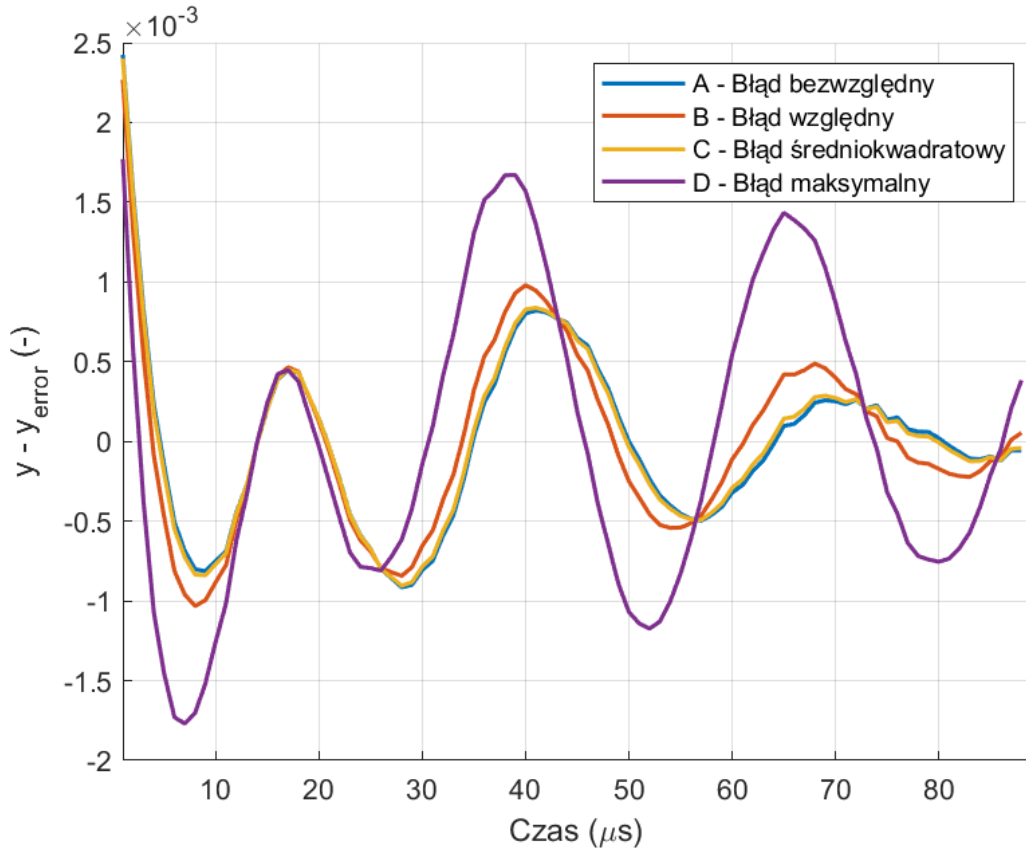
Uzyskane wartości częstości drgań tłumionych oscylują wokół **214 167,300 – 214 676,300**.

Uzyskane wartości dla parametru a zawierają się w zakresie **0,043005 – 0,045699**.



Rysunek 5.4: Wartość średnia przedstawiona względem parametrów szacowanych z błędów dla wymuszenia 200 mV.

Rysunek 5.4 przedstawia średnią wartość odpowiedzi (oznaczoną linią zieloną) nałożoną na cztery krzywe powstałe z podstawienia parametrów pochodzących z wyznaczenia minimów lokalnych błędów.



Rysunek 5.5: Różnica pomiędzy średnią odpowiedzią układu, a krzywą uzyskaną z parametrów modelu wyznaczonych dla minimów lokalnych błędów dla wymuszenia 200 mV.

Rysunek 5.5 przedstawia różnicę pomiędzy średnią wartością odpowiedzi układu RLC na zadane wymuszenie, a odpowiedzią, która powstała na podstawie parametrów wyznaczonych dla minimów lokalnych różnych funkcji błędu. Oscylacyjny charakter krzywych sugeruje, że model odtwarza dynamikę układu, jednak w pewnych obszarach występują lokalne różnice w amplitudzie i fazie. Największe różnice między modelem, a danymi pochodzącymi z eksperymentu pojawiają się w momentach ekstremów funkcji odpowiedzi, co może sugerować drobne niedokładności w estymacji amplitudy lub tłumienia. Wszystkie metody wykazują zbliżony przebieg błędu, co wskazuje na spójny proces przybliżania odpowiedzi układu.

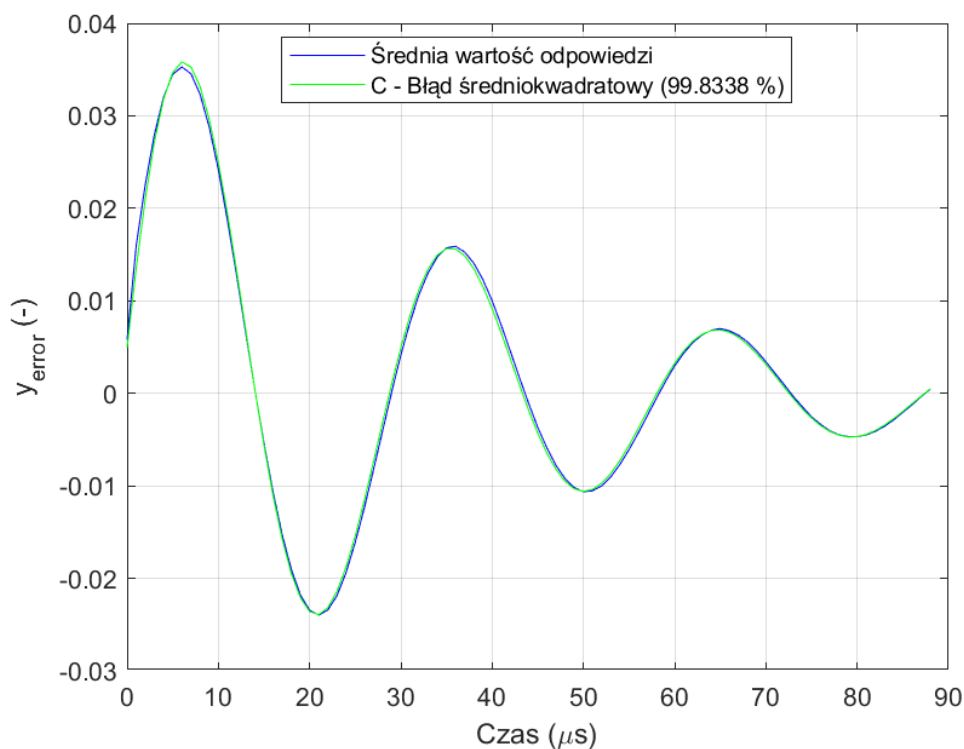
Krzywa odpowiadająca maksymalnej odległości (fioletowa) wykazuje największe odchylenia od pozostałych krzywych. Krzywa powstała z parametrów definiowanych przed minimum lokalne błędu średniokwadratowego (żółty) oraz błędu bezwzględnego (niebieski) mają niemal identyczny kształt, co świadczy o ich podobnym wpływie na wybór parametrów modelu.

Różnice pomiędzy średnią odpowiedzią układu RLC, a krzywymi wyznaczonymi z parametrów uzyskanych dla prezentowanych błędów mieszczą się w zakresie $\pm 2.5 \times 10^{-3}$, co oznacza, że model dobrze odtwarza rzeczywistą odpowiedź układu, a różnice są niewielkie.

Na podstawie przeprowadzonych badań, w tym analizy wykresów błędów identyfikacji (rysunek 5.3 oraz tabeli wyznaczonych parametrów modelu (tabela 5.2), można dokonać oceny skuteczności każdej z metod minimalizacji błędu. Błędy bezwzględny, względny i średniokwadratowy dają bardzo zbliżone wartości.

Błąd maksymalnej odległości charakteryzuje się nieco innym przebiegiem odpowiedzi, co sugeruje, że jego minimalizacja może prowadzić do większych lokalnych odchyleń. Stosunkowo niskie wartości błędów w tabeli wskazują na dobrą jakość identyfikacji dla wszystkich metod.

Analizując wykres różnic między modelem, a rzeczywistą odpowiedzią układu, należy zauważyć, że oscylacyjny charakter różnic wskazuje na drobne przesunięcia fazowe i amplitudowe w modelowanej odpowiedzi.



Rysunek 5.6: Reprezentacja krzywej dopasowanej na podstawie analizy błędów dla wymuszenia 200 mV.

Biorąc pod uwagę uzyskane wyniki, błąd średniokwadratowy (C) oraz błąd bezwzględny (A) wydają się najlepiej odpowiadać wymaganiom identyfikacji parametrów równania modelu oscylatora. Oba te błędy prowadzą do pewnych wartości parametrów tłumienia i częstości drgań tłumionych oraz minimalizują globalne różnice między modelem a rzeczywistą odpowiedzią układu.

Rozważając przypadek, w którym napięciem wymuszającym była wartość 200 mV, najlepszym dopasowaniem parametrów modelu jest błąd średniokwadratowy (C), który zapewnia identyfikację parametrów równania modelu układu RLC. Dopasowanie to przedstawiono na rysunku 5.6.

5.6 Szacowanie parametrów równania dla wartości wymuszeń 100 mV i 300 mV

Wykorzystując proces identyfikacji przeprowadzony wcześniej dla wymuszenia 200 mV, dokonano analogicznej analizy dla wartości 100 mV i 300 mV. Parametry uzyskane w tym procesie przedstawiono odpowiednio w tabeli 5.3 oraz tabeli 5.4.

Tabela 5.3: Wartości tłumienia, częstotliwości oraz amplitudy wyznaczone dla minimów lokalnych dla wymuszenia 100 mV

Funkcja błędu	Tłumienie b	ω	Amplituda a
A – Błąd bezwzględny	27 835,975	214 685,600	0,021543
B – Błąd względny	27 181,854	214 769,400	0,021315
C – Błąd średniokwadratowy	27 842,144	214 684,800	0,021545
D – Błąd maksymalny	21 464,817	215 415,900	0,019240

Dla wymuszenia 100 mV wartości tłumienia mieszczą się w stosunkowo wąskim zakresie (od 21 464 do 27 842). W szczególności:

- Metoda A (błąd bezwzględny) daje wartość 27 835,975,
- Metoda B (błąd względny) – 27 181,854,
- Metoda C (błąd średniokwadratowy) – 27 842,144,
- Metoda D (błąd maksymalny) – najniższą wartość, 21 464,817.

Wartość częstotliwości tłumionej ω pozostaje także zbliżona. To oznacza, że parametry są mało wrażliwe na zmianę funkcji błędu w przypadku niskich wartości wymuszenia.

Analiza wartości amplitudy a wykazuje niewielkie różnice – mieści się ona w zakresie od 0,019 do 0,0215. Najniższą wartość daje metoda D, a najwyższą – metoda C.

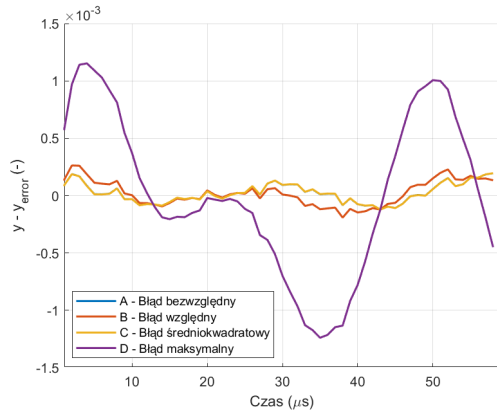
Tabela 5.4: Wartości tłumienia, częstotliwości oraz amplitudy wyznaczone dla minimów lokalnych dla wymuszenia 300 mV

Funkcja błędu	Tłumienie b	ω	Amplituda a
A – Błąd bezwzględny	27 413,9323	214 739,900	0,061809
B – Błąd względny	33 401,2743	213 890,400	0,068427
C – Błąd średniokwadratowy	28 232,6094	214 633,800	0,062779
D – Błąd maksymalny	53 212,2638	209 840,900	0,084764

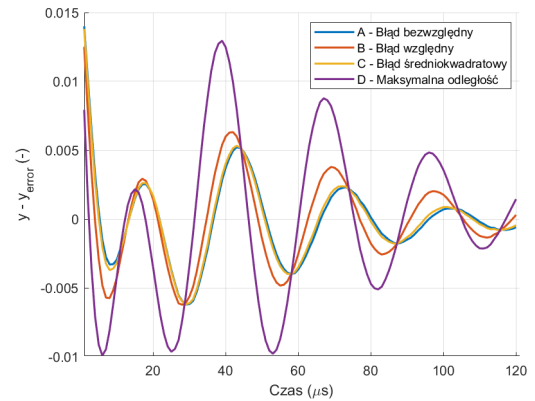
Dla wymuszenia 300 mV wartości tłumienia są znacznie wyższe i mieszczą się w szerszym zakresie (od 27 413 do 53 212), co wskazuje na większą czułość układu na wybraną metodę minimalizacji błędu. W szczególności:

- Metoda A (błąd bezwzględny) daje wartość 27 413,9323,
- Metoda B (błąd względny) – najwyższą spośród metod A–C, 33 401,2743,
- Metoda C (błąd średniokwadratowy) - daje wartość 28 232,6094,
- Metoda D (błąd maksymalny) – zdecydowanie najwyższą wartość, 53 212,2638.

W przypadku amplitudy a zakres wartości mieści się od 0,0618 do 0,0848, czyli jest ponad trzykrotnie wyższy niż dla 100 mV. Najniższą wartość ponownie daje metoda A, natomiast najwyższą – metoda D. Tak duże różnice pomiędzy parametrami mogą oznaczać zmianę charakteru odpowiedzi układu przy wyższych wartościach tłumienia. Analogicznie do przypadku przedstawionego na rysunku 5.4 przeprowadzono porównanie wartości średniej z krzywymi uzyskanymi z parametrów modelu, odpowiadającymi minimom lokalnym błędów. Różnice przedstawiono na rysunkach 5.7 i 5.8.

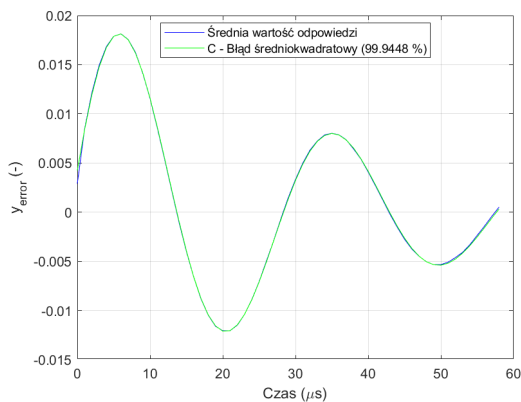


Rysunek 5.7: Różnica pomiędzy średnią odpowiedzią układu, a krzywą uzyskaną z parametrów modelu wyznaczonych dla minimów lokalnych błędów dla wymuszenia 100 mV.

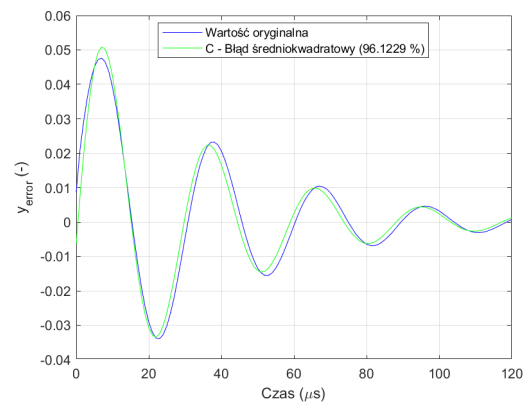


Rysunek 5.8: Różnica pomiędzy średnią odpowiedzią układu, a krzywą uzyskaną z parametrów modelu wyznaczonych dla minimów lokalnych błędów dla wymuszenia 300 mV.

Dla niższych wartości wymuszeń (100 mV oraz 200 mV) przybliżenie modelu przebiega z wysoką dokładnością – uzyskane krzywe pokrywają się ze średnią odpowiedzią układu, a różnica pomiędzy średnimi odpowiedziami układu, a krzywymi uzyskanymi z parametrów modelu wyznaczonych dla minimów lokalnych błędów są niewielkie. Najlepsze dopasowanie krzywej uzyskano dla parametrów wyznaczonych metodą minimalizacji błędu średniokwadratowego (rysunki 5.9 i 5.10).



Rysunek 5.9: Krzywe dopasowane na podstawie analizy błędu średniokwadratowego dla wymuszenia 100 mV.



Rysunek 5.10: Krzywe dopasowane na podstawie analizy błędu średniokwadratowego dla wymuszenia 300 mV.

Tabela 5.5: Porównanie parametrów modelu uzyskanych dla różnych funkcji błędu i poziomów wymuszenia

Wymuszenie	Funkcja błędu	Tłumienie b	ω	Amplituda a
100 mV	A – Błąd bezwzględny	27 835,975	214 685,600	0,021543
	B – Błąd względny	27 181,854	214 769,400	0,021315
	C – Błąd średniokwadratowy	27 842,144	214 684,800	0,021545
	D – Błąd maksymalny	21 464,817	215 415,900	0,019240
200 mV	A – Błąd bezwzględny	27 907,607	214 676,300	0,043005
	B – Błąd względny	28 754,542	214 564,500	0,043648
	C – Błąd średniokwadratowy	28 027,348	214 660,700	0,043096
	D – Błąd maksymalny	31 577,143	214 167,300	0,045699
300 mV	A – Błąd bezwzględny	27 413,932	214 739,900	0,061809
	B – Błąd względny	33 401,274	213 890,400	0,068427
	C – Błąd średniokwadratowy	28 232,609	214 633,800	0,062779
	D – Błąd maksymalny	53 212,264	209 840,900	0,084764

W tabeli 5.5 przedstawiono wartości parametrów modelu dla trzech poziomów wymuszeń: 100 mV, 200 mV i 300 mV. Wraz ze wzrostem wielkości wymuszenia obserwuje się wyraźny wzrost wartości amplitudy oraz tłumienia.

Na podstawie dotychczasowych analiz można stwierdzić, że model opisany równaniem (3.10) dobrze odwzorowuje odpowiedź układu w zakresie wymuszeń od 100 mV do 200 mV. Dla wartości 300 mV nie osiąga zakładanej amplitudy (rysunek 5.10). Dalsze zwiększanie amplitudy impulsów skutkuje pogorszeniem jakości dopasowania, co może wskazywać na pojawienie się efektów nieliniowych lub wypłaszczenia się odpowiedzi, tak jak dla odpowiedzi na impulsy powyżej 300 mV (rysunek 4.6). Zwiększenie długości próbki nie wpływa na poprawę dopasowania. Tak więc badania eksploracyjne wykazały, że równanie (3.10) można stosować jedynie dla wymuszeń w zakresie od 100 mV do 200 mV.

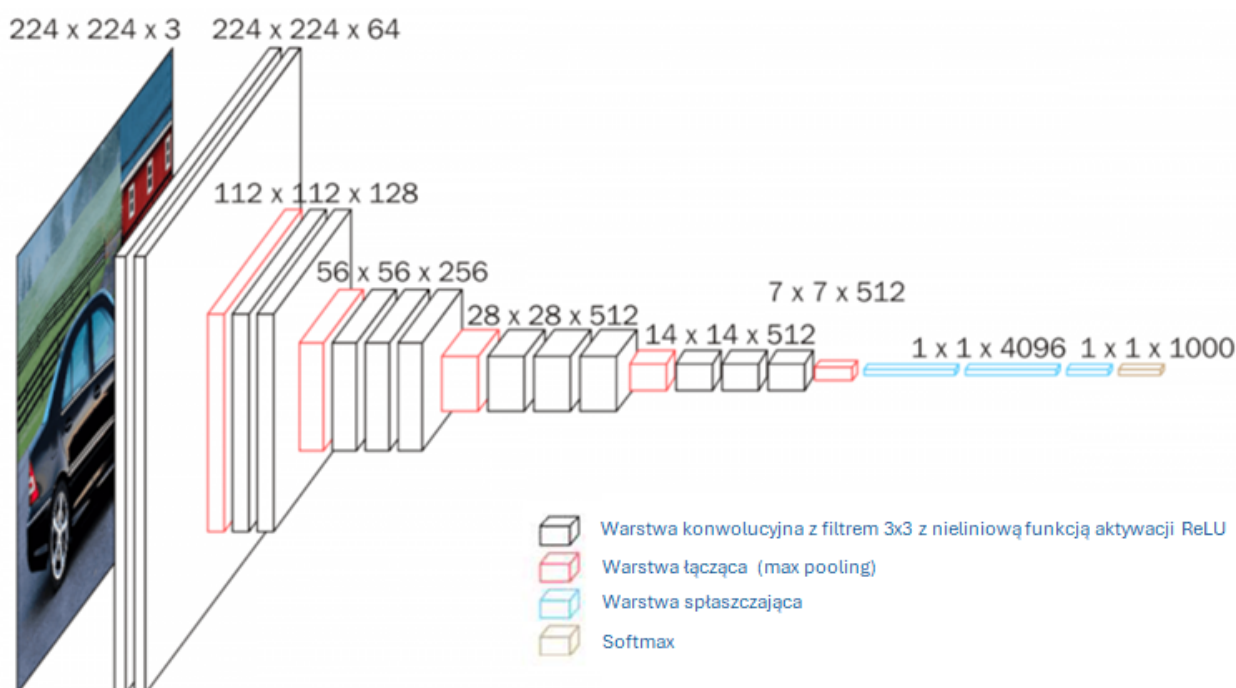
Dodatkowo, po przeprowadzeniu procesu identyfikacji parametrów dla tych samych wielkości wymuszeń, lecz o innej wartości prawdopodobieństwa wystąpienia impulsów, zaobserwowano zmianę parametrów tłumienia b , częstości drgań tłumionych ω oraz wartości amplitudy a , co szczegółowo opisano w rozdziale 7. Oznacza to, że nie tylko poziom amplitudy wymuszenia, ale również rozkład impulsów wpływa na dopasowanie modelu i wyznaczone parametry.

Zjawisko to wskazuje, że parametr b może w istotny sposób zależeć od rodzaju rozkładu wymuszeń. W konsekwencji modyfikacja samego modelu matematycznego w celu uwzględnienia wszystkich możliwych wariantów rozkładów byłaby zadaniem bardzo trudnym, a w praktyce niewykonalnym ze względu na nieskończoną liczbę potencjalnych przypadków. Z tego powodu do rozpoznawania odpowiedzi układu i klasyfikacji rozkładów impulsów konieczne staje się wykorzystanie metod sztucznej inteligencji.

Rozdział 6

Architektura VGG16 i VGG19

Architektury VGG16 oraz VGG19 zostały opracowane w 2014 roku przez Karen Simonyan i Andrew Zissermana z Uniwersytetu Oksfordzkiego [63]. Sieci powstały w ramach badań prowadzonych na potrzeby konkursu *ILSVRC – ImageNet Large Scale Visual Recognition Challenge*, stanowiącego jedno z najważniejszych wydarzeń w dziedzinie rozpoznawania obrazów [55].



Rysunek 6.1: Struktura sieci konwulucyjnej VGG [79].

Celem autorów było zbadanie, w jaki sposób zwiększanie głębokości sieci konwulucyjnej wpływa na jej zdolność do rozpoznawania złożonych wzorców w obrazach. W przeciwieństwie do wcześniejszych architektur, Simonyan i Zisserman postawili na prostotę budowy, opierając sieć niemal wyłącznie na warstwach konwulucyjnych z filtrami o rozmiarze 3×3 oraz warstwach *max pooling*. Dzięki takiemu podejściu uzyskano modele o dużej głębokości, w których wszystkie warstwy konwulucyjne były zbudowane według tej samej zasady czyli wykorzystano filtry 3×3 , przesunięcie filtru o 1, uzupełnienie zerami zapewniające

zachowanie wymiarów obrazu. Taka regularność oznaczała, że architektura była łatwa do implementacji i analizy, a jednocześnie umożliwiała stopniowe zwiększanie poziomu abstrakcji w reprezentacjach cech — od prostych krawędzi i tekstur w pierwszych warstwach, aż po złożone struktury obiektów w warstwach głębszych. Jednolita konstrukcja warstw pozwoliła także na systematyczne pogłębianie sieci, bez konieczności stosowania wielu różnych rozmiarów filtrów czy skomplikowanych połączeń, co wyróżniało VGG na tle bardziej nieregularnych architektur wcześniejszych generacji [63, 73, 37, 19].

Na rysunku 6.1 przedstawiono ogólną strukturę sieci VGG. VGG16 stanowi 13 warstw konwolucyjnych i 3 warstwy w pełni połączone, co daje razem 16 warstw z uczonymi parametrami. W przypadku VGG19 liczby te wynoszą odpowiednio 16 i 3, co łącznie daje 19 warstw. W obu przypadkach większość architektury stanowią warstwy konwolucyjne, których rolą jest automatyczne wyodrębnianie cech obrazu o rosnącym poziomie złożoności. Z kolei trzy ostatnie warstwy mają charakter w pełni połączony (*fully connected*) i pełnią funkcję klasyfikatora: pierwsze dwie warstwy w pełni połączone przekształcają spłaszczonej reprezentację cech w coraz bardziej uogólnione wektory cech o stałej długości (4096), co umożliwia wyodrębnienie semantycznych cech użytecznych dla klasyfikacji. To właśnie w tych końcowych warstwach koncentruje się największa liczba parametrów całej sieci, co sprawia, że są one istotne zarówno dla jej zdolności klasyfikacyjnych, jak i dla całkowitej złożoności obliczeniowej modelu.

W kontekście sieci konwolucyjnych przez *parametry* rozumie się wagi filtrów konwolucyjnych oraz połączeń w warstwach w pełni połączonych. To właśnie te wartości są modyfikowane w procesie uczenia, a ich odpowiednie dostosowanie pozwala sieci rozpoznawać złożone wzorce w danych wejściowych [36, 5]. Liczbę parametrów można oszacować, analizując rozmiar filtrów oraz liczbę neuronów w poszczególnych warstwach. Dla VGG16 przykładowe wartości są następujące:

- pierwsza warstwa konwolucyjna (*Conv1_1*) – około 1,8 tys. parametrów,
- warstwa konwolucyjna średniego poziomu (*Conv3_3*) – około 295 tys.,
- głębsza warstwa konwolucyjna (*Conv5_3*) – ponad 2,3 mln,
- pierwsza warstwa w pełni połączona (*FC1*) – około 102 mln,
- ostatnia warstwa klasyfikacyjna (*FC3*) – około 4,1 mln.

Widać zatem, że to przede wszystkim warstwy w pełni połączone (FC) odpowiadają za ogromną liczbę parametrów całej architektury. Łącznie daje to około 138 mln parametrów dla VGG16 i 144 mln dla VGG19. Tak duża liczba wag stanowi istotne wyzwanie obliczeniowe, ale jednocześnie pozwala sieciom na reprezentowanie bardzo złożonych zależności pomiędzy cechami obrazu.

Obie architektury uzyskały w 2014 roku czołowe wyniki klasyfikacji na zbiorze ImageNet, osiągając dokładność top-5 na poziomie 92,7% dla VGG16 i 93,6% dla VGG19 [63]. Sukces ten przyczynił się do ich popularyzacji zarówno w badaniach, jak i zastosowaniach praktycznych w dziedzinie widzenia komputerowego.

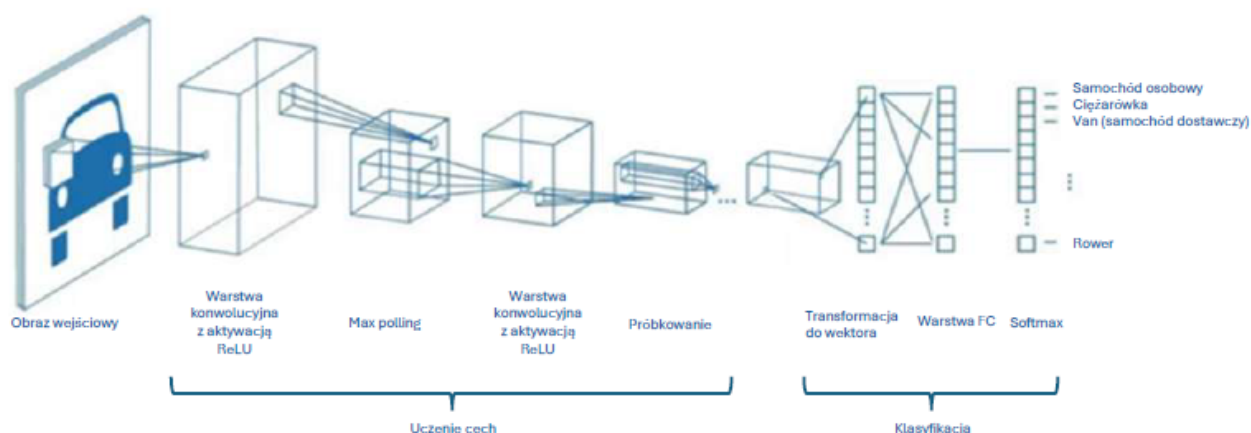
6.1 Schemat działania sieci

Na rysunku 6.2 przedstawiono uproszczony schemat konwolucyjnej sieci neuronowej, którego struktura odpowiada architekturze VGG [63]. Obraz wejściowy podawany jest kolejno do bloków konwolucyjnych, w których stosowane są filtry o niewielkim rozmiarze (3×3) oraz funkcje aktywacji typu ReLU. Takie podejście pozwala na uzyskiwanie złożonych reprezentacji cech przy ograniczonej liczbie parametrów, ponieważ kilka warstw konwolucyjnych z małymi filtrami może odwzorować ten sam obszar recepcyjny, co pojedyncza warstwa z filtrem większym (np. 5×5 czy 7×7), lecz przy znacznie mniejszej liczbie wag.

Po każdym bloku konwolucyjnym stosowana jest operacja próbkowania obrazu, najczęściej w postaci warstwy maksymalnego próbkowania (max pooling) o rozmiarze 2×2 . Jej zastosowanie prowadzi do redukcji wymiarów map cech, zwiększa odporność sieci na niewielkie przesunięcia czy zakłócenia w obrazie oraz pozwala skoncentrować uwagę modelu na najbardziej istotnych fragmentach danych, jednocześnie ograniczając ryzyko przeuczenia. Proces ten stanowi integralny element hierarchicznego uczenia cech i umożliwia stopniowe wyodrębnianie coraz bardziej abstrakcyjnych reprezentacji sygnału wejściowego.

W dalszej części procesu dane są transformowane do wektora (*flatten*) i przekazywane do warstw w pełni połączonych (*fully connected*). W architekturach VGG znajdują się trzy takie warstwy, które pełnią rolę klasyfikatora wysokiego poziomu. Ostatnia z nich jest zakończona funkcją softmax, która przekształca wynik w rozkład prawdopodobieństwa nad wszystkimi klasami. Na schemacie jako przykładowe klasy pokazano: *samochód*, *ciężarówka*, *van* oraz *rower*.

Warto podkreślić, że przedstawiony rysunek ma charakter poglądowy — pokazuje jedynie ogólny przepływ danych przez kolejne etapy sieci (konwolucje, pooling, warstwy w pełni połączone), a nie pełną liczbę warstw ani dokładny układ filtrów i połączeń. W rzeczywistości szczegółowy opis architektury VGG16 i VGG19 obejmuje odpowiednio 16 oraz 19 warstw wagowych. Schemat dobrze ilustruje jednak zasadniczą ideę przetwarzania danych w sieciach konwolucyjnych typu VGG i pozwala na intuicyjne zrozumienie logiki działania tej architektury.



Rysunek 6.2: Przepływ obrazu w sieci konwolucyjnej VGG dla przykładu obrazu samochodu [63, 33, 16, 82].

6.2 Opis warstw modelu

Architektury VGG zakładają zastosowanie wyłącznie filtrów konwolucyjnych o rozmiarze 3×3 , ze skokiem (*stride*) równym 1 oraz wypełnieniem typu *same* (padding), co pozwala zachować stałe wymiary map cech na każdym etapie przetwarzania. Po każdej operacji konwolucyjnej stosowana jest nieliniowa funkcja aktywacji ReLU. W celu redukcji rozmiaru przestrzennego map cech, po każdym bloku konwolucyjnym umieszczana jest warstwa MaxPool o oknie 2×2 przesuwanym co 2 piksele. Dzięki temu sieć może stopniowo wyodrębniać coraz bardziej złożone cechy, przy jednoczesnym ograniczeniu wymiarowości reprezentacji i zmniejszeniu liczby obliczeń [73].

Wejściem do sieci jest obraz o wymiarach 224×224 piksele w trzech kanałach RGB. Zastosowanie pięciu kolejnych warstw MaxPool prowadzi do systematycznej redukcji rozdzielczości przestrzennej map cech w następującej sekwencji:

$$224 \times 224 \rightarrow 112 \times 112 \rightarrow 56 \times 56 \rightarrow 28 \times 28 \rightarrow 14 \times 14 \rightarrow 7 \times 7.$$

Część klasyfikacyjna sieci VGG składa się z trzech w pełni połączonych warstw (*fully connected*, FC). Każda z dwóch pierwszych (FC6, FC7) zawiera 4096 neuronów z aktywacją ReLU, a dodatkowo stosowany jest w nich mechanizm Dropout, który redukuje ryzyko przeuczenia. Dropout to technika regularyzacji polegająca na losowym „wyłączaniu” (zerowaniu) części aktywacji w warstwie podczas uczenia, co ogranicza zjawisko współadaptacji neuronów i poprawia zdolność uogólniania modelu. W fazie uczenia model działa z losową próbą jednostek, co można interpretować jako trenowanie zespołu wielu „rzadszych” pod-sieci współdzielących wagi; w fazie wnioskowania (testu/inferencji) Dropout jest wyłączony i używa się pełnych aktywacji (bez maski), dzięki czemu uzyskuje się uśrednienie efektów wielu konfiguracji.

W praktyce Dropout najczęściej stosuje się w warstwach w pełni połączonych, gdzie korelacje między neuronami są silne. W warstwach konwolucyjnych używa się go rzadziej (z reguły mniejsze prawdopodobieństwo, np. 0,1–0,3), ponieważ lokalna struktura i współdzielenie wag pełnią częściowo rolę regularyzującą. Dropout uzupełnia inne metody ograniczania przeuczenia, takie jak regularyzacja L_2 , wczesne zatrzymanie (*early stopping*) czy *data augmentation*. W architekturach bez normalizacji partiami (Batch-Norm) Dropout bywa szczególnie skuteczny; w sieciach z intensywną normalizacją jego rola może być mniejsza, choć nadal bywa użyteczny w częściach klasyfikacyjnych. Ostatnia warstwa FC (FC8) ma liczbę neuronów odpowiadającą liczbie klas w danym zadaniu klasyfikacyjnym (np. 1 000 w przypadku zbioru ImageNet) i wykorzystuje funkcję aktywacji Softmax do wyznaczenia rozkładu prawdopodobieństwa przynależności do poszczególnych klas.

W przeciwieństwie do wcześniejszej architektury AlexNet, w której stosowano warstwy normalizacji lokalnej (LRN, ang. Local Response Normalization) w celu poprawy uogólniania modelu, w architekturach VGG zrezygnowano z tego rozwiązania. Autorzy uznali, że regularne stosowanie małych filtrów konwolucyjnych w połączeniu z funkcją aktywacji ReLU zapewnia wystarczającą stabilność procesu uczenia. Z tego względu dodatkowa normalizacja nie była konieczna.

Normalizacja danych wejściowych polegała natomiast na zamianie przestrzeni barw z RGB na BGR oraz odjęciu średnich wartości dla każdego kanału, wyznaczonych na podstawie zbioru ImageNet. Takie

podejście było zgodne z konwencją przyjętą w bibliotece Caffe i zostało przejęte do standardowych pakietów implementujących modele VGG.

6.2.1 Sieć VGG16

Architektura VGG16 składa się z pięciu kolejnych bloków konwolucyjnych, po których następuje część klasyfikacyjna złożona z trzech warstw w pełni połączonych. Taka konstrukcja pozwala na stopniowe zwiększanie liczby kanałów (map cech) i jednocześnie redukcję rozdzielczości przestrzennej. W praktyce oznacza to przejście od niskopoziomowych cech (np. krawędzie, tekstury) w pierwszych blokach, aż do cech wysokiego poziomu (kształty obiektów, układy części) w blokach głębszych. Poniżej przedstawiono szczegółowy opis działania poszczególnych bloków.

- **Blok 1 (64 kanały).** Wejściem jest obraz $224 \times 224 \times 3$ w przestrzeni RGB. Dwie warstwy konwolucyjne przy użyciu filtra 3×3 transformują dane do 64 map cech, a funkcja ReLU wprowadza nieliniowość. Dzięki temu sieć może odwzorować proste cechy, takie jak krawędzie czy kontrast lokalny. Warstwa maksymalnego próbkowania MaxPool redukuje wymiar przestrzenny z 224×224 do 112×112 , co zmniejsza liczbę obliczeń przy jednoczesnym zwiększeniu uogólnienia,
- **Blok 2 (128 kanałów).** Dwie kolejne warstwy konwolucyjne zwiększają liczbę map cech z 64 do 128 i pozwalają na rozpoznawanie bardziej złożonych kombinacji cech, np. układów krawędzi. Warstwa MaxPool redukuje wymiar do 56×56 ,
- **Blok 3 (256 kanałów).** Trzy warstwy konwolucyjne zwiększają liczbę map z 128 do 256, a funkcje ReLU po każdej konwolucji umożliwiają wyodrębnianie wielowymiarowych wzorców, takich jak fragmenty kształtów czy tekstury obiektów. Warstwa MaxPool zmniejsza wymiar do 28×28 ,
- **Blok 4 (512 kanałów).** Trzy warstwy konwolucyjne zwiększające liczbę map z 256 do 512 pozwalają uchwycić jeszcze bardziej abstrakcyjne cechy, np. części obiektów lub układy struktur w obrazie. Redukcja wymiaru przestrzennego do 14×14 pozwala zachować wydajność obliczeniową przy zwiększonej liczbie kanałów,
- **Blok 5 (512 kanałów).** Trzy kolejne warstwy konwolucyjne utrzymują liczbę map cech, ale pozwalają na bardzo bogate odwzorowanie cech wysokiego poziomu. Końcowa warstwa MaxPool zmniejsza wymiar przestrzenny do 7×7 , co przygotowuje dane do etapu klasyfikacji,
- **Część klasyfikacyjna.** Wynikiem części konwolucyjnej jest tensor o wymiarach $7 \times 7 \times 512$, który zostaje spłaszczony do wektora 25088 cech. Trafia on następnie do dwóch warstw w pełni połączonych (FC6, FC7), z których każda zawiera 4096 neuronów z aktywacją ReLU. W obu warstwach stosowany jest mechanizm Dropout z prawdopodobieństwem $p = 0,5$, który losowo „wyłącza” część neuronów w trakcie uczenia, co ogranicza ryzyko przeuczenia i poprawia zdolność uogólniania modelu. Ostatnia warstwa (FC8) ma C neuronów, gdzie C odpowiada liczbie klas (np. $C = 1000$ dla ImageNet), i wykorzystuje funkcję Softmax. Dzięki temu wektor cech zamieniany jest na rozkład prawdopodobieństwa przynależności do poszczególnych klas.

6.2.2 Sieć VGG19

Architektura VGG19 stanowi rozszerzenie VGG16. Zasadnicza struktura pozostaje taka sama: pięć bloków konwolucyjnych i część klasyfikacyjna z trzema warstwami w pełni połączonymi. Różnica polega na tym, że w blokach 3–5 dodano po jednej dodatkowej warstwie konwolucyjnej, dzięki czemu liczba warstw konwolucyjnych wzrosła z 13 do 16. Dodanie kolejnych konwolucji zwiększa głębokość modelu, pozwalając na uchwycenie jeszcze bardziej złożonych i hierarchicznych cech obrazu. Część klasyfikacyjna pozostaje identyczna jak w VGG16, ale cała sieć jest wymagająca obliczeniowo i zawiera około 6 milionów dodatkowych parametrów.

- **Blok 1 (64 kanały).** Wejściem jest obraz $224 \times 224 \times 3$. Dwie warstwy konwolucyjne przekształcają go w 64 mapy cech. Funkcje ReLU zapewniają nieliniowość i ułatwiają propagację gradientu. Operacja MaxPool zmniejsza wymiar przestrzenny do 112×112 . Podobnie jak w VGG16, blok ten odpowiada za detekcję najprostszych cech, takich jak krawędzie czy kontrast lokalny,
- **Blok 2 (128 kanałów).** Dwie kolejne warstwy konwolucyjne zwiększają liczbę map cech z 64 do 128, co umożliwia uchwycenie bardziej złożonych wzorców, takich jak zestawienia linii czy faktury. Po operacji MaxPool wymiar przestrzenny zmniejsza się do 56×56 ,
- **Blok 3 (256 kanałów).** W tym bloku występują już cztery warstwy konwolucyjne, każda z aktywacją ReLU. Dodanie czwartej konwolucji względem VGG16 zwiększa zdolność sieci do reprezentowania skomplikowanych kombinacji cech, takich jak fragmenty obiektów. Warstwa MaxPool redukuje rozdzielczość do 28×28 ,
- **Blok 4 (512 kanałów).** Podobnie jak w bloku 3, liczba warstw konwolucyjnych wynosi cztery. Każda przekształca dane w coraz bogatszy zestaw 512 map cech, umożliwiając uchwycenie struktur wysokiego poziomu, np. układów części obiektów. Warstwa MaxPool zmniejsza wymiar przestrzenny do 14×14 .
- **Blok 5 (512 kanałów).** Również w tym bloku występują cztery warstwy konwolucyjne, które pozwalają sieci na wyodrębnianie najbardziej abstrakcyjnych reprezentacji, silnie powiązanych z klasami obiektów. Końcowa warstwa MaxPool zmniejsza wymiar przestrzenny map cech do 7×7 ,
- **Część klasyfikacyjna.** Wynik pięciu bloków konwolucyjnych to tensor $7 \times 7 \times 512$, który po spłaszczeniu daje wektor 25088 cech. Dalej przebieg jest identyczny jak w VGG16: dwie warstwy w pełni połączone (FC6, FC7) zawierające po 4096 neuronów z aktywacją ReLU i mechanizmem Dropout, a następnie końcowa warstwa (FC8) z C neuronami i funkcją Softmax. Dzięki temu możliwe jest przypisanie obrazu do jednej z C klas.

Tabela 6.1: Zestawienie liczby warstw konwolucyjnych w architekturach VGG16 i VGG19

Model	Blok 1	Blok 2	Blok 3	Blok 4	Blok 5	Warstwy FC
VGG16 (D)	2×64	2×128	3×256	3×512	3×512	3
VGG19 (E)	2×64	2×128	4×256	4×512	4×512	3

6.3 Ocena skuteczności klasyfikacji

Skuteczność klasyfikatora ocenia się z wykorzystaniem standardowych metryk powszechnie stosowanych w zadaniach klasyfikacji wieloklasowej [72, 16, 5]. Analizie podlegają zarówno wskaźniki ogólne (np. dokładność całkowita), jak i szczegółowe miary liczone osobno dla każdej klasy. Do opisu jakości klasyfikacji wykorzystano następujące pojęcia: liczba próbek poprawnie zaklasyfikowanych do danej klasy — *True Positive (TP) Prawdziwie dodatni / Trafienie dodatnie (PD)*, liczba próbek prawidłowo rozpoznanych jako nienależące do tej klasy — *True Negative (TN) Prawdziwie ujemny / Trafienie ujemne (PU)*, liczba próbek błędnie przypisanych do danej klasy — *False Positive (FP) Fałszywie dodatni / Błąd I rodzaju (FD)* oraz liczba próbek danej klasy błędnie sklasyfikowanych jako inna kategoria — *False Negative (FN) Fałszywie ujemny / Błąd II rodzaju (FU)* [69, 74].

Przewidywana klasa = Pozytywna		Przewidywana klasa = Negatywna	
Pred. Positive = TP+FP		Pred. Negative = FN+TN	
Rzeczywista klasa = Negatywna Actual Negative = FP + TN	PD Prawdziwie dodatni	FU Fałszywie ujemny	FD Fałszywie dodatni
	PU Prawdziwie ujemny		

Metryki:

$$\text{Precyzja} = \frac{PU}{PU + FD}$$

$$\text{Czułość} = \frac{PD}{PD + FU}$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precyzja} \cdot \text{Czułość}}{\text{Precyzja} + \text{Czułość}}$$

$$\text{Dokładność} = \frac{PD + PU}{PD + FD + FU + PU}$$

Rysunek 6.3: Schemat macierzy pomyłek z definicjami metryk: precyzja, czułość, F1 i dokładność.

Na ich podstawie obliczane są m.in. następujące wskaźniki. Dokładność (Accuracy) to ogólny wskaźnik poprawności klasyfikacji, określający stosunek liczby poprawnie sklasyfikowanych przykładów do liczby wszystkich przykładów. W zadaniu wieloklasowym jest to stosunek sumy prawidłowych klasyfikacji do liczby wszystkich próbek w zbiorze. Precyzja (Precision) jest miarą tego, jaki odsetek przykładów przypisanych przez model do danej klasy rzeczywiście do niej należy. Wysoka precyzja oznacza niski udział fałszywych alarmów (FP). Czułość (Recall) to miara tego, jaki odsetek wszystkich przykładów danej klasy został poprawnie wykryty przez model. Wysoka czułość oznacza, że model rzadko pomija przypadki danej klasy. Ostatnia to wartość F1-score. Jest to miara skuteczności klasyfikatora będąca średnią harmoniczną precyzji i czułości.

6.4 Obraz wejściowy i wstępne przetwarzanie przykładowego obrazu

W celu zobrazowania różnic pomiędzy dwiema głębokimi architekturami konwolucyjnymi VGG16 i VGG19 przeprowadzono analizę działania obu modeli na wybranym obrazie testowym. Analiza odzwierciedla kolejne etapy przetwarzania w sieci konwolucyjnej. Na wejściu obie sieci wymagają obrazu o rozmiarach 224×224 piksele w przestrzeni barw RGB. W praktyce oznacza to konieczność przeskalowania obrazu źródłowego do tej rozdzielczości, przy zachowaniu proporcji barw. Jest to pierwszy krok, zapewniający jednolity format danych wejściowych i umożliwiający bezpośrednie podanie obrazu do sieci neuronowej. W celu porównania sieci VGG16 i VGG19 wykorzystano zdjęcie auta osobowego (rysunek 6.4) (źródło własne).



(a) Oryginalne zdjęcie wykorzystane do klasyfikacji (źródło własne).

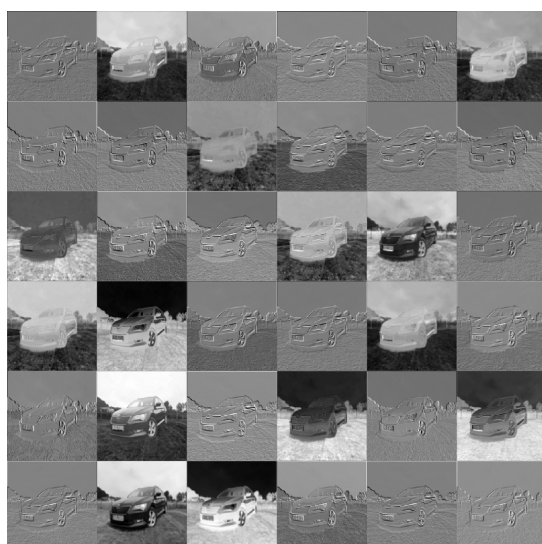


(b) Zdjęcie po konwersji do rozmiaru 224x224.

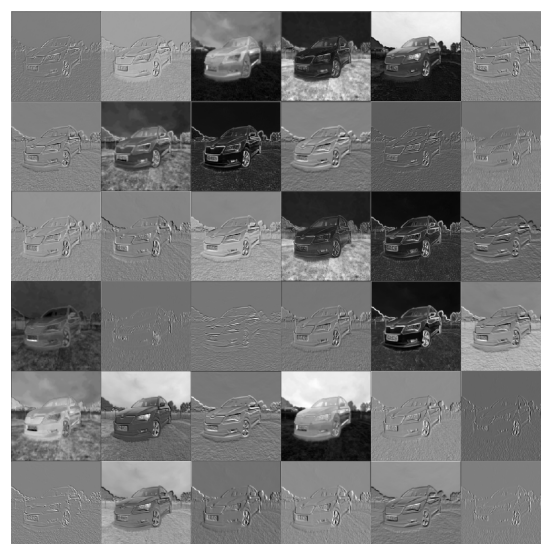
Rysunek 6.4: Obraz wejściowy do sieci.

6.4.1 Mapy cech dla wczesnych i głębokich warstw

Na początkowych poziomach architektury (np. `conv1_1` (rysunek 6.5a), `conv1_2` (rysunek 6.5b)) aktywacje odpowiadają przede wszystkim prostym cechom niskiego poziomu, takim jak krawędzie pionowe i poziome, kontrasty jasności czy elementarne wzorce tekstur. Na rysunku 6.5 przedstawiono pierwsze 36 map cech dla obu sieci w warstwach początkowych. Widać wyraźnie, że niezależnie od głębokości architektury, obie sieci wyodrębniają z obrazu podobne wzorce geometryczne i kontrastowe. Jest to zgodne z ogólną zasadą działania sieci: najniższe warstwy działają analogicznie do filtrów wykrywających krawędzie i tekstury w klasycznych metodach przetwarzania obrazu.



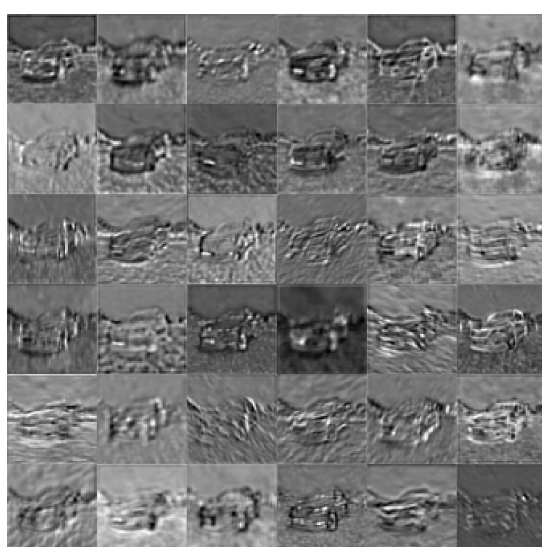
(a) Warstwa wczesna conv1_1 - pierwsze 36 map.



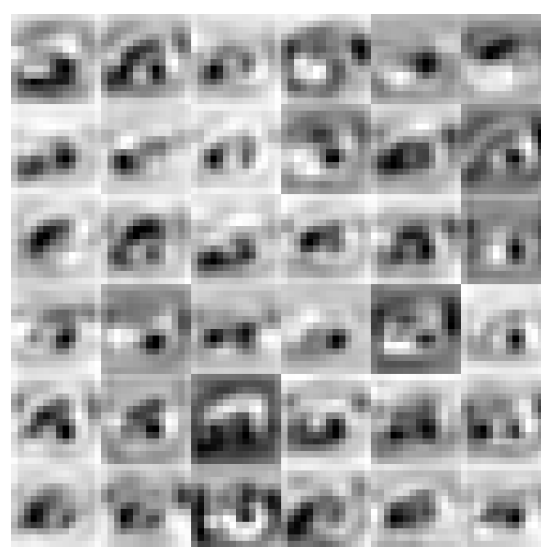
(b) Warstwa wczesna conv1_2 - pierwsze 36 map.

Rysunek 6.5: Wzorce krawędziowe i kontrastowe dla obu sieci dla początkowych warstw sieci.

W miarę przechodzenia w głąb sieci (np. conv3_3 (rysunek 6.6a), conv5_3 (rysunek 6.6b)) reprezentacje stają się coraz bardziej abstrakcyjne. Mapy cech w warstwach głębokich nie odpowiadają już pojedynczym krawędziom, lecz złożonym strukturom, takim jak fragmenty obiektów, układy geometryczne czy specyficzne wzorce charakterystyczne dla danej klasy. Rysunek 6.6 ukazuje pierwsze 36 aktywacji dla wybranych głębszych warstw obu architektur. Zauważalne jest, że VGG19 dzięki dodatkowym warstwom konwolucyjnym generuje bogatsze i bardziej różnicujące mapy cech. Obszary aktywne są bardziej zlokalizowane, a wzorce zawierają większą ilość szczegółów, co świadczy o zdolności sieci do lepszego modelowania zależności hierarchicznych [48, 63, 82, 62, 81, 59, 4].



(a) Warstwa głęboka conv3_3 - pierwsze 36 map.



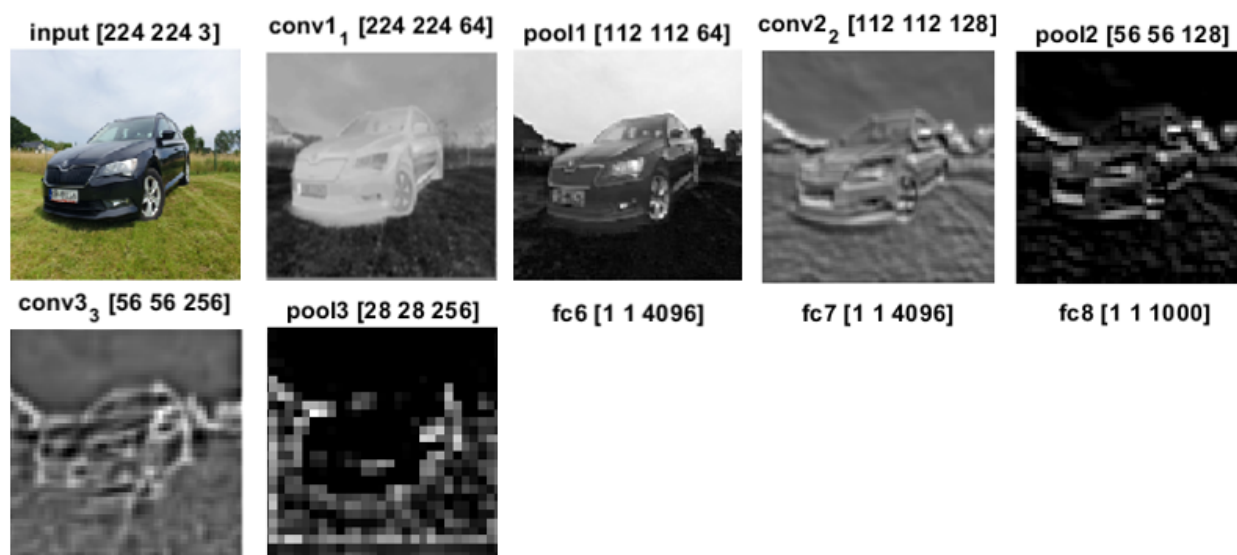
(b) Warstwa głęboka conv5_3 - pierwsze 36 map.

Rysunek 6.6: Wzorce krawędziowe i kontrastowe dla obu sieci dla głębszych warstw sieci.

Różnice pomiędzy VGG16 a VGG19 są subtelne na poziomie warstw początkowych, ale stają się wyraźniejsze w głębszych blokach konwolucyjnych. VGG16 tworzy bardziej ogólne i rozproszone wzorce, natomiast VGG19 potrafi uchwycić detale o wyższym stopniu złożoności. Wynika to bezpośrednio z architektury: dodanie dodatkowych warstw w blokach konwolucyjnych umożliwia stopniowe budowanie coraz bardziej abstrakcyjnych reprezentacji, które mogą poprawiać dokładność klasyfikacji.

6.4.2 Przepływ obrazu w sieci

Na rysunku 6.7 przedstawiono przepływ danych przez wybrane warstwy sieci VGG16, począwszy od obrazu wejściowego aż do końcowej warstwy klasyfikacyjnej. Każdy z etapów przekształca reprezentację obrazu, przechodząc od niskopoziomowych cech wizualnych do abstrakcyjnej reprezentacji semantycznej [63, 36, 16].



Rysunek 6.7: Przepływ obrazu w sieci dla obrazu samochodu: kolejne warstwy od wejścia do wyjścia.

Opis poszczególnych etapów:

1. **input [224 224 3]** – obraz wejściowy, przeskalowany do wymiarów wymaganych przez sieć (224×224 piksele, 3 kanały RGB). Dane te stanowią punkt startowy przetwarzania,
2. **conv1_1 [224 224 64]** – pierwsza warstwa konwolucyjna wyodrębnia podstawowe cechy obrazu, takie jak krawędzie i kontrasty. Obraz pozostaje czytelny, ale widoczna jest jego filtracja,
3. **pool1 [112 112 64]** – pierwsza operacja pooling redukuje rozdzielczość do 112×112 , zachowując najistotniejsze informacje. W efekcie obraz zostaje uproszczony i mniej podatny na szumy,
4. **conv2_2 [112 112 128]** – na tym etapie sieć uczy się bardziej złożonych wzorców: lokalnych tekstur oraz fragmentów obiektów (np. reflektorów, kół),
5. **pool2 [56 56 128]** – druga operacja pooling dodatkowo zmniejsza rozdzielczość, a obraz staje się bardziej skondensowany. Zachowane zostają jedynie najważniejsze regiony aktywności,

6. **conv3_3 [56 56 256]** – w głębszych warstwach konwolucyjnych pojawiają się mapy cech trudne do interpretacji wizualnie. Są to kombinacje prostszych cech, odpowiadające fragmentom obiektów i charakterystycznym wzorcom dla danego obrazu,
7. **pool3 [28 28 256]** – kolejne podpróbkiowanie prowadzi do silnej redukcji szczegółów. Obraz przyjmuje formę uproszczonej „mapy obecności” obiektu względem tła,
8. **fc6 [1 1 4096]** – pierwsza warstwa w pełni połączona spłaszcza obraz do postaci wektora 4096 wartości,
9. **fc7 [1 1 4096]** – druga warstwa w pełni połączona transformuje wektor w jeszcze bardziej abstrakcyjną reprezentację. Neurony odpowiadają za wykrywanie złożonych motywów (np. „samochód” jako kategoria ogólna),
10. **fc8 [1 1 1000]** – końcowa warstwa w pełni połączona odwzorowuje wektor cech na rozkład prawdopodobieństwa 1000 klas z bazy ImageNet. Wynik tej warstwy jest interpretowany przez funkcję *softmax* jako końcowa klasyfikacja.

Analiza rozmiarów aktywacji pozwala dokładniej prześledzić, jak obraz wejściowy jest transformowany przez kolejne warstwy architektury VGG16. Każdy etap zmienia zarówno wymiary przestrzenne danych, jak i liczbę kanałów (map cech), a równocześnie zwiększa poziom abstrakcji reprezentowanych informacji. Szczegółowe zestawienie parametrów przedstawiono w tabeli 6.2.

Tabela 6.2: Porównanie liczby uczonych parametrów w architekturach VGG16 i VGG19 wykorzystanych dla klasyfikacji obrazu

Model	Liczba parametrów
VGG16	138 357 544
VGG19	143 667 240

6.4.3 Grad-CAM – wizualizacja istotności

W celu lepszego zrozumienia mechanizmu podejmowania decyzji przez analizowane sieci zastosowano metodę Grad-CAM (*Gradient-weighted Class Activation Mapping*) [59]. Technika ta umożliwia wizualizację obszarów obrazu wejściowego, które w największym stopniu przyczyniły się do przypisania klasy wynikowej. Mapa istotności, naniesiona w postaci kolorowej nakładki na obraz oryginalny. Metoda pozwala na jakościową interpretację działania modelu i ocenę, czy proces klasyfikacji w sieci bazuje na fragmentach danych wejściowych niosących informację istotną z punktu widzenia znaczenia klasy. Kolor czerwony oznacza obszar, który miał największy wpływ na działanie klasyfikatora. Na rysunku 6.8 przedstawiono wyniki metody Grad-CAM dla obrazu zaklasyfikowanego przez obie sieci do klasy *minivan*. W przypadku VGG16 (rys. 6.8a) obszary aktywacji są bardziej rozproszone, obejmując znaczną część karoserii pojazdu, a także fragmenty otoczenia. Z kolei w VGG19 (rys. 6.8b) aktywacja jest bardziej zwarta i skoncentrowana głównie

w centralnej części samochodu, obejmując m.in. charakterystyczne elementy takie jak przednia szyba i maska. Świadczy to o większej precyzji lokalizacji cech istotnych dla rozpoznania klasy oraz o potencjalnie lepszej zdolności generalizacji głębszej architektury.



(a) Metoda Grad-CAM dla sieci VGG16: klasyfikacja *minivan*.



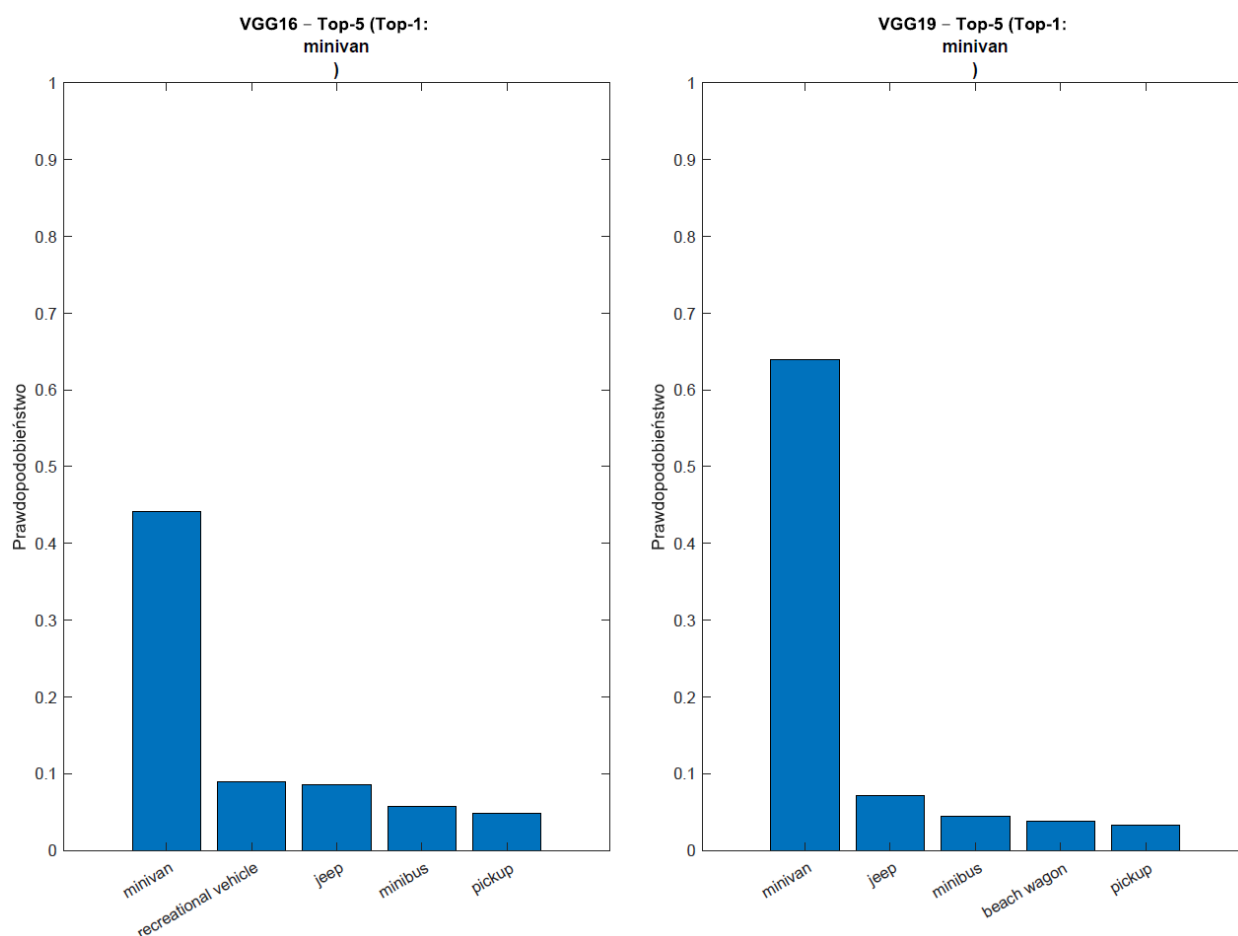
(b) Metoda Grad-CAM dla sieci VGG19: klasyfikacja *minivan*.

Rysunek 6.8: Wizualizacja obszarów istotnych dla klasyfikacji obrazu jako *minivan* z wykorzystaniem metody Grad-CAM.

Wyniki te wskazują, że mimo iż obie sieci poprawnie rozpoznają klasę *minivan*, to sposób podejmowania decyzji różni się – VGG16 polega na bardziej ogólnym zestawie wskazówek wizualnych, podczas gdy VGG19 wykorzystuje bardziej szczegółowe, spójne obszary obrazu.

6.4.4 Wyniki klasyfikacji Top-5

Dla obu architektur wyznaczono listę pięciu najbardziej prawdopodobnych klas (rysunek 6.9) wraz z przypisanymi prawdopodobieństwami. Wynik ten pozwala na ocenę poprawności klasyfikacji w kontekście rzeczywistej etykiety obiektu. W analizowanym przykładzie obie sieci poprawnie rozpoznały główną klasę (*Top-1*), różnice dotyczyły natomiast wartości prawdopodobieństwa oraz kolejności pozostałych etykiet w zestawieniu Top-5. Można zaobserwować, że VGG19, dzięki większej głębokości sieci wykazuje większe wartości prawdopodobieństwa przynależności do klasy.



Rysunek 6.9: Reprezentacja najbardziej prawdopodobnych klas na podstawie zdjęcia.

Tabela 6.3: Porównanie prawdopodobieństw klasyfikacji dla wybranych kategorii w VGG16 i VGG19

Kategoria	VGG16	VGG19
minivan	0.4409	0.6399
recreational vehicle	0.0888	–
jeep	0.0858	0.0719
minibus	0.0572	0.0446
beach wagon	–	0.0378
pickup	0.0476	0.0333

Tabela 6.3 przedstawia porównanie wyników klasyfikacji Top-5 dla dwóch architektur sieci konwolucyjnych: VGG16 oraz VGG19. W tabeli zestawiono kategorie, które pojawiły się w wynikach obu modeli, wraz z odpowiadającymi im prawdopodobieństwami. Warto zauważyć, że w obu przypadkach najwyższe prawdopodobieństwo przypisane zostało kategorii *minivan*, jednak w sieci VGG19 wynik ten jest wyższy (0.6399) niż w VGG16 (0.4409). Świadczy to o większej dokładności klasyfikatora.

Kolejne kategorie różnią się częściowo między modelami. VGG16 wyżej oceniła klasę *recreational vehicle* (0.0888), która nie pojawiła się w zestawieniu VGG19. Z kolei sieć VGG19 uwzględniła kategorię

beach wagon (0.0378), nieuwzględnioną przez VGG16. Wspólne dla obu architektur okazały się klasy *jeep*, *minibus* oraz *pickup*, przy czym dla każdej z nich wartości prawdopodobieństwa były zbliżone, choć nieco wyższe w modelu VGG16 (np. *jeep*: 0.0858 vs 0.0719), bądź odwrotnie w modelu VGG19 (np. *minibus*: 0.0446 vs 0.0572 w VGG16).

Analiza wyników wskazuje, że choć oba modele identyfikują poprawnie tę samą kategorię dominującą, to różnice w pozostałych klasyfikacjach Top-5 świadczą o odmiennych reprezentacjach wewnętrznych cech obrazu. VGG19, dzięki większej liczbie warstw konwolucyjnych, wykazuje wyraźniejszą dominację jednej klasy dla przedstawionego przykładu, podczas gdy VGG16 generuje rozkład prawdopodobieństw bardziej rozproszony pomiędzy kilka pokrewnych kategorii.

Rozdział 7

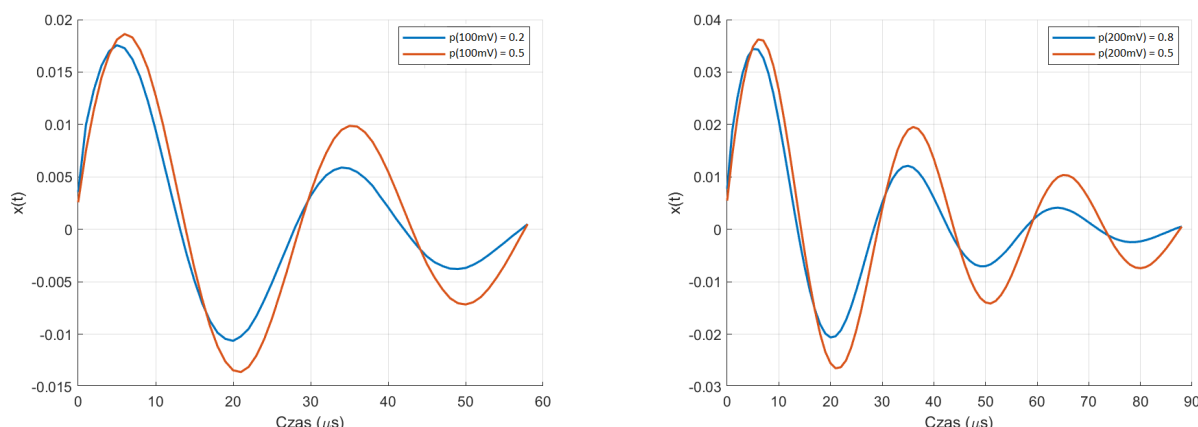
Rozpoznanie odpowiedzi układu na pojednawczy impuls oraz odpowiedzi układu na dwa impulsy następujące po sobie dla $\Delta t < 200\mu s$

Skuteczność metod uczenia maszynowego w dużej mierze zależy od jakości oraz sposobu przygotowania danych wejściowych. Proces ten obejmuje kilka etapów, takich jak: czyszczenie danych, selekcja cech, normalizacja, etykietowanie, a także podział na zbiory treningowe, walidacyjne i testowe. Przygotowanie danych umożliwia skuteczne trenowanie modelu. Zbiór danych powinien zawierać wszystkie istotne warianty sygnałów i klas, które mają być rozpoznawane. Nierównomierny rozkład klas może prowadzić do przeuczenia modelu na klasę dominującą i pogorszenia ogólnej skuteczności. W tej pracy przyjęto 80% danych testowych jako ten zbiór, który służy do uczenia modelu. Drugi zbiór to zbiór walidacyjny (validation set) zawierający 20% danych i ten służy do strojenia hiperparametrów i unikania przeuczenia. Dodatkowo wykorzystano zbiór testowy (test set) zawierający dane, których sieć nie widziała w procesie uczenia. Służy on do ostatecznej oceny jakości modelu, niezależnie od etapu trenowania. W tym rozdziale przedstawiono rozpoznanie odpowiedzi na pojedyncze oraz podwójne impulsy dla $\Delta t < 200\mu s$ w układzie dyskretnym.

7.1 Opis analizowanych przypadków

Analiza średnich wartości odpowiedzi układu, przedstawiona na rysunkach 7.1a i 7.1b wskazuje, że przebiegi uzyskane dla różnych prawdopodobieństw wystąpienia impulsów się różnią. Różnice te dotyczą przede wszystkim wartości parametrów, takich jak tłumienie b , częstość drgań tłumionych ω czy amplituda a .

Wykorzystując algorytm do estymacji parametrów modelu przedstawiony w rozdziale 5 wyznaczono wartości dla poszczególnych prób. Dla każdej konfiguracji wyznaczono parametry tłumienia b , częstość drgań tłumionych ω oraz amplitudy a , przy użyciu funkcji błędu średniokwadratowego (C – błąd średniokwadratowy).



(a) Średnie wartości odpowiedzi układu RLC dla wymuszenia 100 mV o prawdopodobieństwie wystąpienia impulsów $p(100mV) = 0.2$ oraz $p(100mV) = 0.5$.

(b) Średnie wartości odpowiedzi układu RLC dla wymuszenia 200 mV o prawdopodobieństwie wystąpienia impulsów $p(200mV) = 0.8$ oraz $p(200mV) = 0.5$.

Rysunek 7.1: Porównanie średnich wartości odpowiedzi układu RLC dla różnych poziomów wymuszenia i prawdopodobieństwie wystąpienia impulsu.

Tabela 7.1: Parametry tłumienia b , częstości ω i amplitudy a wyznaczone na podstawie minimum funkcji błędu dla wymuszeń 100 mV i 200 mV przy różnych wartościach prawdopodobieństwa $p(\eta_i)$

Wymuszenie	Funkcja błędu	$p(\eta_i)$	Tłumienie b	ω	Amplituda a
100 mV	C – błąd średniokwadratowy	0.2	35 668.610	213 524.000	0.021359
		0.5	21 890.092	215 373.100	0.021478
200 mV	C – błąd średniokwadratowy	0.5	22 128.844	215 348.700	0.042979
		0.8	35 952.993	213 476.300	0.042904

Tabela 7.1 przedstawia wartości parametrów opisujących odpowiedź układu RLC wyznaczone na podstawie analizy minimów lokalnych. W tabeli zestawiono wyniki dla dwóch wartości amplitudy wymuszenia (100 mV oraz 200 mV) oraz dwie konfiguracje prawdopodobieństwa wymuszenia impulsów. Pierwszy to symetryczny rozkład z prawdopodobieństwem 0.5 dla wymuszenia 100 mV oraz 0.5 dla wymuszenia 200 mV. Drugi to asymetryczny rozkład z prawdopodobieństwami 0.2 dla wymuszenia 100 mV i 0.8 dla wymuszenia 200 mV. Tabela 7.1 wskazuje, że parametry odpowiedzi układu RLC ulegają zmianom w zależności od rozkładu impulsów.

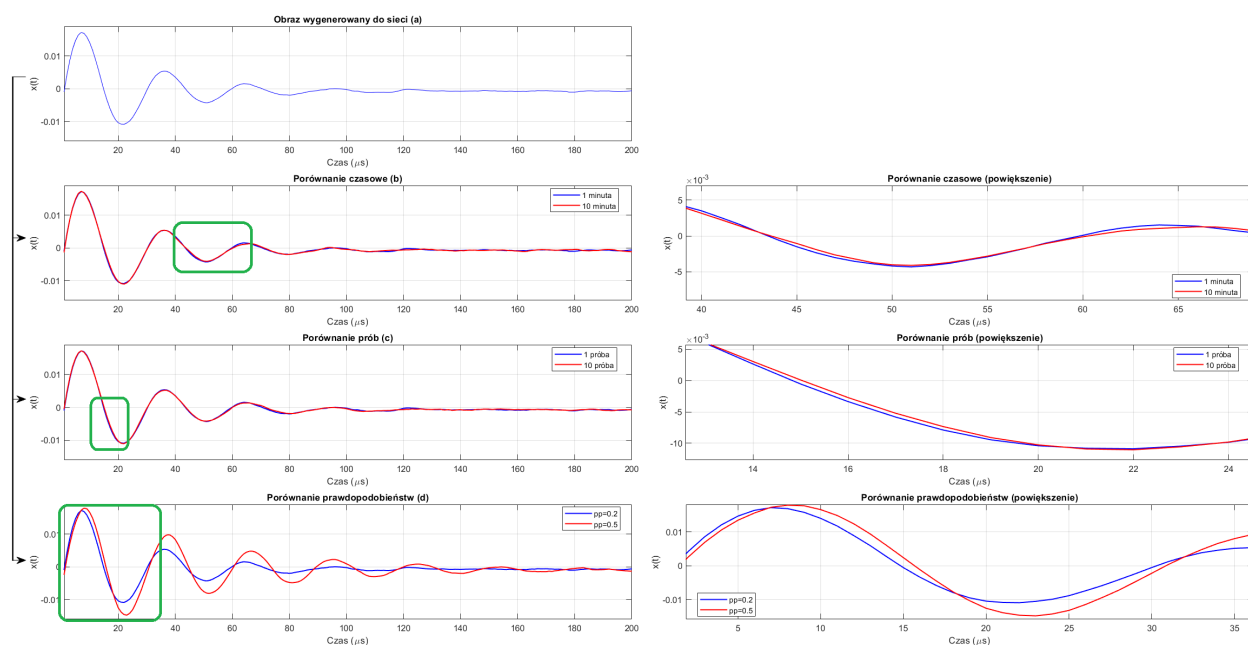
7.2 Przygotowanie zbioru danych

Przygotowanie zbioru danych rozpoczęto od definicji dwóch podstawowych klas: **A** i **B**, odpowiadające odpowiednio pojedynczym odpowiedziom na wymuszenia o wielkości 100 mV i 200 mV. Analiza danych eksperymentalnych pozwala określić udział procentowy poszczególnych klas w próbie pomiarowej.

Tabela 7.2: Liczba pojedynczych wymuszeń układu RLC oraz ich udział procentowy w zarejestrowanej próbie (dla wymuszeń 100 mV, $p(100mV) = 0.5$ oraz 200 mV, $p(200mV) = 0.5$)

Wymuszenie	Liczba impulsów	Udział w próbie [%]
100 mV	2995	49.2680%
200 mV	3023	50.7320%
Suma	6018	100.0000%

Tabela 7.2 przedstawia zestawienie liczby wymuszeń układu RLC dla poszczególnych wielkości wraz z odpowiadającym im udziałem w próbie pomiarowej. Analiza danych wskazuje, że w próbie zarejestrowano odpowiedzi na impulsy, których wymuszenia stanowią wartości: 100 mV (49.27%) oraz 200 mV (50.73%).



Rysunek 7.2: Porównanie pojedynczej odpowiedzi układu RLC na wymuszenie 100 mV oraz wpływu różnych czynników na kształt odpowiedzi.

Na rysunku 7.2 przedstawiono zestawienie przebiegów odpowiedzi układu RLC wzbudzanego losową serią impulsów. Grafika składa się z czterech głównych paneli oznaczonych literami (a)–(d), z których każdy przedstawia inne porównanie odpowiedzi na wymuszenie o wielkości 100 mV. Dodatkowo, dla zwiększenia czytelności i uchwycenia subtelnych różnic, obok paneli (b)–(d) zamieszczono powiększenia wybranych fragmentów przebiegów.

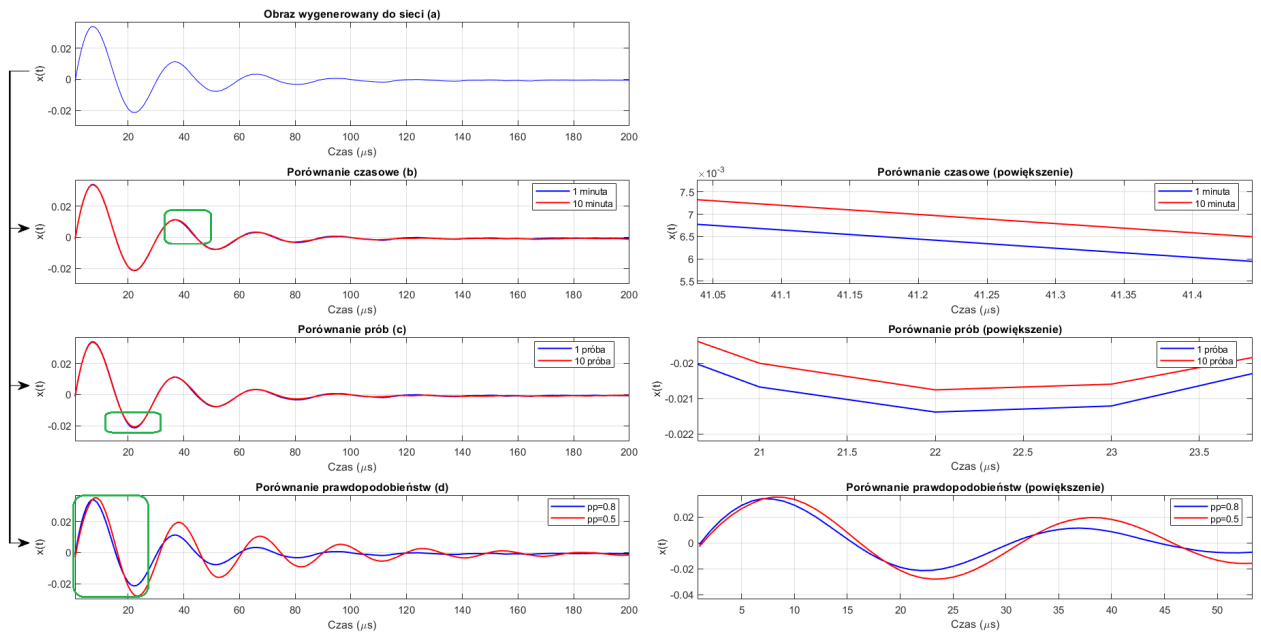
Panel (a) prezentuje odpowiedź układu RLC na wymuszenie 100 mV. Panel (b) ilustruje porównanie odpowiedzi tego samego układu RLC zarejestrowanych w różnych chwilach czasowych. Linia niebieska odpowiada pierwszej odpowiedzi w pierwszej minucie rejestracji, natomiast linia czerwona przedstawia analogiczną odpowiedź z dziesiątej minuty. Sygnały te pozostają do siebie zbliżone, jednak powiększenie wybranego fragmentu (panel prawy) ujawnia subtelne różnice w amplitudzie i przesunięciu fazowym w od-

powiedzi. Takie zmiany, choć z pozoru niewielkie, mogą być istotne przy uwzględnieniu, że poszczególne próby o tych samych rozkładach też się różnią.

Panel (c) przedstawia porównanie przebiegów pochodzących z różnych prób pomiarowych o wymuszeniu 100 mV. Linia niebieska odpowiada pierwszej odpowiedzi w pierwszej próbie, natomiast linia czerwona odpowiada pierwszej odpowiedzi w dziesiątej próbie. Szczegółowa analiza powiększonego fragmentu wskazuje, że kolejne próby mogą różnić się w zakresie niewielkich przesunięć fazowych oraz składowej stałej. Wskazuje to na fakt, że szum zarejestrowany przez kartę pomiarową wprowadza pewną zmienność, która wpływa na początek odpowiedzi układu RLC.

Panel (d) ilustruje wpływ prawdopodobieństwa wystąpienia impulsu pp na kształt odpowiedzi układu. Przedstawiono odpowiedzi dla $p(100mV) = 0.2$ (linia niebieska) oraz $p(100mV) = 0.5$ (linia czerwona). Obie krzywe reprezentują pierwszą odpowiedź układu w pierwszej próbie. Zmiany rozkładów wymuszających drgania powodują wyraźniejsze różnice w początkowej fazie oscylacji, przede wszystkim w zakresie amplitudy. Wskazuje to, że prawdopodobieństwo wystąpienia impulsu kształtuje odpowiedź układu i pomimo tej samej wartości impulsu, każda odpowiedź jest inna.

Rysunek pokazuje, że choć podstawowy kształt odpowiedzi pozostaje niezmienny, obserwuje się subtelne różnice w amplitudzie a , tłumieniu b , przesunięciu fazowym ϕ oraz składowej stałej C w zależności od czasu rejestracji, próby oraz prawdopodobieństwa wystąpienia impulsu. Obraz taki jest reprezentatywnym przykładem danych, które w dalszej części badań podlegają procesowi klasyfikacji, a jego struktura umożliwia sieci neuronowej uchwycenie cech, które później będą wykorzystane do celu klasyfikacji i badania rozkładu.



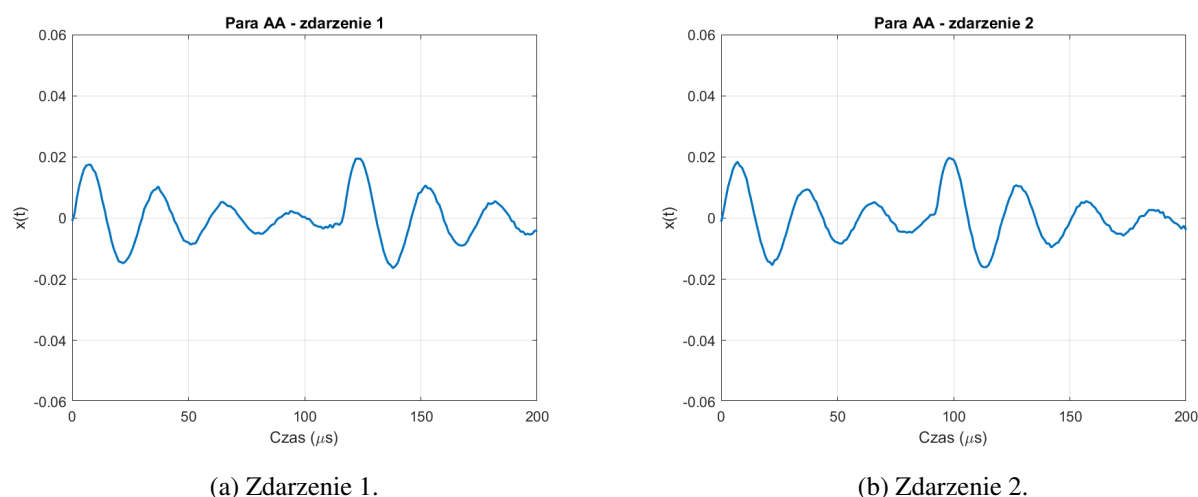
Rysunek 7.3: Porównanie pojedynczej odpowiedzi układu RLC na wymuszenie 200 mV oraz wpływu różnych czynników na kształt odpowiedzi.

W analogiczny sposób przedstawiono porównanie dla wymuszenia 200 mV (rysunek 7.3), gdzie obserwacje są podobne. Rysunki (7.2 i 7.3 przedstawiają złożoność samej odpowiedzi oraz wpływ różnych

czynników – czasu rejestracji, liczby prób oraz wartości prawdopodobieństwa wystąpienia impulsu – na kształt odpowiedzi układu RLC. Badania eksploracyjne pokazują, że różnice dla wymuszenia o wartości 200 mV pomiędzy poszczególnymi przypadkami są większe. Analiza ta pozwala lepiej zrozumieć złożoność badanego procesu, a jednocześnie uzasadnia konieczność stosowania zaawansowanych metod klasyfikacji, zdolnych do rozpoznania odpowiedzi na losową serię impulsów.

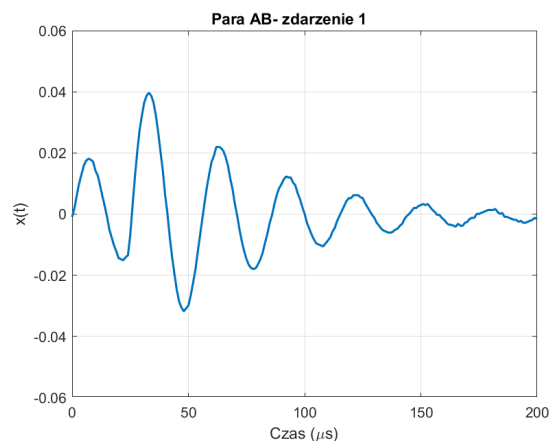
7.2.1 Definicja klas mieszanych (AA, BB, AB/BA)

Klasy AA, BB oraz AB/BA odnoszą się do przypadków, w których nowy impuls pojawia się, zanim poprzedni całkowicie wygaś. Zjawiska te występują z bardzo niskim prawdopodobieństwem i stanowią konsekwencję wcześniej zdefiniowanych wymuszeń. Jak wspomniano w poprzednim podrozdziale, impulsy typu A i B odpowiadają wymuszeniom o amplitudach odpowiednio 100 mV i 200 mV. W zależności od rodzaju nakładających się wymuszeń wyodrębniono trzy główne typy zdarzeń. Klasa AA to zdarzenia, w których dwa kolejne impulsy typu A występują w krótkim odstępie czasowym. Klasa BB to analogiczne zdarzenia dla impulsów typu B. Klasa AB/BA to zdarzenia mieszane, w których nakładają się impulsy różnego typu (A i B lub B i A), co prowadzi do bardziej złożonej odpowiedzi układu. Pierwsze próby wykazały, że ta klasa rozbita na dwie oddzielne klasy AB oraz BA są nierozróżnialne stąd zostały połączone we wspólną klasę. Przykłady zdarzeń należących do klasy AA przedstawiono na Rysunku 7.4.

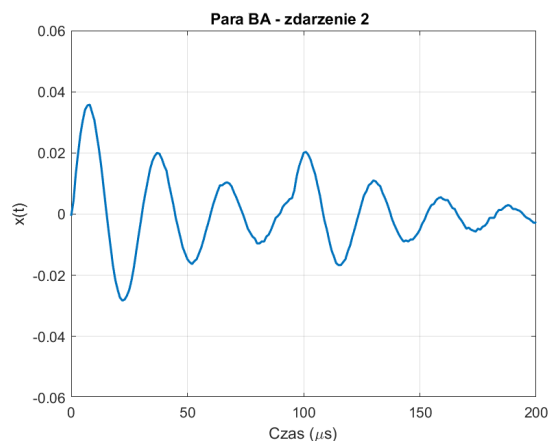


Rysunek 7.4: Przykłady zdarzeń klasy AA zarejestrowane w kolejnych próbach.

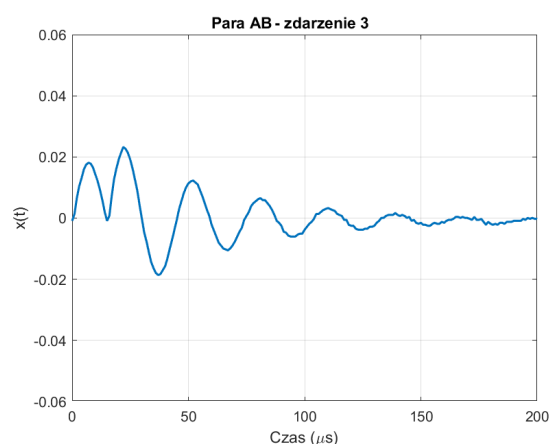
Rysunek 7.5 ilustruje zdarzenia klasy AB/BA, w których impulsy o różnej wielkości, w losowej sekwencji pojawiają się w krótkim odstępie czasowym. Tego typu zdarzenia generują bardziej złożone przebiegi odpowiedzi układu i mogą prowadzić do trudności w klasyfikacji.



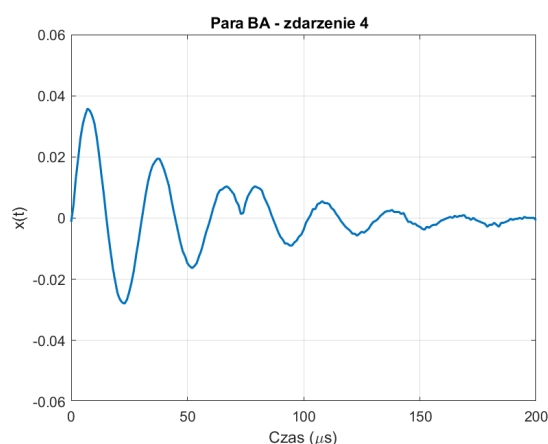
(a) Zdarzenie 1.



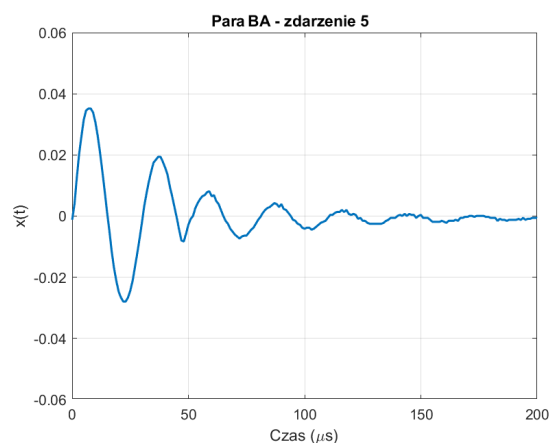
(b) Zdarzenie 2.



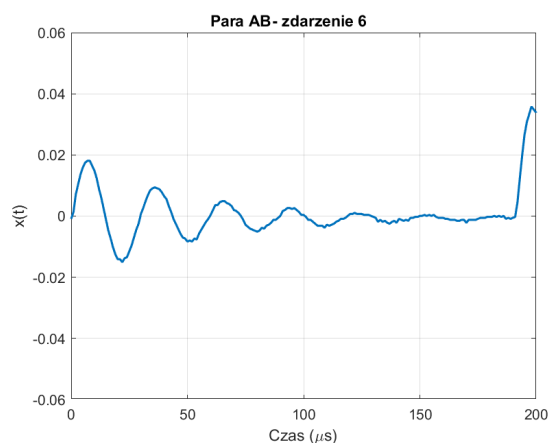
(c) Zdarzenie 3.



(d) Zdarzenie 4.



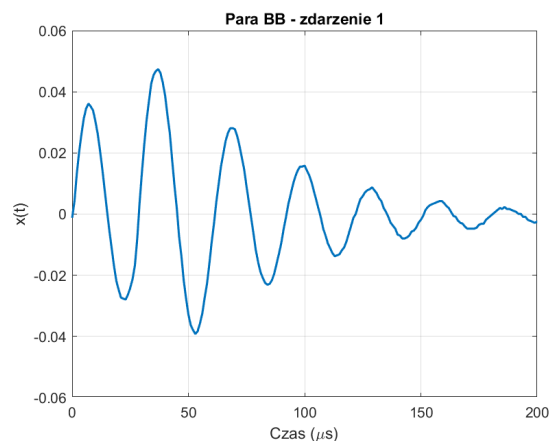
(e) Zdarzenie 5.



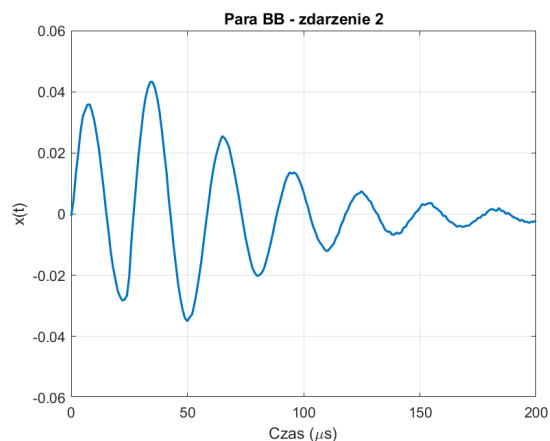
(f) Zdarzenie 6.

Rysunek 7.5: Przykłady zdarzeń klasy AB/BA zarejestrowane w kolejnych próbach.

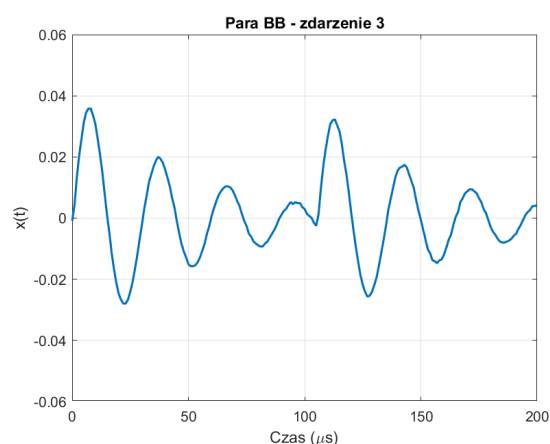
Na rysunku 7.6 przedstawiono zdarzenia klasy BB. Występowanie impulsów typu B w krótkich odstępach czasowych generuje nakładające się odpowiedzi o charakterystycznym kształcie, różniącym się od klasy AA amplitudą odpowiedzi.



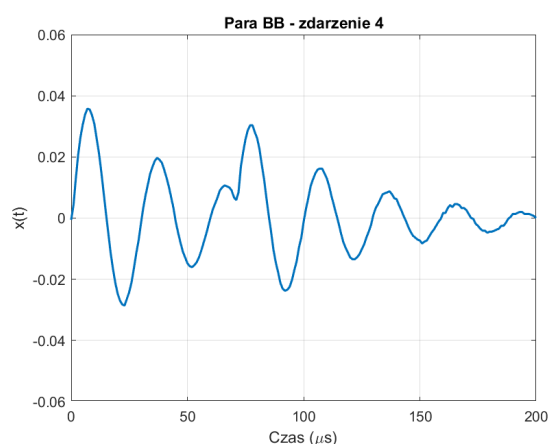
(a) Zdarzenie 1.



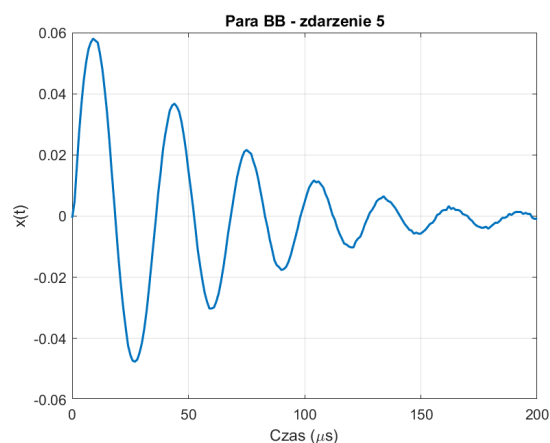
(b) Zdarzenie 2.



(c) Zdarzenie 3.



(d) Zdarzenie 4.



(e) Zdarzenie 5.

(f)

Rysunek 7.6: Przykłady zdarzeń klasy BB zarejestrowane w kolejnych próbach.

W powyższych przypadkach odpowiedź układu ujawnia dodatkowe efekty interferencji między impulsami, a każda para impulsów generuje dwa zestawy parametrów tłumienia, częstotliwości drgań tłumionych oraz amplitudy. Na Rysunku 7.6 (podrysunki a–e) przedstawiono przykłady zdarzeń klasy BB, czyli od-

powiedzi układu na dwa kolejne impulsy o amplitudzie 200 mV. Podobnie jak w przypadku klasy AA obserwuje się nakładanie odpowiedzi, prowadzące do złożonych przebiegów sygnału.

Tabela 7.3: Liczba odpowiedzi złożonych układu RLC oraz ich udział procentowy w zarejestrowanej próbie (dla wymuszeń 100 mV i 200 mV, $p = 0.5$)

Wymuszenie	Liczba odpowiedzi	Udział w próbie[%]
AA	3	0.05%
BB	5	0.08%
AB/BA	6	0.10%
Suma	14	0.23%

Analiza danych przedstawionych w Tabeli 7.3 wskazuje, że liczba złożonych odpowiedzi układu RLC, powstających w wyniku wystąpienia dwóch impulsów następujących po sobie w krótkim odstępie czasowym, jest bardzo ograniczona. W przedstawionej próbie zarejestrowano zaledwie 14 takich przypadków, co stanowi jedynie 0.23% wszystkich odpowiedzi.

W celu potwierdzenia poprawności otrzymanych wyników przeprowadzono analizę porównawczą z rozwiązaniem analitycznym wynikającym z teorii rozkładu wykładniczego. Gęstość prawdopodobieństwa dla rozkładu wykładniczego o parametrze intensywności λ wyrażona jest równaniem 7.1:

$$p(x) = \lambda e^{-\lambda x}, \quad x \geq 0, \quad (7.1)$$

natomiast odpowiadająca jej dystrybuanta przyjmuje postać 7.2:

$$P(x) = 1 - e^{-\lambda x}. \quad (7.2)$$

W badaniach przyjęto wartość intensywności $\lambda = 10$, a następnie obliczono prawdopodobieństwo wystąpienia zdarzenia w przedziale czasowym krótszym niż $200 \mu s$. W tym celu wykonano trzy rodzaje obliczeń w środowisku MatLAB:

1. **Obliczenia analityczne** – polegały na całkowaniu funkcji gęstości $p(x)$ (równanie 7.1) w granicach od 0 do $200 \cdot 10^{-6}$. Wynik tej całki odpowiada wartości dystrybuanty w punkcie $x = 200 \mu s$ (por. równanie 7.2) i wyniósł:

$$\int_0^{200 \cdot 10^{-6}} 10 e^{-10x} dx \approx 0.0019998 \approx 0.2\%. \quad (7.3)$$

2. **Obliczenia numeryczne** – wykorzystano funkcję `integral` środowiska MATLAB, definiując funkcję anonimową odpowiadającą gęstości rozkładu wykładniczego (równanie 7.1). Funkcja anonimowa w środowisku MATLAB jest konstrukcją pozwalającą na zdefiniowanie wyrażenia matematycznego w zwartej postaci, bez konieczności tworzenia oddzielnego pliku funkcji. W analizie opisaną w niniejszej pracy została ona wykorzystana do odwzorowania gęstości $p(x)$. Uzyskany wynik całki numerycznej był zgodny z rozwiązaniem symbolicznym i wyniósł:

$$\approx 0.0019998 \approx 0.2\%.$$

3. **Obliczenia teoretyczne** – bezpośrednio podstawiono wartości do wzoru na dystrybuantę (równanie 7.2):

$$P(200\mu s) = 1 - e^{-\lambda \cdot 200 \cdot 10^{-6}} = 1 - e^{-0.002} \approx 0.0019998 \approx 0.2\%. \quad (7.4)$$

Porównanie wyników uzyskanych w poszczególnych krokach wykazało ich pełną zgodność, ograniczoną jedynie dokładnością numeryczną obliczeń. Niezależnie od przyjętej metody (analitycznej czy numerycznej), wartość prawdopodobieństwa była identyczna i wynosiła około 0.0019998, czyli 0.2%. Oznacza to, że implementacja w MATLAB-ie poprawnie odwzorowuje własności rozkładu wykładniczego, a uzyskane rezultaty można uznać za wiarygodne i zgodne z teorią.

Częstość występowania tego rodzaju zdarzeń jest marginalna w porównaniu do odpowiedzi na pojedyncze impulsy. Niemniej jednak, ich obecność jest ważna dla analizy dynamiki układu, ponieważ prowadzi do nakładania się odpowiedzi i wzajemnych interakcji między impulsami, utrudniając jednoznaczne wyznaczenie parametrów tłumienia, częstotliwości drgań tłumionych oraz amplitudy.

Uwzględnienie takich rzadkich sekwencji w procesie uczenia sieci neuronowych zwiększa ogólność modelu oraz umożliwia poprawne rozpoznawanie także nietypowych zdarzeń w danych pomiarowych, co jest istotne w procesie identyfikacji klas mieszanych AA, BB oraz AB/BA.

7.3 Definicja bazy danych do sieci

Aby zbudować bazę danych do sieci, ważne jest zapewnienie odpowiedniego zbalansowania liczebności klas w zbiorze treningowym. Niezrównoważone klasy mogą spowodować faworyzowanie dominujących kategorii przez model. W taki przypadku stosuje się redukcję liczby przykładów z klas dominujących (*undersampling*) lub wagi klas podczas uczenia modelu.

W badaniach przeprowadzono 20 prób wzbudzenia układu RLC losową serią impulsów o amplitudach 100 mV i 200 mV, przy czym każda z dziesięciu prób charakteryzowała się innym rozkładem prawdopodobieństwa wystąpienia impulsów (Tabela 7.4). Dla każdej z kombinacji prawdopodobieństw wykonano serię pomiarów, które posłużyły do zbudowania bazy danych do uczenia sieci neuronowej. Z dziewięciu prób dla obu prawdopodobieństw stworzono zbiór treningowy, który pozwalał modelowi nauczyć się reprezentacji odpowiedzi układu na impulsy o różnych amplitudach i różnych parametrach losowych. Pozostałą jedną próbę wyodrębniono jako zbiór testowy, służący do weryfikacji skuteczności klasyfikatora dla danych, których model wcześniej nie widział. Takie rozdzielenie prób na zbiór walidacyjny i testowy zapewnia ocenę modelu, zapobiegając przeszacowaniu jego skuteczności. Dzięki temu możliwe jest sprawdzenie zdolności klasyfikatora do rozpoznawania odpowiedzi układu.

Tabela 7.4: Wielkość impulsu i odpowiadające im wartości prawdopodobieństwa wystąpienia impulsu

Wielkość impulsu	Prawdopodobieństwo 1	Prawdopodobieństwo 2
100 mV	0.5	0.2
200 mV	0.5	0.8

Liczba pojedynczych odpowiedzi

Tabele 7.5 i 7.6 przedstawiają liczbę odpowiedzi układu RLC dla poszczególnych prób.

Tabela 7.5: Liczba pojedynczych wymuszeń układu RLC oraz ich udział procentowy w próbie, gdzie prawdopodobieństwo wystąpienia impulsów było równe $p(100mV) = p(200mV) = 0.5$

Próba	Wymuszenie	Liczba odpowiedzi	Udział w próbie [%]
1	100 mV	2995	49.2680
	200 mV	3023	50.7320
	Suma	6018	100.0000
2	100 mV	3088	50.0081
	200 mV	3087	49.9919
	Suma	6175	100.0000
3	100 mV	2935	49.3277
	200 mV	3015	50.6723
	Suma	5950	100.0000
4	100 mV	3044	50.0247
	200 mV	3041	49.9753
	Suma	6085	100.0000
5	100 mV	3024	50.5016
	200 mV	2956	49.4984
	Suma	5980	100.0000
6	100 mV	3015	50.1248
	200 mV	2992	49.8752
	Suma	6007	100.0000
7	100 mV	3000	50.4706
	200 mV	2953	49.5294
	Suma	5953	100.0000
8	100 mV	3056	50.0000
	200 mV	3056	50.0000
	Suma	6112	100.0000
9	100 mV	2883	48.7730
	200 mV	3029	51.2270
	Suma	5912	100.0000
10	100 mV	3128	50.6231
	200 mV	3051	49.3769
	Suma	6179	100.0000

Tabela 7.6: Liczba pojedynczych odpowiedzi układu RLC oraz ich udział procentowy w próbie, gdzie prawdopodobieństwo wystąpienia impulsów było równe $p(100mV) = 0.2$ oraz $p(200mV) = 0.8$

Próba	Wymuszenie	Liczba odpowiedzi	Udział w próbie [%]
1	100 mV	1266	20.8917

ciąg dalszy na następnej stronie

Próba	Wymuszenie	Liczba odpowiedzi	Udział w próbie [%]
	200 mV	4794	79.1083
	Suma	6060	100.0000
2	100 mV	1167	19.9022
	200 mV	4697	80.0978
	Suma	5864	100.0000
3	100 mV	1186	19.1713
	200 mV	4999	80.8287
	Suma	6185	100.0000
4	100 mV	1182	19.8596
	200 mV	4769	80.1404
	Suma	5951	100.0000
5	100 mV	1203	20.2735
	200 mV	4734	79.7265
	Suma	5937	100.0000
6	100 mV	1210	20.1171
	200 mV	4808	79.8829
	Suma	6018	100.0000
7	100 mV	1161	20.0027
	200 mV	4547	79.9973
	Suma	5708	100.0000
8	100 mV	1186	19.6323
	200 mV	4730	80.3677
	Suma	5916	100.0000
9	100 mV	1211	20.4602
	200 mV	4830	79.5398
	Suma	6041	100.0000
10	100 mV	1172	19.5208
	200 mV	4833	80.4792
	Suma	6005	100.0000

Liczba par impulsów

Rzadkie, złożone odpowiedzi układu, wynikające z nakładania się impulsów (AA, BB, AB/BA), zarejestrowano w mniejszej liczbie przypadków. Tabele 7.8 i 7.7 prezentują liczbę par impulsów w poszczególnych próbach. Ich udział procentowy w próbach jest niewielki, co pokazuje, że takie zdarzenia są mało prawdopodobne.

Tabela 7.7: Liczba par odpowiedzi mieszanych w próbach oraz ich udział procentowy, gdzie prawdopodobieństwo wystąpienia impulsów było równe $p(100mV) = p(200mV) = 0.5$

Próba	Para impulsów	Liczba par	Udział w próbie [%]
1	AA	3	0.0500
	BB	5	0.0836

ciąg dalszy na następnej stronie

Próba	Para impulsów	Liczba par	Udział w próbie [%]
	AB_BA	6	0.1002
	Suma	14	0.2338
2	AA	3	0.0485
	BB	3	0.0485
	AB_BA	3	0.0485
	Suma	9	0.1455
3	AA	1	0.0169
	BB	3	0.0506
	AB_BA	12	0.2028
	Suma	16	0.2703
4	AA	1	0.0168
	BB	5	0.0849
	AB_BA	4	0.0679
	Suma	10	0.1696
5	AA	4	0.0670
	BB	6	0.1003
	AB_BA	6	0.1003
	Suma	16	0.2676
6	AA	3	0.0500
	BB	3	0.0500
	AB_BA	3	0.0500
	Suma	9	0.1500
7	AA	5	0.0849
	BB	4	0.0679
	AB_BA	4	0.0679
	Suma	13	0.2207
8	AA	4	0.0670
	BB	3	0.0503
	AB_BA	8	0.1340
	Suma	15	0.2513
9	AA	4	0.0676
	BB	2	0.0338
	AB_BA	4	0.0676
	Suma	10	0.1690
10	AA	3	0.0486
	BB	7	0.1133
	AB_BA	1	0.0162
	Suma	11	0.1781

Tabela 7.8: Liczba par odpowiedzi mieszanych w próbach oraz ich udział procentowy, gdzie prawdopodobieństwo wystąpienia impulsów było równe $p(100mV) = 0.2$ oraz $p(200mV) = 0.8$

Próba	Para impulsów	Liczba par	Udział w próbie [%]
1	AA	1	0.0165
	BB	15	0.2475
	AB_BA	2	0.0330
	Suma	18	0.2970
2	AA	0	0.0000
	BB	5	0.0853
	AB_BA	6	0.1024
	Suma	11	0.1877
3	AA	0	0.0000
	BB	11	0.1779
	AB_BA	5	0.0808
	Suma	16	0.2587
4	AA	1	0.0168
	BB	7	0.1176
	AB_BA	2	0.0336
	Suma	10	0.1680
5	AA	2	0.0337
	BB	6	0.1011
	AB_BA	5	0.0842
	Suma	13	0.2190
6	AA	0	0.0000
	BB	10	0.1662
	AB_BA	1	0.0166
	Suma	11	0.1828
7	AA	0	0.0000
	BB	3	0.0514
	AB_BA	3	0.0514
	Suma	6	0.1028
8	AA	0	0.0000
	BB	5	0.0827
	AB_BA	2	0.0331
	Suma	7	0.1158
9	AA	0	0.0000
	BB	4	0.0676
	AB_BA	1	0.0169
	Suma	5	0.0845
10	AA	0	0.0000
	BB	15	0.2498
	AB_BA	5	0.0833
	Suma	20	0.3331

7.4 Generowanie klas

Chcąc skutecznie przeprowadzić trening klasyfikatora, niezbędne jest przygotowanie odpowiednio przygotowanej bazy danych, odwzorowującej charakterystykę badanego układu. Baza ta została opracowana na podstawie zarejestrowanych odpowiedzi układu RLC na wymuszenia losowe o znanej wielkości impulsu η_i oraz różnych wartości prawdopodobieństwa.

W pracy przyjęto, że pierwsze dziewięć prób dla rozkładów $p(100 \text{ mV}) = p(200 \text{ mV}) = 0.5$ oraz $p(100 \text{ mV}) = 0.2$ i $p(200 \text{ mV}) = 0.8$ będzie wykorzystane do przygotowania zbioru treningowego. Pozostałe próby dla obu rozkładów prawdopodobieństwa wraz z wszystkimi klasami mieszanymi, odpowiednio połączone w 5 klas, zostaną wykorzystane do testów klasyfikatora. Wspomniane dane z 18 prób przeniesiono do katalogu o nazwie `Klasa_A` oraz `Klasa_B` w zależności od wymuszenia (czyli w danym katalogu znajdowały się odpowiedzi dla rozkładów $p(100 \text{ mV}) = 0.2$ oraz $p(100 \text{ mV}) = 0.5$ oraz $p(200 \text{ mV}) = 0.5$ i $p(200 \text{ mV}) = 0.8$).

Kolejnym etapem przygotowania danych było scalanie przebiegów zapisanych w wielu plikach tekstowych w jeden wspólny plik. Celem programu było wygenerowanie plików zbiorczych reprezentujących klasy A oraz B, zawierającego losową próbę wierszy z każdego z wczytanych plików znajdujących się w danym katalogu. Tworzone są dwa pliki zbiorcze `Klasa_A.txt` oraz `Klasa_B.txt` w tym samym katalogu (`ouPDut_file`), do którego zostaną przeniesione wybrane odpowiedzi z plików źródłowych. Parametr `lines_to_move` określa maksymalną liczbę losowych wierszy, jakie mają zostać pobrane z każdego pliku. Po wczytaniu całego pliku następuje losowanie indeksów wierszy do przeniesienia:

- jeżeli liczba dostępnych wierszy jest większa lub równa `lines_to_move`, wybieranych jest losowo `lines_to_move` indeksów bez powtórzeń przy użyciu funkcji `randperm`,
- jeżeli plik zawiera mniej niż `lines_to_move` wierszy, do zbioru wynikowego trafiają wszystkie dostępne linie zawierające pojedynczą odpowiedź układu na dane wymuszenie.

Wybrane linie z każdego pliku są zapisywane w pliku zbiorczym `Klasa_A.txt` oraz `Klasa_B.txt`. Proces powtarza się dla wszystkich plików w katalogu, dzięki czemu uzyskany plik wynikowy zawiera jednolity zestaw danych o kontrolowanej liczebności próbek pochodzących z różnych źródeł. Dzięki tej procedurze możliwe jest szybkie przygotowanie reprezentatywnego zbioru danych wejściowych, który zachowuje losowość wyboru próbek oraz jednocześnie kontroluje ich maksymalną liczebność. Taka forma organizacji danych ułatwia dalsze etapy przetwarzania, takie jak segmentacja, wizualizacja oraz uczenie sieci neuronowych.

Kolejnym krokiem po przygotowaniu klas A i B jest wygenerowanie trzech zbiorów odpowiedzi: `Klasa_AA.txt`, `Klasa_BB.txt` oraz `Klasa_AB_BA.txt`, stanowiących odpowiednio superpozycje odpowiedzi typu A na A, B na B oraz mieszaniny A z B w losowej kolejności (ApB lub BpA). Algorytm pracuje na wcześniej utworzonych plikach przechowujących odpowiedzi na losową serię impulsów. Aby zapewnić zgodność, obie macierze są przycinane do wspólnej liczby wierszy $N = \min(N_A, N_B)$. Następnie tworzone są trzy pliki (`Klasa_AA.txt`, `Klasa_BB.txt`, `Klasa_AB_BA.txt`) oraz inicjalizowane są liczniki `count_AA`, `count_BB`, `count_AB_BA`.

Niech $\mathbf{a} = [a_0, \dots, a_{L-1}]$ i $\mathbf{b} = [b_0, \dots, b_{L-1}]$ oznaczają odpowiadające sobie wiersze z $\mathbf{A}$ i $\mathbf{B}$ o tej samej długości L . Dla całkowitego opóźnienia $d \in \{1, \dots, 200\}$ definiuje się superpozycję z obcięciem (brak wyjścia poza długość wektora):

$$\text{ApB}[n; d] = \begin{cases} a_n + b_{n-d}, & n \geq d, \\ a_n, & n < d, \end{cases} \quad (7.5)$$

$$\text{BpA}[n; d] = \begin{cases} b_n + a_{n-d}, & n \geq d, \\ b_n, & n < d, \end{cases} \quad (7.6)$$

$$\text{ApA}[n; d] = \begin{cases} a_n + a_{n-d}, & n \geq d, \\ a_n, & n < d, \end{cases} \quad (7.7)$$

$$\text{BpB}[n; d] = \begin{cases} b_n + b_{n-d}, & n \geq d, \\ b_n, & n < d. \end{cases} \quad (7.8)$$

Dzięki pętli od 1 do 200 program syntetyzuje przypadki nakładania się odpowiedzi w krótkim odstępie czasowym, co naśladuje sytuację dwóch impulsów następujących po sobie. W każdej iteracji wykonywane są dwie realizacje (pętla $k = 1, 2$), co zwiększa różnorodność zbioru bez zwiększania wymagań pamięciowych. Program tworzy:

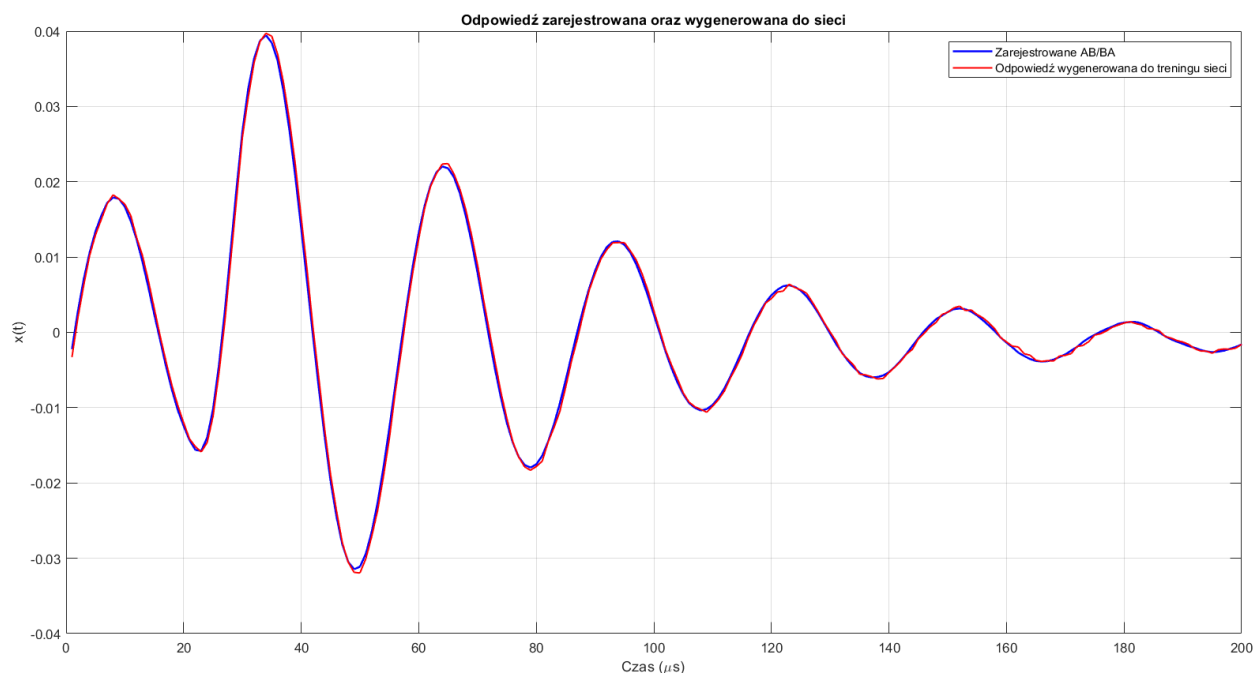
- jeden wektor ApA zapisywany do `Klasa_AA.txt`,
- jeden wektor BpB zapisywany do `Klasa_BB.txt`,
- jeden wektor klasy mieszanej do `Klasa_AB_BA.txt`, wybierany losowo z prawdopodobieństwem $p = 0.5$ między ApB i BpA.

Każdy wektor wynikowy jest zapisywany we wcześniej przygotowanych plikach tekstowych, a odpowiadający licznik jest inkrementowany. Dla skrajnych opóźnień, gdy $d \geq L$, pętla zostaje przerwana, aby uniknąć wyjścia poza zakres sygnału.

Proces przygotowania danych obejmuje dwa kolejne etapy: najpierw scalenie odpowiedzi w klasy podstawowe A i B, a następnie wygenerowanie klas mieszanych poprzez superpozycję sygnałów. Dzięki temu w sposób kontrolowany i powtarzalny syntetyzowane są klasy (AA, BB) oraz mieszana (AB/BA), co pozwala wzbogacić zbiór treningowy o przykłady nakładania się odpowiedzi układu RLC i poprawić czułość klasyfikatora na występowanie tych klas.

W celu redukcji szumu i wygładzenia przebiegów zastosowano filtrację metodą Savitzky'ego–Golaya o parametrach długość ramki filtru: 11 próbek, rząd wielomianu: 2. Zastosowanie tej metody pozwala zachować cechy sygnału, takie jak maksima lokalne czy nachylenia zboczy, przy jednoczesnym ograniczeniu losowych fluktuacji związanych z szumem zarejestrowanym przez kartę pomiarową. Filtr Savitzky'ego–Golaya jest powszechnie stosowany w analizie danych eksperymentalnych, gdyż umożliwia wygładzanie sygnału bez istotnej deformacji jego kształtu, co ma istotne znaczenie dla poprawnego przebiegu procesu uczenia [57, 71, 58].

Weryfikacja poprawności przyjętej metody generowania kombinacji sygnałów dokonano porównania ich przebiegu z odpowiedziami układu RLC zarejestrowanymi eksperymentalnie. Celem tej weryfikacji było sprawdzenie, czy wygenerowane klasy mieszane mogą w wystarczającym stopniu odwzorować charakter rzeczywistych odpowiedzi, tak aby mogły być wykorzystane w procesie uczenia klasyfikatora. Na rysunku 7.7 zaprezentowano przykład porównania odpowiedzi układu dla klasy mieszanej AB/BA zarejestrowanej przez kartę pomiarową z przebiegiem wygenerowanym na potrzeby rozszerzenia zbioru treningowego.



Rysunek 7.7: Porównanie pojedynczej odpowiedzi układu RLC dla klasy mieszanej AB zarejestrowanej przez kartę pomiarową (kolor niebieski) oraz klasą wygenerowaną w celu treningu klasyfikatora (kolor czerwony).

Widoczna zgodność przebiegów potwierdza poprawność zastosowanej metody generowania obrazów, które następnie wykorzystywane są w procesie uczenia sieci. Dzięki temu możliwe jest rozszerzenie dostępnego zbioru danych o próbki syntetyczne, które zachowują istotne cechy rzeczywistych odpowiedzi układu, a tym samym wspomagają proces klasyfikacji.

Po potwierdzeniu metody, każda klasa (A, B, AA, AB/BA, BB) została przetworzona oddzielnie i przedstawiona w postaci wykresu, następnie zapisana jako obraz rastrowy w przestrzeni RGB. Wizualizacje wygenerowano za pomocą funkcji `plot` w środowisku MATLAB, przy czym usunięto osie oraz oznaczenia, aby uprościć strukturę obrazu. Dodatkowo wprowadzono stałe ograniczenie osi Y do zakresu $[-0.06, 0.06]$, co umożliwia ujednolicenie skali sygnałów i eliminuje wpływ różnic amplitudowych. Długość sygnału w obrazie ograniczono do 200 próbek, co – zgodnie z analizą klas mieszanych przedstawioną na rysunkach 7.4–7.6 – stanowi wartość wystarczającą do uchwycenia przypadków mieszanych.



(a) Klasa A.



(b) Klasa AA.



(c) Klasa AB.



(d) Klasa BA.



(e) Klasa B.



(f) Klasa BB.

Rysunek 7.8: Przykłady wygenerowanych obrazów dla poszczególnych klas: (a) A, (b) AA, (c) AB, (d) BA, (e) B oraz (f) BB.

Na rysunku 7.8 zaprezentowano przykłady wygenerowanych obrazów reprezentujących dane wejściowe do procesu treningu klasyfikatora. Każdy obraz odpowiada pojedynczej odpowiedzi układu RLC dla utwo-

rzonych klas. Widoczne klasy obejmują zarówno pojedyncze impulsy (**A**, **B**), jak i rzadko występujące sekwencje podwójnych impulsów (**AA**, **BB**, **AB/BA**). Zauważalne różnice pomiędzy klasami polegają przede wszystkim na odmiennej strukturze sygnału i poziomie amplitudy, co znajduje odzwierciedlenie w uzyskanych reprezentacjach wizualnych.

7.5 Przygotowanie klasyfikatora do treningu

Wybrane sieci VGG16 oraz VGG19 uruchomiono w środowisku MatLAB. Algorytm składa się z wczytania i etykietowania zbioru obrazów, losowego podziału na zbiory uczący i walidacyjny, dopasowania wymiaru wejściowego obrazów oraz augmentacji danych, modyfikacji końcowej warstwy softmax do liczby klas oraz konfiguracji procesu uczenia z walidacją w trakcie treningu.

Obrazy są wczytywane za pomocą funkcji `imageDatastore` z włączoną obsługą podkatalogów oraz opcją `LabelSource=foldernames`, co oznacza, że nazwa folderu jest interpretowana jako etykieta klasy. Taki sposób organizacji danych jest związane z procesem generowania klas do treningu i minimalizuje ryzyko błędów etykietowania, a sam `Datastore` umożliwia strumieniowe przetwarzanie dużych zbiorów. Zbiór podzielono na część uczącą (80%) i walidacyjną (20%) poleceniem:

```
(imdsTrain, imdsValidation) = splitEachLabel(imds, 0.8, 'randomized');
```

Losowość podziału redukuje korelację między próbkami obu zbiorów i ogranicza ryzyko przeuczenia na specyficznych podzbiorach danych. Zachowanie proporcji klas po podziale zmniejsza ryzyko faworyzowania klas.

Sieć VGG16 czy też VGG19 wymaga plików wejściowych o rozmiarze $224 \times 224 \times 3$, stąd tworzymy bufor danych `augmentedImageDatastore` z odpowiednim rozmiarem obrazu `inputSize`. W celu zwiększenia różnorodności próbek i poprawy odporności na wariancję w danych dodatkowo stosujemy augmentację w trakcie wczytywania obrazów:

- odbicie w poziomie: `RandXReflection = true`,
- rotacja losowa: $\theta \sim \mathcal{U}(-15^\circ, 15^\circ)$,
- skalowanie: $s \sim \mathcal{U}(0.9, 1.1)$,
- translacje: $\Delta x, \Delta y \sim \mathcal{U}(-5, 5)$ pikseli.

Tak dobrany zestaw transformacji modeluje realistyczne zmienności geometryczne bez wprowadzania artefaktów, które mogłyby zaburzać rozpoznawanie cech dla klasyfikacji.

Liczbę neuronów w ostatniej w pełni połączonej warstwie (`fullyConnectedLayer`) dopasowano do liczby klas $K = |\text{categories}(\text{imds.Labels})|$. Zwiększono współczynniki uczenia tylko dla tej warstwy (`WeightLearnRateFactor` i `BiasLearnRateFactor = 10`), co podkreśla adaptację części klasyfikacyjnej przy zachowaniu użytecznych reprezentacji ekstraktora cech. Kończącą warstwą jest `classificationLayer`.

Uczenie realizujemy optymalizatorem sgd (stochastic gradient descent with momentum) z parametrami:

$$\text{MiniBatchSize} = 32, \quad \text{MaxEpochs} = 350, \quad \text{InitialLearnRate} = 5 \cdot 10^{-4}, \quad \text{L2} = 10^{-4}.$$

Losowe tasowanie minibatchy w każdej epoce ('Shuffle'='every-epoch') poprawia mieszanie przykładów i stabilność zbieżności. Walidacja odbywa się na `augimdsValidation` co 20 iteracji (`ValidationFrequency=20`), co umożliwia monitorowanie metryk i wczesne wykrycie przeuczenia. Włączony 'Verbose' oraz wykres 'training-progress' dostarczają bieżącej informacji o przebiegu uczenia.

Ostatecznie uczymy sieć wywołaniem:

```
trainedNet = trainNetwork(augimdsTrain, layers, options).
```

W praktyce rekomendowane jest utrzymanie stałego podziału walidacyjnego dla porównywalności eksperymentów oraz obserwację krzywych straty/dokładności (loss/accuracy) w celu ewentualnej korekty tempa uczenia lub zakresów augmentacji. W tabeli 7.9 przedstawiono podsumowanie parametrów procesu treningu dla sieci VGG16

Tabela 7.9: Parametry procesu treningu sieci VGG16

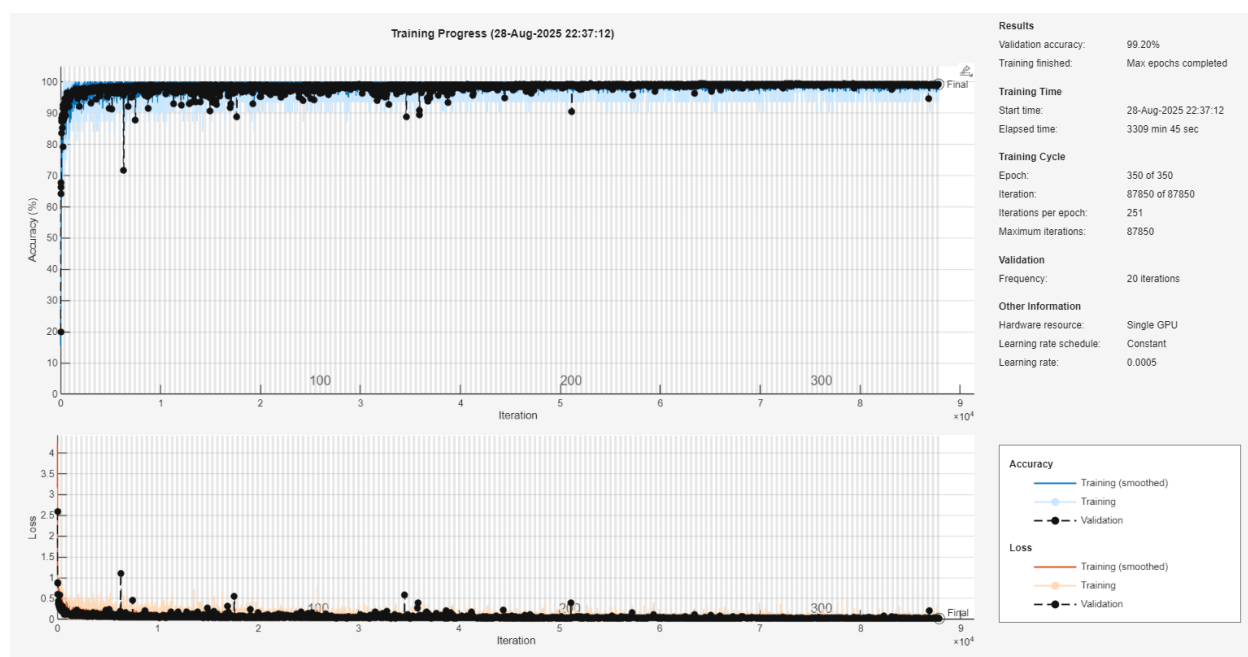
Parametr	Wartość / Zakres
Wejście sieci	$224 \times 224 \times 3$
Augmentacja	odbicie X, rotacja $\pm 15^\circ$, skala $[0.9, 1.1]$, przesunięcia ± 5 px
Podział danych	80% trening / 20% walidacja
Warstwa FC końcowa	liczba neuronów = liczba klas K
Współczynnik uczenia (ostatnia FC)	$\times 10$ dla wag i biasów
Optymalizator	SGDM
Rozmiar mini-batch	32
Liczba epok	350
Początkowe LR	$5 \cdot 10^{-4}$
Regularyzacja L2	$1 \cdot 10^{-4}$
Walidacja	co 20 iteracji

7.6 Trening klasyfikatora z wykorzystaniem danych dla układu RLC

Jak wspomniano w rozdziałach powyżej, trening klasyfikatora opierał się o pięć klas odpowiadających różnym kombinacjom impulsów: Klasa_A, Klasa_AA, Klasa_AB_BA, Klasa_B oraz Klasa_BB. Każda z klas została zrównoważona pod względem liczności. W zbiorze testowym znalazło się łącznie 2012 przykładów, przy czym liczba próbek przypadających na pojedynczą klasę wynosiła od 402 do 403 obrazów. Podobny rozkład zastosowano dla zbiorów treningowego i walidacyjnego, co zapewniło równomierne obciążenie modelu wszystkimi klasami na każdym etapie uczenia. Dzięki wyrównanemu rozkładowi danych wyniki metryk takich jak precyzja, czułość czy miara F1 odzwierciedlają faktyczną zdolność modelu do rozróżniania sygnałów odpowiadających wszystkim typom impulsów.

7.6.1 Sieć VGG16

Podczas procesu uczenia monitorowano dokładność klasyfikatora na zbiorze walidacyjnym. Funkcję straty oraz dokładność przedstawiono na rysunku 7.9. Na podstawie wyników weryfikacji obserwowano stabilizację krzywych określających zmiany wartości funkcji straty oraz dokładności klasyfikacji w kolejnych epokach dla zbiorów treningowego i walidacyjnego. Krzywe się pokrywają co sugeruje prawidłową proces treningu modelu. Zastosowanie augmentacji pozwoliło na redukcję przeuczenia.

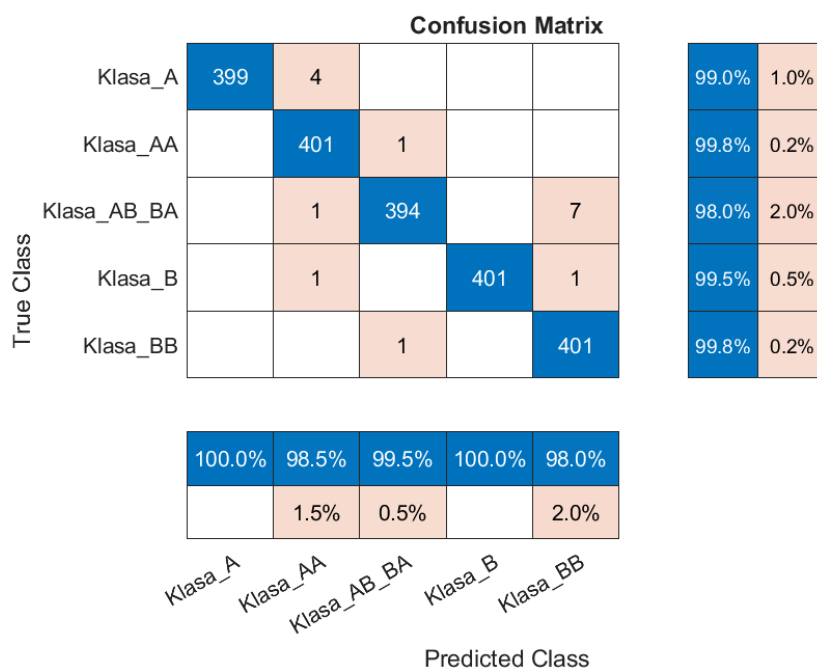


Rysunek 7.9: Przebieg procesu uczenia sieci VGG16.

Jako funkcję straty zastosowano *uśrednioną entropię krzyżową* (cross-entropy loss), która jest standardowym rozwiązaniem w problemach klasyfikacji wieloklasowej. Funkcja ta mierzy rozbieżność pomiędzy rozkładem prawdopodobieństwa przewidzianym przez sieć (na wyjściu warstwy softmax), a rzeczywistymi etykietami klas. Minimalizacja entropii krzyżowej prowadzi do zwiększenia pewności predykcji dla poprawnej klasy, a tym samym do poprawy skuteczności klasyfikatora. W trakcie uczenia funkcja straty była wyznaczana po każdej iteracji na zbiorze treningowym oraz dodatkowo, w regularnych odstępach (co 20 iteracji), na zbiorze walidacyjnym.

Drugą istotną miarą była dokładność klasyfikacji, definiowana jako stosunek liczby poprawnie zaklasyfikowanych próbek do liczby wszystkich próbek w danym zbiorze. Dokładność była liczona zarówno dla zbioru treningowego, jak i walidacyjnego. Na rysunku 7.9 przedstawiono przebieg obu metryk w czasie. Należy podkreślić, że raportowana końcowa dokładność walidacyjna na poziomie 99,2% odnosi się do **ostatniej warstwy klasyfikatora sieci (softmax)**, a nie do średniej wartości z całego procesu uczenia. Oznacza to, że wynik ten został osiągnięty w finalnym modelu po zakończeniu ostatniej epoki (350), a nie w trakcie wcześniejszych etapów uczenia. Przebieg krzywych dokładności i straty na rysunku wskazuje, że proces uczenia przebiegał bez wyraźnych oznak przeuczenia, a wartości obu miar uległy ustabilizowaniu już po kilku pierwszych epokach.

W środowisku MatLab, macierz pomyłek, oprócz klasycznej reprezentacji w postaci centralnego kwadratu przedstawiającego liczby zliczeń (poprawnych i błędnych klasyfikacji), zawiera również dwa dodatkowe panele z wartościami procentowymi. Prostokąt umieszczony po prawej stronie kwadratu prezentuje **czułość (Recall)** dla każdej klasy, czyli odsetek poprawnie rozpoznanych przykładów spośród wszystkich przykładów rzeczywistych. Z kolei prostokąt umieszczony poniżej odpowiada za **precyzję (Precision)**, tj. odsetek przykładów poprawnie zaklasyfikowanych do danej klasy spośród wszystkich próbek przypisanych do niej przez model. Każdy z dodatkowych paneli zawiera dwie wartości liczbowe. W przypadku panelu po prawej stronie są to dwie kolumny, natomiast w przypadku panelu dolnego dwa wiersze. Interpretacja jest następująca: wartość umieszczona w lewej kolumnie lub górnym wierszu odpowiada wskaźnikowi pozytywnemu (odpowiednio czułość lub precyzja), natomiast wartość po prawej lub w dolnym wierszu stanowi jego dopełnienie do 100%, czyli udział błędnych klasyfikacji. Takie przedstawienie metryk pozwala intuicyjnie ocenić zarówno skuteczność klasyfikatora w rozpoznawaniu poszczególnych klas, jak i wielkość popełnianych błędów.



Rysunek 7.10: Zestawienie wyników klasyfikacji po procesie uczenia sieci VGG16 dla układu RLC.

Rysunek 7.10 prezentuje macierz pomyłek dla pięciu klas: Klasa_A, Klasa_AA, Klasa_AB_BA, Klasa_B oraz Klasa_BB. Suma elementów na przekątnej wynosi 1996, a liczba pomyłek 16, co daje dokładność całkowitą równą $\frac{1996}{2012} = 0.9920$ (tj. 99.20%). Dominacja wartości na przekątnej wskazuje na bardzo dobrą klasyfikację. Z prawego panelu (czułość) oraz dolnego panelu (precyzja) odczytujemy:

- **Klasa_A:** czułość = 99.0%, precyzja = 100.0%. Błędy to 4 przypadki zaklasyfikowane jako Klasa_AA.
- **Klasa_AA:** czułość = 99.8%, precyzja = 98.5%. Jedna obserwacja błędnie przypisana do Klasa_AB_BA; 6 fałszywych przypadków (FD) pochodzi z innych klas.

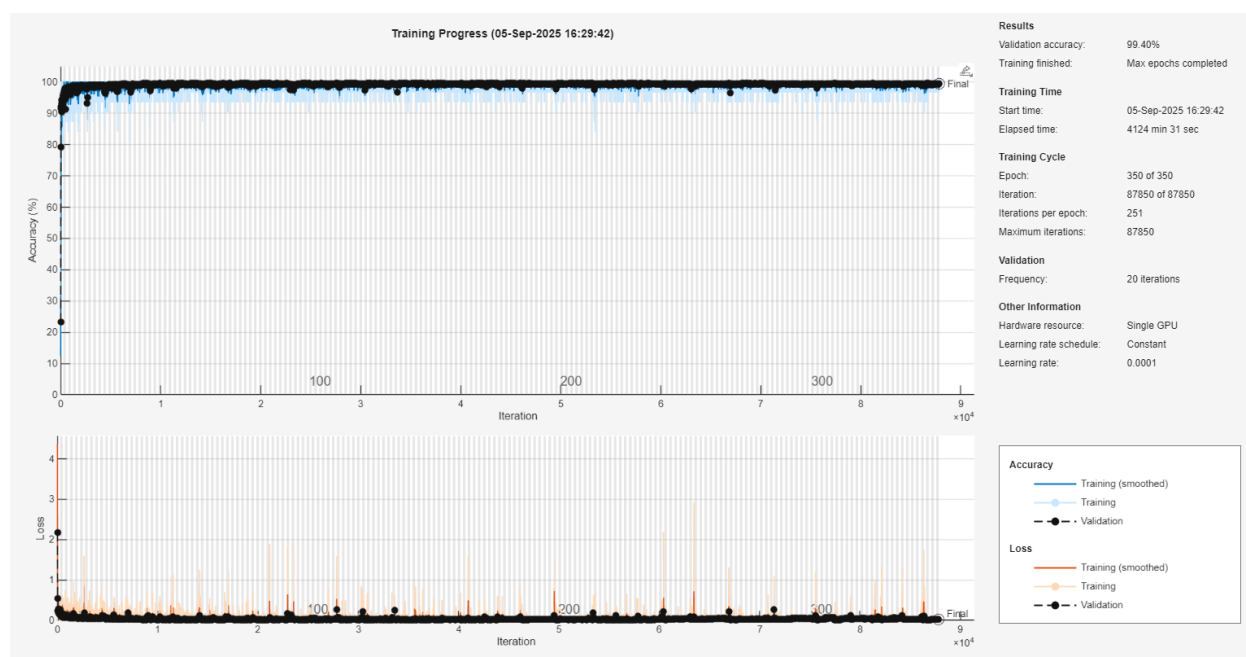
- **Klasa_AB_BA**: czułość = 98.0%, precyzja = 99.5%. To najłabszy (choć nadal bardzo dobry) wynik czułości; 8 pomyłek dzieli się na 1 przypadek do Klasa_A oraz 7 do Klasa_BB.
- **Klasa_B**: czułość = 99.5%, precyzja = 100.0%. Dwa błędy to pojedyncze omyłki do Klasa_A i Klasa_BB.
- **Klasa_BB**: czułość = 99.8%, precyzja = 98.0%. Jedyne błędy to klasyfikacja do Klasa_AB_BA; 8 fałszywych przypadków pochodzi z innych klas.

Wszystkie klasy osiągają bardzo wysokie wyniki — precyzja mieści się w przedziale 98.0%–100.0%, a czułość w przedziale 98.0%–99.8%. Jest to zgodne z raportowanymi metrykami liczbowymi. Najsilniejsza para pomyłek to Klasa_AB_BA vs Klasa_BB (7 przypadków). Drugi zauważalny motyw to pomyłki pomiędzy Klasa_A i Klasa_AA. Obydwa wzorce sugerują, że klasy zbliżone co do wartości amplitudy lub o podobnym wzorcu cech są najtrudniejsze do rozróżnienia.

Wyniki zbiorcze wskazują, że model uzyskał łączny F1-score równy 0.992, przy relatywnie niewielkiej liczbie błędnych klasyfikacji (16 fałszywych pozytywów oraz 16 fałszywych negatywów w skali całego zbioru testowego). Tak wysoka skuteczność ogólna potwierdza, że zaproponowany system klasyfikacji jest dobrze dopasowany do zadania i może być z powodzeniem wykorzystywany do dalszej analizy odpowiedzi układu.

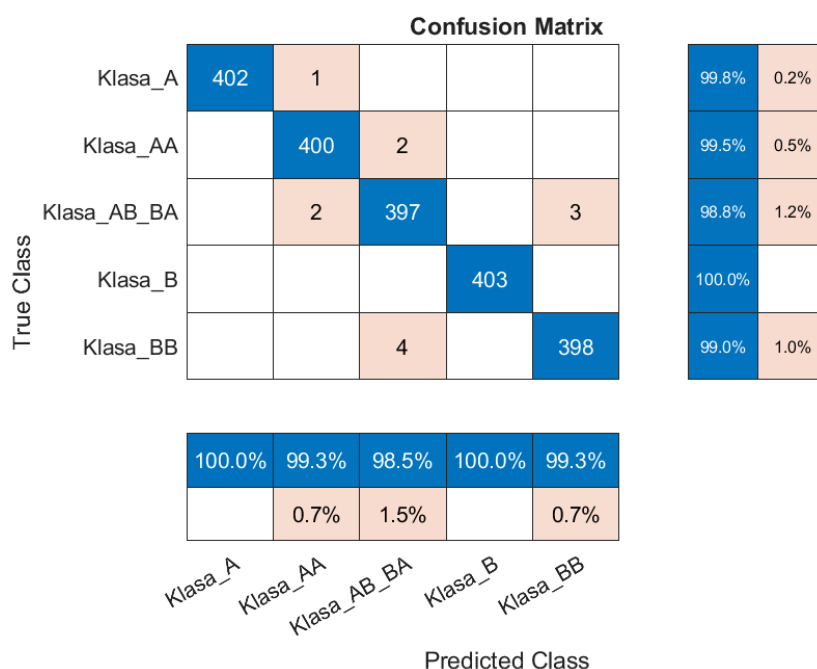
7.6.2 Sieć VGG19

Tak jak w przypadku sieci VGG16, monitorowano dokładność klasyfikatora na zbiorze walidacyjnym. Proces został przedstawiony na rysunku 7.11. Krzywe reprezentujące dokładność i funkcja straty wykazują szybki wzrost jakości modelu na początkowym etapie, a następnie stabilizację, co świadczy o prawidłowym procesie uczenia. Dzięki zastosowaniu augmentacji danych nie zaobserwowano istotnego przeuczenia.



Rysunek 7.11: Reprezentacja procesu trenowania sieci dla architektury VGG19.

Po zakończonym procesie treningu sieć osiągnęła dokładność walidacyjną na poziomie 99.4%. Wartość ta została obliczona na podstawie porównania etykiet rzeczywistych z etykietami przewidywanymi. Dla szczegółowej oceny wyników wygenerowano macierz pomyłek, którą przedstawiono na rysunku 7.12.



Rysunek 7.12: Zestawienie wyników klasyfikacji po procesie uczenia sieci VGG19 dla układu RLC.

Rysunek 7.12 przedstawia wyniki klasyfikacji dla pięciu klas: Klasa_A, Klasa_AA, Klasa_AB_BA, Klasa_B oraz Klasa_BB. Suma elementów na przekątnej wynosi 2000, przy 12 fałszywych pozytywach (FD) i 12 fałszywych negatywach (FU). Całkowita dokładność wynosi $\frac{2000}{2024} \approx 0.9940$ (99.40%). Podobnie jak w przypadku VGG16, dominacja wartości na przekątnej potwierdza bardzo wysoką skuteczność modelu. Z prawego panelu (czułość) oraz dolnego panelu (precyzja) odczytujemy:

- **Klasa_A**: czułość = 99.8%, precyzja = 100.0%. Jedyny błąd to pojedyncze pominięcie przykładu tej klasy,
- **Klasa_AA**: czułość = 99.5%, precyzja = 99.3%. Pojawiły się dwie pomyłki do Klasa_AB_BA oraz trzy fałszywe przypisania z innych klas,
- **Klasa_AB_BA**: czułość = 98.8%, precyzja = 98.5%. To najniższa (choć nadal wysoka) skuteczność – pięć pomyłek obejmuje dwie klasyfikacje jako Klasa_AA oraz trzy jako Klasa_BB,
- **Klasa_B**: czułość = 100.0%, precyzja = 100.0%. Brak błędnych klasyfikacji w tej klasie,
- **Klasa_BB**: czułość = 99.0%, precyzja = 99.3%. Cztery przypadki błędnie przypisano do Klasa_AB_BA, a trzy fałszywe przykłady pochodziły z innych klas.

Podsumowując, wszystkie klasy uzyskały bardzo wysoką precyzję i czułość – większość wartości przekracza 99%, a najniższe wskaźniki oscylują wokół 98.5%. Miara F1-score utrzymuje się na poziomie powyżej 0.986, co potwierdza stabilność modelu.

7.6.3 Porównanie wyników VGG16 i VGG19

W celu pełniejszej oceny jakości klasyfikacji zestawiono wyniki uzyskane przez obie architektury (VGG16 oraz VGG19). Porównując obie sieci, można zauważyć, że w procesie trenowania klasyfikatora:

- Zarówno VGG16, jak i VGG19 osiągają wysokie wartości precyzji i czułości (powyżej 0.98) dla wszystkich klas, co świadczy o stabilności klasyfikacji,
- Sieć VGG16 uzyskała globalny F1-score na poziomie 0.992, natomiast VGG19 osiągnęła wyższy wynik 0.994. Różnica ta wskazuje, że dodatkowe warstwy konwolucyjne w VGG19 mogą nieznacznie poprawiać zdolność sieci do rozróżniania trudniejszych przypadków,
- Największym wyzwaniem dla obu architektur okazała się klasa `Klasa_AB_BA`. W przypadku VGG16 czułość wyniosła 0.980, a precyzja 0.995, natomiast dla VGG19 uzyskano czułość 0.988 i precyzję 0.985. Choć wartości te różnią się nieznacznie, widoczne jest, że VGG19 lepiej radziła sobie z wychwytywaniem większej liczby przykładów tej klasy,
- Klasy `Klasa_A` i `Klasa_B` osiągnęły w obu modelach najlepsze rezultaty, często z precyzją i czułością równą 1.0, co wskazuje na brak błędnych klasyfikacji lub ich minimalną liczbę.

Jeśli chodzi o proces uczenia, to:

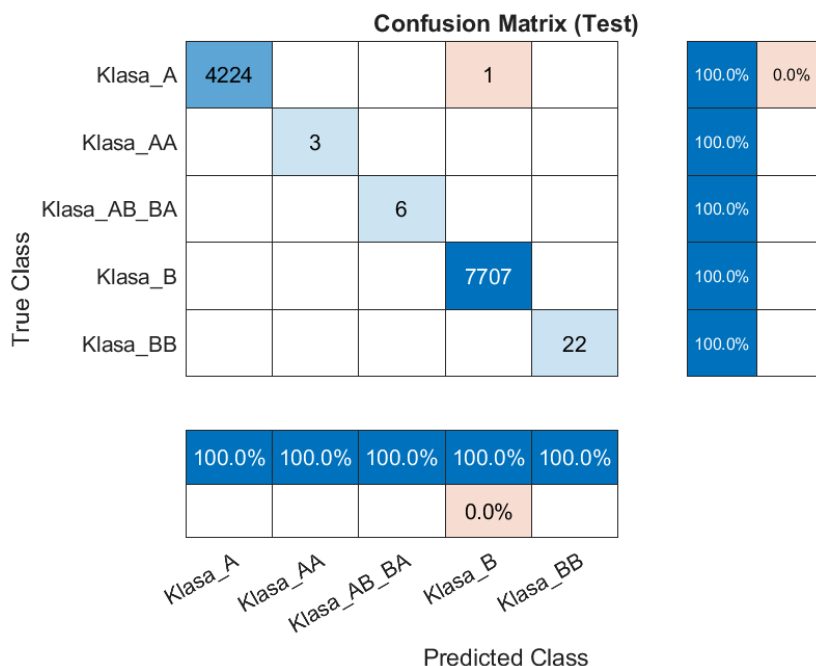
- W przypadku VGG16 uzyskano dokładność walidacyjną równą 99.2%, przy stabilnej krzywej uczenia i niewielkiej liczbie błędnych klasyfikacji (16 FD i 16 FU).
- VGG19 osiągnęła jeszcze wyższą dokładność walidacyjną 99.4%, co sugeruje lepsze dopasowanie modelu do danych treningowych. Liczba błędów była również niższa (12 FD i 12 FU).

7.7 Skuteczności klasyfikatora na próbach spoza zbioru walidacyjnego

Po etapie treningu i walidacji, przeprowadzonych zarówno na danych pochodzących z eksperymentu (klasy podstawowe A i B), jak i na danych przygotowanych sztucznie (klasy mieszane), dokonano weryfikacji działania klasyfikatora na danych spoza zbioru treningowego. Celem tego etapu było sprawdzenie, w jakim stopniu wytrenowane modele zachowują zdolność poprawnego rozróżniania klas. Do analizy wykorzystano dziesiątą serię pomiarową, obejmującą dwa różne rozkłady prawdopodobieństwa wystąpienia impulsu $p(100mV) = p(200mV) = 0.5$ oraz $p(100mV) = 0.2$ i $p(200mV) = 0.8$. Zarejestrowane przebiegi czasowe zostały poddane identycznej procedurze przetwarzania jak w przypadku zbioru treningowego, to znaczy podzielone na segmenty, przypisane do klas (`Klasa_A`, `Klasa_AA`, `Klasa_AB_BA`, `Klasa_B`, `Klasa_BB`), a następnie przekształcone do postaci obrazów. Tak przygotowane dane posłużyły jako wejście do klasyfikatorów opartych na architekturach VGG16 i VGG19, umożliwiając porównanie skuteczności obu modeli w warunkach eksperymentalnych.

7.7.1 Wyniki dla sieci VGG16

Rysunek 7.13 przedstawia macierz pomyłek uzyskaną dla klasyfikacji danych spoza zbioru treningowego. Wartości na przekątnej wskazuje, że sieć niemal bezbłędnie rozpoznaje poszczególne klasy. Łącznie poprawnie sklasyfikowano 11 963 obrazów, podczas gdy liczba błędów wyniosła zaledwie 1. Oznacza to, że dokładność całkowita klasyfikacji wyniosła $\frac{11963}{11964} = 0.9999$, czyli 99.99%.



Rysunek 7.13: Zestawienie wyników klasyfikacji po procesie testowania sieci VGG16 dla układu RLC na danych spoza zbioru treningowego.

Analizując prawe i dolne panele (czułość i precyzja) można zauważyć, że dla większości klas zarówno precyzja, jak i czułość osiągają wartość równą 100%. Wyjątek stanowią Klasa_A i Klasa_B:

- **Klasa_A:** czułość = 99.98%, precyzja = 100%. Pojedynczy błąd polegał na błędnym zaklasyfikowaniu jednej próbki jako Klasa_B,
- **Klasa_B:** czułość = 100%, precyzja = 99.99%.
- **Klasa_AA, Klasa_AB_BA, Klasa_BB.** Dla tych kategorii zarówno precyzja, jak i czułość osiągnęły wartość 100%, co oznacza brak błędów klasyfikacji.

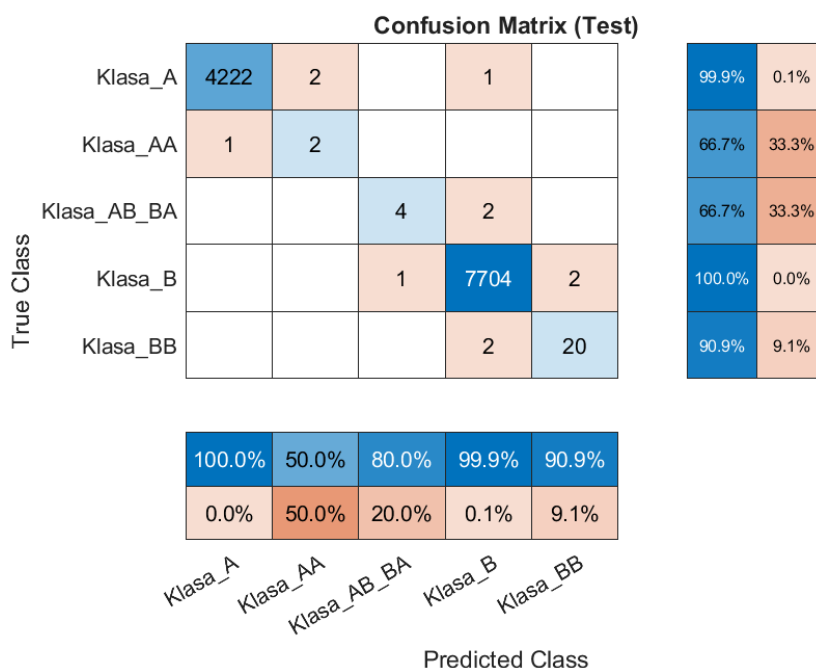
Wysoka zgodność klasyfikacji na danych spoza zbioru treningowego wskazuje, że model nie tylko dobrze odwzorował strukturę danych treningowych, ale również zachował zdolność generalizacji danych pochodzących spoza tego zbioru. Pojedyncze błędy dotyczyły jedynie klas podstawowych (A i B) i miały charakter marginalny ($< 0.01\%$ wszystkich próbek). Wyniki te potwierdzają, że przygotowany system klasyfikacji jest w stanie skutecznie rozpoznawać różne typy impulsów także w praktycznych warunkach pomiarowych.

Uzyskane wyniki, zarówno w procesie treningu, jak i podczas klasyfikacji przypadków spoza zbioru testowego, charakteryzują się wyjątkowo wysoką dokładnością (odpowiednio 99.2% oraz 99.99%). Tak dobre rezultaty można uzasadnić kilkoma czynnikami. Odpowiednie przygotowanie danych wejściowych (segmentacja, standaryzacja, równomierny rozkład klas) pozwoliło na ograniczenie efektu niezrównoważonych zbiorów, które mogłyby wprowadzać stronniczość w działaniu modelu.

Wysoka zgodność wyników na danych spoza zbioru z rezultatami treningowymi sugeruje, że model nauczył się rzeczywistych zależności obecnych w sygnałach, a nie jedynie zapamiętał wzorce ze zbioru uczącego. Fakt ten potwierdza prawidłowość procesu przygotowania danych oraz adekwatność zastosowanej architektury sieci konwolucyjnych do rozwiązywanego problemu.

7.7.2 Wyniki dla sieci VGG19

Rysunek 7.14 przedstawia macierz pomyłek uzyskaną dla klasyfikacji danych spoza zbioru testowego z wykorzystaniem sieci VGG19. Dane wykorzystane do tej klasyfikacji są tymi samymi, które użyto dla sieci VGG16. Podobnie jak w przypadku architektury VGG16, na przekątnej macierzy pokazana jest liczba prawidłowych klasyfikacji, jednak liczba błędów jest nieco wyższa, zwłaszcza w odniesieniu do klas mieszanych i słabiej reprezentowanych. Łącznie poprawnie sklasyfikowano 11 963 obrazów, przy 11 błędach. Dokładność całkowita klasyfikacji wyniosła $\frac{11963}{11974} = 0.9991$ (tj. 99.91%).



Rysunek 7.14: Zestawienie wyników klasyfikacji po procesie testowania sieci VGG19 dla układu RLC na danych spoza zbioru treningowego.

Analiza paneli bocznych (czułość i precyzja) wskazuje, że dla głównych klas (Klasa_A i Klasa_B) wartości precyzji i czułości są bardzo wysokie, powyżej 99.9%, co oznacza, że błędy klasyfikacji występowały sporadycznie i miały marginalny charakter. Natomiast klasy mieszane (Klasa_AA, Klasa_AB_BA oraz Klasa_BB) cechowały się wyraźnie niższymi wskaźnikami.

- **Klasa_A**: czułość = 99.93%, precyzja = 99.98%. Pojedyncze błędy polegały na błędnym przypisaniu 3 próbek do innych kategorii oraz 1 przypadku fałszywego przypisania,
- **Klasa_B**: czułość = 99.96%, precyzja = 99.94%. Sieć niemal idealnie rozpoznawała tę kategorię – 3 próbki zostały pomyłone, a 5 błędnie przypisano do tej klasy,
- **Klasa_AA**: czułość = 66.7%, precyzja = 50.0%. Niewielka liczność tej klasy sprawiła, że pojedyncze błędy silnie obniżyły metryki – jedna próbka została pominięta, a dwie inne błędnie przypisano do tej kategorii,
- **Klasa_AB_BA**: czułość = 66.7%, precyzja = 80.0%. Dwie próbki nie zostały rozpoznane, jedna została błędnie przypisana do tej klasy, a cztery sklasyfikowano poprawnie,
- **Klasa_BB**: czułość = 90.9%, precyzja = 90.9%. Dwa przypadki zostały pominięte oraz dwa błędnie przypisane do tej kategorii.

Wyniki zbiorcze (Suma) wskazują, że spośród 11 974 analizowanych obrazów poprawnie sklasyfikowano 11 962, przy 11 fałszywych. Łączny F1-score osiągnął wartość 0.9991, co oznacza bardzo wysoką skuteczność klasyfikatora mimo lokalnych trudności w odróżnianiu niektórych klas mieszanych.

Podsumowując, sieć VGG19 bardzo dobrze radzi sobie z rozpoznawaniem klas podstawowych (A i B), natomiast w przypadku klas mieszanych (AA, AB_BA, BB) pojawiają się problemy, które obniżają wartości metryk. Może to wynikać z niewielkiej liczności próbek w tych kategoriach oraz dużego podobieństwa ich cech do sygnałów innych klas. W przypadku klas mieszanych, sieć nie nauczyła się wzorców.

7.7.3 Podsumowanie wyników testów klasyfikatora

Do treningu oraz testów klasyfikatora wykorzystano te same zbiory danych. Uzyskane wyniki (tabela 7.10) wskazują, że obie architektury osiągnęły wysoką skuteczność klasyfikacji zarówno w procesie treningu, jak i w testach na danych pochodzących spoza zbioru testowego. Sieć VGG19 charakteryzowała się nieco wyższą dokładnością treningową (99.4% wobec 99.2% dla VGG16), co potwierdza jej większą zdolność reprezentacyjną wynikającą z dodatkowych warstw konwolucyjnych. Jednak w przypadku testów na danych spoza zbioru wyższe wyniki uzyskała sieć VGG16 (99.99% wobec 99.91% dla VGG19). Może to wynikać z faktu, że bardziej złożona architektura VGG19 była bardziej podatna na przeuczenie i gorzej generalizowała na klasach rzadszych (Klasa_AA, Klasa_AB_BA, Klasa_BB). W efekcie, mimo lepszych wyników treningowych, VGG19 wykazała mniejszą dokładnością w klasyfikowaniu klas mieszanych.

Tabela 7.10: Porównanie dokładności klasyfikacji dla sieci VGG16 i VGG19 w procesie uczenia i testów

Sieć	Dokładność treningu [%]	Dokładność testów [%]
VGG16	99.2	99.99
VGG19	99.4	99.91

Przyczyny słabszej jakości rozpoznawania klas mieszanych przez VGG19 można wiązać z czynnikami:

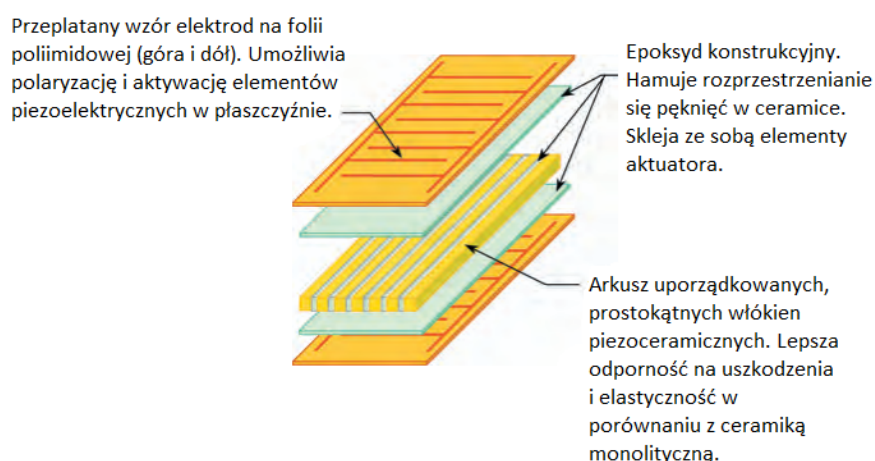
- **Podobieństwo cech sygnałów** – klasy mieszane wykazują wysoki poziom podobieństwa do klas bazowych, co wprowadza niejednoznaczności i sprzyja błędnym klasyfikacjom.
- **Większa głębokość sieci VGG19** – dodatkowe warstwy konwolucyjne mogły prowadzić do nadmiernego dopasowania modelu do wzorców ze zbioru treningowego, kosztem zdolności generalizacji dla trudniejszych przypadków.

Podsumowując, sieć VGG16 okazała się modelem stabilnym i dobrze dopasowanym do charakterystyki badanego problemu, co potwierdzają uzyskane wyniki testowe. Z kolei architektura VGG19, mimo swojej teoretycznej przewagi związanej z większą głębokością, wymagałaby zastosowania dodatkowych mechanizmów regularizacji lub rozszerzenia zbioru danych treningowych, aby osiągnąć porównywalny poziom generalizacji. Należy jednak podkreślić, że uzyskany model nie generalizuje wyników na inne rozkłady prawdopodobieństwa wystąpienia impulsów. W celu uwzględnienia tego typu przypadków konieczne jest ich włączenie do zbioru uczącego i walidacyjnego.

Rozdział 8

Klasyfikacja odpowiedzi belki kompozytowej FR-4 z piezoelektrykiem Macro Fiber Composite™ na losową serię impulsów

W następnym etapie prac badawczych analizowano układ ciągły. Badaniu poddano belkę kompozytową z piezoelektrykiem Macro Fiber Composite™ (MFC) [11]. MFC jest nowoczesnym materiałem piezoelektrycznym, pełniącym jednocześnie rolę elementu wykonawczego *aktuatora* oraz czujnika *sensora*. Technologia MFC została opracowana przez NASA w 1996 roku, natomiast od 2002 roku jest komercjalizowana przez firmę Smart Material Corp., która odpowiada za rozwój i dystrybucję tych kompozytów. Obecnie dostępnych jest ponad 20 typów MFC, różniących się wymiarami, parametrami elektrycznymi oraz trybem pracy.

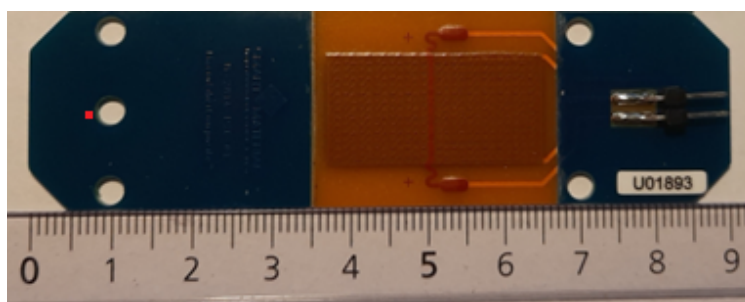


Rysunek 8.1: Struktura materiału piezoelektrycznego [65].

Konstrukcja MFC (rysunek 8.1) opiera się na prostokątnych prętach piezoceramicznych, umieszczonych pomiędzy warstwami kleju, elektrod oraz folii polimidowej. Elektrody rozmieszczone są w formie wzoru

interdigitowanego (IDE), co umożliwia bezpośrednie przekazywanie napięcia do włókien ceramicznych. Dzięki temu możliwe jest prowadzenie polaryzacji i pomiarów w płaszczyźnie materiału. Taka budowa zapewnia wysoką wytrzymałość, odporność na uszkodzenia oraz gotowość do użycia. Grubość rzędu kilku dziesiątych milimetra sprawia, że MFC stanowi ciekawą i elastyczną strukturę, którą można zarówno przyłutować do powierzchni, jak i wbudować w kompozyty konstrukcyjne.

Istotną cechą jest możliwość jednoczesnego działania zarówno jako aktuator, jak i sensor, co znacząco zwiększa zakres zastosowań. MFC charakteryzuje się także skuteczną konwersją energii elektrycznej w mechaniczną i odwrotnie, co pozwala na jego wykorzystanie w systemach tłumienia drgań, pozyskiwania energii czy aktywnej kontroli konstrukcji. Może działać jak również jako czujnik. W stanie beznapięciowym MFC działa jak bardzo czuły tensometr, zdolny do detekcji odkształceń, drgań, hałasu oraz pozyskiwania energii z wibracji (*energy harvesting*). Ponadto sam MFC może działać w dwóch trybach: d_{33} (rozszerzanie) oraz d_{31} (kurczenie). W niniejszej pracy zastosowano model **M-2814-P2**, należący do grupy P2 (d_{31}), który charakteryzuje się odkształceniem ujemnym (kurczeniem). Wybór aktuatora działającego w trybie d_{31} jest uzasadniony. Konfiguracja ta jest szczególnie efektywna przy wzbudzaniu elementów smukłych, w których dominujące odkształcenia mają charakter zginania. Tryb d_{31} pozwala na generowanie sił wewnątrz materiału w kierunku zgodnym z głównymi liniami naprężeń zginających, co przekłada się na wyraźniejsze odpowiedzi układu na i powtarzalność wyników.



Rysunek 8.2: Belka z laminatu FR-4 (90 mm × 25 mm × 1 mm) z zamocowanym aktuatorem MFC M-2814-P2 oraz punktem pomiarowym.

Wybór aktuatora (rysunek 8.2) minimalizuje ryzyko wprowadzania do układu zbyt dużych, nieliniowych deformacji, które mogłyby zaburzać analizę zarejestrowanych odpowiedzi. Jednocześnie materiał ten zapewnia wystarczającą energię wzbudzenia w badanym zakresie napięć (tabela 8.1). Wybór elementu MFC pracującego w trybie d_{31} wynikał z potrzeby uzyskania układu o dużej czułości, zdolnego do odwzorowania odkształceń smukłej belki, przy jednoczesnej możliwości kontrolowanego wzbudzania drgań. Konfiguracja ta okazała się najlepiej dostosowana do charakteru prowadzonych badań, w których rolę odgrywają subtelne różnice w odpowiedziach na losowe wymuszenia impulsowe.

Do najważniejszych zalet technologii Macro Fiber Composite należy wysoka elastyczność oraz niewielka grubość. Dodatkowo warstwowa konstrukcja oparta na folii polimidowej i klejach epoksydowych zapewnia dużą trwałość oraz odporność na działanie czynników środowiskowych. Piezoelektryk **M-2814-P2** (rysunek 8.2), należący do grupy P2 (d_{31}). Posiada on aktywną powierzchnię o wymiarach 28 × 14 mm (1,1 cala × 0,55 cala), pojemność 30, 78 nF, odkształcenie swobodne –630 ppm oraz siłę blokującą –76 N.

Aktuator MFC M-2814-P2 został zamocowany na belce wykonanej z laminatu epoksydowego wzmocnionego włóknem szklanym (FR-4). Belka ta posiada wymiary: **90 mm × 25 mm**, przy grubości **1 mm** (40 mil).

FR-4 jest standardowym laminatem stosowanym w elektronice i konstrukcjach lekkich. Do jego głównych właściwości należą: gęstość ok. $1,85 \text{ g/cm}^3$, moduł Younga w zakresie 16–24 GPa, wytrzymałość na zginanie rzędu 340–480 MPa, współczynnik Poissona $\nu \approx 0,13$ – $0,20$ oraz maksymalna temperatura pracy 130–140°C. Wybór materiału FR-4 zamiast klasycznych metali, takich jak aluminium czy stal, podyktowany był kilkoma względami. Przede wszystkim FR-4 ma znacznie mniejszy moduł Younga niż metale, co sprawia, że nawet przy niewielkich wymuszeniach generowanych przez aktuator MFC odkształcenia belki są łatwo zauważalne i możliwe do dokładnej analizy. W przypadku aluminium ($E \approx 70 \text{ GPa}$) czy stali ($E \approx 200 \text{ GPa}$) ta sama siła aktuatora prowadziłaby do znacznie mniejszych ugięć, co utrudniałoby rejestrację zjawisk w warunkach laboratoryjnych. Ponadto próby wykazały, że stosowanie materiałów takich jak aluminium bądź stal wymagałoby zastosowania wzmacniacza sygnału bądź innego aktuatora.

FR-4 jest materiałem o dużej stabilności, dobrej odporności termicznej oraz łatwości obróbki. Warto również podkreślić, że laminat FR-4, dzięki swojej warstwowej strukturze i stosunkowo niskiej gęstości, jest reprezentatywnym materiałem dla nowoczesnych konstrukcji kompozytowych stosowanych w lotnictwie czy inżynierii mechanicznej. Dzięki temu wyniki uzyskane podczas badań mogą być w przyszłości odniesione do bardziej zaawansowanych aplikacji inżynierskich.

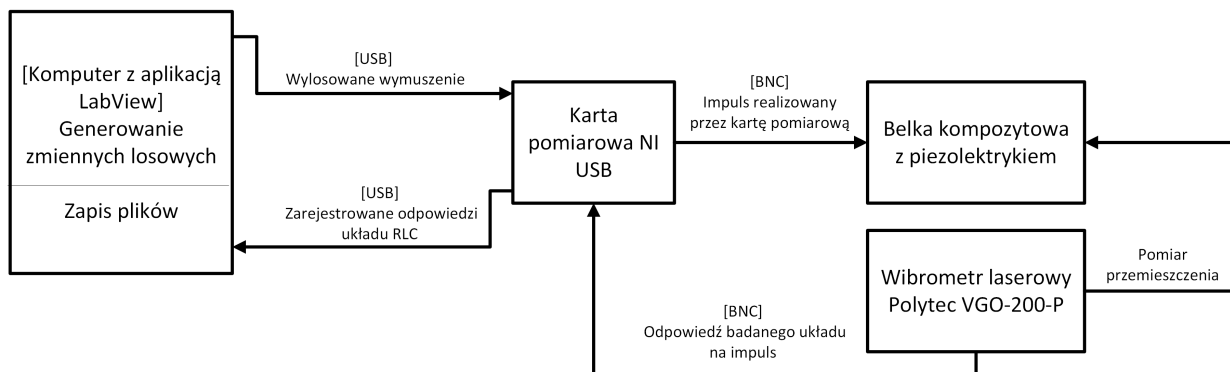
8.1 Stanowisko pomiarowe

Do przeprowadzenia badań eksperymentalnych zaprojektowano i wykonano stanowisko pomiarowe. Konstrukcja została zaprojektowana w środowisku Autodesk Inventor. Podstawę stanowiska stanowi płyta aluminiowa o grubości 10 mm, wycinana strumieniem wody w celu zapewnienia wysokiej dokładności wymiarowej. Jako materiał bazowy zastosowano stop aluminium serii EN AW-5083 (AlMg4.5Mn), powszechnie stosowany w budowie stanowisk laboratoryjnych ze względu na dobrą obrabialność i wysoką tolerancję wymiarową. Pod płytą zamocowano cztery wibroizolatory (czarne stopy), które mają na celu ograniczenie wpływu drgań zewnętrznych na pomiary oraz odizolowanie stanowiska od drgań własnych podłoża. Do zamocowania belki kompozytowej z aktuatorem MFC wykorzystano dwa aluminiowe piedestały, umożliwiające regulację położenia uchwytów w kierunku osiowym. Pomiędzy belką a aluminiowym uchwytem zastosowano przekładkę z twardej gumy technicznej, co zmniejsza ryzyko uszkodzenia próbki i równocześnie kompensuje lokalne naprężenia montażowe. Połączenia śrubowe wykonano z użyciem standardowych śrub klasy wytrzymałości 8.8. Punkt pomiarowy znajduje się na wolnym końcu belki (rys. 8.2), co umożliwia rejestrację odkształceń belki.

8.2 Opis stanowiska badawczego

Za rejestrację odpowiedzi odpowiadał wibrometr laserowy, a badanym obiektem była belka kompozytowa z piezoelektrykiem. Do rejestracji drgań wykorzystano wibrometr laserowy jednopunktowy Polytec VibroGo VGO-200P [80], który bazuje na zasadzie efektu Dopplera. Umożliwia on bezkontaktowy pomiar

przemieszczeń, prędkości oraz przyspieszeń drgań w szerokim zakresie częstotliwości do 320 kHz. Maksymalna prędkość pomiarowa do 6 m/s. Odległość pomiarowa: do 30 m. Zaletą zastosowania wibrometru laserowego w badaniach belki kompozytowej jest eliminacja wpływu dodatkowej masy czujników kontaktowych na odpowiedź badanego układu. Urządzenie charakteryzuje się kompaktową konstrukcją i klasą szczelności IP64, dzięki czemu może być stosowane zarówno w warunkach laboratoryjnych, jak i w środowisku przemysłowym. Ponadto, wyposażenie w ekran dotykowy, autofokus oraz możliwość bezprzewodowego sterowania ułatwia konfigurację pomiaru i poprawia jakość sygnału. Wbudowany rejestrator pozwala na zapis danych bezpośrednio w urządzeniu.



Rysunek 8.3: Uproszczony schemat doświadczenia z belką kompozytową.



Rysunek 8.4: Stanowisko pomiarowe z belką kompozytową.

Schemat układu pomiarowego przedstawiono na rysunku 8.3. Zrealizowane stanowisko badawcze składało się z komputera z aplikacją *LabVIEW*, odpowiedzialnego za generowanie losowych sekwencji impulsów oraz rejestrację odpowiedzi badanego układu, karty pomiarowej NI USB, belki kompozytowej z piezoelektrykiem MFC, pełniącej rolę badanego obiektu, wibrometru laserowego Polytec VGO-200P, umożliwiającego bezkontaktowy pomiar przemieszczeń. Całość układu zaprezentowano na fotografii (rys. 8.4). Stanowisko umożliwiało rejestrację odpowiedzi belki na pojedyncze impulsy oraz serie impulsów o charakterze losowym, analogicznie jak w przypadku wcześniejszych eksperymentów z układem RLC.

8.3 Przygotowanie zbioru danych

W tabeli 8.1 zestawiono przykładowe parametry wykorzystane w doświadczeniach z belką kompozytową z piezoceramicznym elementem MFC. W eksperymentach rozpatrzono dziesięć poziomów napięć wzbudzających U_i , odpowiadających kolejnym amplitudom impulsów η_i . Zakres wartości obejmuje przedział od 1 V do 10 V, co pozwala na analizę odpowiedzi układu zarówno w warunkach słabego, jak i silnego wymuszenia.

Tabela 8.1: Parametry doświadczenia wykorzystane w doświadczeniu z belką kompozytową z piezoceramiem MFC

Parametr	Wartość
$U_1 (\eta_1)$	1 V
$U_2 (\eta_2)$	2 V
$U_3 (\eta_3)$	3 V
$U_4 (\eta_4)$	4 V
$U_5 (\eta_5)$	5 V
$U_6 (\eta_6)$	6 V
$U_7 (\eta_7)$	7 V
$U_8 (\eta_8)$	8 V
$U_9 (\eta_9)$	9 V
$U_{10} (\eta_{10})$	10 V
$p(\eta_i)$	0.1 (dla każdego $i = 1, \dots, 10$)
t_{pomiar}	600 s
Intensywność λ	10
Częstotliwość próbkowania	100 kHz

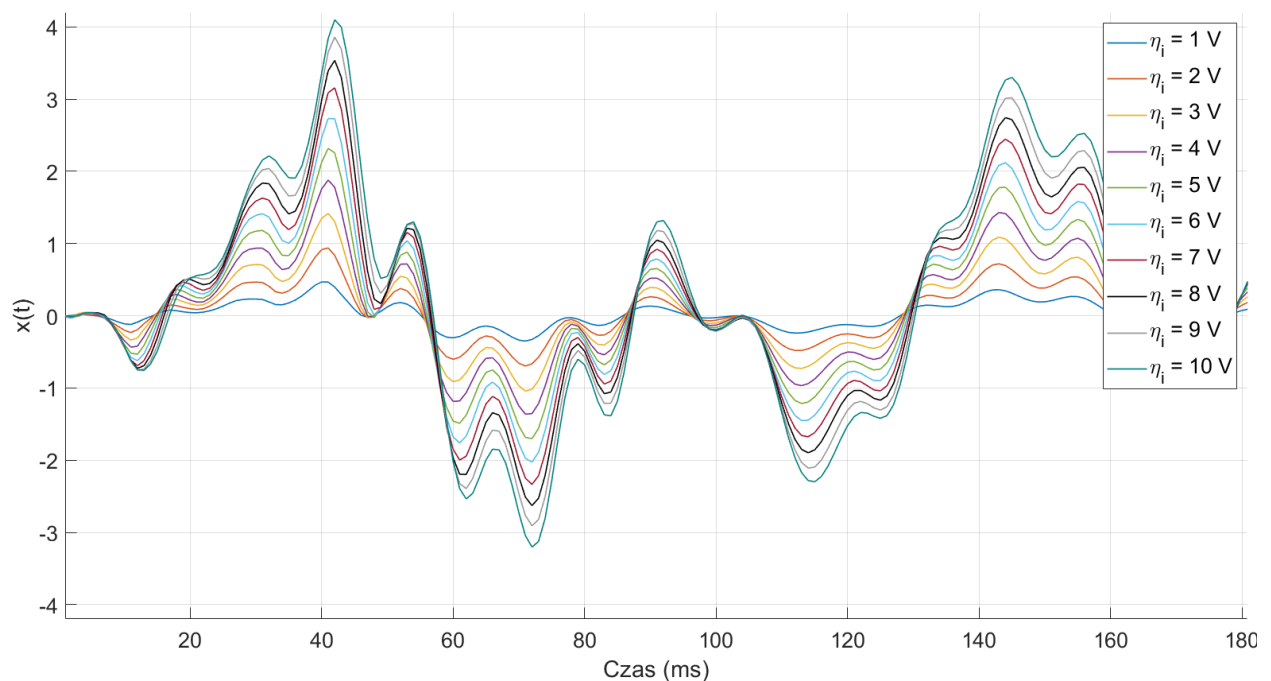
Każdą próbę realizowano przez czas $t_{\text{pomiar}} = 600$ s, a intensywność określono wartością $\lambda = 10$. Rejestracja odpowiedzi prowadzona była z częstotliwością próbkowania 100 kHz. Wybór częstotliwości próbkowania 100 kHz wynika z charakterystyki piezoceramicznego aktuatora MFC (M-2814-P2) [65]. Zgodnie z notą katalogową element ten posiada szerokie pasmo pracy, sięgające od kilku Hz do kHz dla niskich napięć nawet do 750 kHz przy wysokich napięciach elektrycznych. W doświadczeniach stosowano jednak

znacznie niższe napięcia wzbudzające (od 1 V do 10 V, tabela 8.1), co ogranicza energię wprowadzoną do układu.

Wybór próbkowania na poziomie 100 kHz zapewnia bezpieczny margines w stosunku do kryterium Nyquista. Wybrana częstotliwość umożliwia rejestrację także składowych sygnału generowanych przy wzbudzeniach impulsowych. Dzięki temu możliwe jest wierne odtworzenie dynamiki układu, uniknięcie aliasingu oraz uchwycenie subtelnych różnic w odpowiedziach, które są istotne w dalszej analizie i klasyfikacji.

Wymuszenie zastosowane w badaniach miało charakter zbliżony do tego, które opisano w rozdziale 4.3, z tą różnicą, że zakres napięcia wymuszającego został zwiększony do przedziału od 1 do 10 V. Wynika to z faktu, że przy zastosowaniu takich samych amplitud impulsów jak w przypadku układu RLC, rejestrowane odpowiedzi belki z aktuarem MFC były zbliżone poziomem do szumów pomiarowych oraz zakłóceń pochodzących z otoczenia, co uniemożliwiało ich jednoznaczną identyfikację. Zwiększenie amplitudy o rząd wielkości pozwoliło uzyskać odpowiedzi o wyraźniejszym stosunku sygnału do szumu.

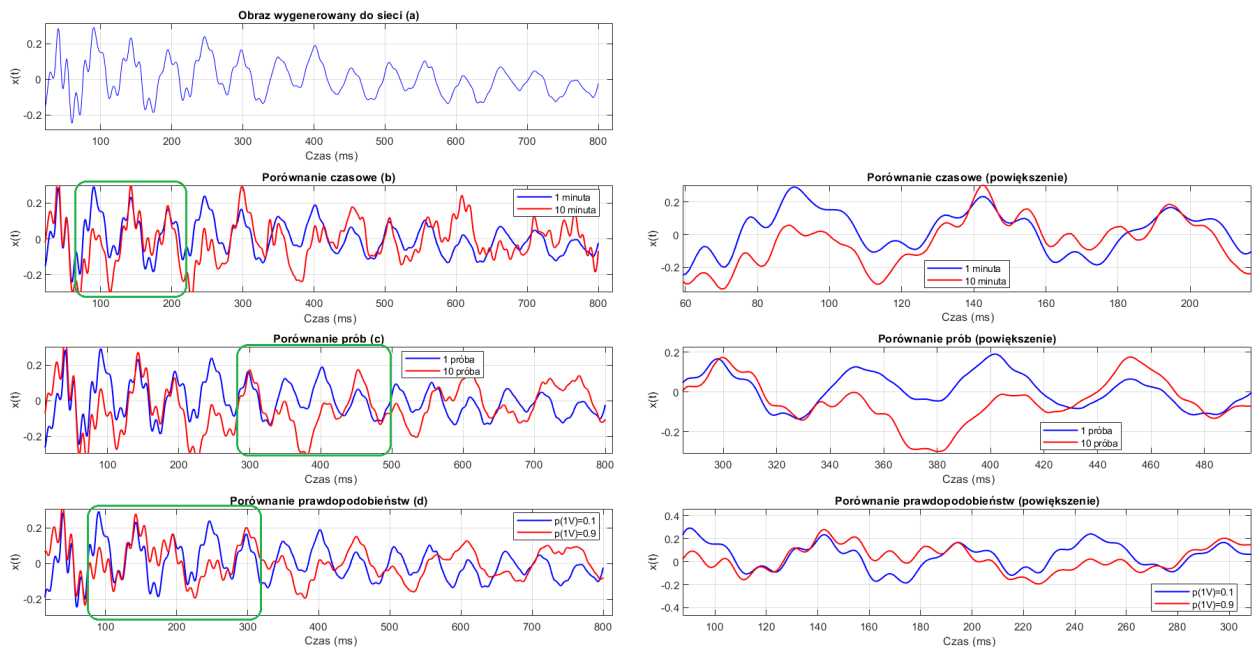
Na rysunku 8.5 przedstawiono średnie wartości odpowiedzi belki kompozytowej FR-4 z piezoelektrykiem MFC, dla dziesięciu poziomów wymuszeń impulsowych opisanych w tabeli 8.1. Każda krzywa odpowiada średniej wartości odpowiedzi uzyskanej z dziesięciu prób dla danego poziomu napięcia η_i , gdzie $\eta_1 = 1 \text{ V}$, $\eta_2 = 2 \text{ V}$, ..., $\eta_{10} = 10 \text{ V}$, $p(\eta_i)=0.1$.



Rysunek 8.5: Średnia wartość odpowiedzi belki kompozytowej na wymuszenia w zakresie $\eta_1 = 1 \text{ V}$, ..., $\eta_{10} = 10 \text{ V}$, w chwili $t \in \langle 0, 180 \rangle \text{ ms}$.

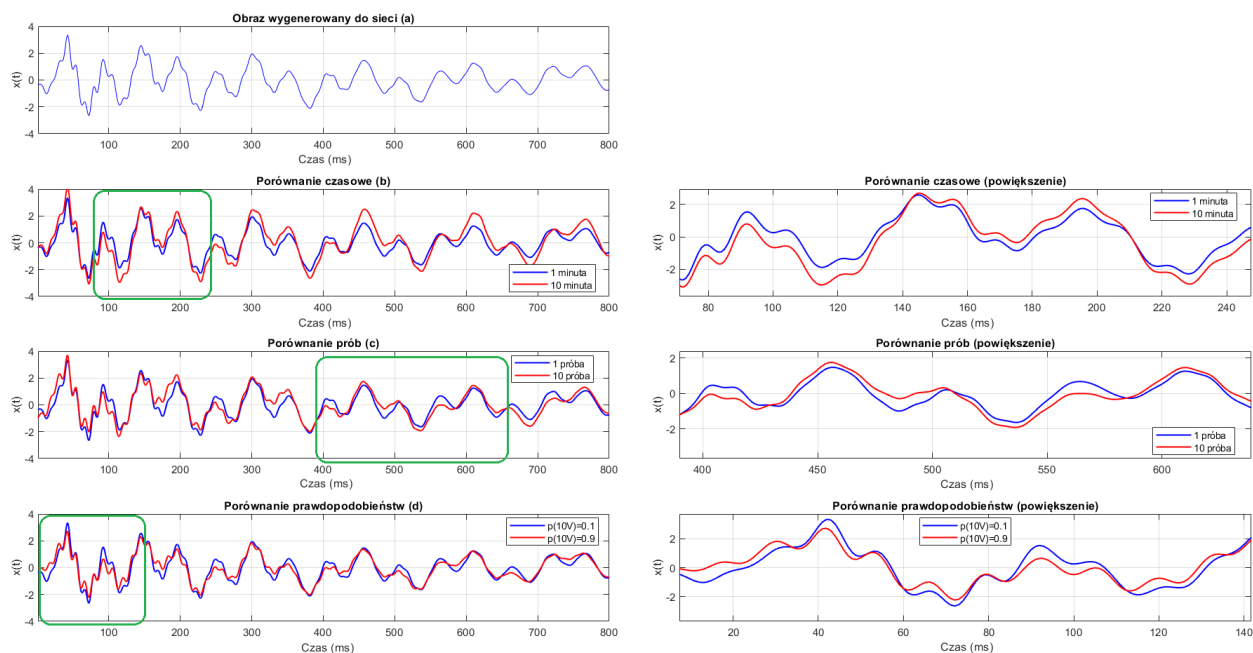
Analizując przebiegi można zauważyć, że wszystkie odpowiedzi charakteryzują się zbliżonym kształtem. Wzrost wartości wymuszenia prowadzi do proporcjonalnego zwiększania amplitudy drgań. Charakterystyczne piki odpowiedzi są zachowane w tej samej chwili czasowej, a kształt sygnału nie ulega deformacjom przy wzroście poziomu napięcia. Można również zaobserwować niewielkie różnice pomiędzy krzywymi wynikającą z naturalnej zmienności kolejnych prób oraz obecności szumów pomiarowych. Dzięki

uśrednieniu z dziesięciu prób zminimalizowano wpływ losowych fluktuacji, szumu i pokazano zależność pomiędzy poziomem impulsu jako wymuszenia układu a amplitudą odpowiedzi. Belka kompozytowa FR-4 zintegrowana z piezoelektrykiem MFC wykazuje odpowiedzi, w których amplituda skaluje się z poziomem wymuszenia. Potwierdza to przydatność tego układu jako obiektu do dalszych badań.



Rysunek 8.6: Porównanie pojedynczej odpowiedzi belki z piezoelektrykiem na wymuszenie 1 V oraz wpływ różnych czynników na kształt odpowiedzi.

Na rysunku 8.6 przedstawiono zestawienie odpowiedzi belki kompozytowej wzbudzanej serią impulsów o amplitudzie 1 V. Panel (a) prezentuje pojedynczą odpowiedź belki na wymuszenie. Widoczny jest oscylacyjny przebieg o malejącej amplitudzie, co świadczy o obecności tłumienia. Panel (b) pokazuje porównanie odpowiedzi zarejestrowanych w dwóch chwilach. Jest to pierwsza odpowiedź w pierwszej minucie (linia niebieska) oraz pierwsza odpowiedź po 10 minutach (linia czerwona). W powiększeniu (prawy panel) można dostrzec różnice w amplitudzie szczególnie w okolicach maksymalnych wartości sygnału. Oznacza to, że odpowiedź belki może się zmieniać w czasie rejestracji, nawet przy zachowaniu tej samej wartości wymuszenia i tej samej próbie. Panel (c) przedstawia odpowiedzi z różnych prób pomiarowych. Linia niebieska odpowiada pierwszej odpowiedzi w pierwszej próbie, natomiast linia czerwona pierwszej odpowiedzi w dziesiątej próbie. Przebiegi mają podobny kształt, lecz w powiększeniu widoczne są różnice w poziomie amplitudy oraz w niewielkim przesunięciu całego sygnału w osi pionowej. Panel (d) ilustruje wpływ zmiany prawdopodobieństwa pojawienia się impulsu na odpowiedź belki. Pokazano pierwszą odpowiedź dla $p(1V) = 0.1$ (linia niebieska) i $p(1V) = 0.9$ (linia czerwona). Oba przebiegi zachowują ten sam charakter oscylacyjny, lecz różnią się amplitudą i początkowym kształtem drgań. W analogiczny sposób przedstawiono porównanie dla wymuszenia 10 V (rysunek 8.7), gdzie obserwacje są podobne. Rysunki (8.6 i 8.7 przedstawiają pojedyncze odpowiedzi oraz wpływ różnych czynników na kształt odpowiedzi belki kompozytowej.



Rysunek 8.7: Porównanie pojedynczej odpowiedzi belki z piezoelektrykiem na wymuszenie 10 V oraz wpływ różnych czynników na kształt odpowiedzi.

Obserwacje poczynione na podstawie tych przebiegów są analogiczne również dla pozostałych przypadków, dlatego w treści rozdziału zaprezentowano jedynie te dwa przykłady. Z porównanie wyników z badań eksperymentalnych wynika, że istnieją subtelne różnice w odpowiedzi belki pomiędzy próbami. Odpowiedzi układu ciągle zależą również od rozkładu i od wielkości impulsu. Zmiana wartości $p(\eta_i)$ prowadzi do różnic w początkowej fazie odpowiedzi, szczególnie w zakresie amplitud. Efekt ten jest przedstawiony na rysunku 8.7.

Należy zauważyć, że klasyczne modele matematyczne belek (np. równania Eulera–Bernoulliego czy Timoshenki) opisują układy w sposób deterministyczny i liniowy [30, 43, 29]. W przedstawionych wynikach pojawiają się jednak wyraźne odchylenia od modelu: zmienność między próbami, zależność od prawdopodobieństwa impulsu, a także nieliniowy charakter tłumienia przy wyższych wymuszeniach. Oznacza to, że budowa jednoznacznego modelu matematycznego byłaby bardzo trudna i wymagałaby uwzględnienia złożonych efektów stochastycznych oraz nieliniowych. Z tego powodu w dalszej części pracy wykorzystano metody sztucznej inteligencji, które pozwalają na uchwycenie różnic w odpowiedziach i są lepiej przystosowane do analizy systemów złożonych, losowych i nieliniowych. Dzięki temu możliwe jest przeprowadzenie klasyfikacji odpowiedzi belki w oparciu o dane empiryczne, bez konieczności konstruowania analitycznego modelu matematycznego.

Rozwijając podejście oparte na mechanice stochastycznej, konieczne jest uwzględnienie dużej liczby zróżnicowanych przypadków wymuszeń. W tabeli 8.3 przedstawiono zestawienie losowych serii impulsów, które posłużyły jako dane wejściowe do eksperymentów. Analiza ogranicza się jednak do takich sekwencji, w których kolejne impulsy pojawiają się dopiero po całkowitym wygaśnięciu odpowiedzi układu na impuls poprzedni, co pozwala uniknąć efektu nakładania się sygnałów.

8.3.1 Definicja klas i prób pomiarowych

Tabela 8.2 przedstawia definicję klas wykorzystanych w procesie analizy i klasyfikacji odpowiedzi belki kompozytowej FR-4 z piezoelektrykiem MFC. Każdej wartości wymuszenia η_i odpowiada odrębna klasa. W eksperymencie zdefiniowano dziesięć klas odpowiadających napięciom od 1 V do 10 V, oznaczonym kolejno jako $U_1, U_2, \dots, U_{10}$. Dzięki takiej strukturze klasyfikacja może być prowadzona wprost w odniesieniu do poziomu wymuszenia impulsowego.

Tabela 8.2: Definicja klas w doświadczeniu z belką kompozytową z piezoceramiką MFC

Wartość impulsu	Oznaczenie klasy
$U_1 (\eta_1)$	Klasa 1
$U_2 (\eta_2)$	Klasa 2
$U_3 (\eta_3)$	Klasa 3
$U_4 (\eta_4)$	Klasa 4
$U_5 (\eta_5)$	Klasa 5
$U_6 (\eta_6)$	Klasa 6
$U_7 (\eta_7)$	Klasa 7
$U_8 (\eta_8)$	Klasa 8
$U_9 (\eta_9)$	Klasa 9
$U_{10} (\eta_{10})$	Klasa 10

Tabela 8.3 przedstawia zestawienie wszystkich przypadków losowych serii impulsów zastosowanych w badaniach belki kompozytowej. Każdy wiersz odpowiada odrębnej próbie eksperymentalnej, w której zdefiniowano zarówno zbiór możliwych amplitud impulsów η_i , jak i przypisane im prawdopodobieństwa wystąpienia. Liczba przeprowadzonych prób wskazuje na szeroki zakres analizowanego materiału badawczego. W rozprawie szczegółowo zaprezentowano jedynie wybrane przypadki. Należy podkreślić, że obserwacje uzyskane dla przedstawionych w tej pracy przykładów są zgodne z wynikami wszystkich pozostałych prób, co pozwala traktować przedstawione rezultaty jako charakterystyczne dla całego zbioru badań.

Tabela 8.3: Przypadki losowych serii impulsów dla doświadczenia z belką kompozytową

Nr próby	Amplitudy (η_i) (V)	Prawdopodobieństwo $p(\eta_i)$	Liczba prób
1	1,2,3,4,5,6,7,8,9,10	0.1	10
2	1,3,5,7,9	0.2	3
3	2,4,6,8,10	0.2	3
4	1,2,3,4	0.25	3
5	7,8,9,10	0.25	3
6	2,4,7,10	0.25	3
7	2,4,6,8	0.25	3
8	1,3,7,9	0.25	3
9	1,10	0.5	3

ciąg dalszy na następnej stronie

Nr próby	Amplitudy (η_i) (V)	Prawdopodobieństwo $p(\eta_i)$	Liczba prób
10	3,8	0.5	3
11	2,6,10	0.3333	3
12	1,5,6	0.3333	3
13	3,4,5,6	0.25	3
14	1,5,9,10	0.25	3
15	2,3,8,9	0.25	3
16	5,6,7,8	0.25	3
17	3,6,9	0.3333	3
18	2,5,8,11	0.25	3
19	1,2,3,4,5	0.2	3
20	1–8	0.125	3
21	3–10	0.125	3
22	1,2,4,5,6,7,9,10	0.125	3
23	3,7,10	0.7, 0.2, 0.1	3
24	3,7,10	0.1, 0.6, 0.3	3
25	3,7,10	0.9, 0.05, 0.05	3
26	3,7,10	0.05, 0.05, 0.9	3
27	1,2,5	0.15, 0.25, 0.6	3
28	1,2,10	0.15, 0.25, 0.6	3
29	1,2,10	0.9, 0.05, 0.05	3
30	6,9,10	0.9, 0.05, 0.05	3
31	1,4,8,10	0.5, 0.1, 0.25, 0.15	3
32	2,8,10	0.8, 0.15, 0.05	3

Czas trwania pojedynczego eksperymentu ustalono na 600 s. Dane pomiarowe przechowywano w postaci plików tekstowych, a następnie przetwarzano algorytmem wyodrębniającym pojedyncze odpowiedzi (rozdział 4.4), przygotowując je do dalszej analizy.

8.3.2 Definicja bazy danych dla sieci

Do treningu klasyfikatora wybrano pięć zestawów prób, które obejmują różne kombinacje amplitud i prawdopodobieństw. Ich parametry zestawiono w tabeli 8.4.

Tabela 8.4: Przypadki wybrane do treningu i walidacji klasyfikatora dla belki kompozytowej

Nr próby	Amplitudy (η_i) (V)	Prawdopodobieństwo $p(\eta_i)$	Liczba prób
2	1, 3, 5, 7, 9	0.2	3
3	2, 4, 6, 8, 10	0.2	3
19	1, 2, 3, 4, 5	0.2	3
4	1, 2, 3, 4	0.25	3
5	7, 8, 9, 10	0.25	3

Do treningu i walidacji klasyfikatora wybrano pięć zestawów prób: nr 2, 3, 19, 4 oraz 5. Ich wspólną cechą jest równomierny rozkład prawdopodobieństw impulsów ($p(\eta_i) = 0.2$ lub $p(\eta_i) = 0.25$), co oznacza, że każdy z impulsów w danym zbiorze występuje z jednakowym prawdopodobieństwem. Dzięki temu uzyskane dane treningowe są dobrze zrównoważone i pozwalają uniknąć sytuacji, w której klasyfikator uczy się przewagi jednej amplitudy kosztem innych. Do weryfikacji działania wytrenowanego klasyfikatora wykorzystano trzy zestawy prób przedstawione w tabeli 8.5.

Tabela 8.5: Przypadki testowe wykorzystane do testów działania klasyfikatora

Nr próby	Amplitudy (η_i) (V)	Prawdopodobieństwo $p(\eta_i)$	Liczba prób
1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	0.1	3
6	2, 4, 7, 10	0.25	3
13	3, 4, 5, 6	0.25	3

8.3.3 Generowanie obrazów dla sieci z danych eksperymentalnych

Baza danych została opracowana na podstawie zarejestrowanych odpowiedzi opisanych w podrozdziale 8.3.2. Dane uporządkowano w katalogach według wartości η_i , a następnie scalono odpowiedzi zapisane w wielu plikach tekstowych w pojedyncze pliki odpowiadające klasom. W rezultacie utworzono pliki zbiorcze `Klasa_1.txt`, ..., `Klasa_10.txt`. Proces wyboru odpowiedzi przebiega analogicznie jak w przypadku układu RLC lecz jest pominięty fragment dotyczący generowania klas mieszanych. W badaniach belki kompozytowej rozpatrywane są tylko przypadki pojedynczych odpowiedzi na wymuszenie. Parametr `lines_to_move` określa maksymalną liczbę losowych wierszy pobieranych z każdego pliku i wpisywanych do plików zbiorczych. Dzięki temu każdy plik wynikowy zawiera zestaw danych o kontrolowanej liczbie próbek pochodzących z różnych prób. Podobnie jak w przypadku danych z układu RLC, przed dalszym przetwarzaniem zastosowano filtrację metodą Savitzky'ego–Golaya o parametrach: długość ramki filtru 11 próbek, rząd wielomianu 2 [57, 71, 58].

Na podstawie plików zbiorczych wygenerowano obrazy w przestrzeni RGB. Wizualizacje utworzono w środowisku MATLAB za pomocą funkcji `plot`, usuwając osie i oznaczenia w celu uproszczenia struktury obrazu. Skala osi Y została ograniczona do przedziału $[-4, 4]$, co ujednoliciło porównania sygnałów i zniwelowało różnice amplitudowe. Długość sygnału widoczna na obrazie została ograniczona do 800 próbek. Na rysunku 8.8 przedstawiono przykładowe obrazy wykorzystywane w klasyfikacji odpowiedzi belki kompozytowej na losową serię impulsów.



(a) Klasa 1.



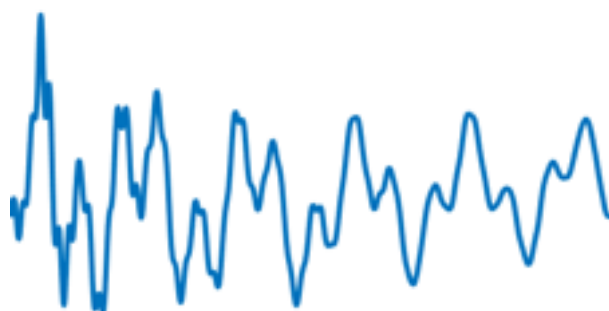
(b) Klasa 2.



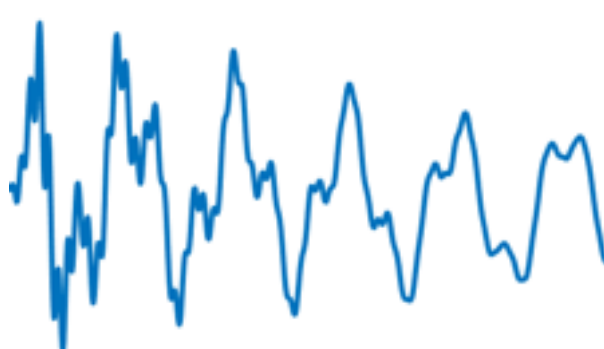
(c) Klasa 3.



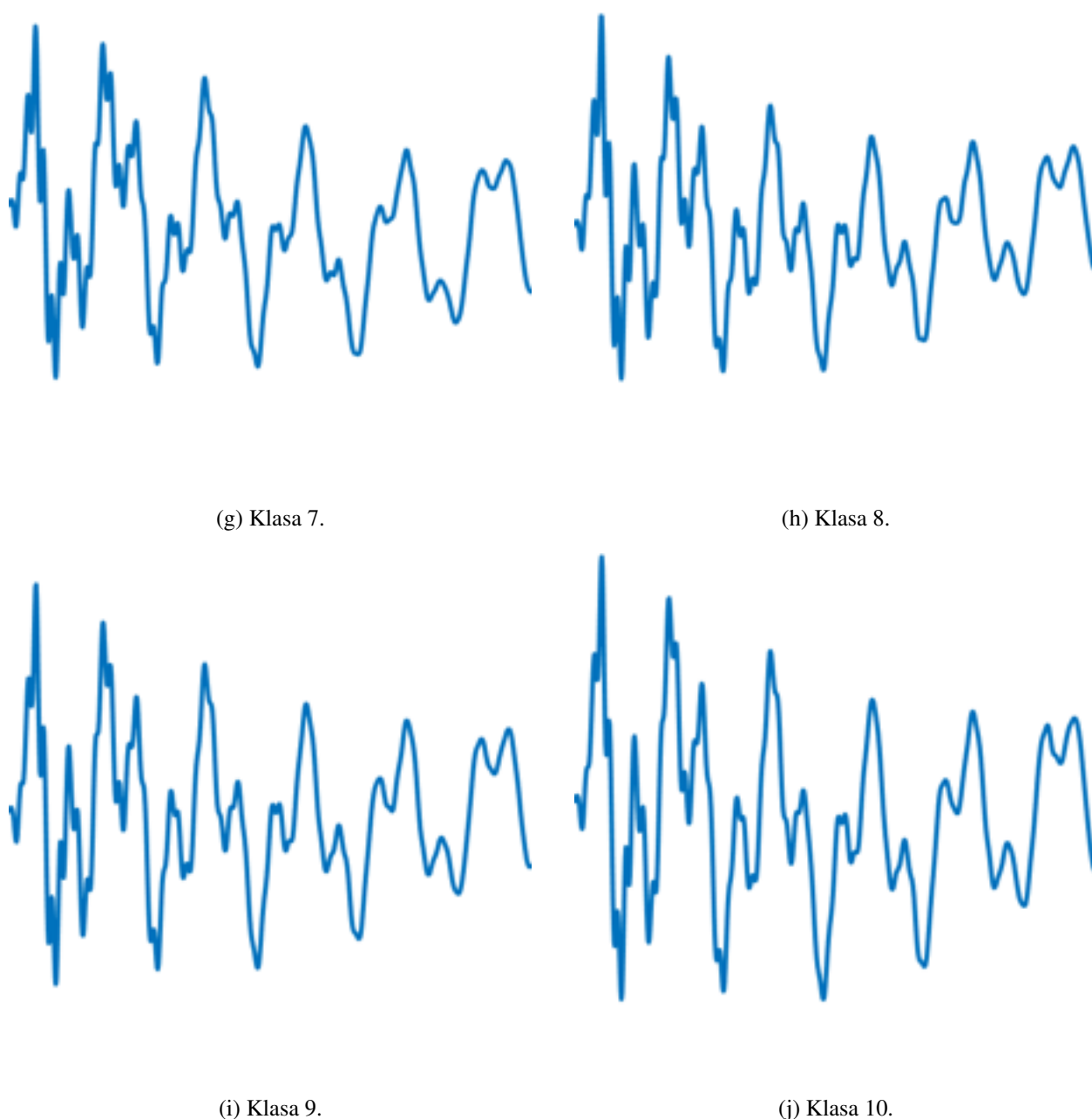
(d) Klasa 4.



(e) Klasa 5.



(f) Klasa 6.



Rysunek 8.8: Przykłady wygenerowanych obrazów (cd.).

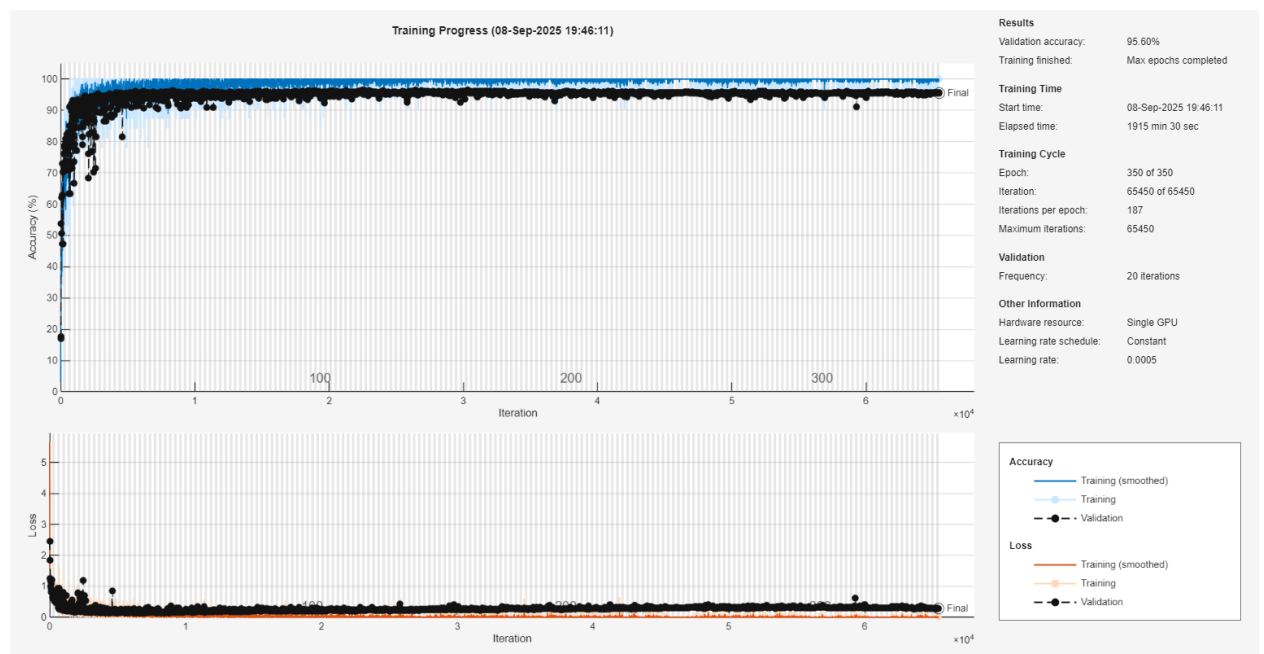
Tabela 8.6 przedstawia zestawienie liczby obrazów przypisanych do poszczególnych klas w zbiorach użytych podczas badań z belką kompozytową. W eksperymentach wykorzystano dwa warianty danych: zbiór bazowy oraz zbiór powiększony dwukrotnie, obejmujący większą liczbę próbek treningowych i walidacyjnych. Zbiór testowy pozostał niezmieniony we wszystkich analizach, co pozwoliło na bezpośrednie porównanie jakości klasyfikacji pomiędzy różnymi architekturami sieci (VGG16, VGG19) oraz pomiędzy wersjami zbioru uczącego. Dzięki takiemu podejściu możliwa była ocena wpływu liczby próbek treningowych na zdolność generalizacji klasyfikatora. Podobnie jak w przypadku układu RLC, zbiór treningowy i walidacyjny został podzielony w proporcji 80/20.

Tabela 8.6: Liczba obrazów wykorzystanych w procesie trenowania, walidacji i testowania klasyfikatorów dla poszczególnych klas

Klasa	Zbiór treningowy + walidacyjny (zbiór bazowy)	Zbiór treningowy + walidacyjny (zbiór powiększony 2x)	Zbiór testowy
Klasa 1	750	1500	1500
Klasa 2	750	1500	1500
Klasa 3	750	1500	2700
Klasa 4	750	1500	2700
Klasa 5	750	1500	2700
Klasa 6	750	1500	1800
Klasa 7	750	1500	1800
Klasa 8	750	1500	2700
Klasa 9	750	1500	1800
Klasa 10	750	1500	1800

8.4 Sieć VGG16 – proces uczenia oraz klasyfikacji danych spoza zbioru walidacyjnego

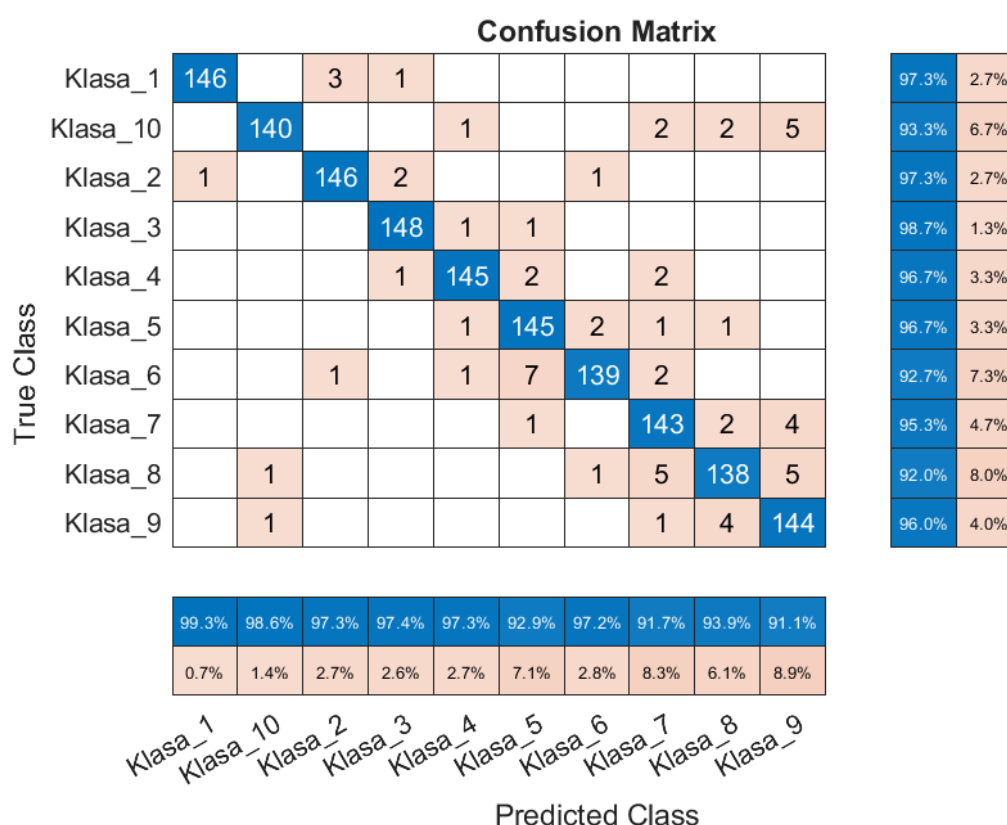
Proces uczenia sieci, testów oraz i konfiguracji jest analogiczny do danych dla układu RLC (podrozdział 7.5), lecz wymagał trenowania klasyfikatorów na nowych zbiorach danych. Na rysunku 8.9 widoczny jest charakterystyczny przebieg metryk dokładności i funkcji straty w procesie uczenia sieci VGG16. Krzywe odnoszące się do zbioru treningowego oraz walidacyjnego są w dużej mierze równoległe, co świadczy o stabilnym procesie optymalizacji modelu i podobnym tempie uczenia się.



Rysunek 8.9: Reprezentacja procesu trenowania sieci VGG16 dla doświadczenia z belką kompozytową.

Zauważalny jest jednak stały odstęp pomiędzy wartościami osiąganymi na zbiorze treningowym i walidacyjnym. Dokładność treningowa zbliża się do wartości bliskich 100%, natomiast dokładność walidacyjna stabilizuje się na poziomie 95.6%. Tego rodzaju różnica, określana jako *generalization gap*, wskazuje, że sieć dopasowała się bardzo dobrze do danych treningowych, jednak jej zdolność uogólniania na nowe, niewidziane wcześniej próbki pozostaje nieco ograniczona. Warto podkreślić, że utrzymująca się, stosunkowo niewielka różnica pomiędzy wynikami dla obu zbiorów jest zjawiskiem naturalnym i oczekiwanym w procesie uczenia głębokich sieci neuronowych gdy klasy są podobne. Tylko w sytuacjach, gdy odległość pomiędzy tymi krzywymi narasta wraz z liczbą epok, można mówić o silnym przeuczeniu wymagającym interwencji.

Rysunek 8.10 przedstawia macierz pomyłek uzyskaną w procesie walidacji podczas treningu z wykorzystaniem architektury VGG16. Liczba sklasyfikowanych przypadków przedstawiona na przekątnej wskazuje na wysoką skuteczność klasyfikatora — większość próbek została poprawnie przypisana do odpowiednich klas. Liczba błędów jest stosunkowo niewielka i dotyczy głównie klas o większym stopniu podobieństwa cech (Klasa_5, Klasa_6, Klasa_7, Klasa_8 oraz Klasa_9). Pomyłki polegały głównie na zamienianym przypisywaniu próbek do klas sąsiednich, co potwierdza trudność rozróżnienia niektórych wzorców odpowiedzi belki.

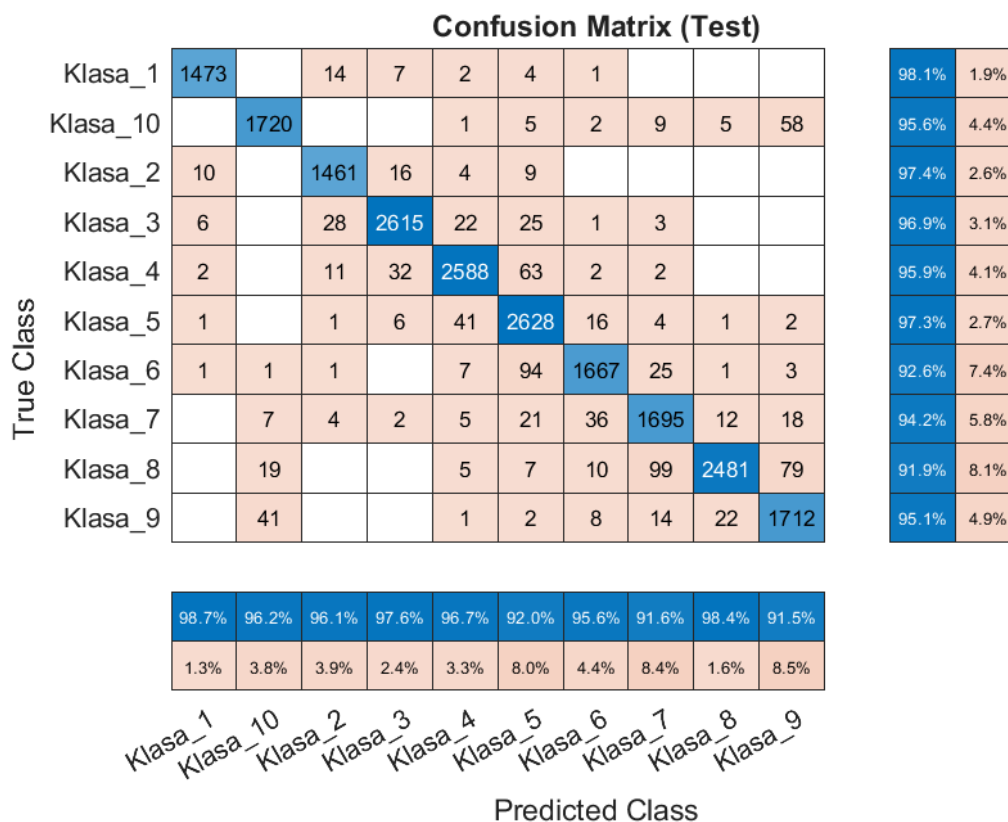


Rysunek 8.10: Macierz pomyłek uzyskana w procesie walidacji podczas treningu sieci VGG16 dla doświadczenia z belką kompozytową.

Z prawego panelu (czułość) i dolnego panelu (precyzja) można odczytać, że większość klas osiągnęła wyniki powyżej 95%. Najwyższą skuteczność odnotowano dla Klasa_1 i Klasa_3, gdzie zarówno pre-

cyzja, jak i czułość przekroczyły 98%. Najniższe wartości dotyczą Klasa_8 (precyzja 92%, czułość 92%) oraz Klasa_9 (precyzja 91%, czułość 96%). Oznacza to, że w przypadku tych klas klasyfikator częściej popełniał błędy, co potwierdza także obserwowana liczba błędów w macierzy pomyłek. Wynik zbiorczy pokazuje średnią wartość $F1 = 0.956$, co oznacza, że sieć VGG16 dobrze nauczyła się odwzorowywać strukturę danych w procesie treningu. Macierz pomyłek wskazuje przy tym, że największe trudności sprawiają klasy o podobnych cechach sygnału, jednak poziom błędów pozostaje niski i nie wpływa znacząco na jakość ogólną klasyfikacji.

W kolejnym kroku wykorzystano wytrenowaną sieć na danych spoza zbioru uczącego. Rysunek 8.11 przedstawia macierz pomyłek uzyskaną w procesie testowania klasyfikatora VGG16. Analizując przekątną macierzy można zauważyć, że sieć poprawnie rozpoznaje zdecydowaną większość przypadków, co potwierdzają wysokie wartości dokładności dla wszystkich klas. Najlepsze wyniki uzyskano m.in. dla klas Klasa_1, Klasa_3, Klasa_4 oraz Klasa_5, gdzie odsetek poprawnych klasyfikacji przekracza 96%. Nieco niższe wartości odnotowano w przypadku klas Klasa_6 oraz Klasa_8, których dokładności wynoszą odpowiednio 92.6% i 91.9%.



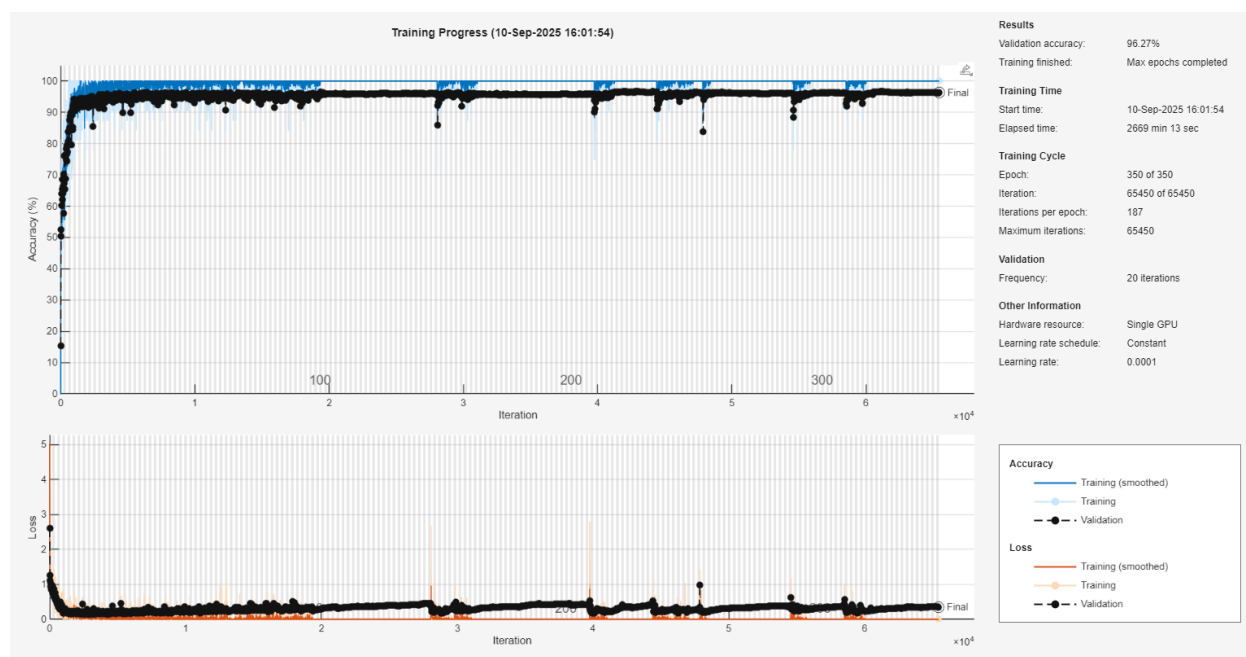
Rysunek 8.11: Zestawienie wyników klasyfikacji w procesie testowania sieci VGG16 dla doświadczenia z belką kompozytową.

Błędne klasyfikacje koncentrują się głównie pomiędzy klasami o zbliżonej charakterystyce. Przykładowo, próbki Klasa_6 są czasami przypisywane do Klasa_7 i Klasa_8, natomiast część sygnałów z Klasa_8 błędnie klasyfikowana jest jako Klasa_9. Pomimo tych rozbieżności udział błędnych klasyfikacji w całym zbiorze testowym pozostaje niewielki. Warto podkreślić, że ogólna skuteczność całego

modelu, liczona jako średnia miara F1, wyniosła 0.950. Oznacza to, że sieć VGG16 dobrze radzi sobie z rozróżnianiem badanych klas i zapewnia wysoką jakość klasyfikacji w przypadku większości kategorii. Podsumowując, proces testowania pokazał, że VGG16 jest stabilnym i skutecznym klasyfikatorem w warunkach eksperymentalnych, a główne błędy dotyczą klas o zbliżonej charakterystyce sygnałów. Rezultaty te wskazują na konieczność zwiększenia liczby przykładów lub lepszego zrównoważenia zbioru danych w celu poprawy równomierności jakości klasyfikacji.

8.5 Sieć VGG19 – proces uczenia oraz klasyfikacji danych spoza zbioru walidacyjnego

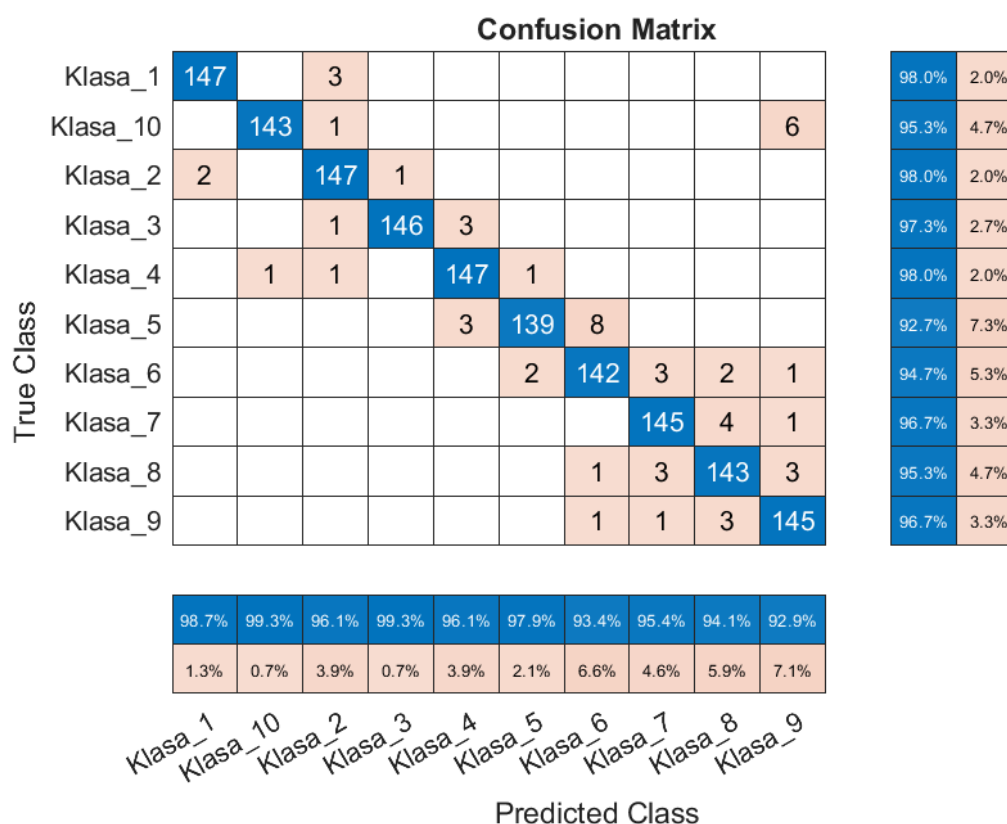
Analogicznie jak w sieci VGG16, rysunek 8.12 przedstawia wyniki treningu dla sieci VGG19. W tym przypadku dokładność walidacyjna osiągnęła poziom 96.3%, co sugeruje, że głębsza architektura cechuje się większą zdolnością reprezentacyjną w porównaniu do sieci VGG16.



Rysunek 8.12: Reprezentacja procesu trenowania sieci dla architektury VGG19 dla doświadczenia z belką kompozytową.

Analizując wykres błędu (*loss*) można zauważyć, że krzywa treningowa szybko osiągnęła niski poziom i ustabilizowała się. Natomiast przebieg błędu walidacyjnego jest bardziej rozproszony i w wybranych momentach wykazuje skoki, co sugeruje okresowe przeuczenie modelu względem danych treningowych. Zjawisko to wynika z większej liczby warstw konwolucyjnych i gęstych połączeń charakterystycznych dla VGG19, co zwiększa jej zdolność dopasowania do zbioru treningowego, ale równocześnie ogranicza zdolność generalizacji. Pomimo wysokiej dokładności walidacyjnej, zaobserwowane oznaki przeuczenia mogą sugerować, że sieć VGG19 nie będzie równie skuteczna w testach na danych nowych i nieznanach. Rysunek 8.13 przedstawia macierz pomyłek uzyskaną w procesie trenowania klasyfikatora opartego na architek-

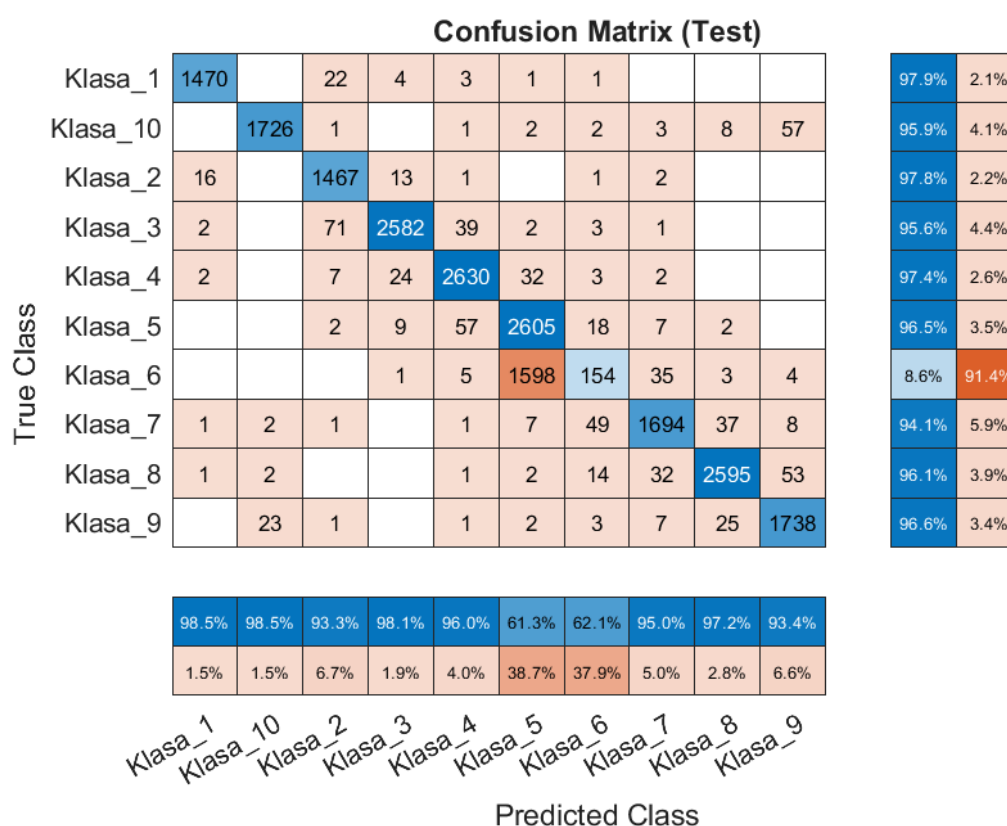
turze VGG19. Sieć skutecznie rozpoznaje większość klas, choć w porównaniu do VGG16 zauważalne są nieco większe różnice między klasami pod względem liczby błędów.



Rysunek 8.13: Zestawienie wyników klasyfikacji w procesie trenowania sieci VGG19 dla doświadczenia z belką kompozytową.

Najwyższą skuteczność odnotowano dla klas Klasa_3 i Klasa_10, gdzie precyzja osiągnęła wartość ≈ 0.993 , a czułość odpowiednio 0.973 oraz 0.953. Wysokie wskaźniki uzyskała również Klasa_1 ($F1 = 0.983$). Z kolei nieco niższe rezultaty odnotowano dla klas Klasa_5, Klasa_6 i Klasa_9, gdzie wartości czułości spadła poniżej 0.95, a F1-score wyniósł odpowiednio 0.952, 0.940 oraz 0.948. Wskazuje to, że sieć miała trudności z pełnym rozpoznaniem wszystkich próbek należących do tych kategorii, co mogło wynikać ze zwiększonego podobieństwa ich cech do innych klas. Podobnie jak dla sieci VGG16, wykorzystując sieć VGG19 przeprowadzano weryfikację sieci na tym samym zbiorze danych testowych. Rysunek 8.14 przedstawia macierz pomyłek uzyskaną w procesie testowania klasyfikatora VGG19. Analizując przekątną macierzy można zauważyć, że sieć dobrze radzi sobie z rozpoznawaniem większości klas, w szczególności Klasa_1, Klasa_2, Klasa_3, Klasa_4, Klasa_5, Klasa_7, Klasa_8, Klasa_9 oraz Klasa_10, gdzie dokładności klasyfikacji przekraczają 94%. Wysokie wartości precyzji i czułości dla tych kategorii potwierdzają stabilność modelu i jego zdolność do uogólniania wzorców dla dobrze reprezentowanych klas. Największe problemy wystąpiły w przypadku klasy Klasa_6. Sieć poprawnie sklasyfikowała jedynie 154 próbek, podczas gdy aż 1598 przypadków zostało błędnie przypisanych do innych klas, głównie Klasa_5 oraz Klasa_7. Wartość czułości dla tej kategorii spadła do 0.086, a F1-score wyniósł zaledwie 0.151. Sugeruje to istotne trudności w rozróżnianiu cech charakterystycznych

tej klasy, prawdopodobnie związane z podobieństwem sygnałów do sąsiednich kategorii oraz większym zróżnicowaniem danych testowych.



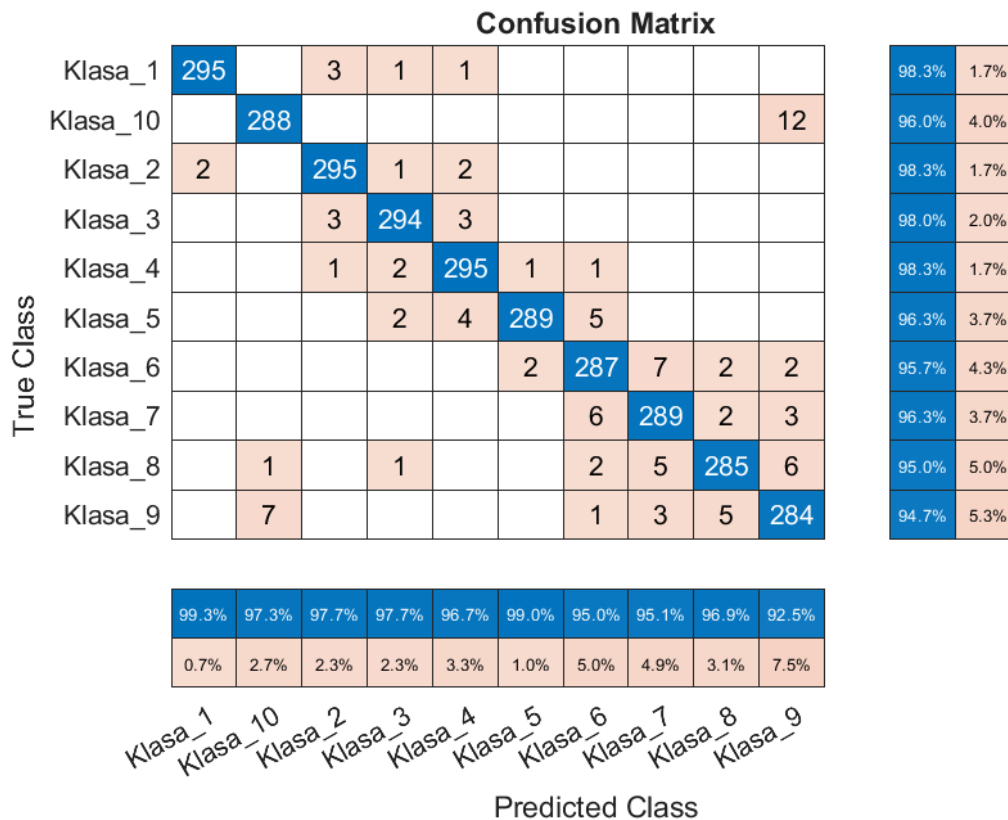
Rysunek 8.14: Zestawienie wyników klasyfikacji w procesie testu sieci VGG19 dla doświadczenia z belką kompozytową.

Ogólna dokładność klasyfikatora na całym zbiorze testowym wyniosła 88.86%, co świadczy o dobrej jakości działania sieci VGG19, choć jest to wynik niższy niż w przypadku architektury VGG16. Należy jednak podkreślić, że globalna wartość nie oddaje problemów związanych z poszczególnymi kategoriami – szczególnie z klasą *Klasa_6*, której niska skuteczność znacząco obniża równomierność jakości klasyfikacji w ramach wszystkich rozważanych przypadków.

8.6 Sieć VGG19 – proces uczenia na większym zbiorze treningowym i walidacyjnym oraz porównanie klasyfikacji na danych spoza zbioru walidacyjnego

Otrzymane wyniki dla obu sieci zostały uzyskane wskutek testowania i walidacji klasyfikatorów na tych samych zbiorach treningowych i walidacyjnych, a następnie sprawdzane na danych pochodzących spoza zbioru walidacyjnego. Ze względu na niską dokładność sieci VGG19 (88.86%) podjęto decyzję o wytrenowaniu nowego klasyfikatora, zakładając dwukrotne zwiększenie zbioru treningowego oraz walidacyjnego. Należy podkreślić, że wcześniej wykorzystany zbiór (bazowy) został rozszerzony o nowe przypadki, na-

tomiast dane pochodzące ze zbioru testowego były tymi samymi, których użyto do testów VGG16 oraz VGG19. Rysunek 8.15 przedstawia macierz pomyłek uzyskaną w procesie trenowania sieci VGG19 dla doświadczenia z belką kompozytową, przy czym w tym przypadku zastosowano dwukrotnie zwiększony zbiór treningowy i walidacyjny. Analiza macierzy wskazuje, że sieć poprawnie rozpoznaje zdecydowaną większość przypadków, o czym świadczą wysokie wartości na przekątnej. Najlepsze wyniki uzyskano dla klas Klasa_1, Klasa_2, Klasa_3, Klasa_4 oraz Klasa_5, gdzie dokładności klasyfikacji osiągały wartości zbliżone do 99%.

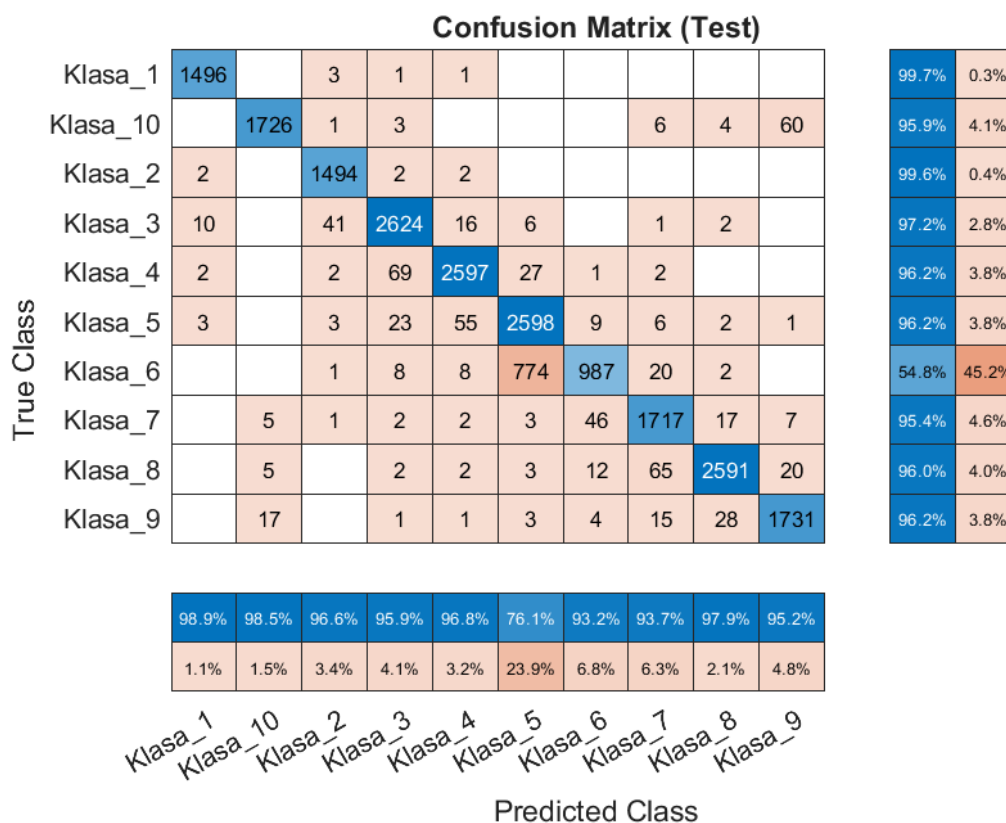


Rysunek 8.15: Zestawienie wyników klasyfikacji w procesie trenowania sieci VGG19 dla doświadczenia z belką kompozytową na dwukrotnie zwiększonym zbiorze treningowym i walidacyjnym.

Błędne klasyfikacje mają charakter incydentalny i koncentrują się przede wszystkim pomiędzy klasami sąsiadującymi, co sugeruje, że rozbieżności wynikają z podobieństwa cech sygnałów. Najczęściej mylone są przypadki należące do klas Klasa_6, Klasa_7, Klasa_8 oraz Klasa_9, gdzie udział błędnych klasyfikacji wynosi od 3 do 7%. Mimo to, dla każdej klasy wartość czułości przekracza 94%.

Podsumowując, powiększenie zbioru danych przełożyło się na poprawę procesu uczenia. Sieć VGG19 uzyskała wysoką ogólną dokładność klasyfikacji na poziomie 96.70%.

Rysunek 8.16 przedstawia macierz pomyłek uzyskaną w procesie testowania klasyfikatora VGG19 po dwukrotnym zwiększeniu zbioru treningowego i walidacyjnego. Ogólna dokładność klasyfikacji wzrosła w tym przypadku do około 93.14%, co oznacza zauważalną poprawę względem wcześniejszych wyników. Zwiększenie liczby próbek umożliwiło sieci lepsze uogólnienie cech charakterystycznych badanych sygnałów i zmniejszyło wpływ błędnych przypisań w większości kategorii.



Rysunek 8.16: Zestawienie wyników klasyfikacji w procesie testowania sieci VGG19 (trenowanym na zwiększonym zbiorze treningowym i walidacyjnym) dla doświadczenia z belką kompozytową.

Najwyższe wartości dokładności osiągnięto dla klas Klasa_1, Klasa_2, Klasa_3, Klasa_4, Klasa_5, Klasa_7, Klasa_8 oraz Klasa_9, gdzie skuteczność klasyfikacji przekracza 95%. Szczególnie dobre wyniki odnotowano w przypadku Klasa_1 i Klasa_2, które uzyskały odpowiednio 99.7% i 99.6% poprawnych klasyfikacji. Największe problemy wciąż występują dla klasy Klasa_6, która charakteryzuje się niższą dokładnością (ok. 54.8%) i wysokim odsetkiem błędnych klasyfikacji (głównie do klas Klasa_5 oraz Klasa_7). W porównaniu z wcześniejszymi wynikami nastąpiła jednak poprawa – czułość dla tej klasy wzrosła, a F1-score osiągnął wartości powyżej 0.6. Oznacza to, że zwiększenie zbioru treningowego pozwoliło sieci lepiej uchwycić charakterystykę tej kategorii, choć nadal pozostaje ona najbardziej problematyczna.

Tabela 8.7: Porównanie dokładności klasyfikacji dla sieci VGG16 i VGG19 w procesie treningu i testów dla doświadczenia z belką kompozytową

Sieć	Dokładność treningu [%]	Dokładność testów [%]
VGG16	93.51	95.42
VGG19	96.27	88.86
VGG19 – dwukrotnie zwiększony zbiór	96.70	93.14

Rozszerzenie zbioru treningowego i walidacyjnego przełożyło się na wyższą ogólną skuteczność sieci VGG19 oraz poprawę jakości klasyfikacji dla trudniejszych klas, w tym szczególnie dla Klasa_6. Wyniki

te potwierdzają, że zwiększenie liczby próbek poprawia zdolności generalizacji modelu dla niewidzianych przypadków.

Tabela 8.7 przedstawia wyniki dla sieci VGG16 i VGG19. Obie architektury osiągnęły wysoką skuteczność w procesie treningu, przy czym VGG19 uzyskała wyższe wartości dokładności w porównaniu do VGG16 (odpowiednio 96.27% oraz 93.51%). Jednak w testach to VGG16 osiągnęła lepszy wynik (95.42% vs 88.86%), co wskazuje na większą zdolność generalizacji tego modelu. Niższe wyniki testowe VGG19 wynikały przede wszystkim z trudności w klasyfikacji klasy `Klasa_6` oraz większej wariancji w rozpoznawaniu mniej reprezentatywnych kategorii. Po dwukrotnym zwiększeniu zbioru treningowego i walidacyjnego dokładność VGG19 w treningu wzrosła do 96.70%, a dokładność testowa do 93.14%. Oznacza to, że powiększenie zbioru danych pozwoliło na lepsze uogólnienie cech i ograniczyło problemy związane z klasyfikacją trudnych klas, przy czym `Klasa_6` w dalszym ciągu pozostaje najbardziej problematyczna. Uzyskane rezultaty wskazują, że zwiększenie rozmiaru zbioru uczącego i walidacyjnego poprawia jakość działania modelu VGG19. Jednocześnie sugerują, że dodatkowe mechanizmy regularyzacji (np. wagi klas, augmentacja ukierunkowana na klasy trudne, silniejszy dropout) mogą być konieczne, aby całkowicie zniwelować różnice względem VGG16, uniknąć przeuczenia i poprawić stabilność klasyfikacji we wszystkich kategoriach.

Rozdział 9

Podsumowanie i wnioski końcowe

Celem rozprawy było opracowanie metody klasyfikacji odpowiedzi układów dynamicznych – dyskretnego (RLC) i ciągłego (belki kompozytowej) – na losową serię impulsów o znanej wielkości η_i . Metoda ta opiera się na przekształceniu zarejestrowanych odpowiedzi do postaci obrazów oraz ich analizie z wykorzystaniem konwolucyjnych sieci neuronowych (VGG16 i VGG19). W rozdziale zestawiono najważniejsze wnioski płynące z przeprowadzonych analiz, podsumowano stopień realizacji założonych celów oraz wskazano potencjalne kierunki dalszych badań.

Pierwszym zrealizowanym celem było przeprowadzenie eksperymentu dla układu dyskretnego (układ RLC). Analogie elektromechaniczne pozwoliły na stworzenie reprezentatywnego układu dynamicznego.

Drugim zrealizowanym celem było zbadanie wpływu wielkości η_i i prawdopodobieństwa wystąpienia $p(\eta_i)$ na kształt odpowiedzi. Odnotowano zmienność wartości tłumienia b , częstotliwości drgań tłumionych ω , amplitudy a , a także fazy początkowej ϕ oraz składowej stałej C . Dodatkowo, zarejestrowany szum pomiarowy wprowadzał zmienność początkowych wartości, co wpływało na amplitudy odpowiedzi. W konsekwencji, odpowiedzi układu nie były identyczne. Ponadto estymacja parametrów modelu matematycznego na podstawie danych eksperymentalnych wskazała, że model deterministyczny może poprawnie odwzorowywać odpowiedzi dla impulsów o niskiej amplitudzie (100 mV, 200 mV) oraz w przypadku pojedynczych wymuszeń. Model ten nie uwzględnia jednak efektów nieliniowych ani nakładających się impulsów, dlatego jego zasadność stosowania ogranicza się do prostszych przypadków. Eksperymenty ograniczono do dwóch wartości impulsów, co pozwoliło uwzględnić w badaniach zdarzenia, w których kolejny impuls pojawia się zanim odpowiedź na poprzedni wygaśnie. Wykrywanie tego rodzaju zdarzeń umożliwia ocenę, czy przebieg procesu ulega zmianie.

W dalszej kolejności zrealizowano przetworzenie odpowiedzi układu do formatu umożliwiającego klasyfikację. Opracowano procedurę segmentacji sygnału oraz jego transformacji do postaci obrazów, które mogły zostać poddane analizie przez konwolucyjne sieci neuronowe. Zabieg ten umożliwił wykorzystanie potencjału metod głębokiego uczenia w dziedzinie, w której inne metody analizy sygnałów mogłyby okazać się niewystarczające. Zastosowanie sieci VGG16 i VGG19 w środowisku MatLAB pozwoliły na przeprowadzenie klasyfikacji odpowiedzi układu na losową serię impulsów.

Realizacja celu dotyczącego zastosowania metod uczenia maszynowego wykazała, że sieci konwolucyjne sprawdzają się w klasyfikacji rozkładów impulsów. W przypadku danych z układu RLC obie archi-

tektury uzyskały bardzo wysoką skuteczność (dokładność testowa $> 99.9\%$). Różnice pomiędzy architekturami VGG16 i VGG19 były niewielkie.

Znacznie bardziej złożone badania zostały zaprojektowane dla belki kompozytowej. W analizach uwzględniono dziesięć różnych wartości impulsów wzbudzających drgania układu. Do uczenia klasyfikatora wybrano kilka rozkładów. Do testów klasyfikatora wykorzystano rozkłady, które nie były w zbiorze treningowym i walidacyjnym. W tym przypadku VGG16 osiągnęła średni F1-score równy 0.956 w procesie uczenia oraz 0.950 w testach, co potwierdza jej stabilność i zdolność do uogólniania wzorców sygnałów. Dla porównania, VGG19 uzyskała nieco wyższy wynik w procesie uczenia ($F1 = 0.963$), lecz jej skuteczność w testach znacząco spadła do 0.886. Szczególnie problematyczna okazała się `Klasa_6`, dla której czułość spadła do 0.086, a F1-score do 0.151. Oznacza to, że model nie nauczył się poprawnie odróżniać tej kategorii od klas sąsiednich, co było związane zarówno z dużym podobieństwem cech sygnałów jak i problemem generalizacji nowych przypadków.

Po dwukrotnym zwiększeniu zbioru treningowego i walidacyjnego dokładność VGG19 w treningu wzrosła do 96.7%, a w testach do 93.1%. Jednocześnie poprawie uległa skuteczność klasyfikacji dla trudniejszych kategorii, w tym `Klasa_6`. Wyniki te dowodzą, że powiększenie zbioru danych zwiększa zdolność generalizacji VGG19 i redukuje ryzyko przeuczenia, choć pełna równowaga między wszystkimi klasami wymaga dodatkowych mechanizmów regularyzacji (np. wag klas, augmentacji ukierunkowanej na klasy trudne czy silniejszego dropout).

Przeprowadzone badania pokazały ograniczenia modeli stochastycznych. W praktyce odpowiedzi badanych układów wykazywały istotną zmienność między próbami, zależność od prawdopodobieństwa wystąpienia impulsów oraz nieliniowy charakter tłumienia przy wyższych wymuszeniach. Zjawiska te wskazują, że budowa jednoznacznego modelu analitycznego wymagałaby uwzględnienia złożonych efektów stochastycznych i nieliniowych. Z tego względu zastosowanie metod sztucznej inteligencji okazało się właściwym wyborem – umożliwiło uchwycenie subtelnych różnic w odpowiedziach i przeprowadzenie skutecznej klasyfikacji w oparciu o dane eksperymentalne, bez konieczności konstruowania pełnego modelu matematycznego. Wskazuje to, że dalsze badania powinny rozwijać kierunek mechaniki stochastycznej i metod uczenia maszynowego, z naciskiem na analizę dużej liczby zróżnicowanych przypadków wymuszeń.

9.1 Najważniejsze wnioski

1. Wyniki pochodzące z eksperymentu z wykorzystaniem układu RLC stanowią wiarygodną platformę do badań nad klasyfikacją odpowiedzi na losową serię impulsów,
2. Wymuszenia w postaci trójkątnych impulsów generowanych kartą NI USB 6251 okazały się dobrym przybliżeniem delty Diraca, umożliwiając realizację wymuszeń impulsowych w warunkach laboratoryjnych,
3. Parametry odpowiedzi układu (tłumienie, częstotliwość, amplituda, faza, składowa stała) w układzie dyskretnym wykazywały wyraźną zmienność pomiędzy próbami i rozkładami. Zjawisko to odzwierciedla losowy charakter badanego problemu oraz ograniczenia modeli stochastycznych, które nie uwzględniają efektów nieliniowych,

4. Przekształcenie odpowiedzi do postaci obrazów oraz zastosowanie konwolucyjnych sieci neuronowych (VGG16, VGG19) umożliwiło skuteczną klasyfikację odpowiedzi układu na wymuszenia impulsowe. Dla układu RLC obie architektury uzyskały niemal idealne wyniki (dokładność testowa $> 99,9\%$), co potwierdziło zasadność przyjętego podejścia,
5. W przypadku układu ciągłego (belki kompozytowej) architektura VGG16 uzyskała wysoką skuteczność (średnia miara $F_1 \approx 0,95$), wykazując stabilność i odporność na przeuczenie. Sieć VGG19, mimo nieco lepszych wyników treningowych ($F_1 \approx 0,963$), początkowo charakteryzowała się słabszą generalizacją (test $F_1 \approx 0,886$). Dopiero powiększenie zbioru danych do uczenia i walidacji pozwoliło poprawić jej skuteczność do $93,1\%$, co jednoznacznie wskazuje na kluczową rolę liczności i reprezentatywności danych w uczeniu głębokich sieci,
6. Największe trudności klasyfikatorów wystąpiły w przypadku klas o podobnych cechach odpowiedzi (np. Klasa_6 dla układu ciągłego). Niska czułość i wartość miary F_1 dla tych przypadków dowodzą, że konieczne jest dalsze rozwijanie metod równoważenia danych, augmentacji ukierunkowanej na klasy trudne oraz stosowanie mechanizmów regularyzacyjnych (m.in. wag klas, warstw dropout, transfer learning). Są to kierunki badań niezbędne dla poprawy stabilności i uniwersalności klasyfikatorów,
7. Zastosowanie metod sztucznej inteligencji stanowi skuteczne i elastyczne podejście, pozwalające na klasyfikację w oparciu o dane empiryczne bez konieczności budowy złożonego modelu matematycznego.

9.2 Kierunki dalszych badań

Analiza wyników klasyfikacji dla obu architektur wskazuje na wysoką skuteczność zarówno w procesie treningu, jak i w testach, jednak różnice pomiędzy sieciami VGG16 i VGG19 są istotne z punktu widzenia dalszych badań. W przypadku układów dyskretnych obie architektury sprawdziły się bardzo dobrze, natomiast w analizie układów ciągłych VGG16 charakteryzowała się większą stabilnością i odpornością na przeuczenie. VGG19, mimo wyższej dokładności treningowej, ujawniła problemy z generalizacją na nowych danych, co szczególnie widoczne było w klasach rzadkich (Klasa_1, Klasa_2) oraz w klasie Klasa_6, której sygnały silnie przypominają sąsiednie kategorie. Wskazuje to, że dalsze prace powinny koncentrować się na równoważeniu zbiorów danych oraz zwiększaniu zdolności modeli do uogólniania.

Jednym z istotnych kierunków rozwoju jest zwiększenie liczności danych, zwłaszcza dla klas słabo reprezentowanych. Niewystarczająca liczba próbek prowadzi do niestabilności metryk i fragmentarycznego uczenia się cech. Rozwiązaniem może być zarówno uzupełnienie danych o nowe pomiary, jak i zastosowanie odpowiednio zaprojektowanej augmentacji (*data augmentation*). Modyfikacje obejmują m.in. przesunięcia w czasie, wprowadzanie kontrolowanego szumu, zmiany parametrów okien czasowych czy transformacje w dziedzinie częstotliwości. Coraz częściej stosuje się także techniki takie jak *MixUp*, *CutMix* czy *SpecAugment*, które dodatkowo zwiększają zdolność modeli do generalizacji.

Kolejnym obszarem badań jest modyfikacja funkcji straty. Standardowa entropia krzyżowa marginalizuje klasy rzadkie, dlatego warto stosować wagi klas (*class weights*) lub funkcje straty ukierunkowane na trudne przykłady. Dodatkowo korzystne może być wygładzanie etykiet, które ogranicza nadmierną pewność predykcji i poprawia zdolność generalizacji. W przypadku głębszych architektur, takich jak VGG19, konieczne jest także zastosowanie silniejszych mechanizmów regularyzacji (wyższy dropout, penalizacja wag, DropBlock), elastycznych harmonogramów tempa uczenia oraz wcześniejszego zatrzymywania treningu (*early stopping*). Interesującym kierunkiem jest także łączenie modeli w zespoły lub wykorzystanie destylacji wiedzy (*knowledge distillation*), co pozwala uzyskać lepsze i bardziej stabilne klasyfikatory.

W kontekście dalszego rozwoju warto rozważyć strategie uczenia ukierunkowane na trudne przypadki (*hard example mining*), w których sieć częściej konfrontowana jest z próbkami błędnie klasyfikowanymi. Alternatywą jest uczenie stopniowe (*curriculum learning*), polegające na wprowadzaniu do treningu przykładów od najłatwiejszych do coraz trudniejszych, co sprzyja uporządkowanemu kształtowaniu reprezentacji cech. Obiecującym kierunkiem jest także uczenie kontrastowe (*contrastive learning*), które zwiększa separację pomiędzy klasami w przestrzeni cech i może okazać się szczególnie przydatne w przypadku kategorii o dużym podobieństwie sygnałów.

W literaturze pojawia się również koncepcja łączenia różnych architektur sieci. W kontekście VGG16 i VGG19 można rozważyć podejście kaskadowe, w którym VGG16 pełni rolę szybkiego klasyfikatora wstępnego, a VGG19 analizuje jedynie przypadki trudne. Inną strategią jest równoległe uczenie obu sieci i łączenie wyników (*ensemble*), co poprawia stabilność klasyfikacji. Możliwe jest również łączenie cech uzyskanych z obu architektur lub zastosowanie destylacji wiedzy, gdzie głębsza sieć pełni rolę nauczyciela, a płytsza — ucznia. Takie rozwiązania pozwalają łączyć zalety obu architektur i zwiększać skuteczność klasyfikacji [5, 6, 63, 55, 36].

Ostatnim istotnym kierunkiem badań jest poprawa interpretowalności klasyfikatorów. Metody takie jak *Grad-CAM*, *Score-CAM* czy analiza wrażliwości (*occlusion sensitivity*) pozwalają sprawdzić, czy sieć podejmuje decyzje na podstawie istotnych fragmentów sygnału. Analiza ablacjami (*ablation study*) umożliwia z kolei identyfikację warstw i komponentów sieci odpowiedzialnych za skuteczność klasyfikacji. Rozwój tego typu metod zwiększa zaufanie do modeli i ułatwia ich praktyczne zastosowanie.

Przeprowadzone badania potwierdzają, że metody oparte na sztucznej inteligencji stanowią skuteczne narzędzie do klasyfikacji odpowiedzi układów dynamicznych. Dalszy rozwój powinien koncentrować się na zrównoważeniu danych, usprawnieniu procesu uczenia i regularyzacji, a także na hybrydowych podejściach oraz zwiększaniu interpretowalności modeli. Takie działania pozwolą zwiększyć stabilność klasyfikacji i rozszerzyć możliwości zastosowań w diagnostyce, monitorowaniu stanu konstrukcji oraz analizie zjawisk losowych. Ostatecznie można stwierdzić, że opracowana metoda spełnia założone cele, a jej interdyscyplinarny charakter zwiększa potencjał zastosowań w przyszłych pracach badawczych i inżynierskich.

Bibliografia

- [1] B. Allahverdiev i H. Tuna. “Discrete Dirac equations”. W: *Turkish Journal of Mathematics* 49.3 (2025).
- [2] K. Banasiak. “Wykrywanie sygnałów impulsowych w złożonych warunkach ich akwizycji”. W: *Pomiary Automatyka Kontrola* 60.9 (2014), s. 722–725.
- [3] J. Baranowski i G. Czajkowski. *Układy elektroniczne. Cz. 2: Układy analogowe nieliniowe i impulsowe*. 4 wyd. Warszawa, Polska: WNT, 2004. ISBN: 83-204-2960-9.
- [4] D. Bau i in. “Network Dissection: Quantifying Interpretability of Deep Visual Representations”. W: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017. arXiv: 1704.05796. URL: <https://arxiv.org/abs/1704.05796>.
- [5] C. M. Bishop. *Pattern Recognition and Machine Learning*. New York: Springer, 2006. ISBN: 978-0387310732.
- [6] L. Breiman. “Random Forests”. W: *Machine Learning* 45.1 (2001), s. 5–32. DOI: 10.1023/A:1010933404324.
- [7] B. Chatterjee i A. Bhattacharyya. “Measurement of an impulse response of a system with a random input”. W: *IEEE Transactions on Automatic Control* 8.2 (1963), s. 186–187. DOI: 10.1109/TAC.1963.1105529.
- [8] F. Coutinho, Y. Nogami i F. Toyama. “Unusual situations that arise with the Dirac delta function and its derivative”. W: *Revista Brasileira de Ensino de Física* 31.4 (2009).
- [9] T. Czaplicki. “Przegląd elektrotechniczny”. W: *Przegląd Elektrotechniczny* (1952), s. 2–7.
- [10] O. Darrigol. *A History of Optics: From Greek Antiquity to the Nineteenth Century*. Oxford: Oxford University Press, 2012. ISBN: 978-0-19-964437-7.
- [11] T. P. Daue i J. Kunzmann. “Piezoelectric Vibration Energy Harvester: State of the Art, System Implementations and Economics”. Conference presentation, 10th CMCEE, May 2012. Maj 2012. URL: https://smart-material.com/wp-content/uploads/2025/02/Smart_CMCEE_2012.pdf (term. wiz. 14.09.2025).
- [12] P. A. M. Dirac. *The Principles of Quantum Mechanics*. 4th (revised). Oxford, UK: Oxford University Press, 1958. ISBN: 0-19-852011-5.
- [13] J. Ellis. “Proper time and the Dirac equation”. W: *Journal of Physics A: Mathematical and General* 14.11 (1981), s. 2917–2925.

- [14] U. Erkaboev i in. "Calculation of the Fermi–Dirac function distribution in two-dimensional semiconductor materials at high temperatures and weak magnetic fields". W: *Nano* 16.9 (2021).
- [15] N. Frankowska, P. Frankiewicz i A. Ozga. "Classification of distributions of the values of impulses forcing vibrations of an oscillator". W: *IEEE Access* 12 (2024), s. 158731–158741.
- [16] I. Goodfellow, Y. Bengio i A. Courville. *Deep Learning*. Cambridge, MA: MIT Press, 2016. ISBN: 978-0262035613. URL: <http://www.deeplearningbook.org>.
- [17] X. L. Guo i D. S. Li. "Experiment study of structural random loading identification by the inverse pseudo excitation method". W: *Structural Engineering and Mechanics* 18.6 (2004), s. 791–806.
- [18] K. Haška, D. Zorica i S. M. Cvetićanin. "Fractional RLC circuit in transient and steady state regimes". W: *Communications in Nonlinear Science and Numerical Simulation* 96 (2021), s. 105670.
- [19] K. He i in. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". W: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, s. 1026–1034. DOI: 10.1109/ICCV.2015.123. URL: https://www.cv-foundation.org/openaccess/content_iccv_2015/html/He_Delving_Deep_into_ICCV_2015_paper.html.
- [20] R. Iwankiewicz. "Dynamic response of linear structures to random trains of impulses with random spatial shapes". W: *Journal de Mécanique Théorique et Appliquée* 4.4 (1985), s. 485–504.
- [21] R. Iwankiewicz. *Metody stochastyczne w zagadnieniach układów dynamicznych poddanych losowym seriom impulsów*. Wrocław: Politechnika Wrocławska, Instytut Materiałoznawstwa i Mechaniki Technicznej, 1993.
- [22] R. Iwankiewicz i S. R. K. Nielsen. "Dynamic Response of Non-Linear Systems to Renewal Impulses by Path Integration". W: *Journal of Sound and Vibration* 195 (1996), s. 175–193.
- [23] R. Iwankiewicz i K. Sobczyk. "Dynamic response of linear structures to correlated random impulses". W: *Journal of Sound and Vibration* 86.3 (1983), s. 303–317.
- [24] M. Jabłoński i A. Ozga. "Determining the Distribution of Values of Stochastic Impulses Acting on a Discrete System in Relation to Their Intensity". W: *Acta Physica Polonica A* 121.1A (2012), A174–A178. DOI: 10.12693/APhysPolA.121.174.
- [25] M. Jabłoński i A. Ozga. *Distribution of Random Pulses Acting on a Vibrating System as a Function of Its Motion*. Kraków, Poland: AGH University of Science i Technology Press, 2013. ISBN: 978-83-7464-660-4.
- [26] M. Jabłoński i A. Ozga. "Distribution of Stochastic Impulses Acting on an Oscillator as a Function of Its Motion". W: *Acta Physica Polonica A* 118.1 (2010), s. 74–77. DOI: 10.12693/APhysPolA.118.74.
- [27] M. Jabłoński i A. Ozga. "Statistical characteristics of vibrations of a string forced by stochastic forces". W: *Mechanics (AGH University of Science and Technology)* 27.1 (2008), s. 1–7.

- [28] M. Jabłoński i in. “Determining the Distribution of Stochastic Impulses Acting on a High Frequency System through an Analysis of its Vibrations”. W: *Acta Physica Polonica A* 119.6A (2011), s. 977–980. DOI: 10.12693/APhysPolA.119.977.
- [29] A. Jafarzadeh Jazi, B. Shahriari i K. Torabi. “Exact closed form solution for the analysis of the transverse vibration mode of a Nano-Timoshenko beam with multiple concentrated masses”. W: *International Journal of Mechanical Sciences* 131 (2017), s. 728–743. DOI: 10.1016/j.ijmecsci.2017.08.023.
- [30] S. Kaliski. *Drgania i fale w ciałach stałych*. Warszawa, Polska: Państwowe Wydawnictwo Naukowe, 1966.
- [31] G. Klambauer i in. “Self-Normalizing Neural Networks”. W: *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*. 2017. arXiv: 1706.02515. URL: <https://proceedings.neurips.cc/paper/6698-self-normalizing-neural-networks.pdf>.
- [32] M. Kline. *Mathematical Thought from Ancient to Modern Times*. New York: Oxford University Press, 1972. ISBN: 0-19-501496-0.
- [33] A. Krizhevsky, I. Sutskever i G. E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. W: *Advances in Neural Information Processing Systems 25 (NeurIPS 2012)*. 2012, s. 1097–1105. arXiv: 1207.0580.
- [34] M. Kuhn i K. Johnson. *Applied Predictive Modeling*. New York: Springer, 2013. ISBN: 978-1-4614-6848-6. DOI: 10.1007/978-1-4614-6849-3.
- [35] E. De Lauro i in. “Analogical model for mechanical vibrations in flue organ pipes inferred by independent component analysis”. W: *Journal of the Acoustical Society of America* 122.4 (2007), s. 2413–2424.
- [36] Y. LeCun, Y. Bengio i G. Hinton. “Deep learning”. W: *Nature* 521 (2015), s. 436–444. DOI: 10.1038/nature14539.
- [37] M. Lin, Q. Chen i S. Yan. “Network In Network”. W: *arXiv* (2013). arXiv: 1312.4400. URL: <https://arxiv.org/abs/1312.4400>.
- [38] J. Lützen. *The Prehistory of the Theory of Distributions*. New York: Springer, 1982.
- [39] A. L. Maas, A. Y. Hannun i A. Y. Ng. “Rectifier Nonlinearities Improve Neural Network Acoustic Models”. W: *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language Processing (WDLASL)*. Atlanta, GA, USA, 2013. URL: https://ai.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf.
- [40] J. Martino i K. Harri. “Virtual shaker modeling and simulation, parameters estimation of a high damped electrodynamic shaker”. W: *International Journal of Mechanical Sciences* 151 (2019), s. 375–384.
- [41] MathWorks. *lsqcurvefit — Solve nonlinear curve-fitting (data-fitting) problems in least-squares sense*. Optimization Toolbox documentation. MathWorks. URL: <https://www.mathworks.com/help/optim/ug/lsqcurvefit.html> (term. wiz. 14.09.2025).

- [42] K. Mazur-Śniady i P. Śniady. "Dynamic response of linear structures to random streams of arbitrary impulses in time and space". W: *Journal of Sound and Vibration* 110.1 (1986), s. 59–68.
- [43] P. M. Morse i H. Feshbach. *Methods of Theoretical Physics*. New York, USA: McGraw–Hill, 1953.
- [44] National Instruments. *PCI/PCIe/PXI/PXIe/USB-6251 Specifications*. 2025. URL: <https://www.ni.com/docs/en-US/bundle/pci-pcie-pxi-pxie-usb-6251-specs/page/specs.html> (term. wiz. 25.06.2025).
- [45] A. Ozga. *Identyfikacja rozkładu losowych obciążeń impulsowych w liniowych dyskretnych układach dynamicznych*. Kraków, Polska: Wydawnictwa AGH, 2019. ISBN: 978-83-7464-997-1.
- [46] A. Ozga. "Optimization of Parameters for a Damped Oscillator Excited by a Sequence of Random Pulses". W: *Archives of Acoustics* 39.4 (2014), s. 645–652. DOI: 10.2478/aoa-2014-0070.
- [47] A. Ozga. "Wykorzystanie współczynników korelacji do analizy drgań oscylatorów wymuszonych losową serią impulsów". W: *Przegląd Mechaniczny* 10 (2016), s. 29–32. DOI: 10.15199/148.2016.10.3.
- [48] A. Ozga i M. Sulewski. "Application of Unsupervised Learning Algorithms for Analysis the Vibrations of an Oscillator Forced by a Random Series of Impulses". W: *Vibrations in Physical Systems* 34.1 (2023), s. 2023121. DOI: 10.21008/j.0860-6897.2023.1.21.
- [49] A. Plyukhin. "Brownian oscillator with time-dependent strength: a delta function protocol". W: *Journal of Statistical Mechanics: Theory and Experiment* 2 (2024).
- [50] P. Ramachandran, B. Zoph i Q. V. Le. "Searching for Activation Functions". W: *arXiv* (2017). arXiv: 1710.05941. URL: <https://arxiv.org/abs/1710.05941>.
- [51] M. Rivera i in. "Numerical resolution of the hyperbolic heat equation using smoothed mathematical functions instead of Heaviside and Dirac delta distributions". W: *International Communications in Heat and Mass Transfer* 46 (2013), s. 7–12.
- [52] J. B. Roberts. "System response to random impulses". W: *Journal of Sound and Vibration* 24.1 (1972), s. 23–34.
- [53] J. B. Roberts. "The response of linear vibratory systems to random impulses". W: *Journal of Sound and Vibration* 2.4 (1965), s. 375–390. DOI: 10.1016/0022-460X(65)90116-1.
- [54] J. B. Roberts i P. D. Spanos. *Random Vibration and Statistical Linearization*. Mineola, NY: Dover Publications, 2003. ISBN: 978-0-486-42513-8.
- [55] O. Russakovsky i in. "ImageNet Large Scale Visual Recognition Challenge". W: *International Journal of Computer Vision* 115.3 (2015), s. 211–252. DOI: 10.1007/s11263-015-0816-y. URL: <https://link.springer.com/article/10.1007/s11263-015-0816-y>.
- [56] B. Sapiński, Ł. Jastrzębski i J. Gołdasz. "Electrical harmonic oscillator with MR damper and energy harvester operating as TMD: Experimental study". W: *Mechatronics* 66 (2020), s. 102324.
- [57] A. Savitzky i M. J. E. Golay. "Smoothing and Differentiation of Data by Simplified Least Squares Procedures". W: *Analytical Chemistry* 36.8 (1964), s. 1627–1639. DOI: 10.1021/ac60214a047.

- [58] R. W. Schafer. “What Is a Savitzky–Golay Filter? [Lecture Notes]”. W: *IEEE Signal Processing Magazine* 28.4 (2011), s. 111–117. DOI: 10.1109/MSP.2011.941097.
- [59] R. R. Selvaraju i in. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”. W: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017. arXiv: 1610.02391. URL: <https://arxiv.org/abs/1610.02391>.
- [60] S. Shen. “Starting from the Dirac Equation”. W: *Topological Insulators: Dirac Equation in Condensed Matter*. 2 wyd. Springer, 2017, s. 17–32.
- [61] M. Sherman, G. Kerr i G. González-Parra. “Analytical solutions of linear delay-differential equations with Dirac delta function inputs using the Laplace transform”. W: *Computational & Applied Mathematics* 42.268 (2023).
- [62] K. Simonyan, A. Vedaldi i A. Zisserman. “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”. W: *arXiv* (2013). arXiv: 1312.6034. URL: <https://arxiv.org/abs/1312.6034>.
- [63] K. Simonyan i A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. W: *arXiv* (2014). arXiv: 1409.1556 [cs.CV]. URL: <https://arxiv.org/abs/1409.1556> (term. wiz. 13.09.2025).
- [64] B. Skalmierski i A. Tylikowski. *Stochastic Processes in Dynamics*. Berlin: Springer, 1982.
- [65] Smart Material Corp. *Macro Fiber Composite (MFC) — Datasheet (v2.4)*. Product datasheet. Lut. 2025. URL: https://smart-material.com/wp-content/uploads/2025/02/MFC_V2.4-datasheet-web.pdf (term. wiz. 19.09.2025).
- [66] C. Smith i M. N. Wise. *Energy and Empire: A Biographical Study of Lord Kelvin*. Cambridge: Cambridge University Press, 1989. ISBN: 978-0-521-26173-9.
- [67] K. Sobczyk. “Metody stochastyczne w mechanice: stan i tendencje rozwojowe”. W: *Mechanika Teoretyczna i Stosowana* (1983).
- [68] L. Socha i T. T. Soong. *Linearization in Analysis of Nonlinear Stochastic Systems*. Publisher not specified. 1991.
- [69] M. Sokolova i G. Lapalme. “A systematic analysis of performance measures for classification tasks”. W: *Information Processing & Management* 45.4 (2009), s. 427–437. DOI: 10.1016/j.ipm.2009.03.002.
- [70] D. Sokolovski i R. Mayato. “Superluminal transmission via entanglement, superoscillations, and quasi-Dirac distributions”. W: *Physical Review A* 81.2 (2010).
- [71] J. Steinier, Y. Termonia i J. Deltour. “Smoothing and Differentiation of Data by Simplified Least Square Procedure”. W: *Analytical Chemistry* 44.11 (1972). Komentarz/korekta tablic współczynników SG, s. 1906–1909. DOI: 10.1021/ac60319a045.
- [72] M. Sulewski i A. Ozga. “Application of the Forest Classifier Method for Description of Movements of an Oscillator Forced by a Stochastic Series of Impulses”. W: *Journal of Theoretical and Applied Mechanics* 61.4 (2023), s. 819–831. DOI: 10.15632/jtam-pl/172966.

- [73] C. Szegedy i in. "Going Deeper with Convolutions". W: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Czer. 2015, s. 1–9. DOI: 10.1109/CVPR.2015.7298594. URL: https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Szegedy_Going_Deeper_With_2015_CVPR_paper.html.
- [74] A. Tharwat. "Classification assessment methods: a detailed tutorial". W: *Applied Computing and Informatics* 14.1 (2018), s. 1–38. DOI: 10.1016/j.aci.2018.08.003.
- [75] A. Tylikowski. "Stability of a non-linear rectangular plate". W: *Journal of Applied Mechanics – Transactions of the ASME* 45 (1978), s. 583–585.
- [76] A. Tylikowski i W. Marowski. "Vibration of a nonlinear single degree of freedom system due to Poissonian impulse excitation". W: *International Journal of Non-Linear Mechanics* 21 (1986), s. 229–238.
- [77] I. S. Tyurin. "On the accuracy of the Gaussian approximation". W: *Doklady Mathematics* 80.3 (2009), s. 840–843. DOI: 10.1134/S1064562409060155.
- [78] V. Vapnik. *Statistical Learning Theory*. New York: Wiley, 1998. ISBN: 978-0471030034.
- [79] P. Varshney. *VGGNet-16 Architecture: A Complete Guide*. Kaggle Notebook. 2020. URL: <https://www.kaggle.com/code/blurredmachine/vggnet-16-architecture-a-complete-guide> (term. wiz. 13.09.2025).
- [80] VibroGo® — *Truly portable laser vibration measurement*. Product page, single-point laser Doppler vibrometer. Polytec GmbH. URL: <https://www.polytec.com/int/vibrometry/products/single-point-vibrometers/vibrogo> (term. wiz. 14.09.2025).
- [81] J. Yosinski i in. "Understanding Neural Networks Through Deep Visualization". W: *ICML 2015 Deep Learning Workshop*. 2015. arXiv: 1506.06579. URL: <https://arxiv.org/abs/1506.06579>.
- [82] M. D. Zeiler i R. Fergus. "Visualizing and Understanding Convolutional Networks". W: *Computer Vision – ECCV 2014*. T. 8689. Lecture Notes in Computer Science. Springer, 2014, s. 818–833. DOI: 10.1007/978-3-319-10590-1_53.
- [83] S. Zhou, C. Jean-Mistral i S. Chesné. "Electromagnetic shunt damping with negative impedances: Optimization and analysis". W: *Journal of Sound and Vibration* 445 (2019), s. 188–203.
- [84] D. Zorica i S. M. Cvetićanin. "Dissipative and generative fractional RLC circuits in the transient regime". W: *Applied Mathematics and Computation* 459 (2023), s. 128227.

Spis rysunków

4.1	Uproszczony schemat doświadczenia z układem RLC.	21
4.2	Widok aplikacji LabVIEW do przeprowadzania doświadczeń. Odpowiedź układu RLC na jeden z impulsów z losowej serii impulsów.	22
4.3	Fragment zarejestrowanych danych z doświadczenia zdefiniowanego w tabeli 4.2.	23
4.4	Realizacja delty Diraca przy użyciu karty pomiarowej NI USB 6251.	24
4.5	Odpowiedź układu na pojedynczy impuls zarejestrowana przy użyciu karty pomiarowej NI USB 6251.	25
4.6	Wartości średnie odpowiedzi układu RLC na wymuszenia od 100 mV do 1 000 mV.	26
5.1	Wykres słupkowy dla wartości początkowej o wielkości impulsu 200 mV.	28
5.2	Wartość średnia dla wymuszenia o wartości 200 mV.	29
5.3	Wartości błędów znormalizowanych w poszczególnych iteracjach dla wymuszenia 200 mV.	34
5.4	Wartość średnia przedstawiona względem parametrów szacowanych z błędów dla wymuszenia 200 mV.	35
5.5	Różnica pomiędzy średnią odpowiedzią układu, a krzywą uzyskaną z parametrów modelu wyznaczonych dla minimów lokalnych błędów dla wymuszenia 200 mV.	36
5.6	Reprezentacja krzywej dopasowanej na podstawie analizy błędów dla wymuszenia 200 mV.	37
5.7	Różnica pomiędzy średnią odpowiedzią układu, a krzywą uzyskaną z parametrów modelu wyznaczonych dla minimów lokalnych błędów dla wymuszenia 100 mV.	39
5.8	Różnica pomiędzy średnią odpowiedzią układu, a krzywą uzyskaną z parametrów modelu wyznaczonych dla minimów lokalnych błędów dla wymuszenia 300 mV.	39
5.9	Krzywe dopasowane na podstawie analizy błędu średniokwadratowego dla wymuszenia 100 mV.	39
5.10	Krzywe dopasowane na podstawie analizy błędu średniokwadratowego dla wymuszenia 300 mV.	39
6.1	Struktura sieci konwolucyjnej VGG [79].	41
6.2	Przepływ obrazu w sieci konwolucyjnej VGG dla przykładu obrazu samochodu [63, 33, 16, 82].	43
6.3	Schemat macierzy pomyłek z definicjami metryk: precyzja, czułość, F1 i dokładność.	47
6.4	Obraz wejściowy do sieci.	48
6.5	Wzorce krawędziowe i kontrastowe dla obu sieci dla początkowych warstw sieci.	49
6.6	Wzorce krawędziowe i kontrastowe dla obu sieci dla głębszych warstw sieci.	49

6.7	Przepływ obrazu w sieci dla obrazu samochodu: kolejne warstwy od wejścia do wyjścia. . .	50
6.8	Wizualizacja obszarów istotnych dla klasyfikacji obrazu jako <i>minivan</i> z wykorzystaniem metody Grad-CAM.	52
6.9	Reprezentacja najbardziej prawdopodobnych klas na podstawie zdjęcia.	53
7.1	Porównanie średnich wartości odpowiedzi układu RLC dla różnych poziomów wymuszenia i prawdopodobieństwie wystąpienia impulsu.	56
7.2	Porównanie pojedynczej odpowiedzi układu RLC na wymuszenie 100 mV oraz wpływu różnych czynników na kształt odpowiedzi.	57
7.3	Porównanie pojedynczej odpowiedzi układu RLC na wymuszenie 200 mV oraz wpływu różnych czynników na kształt odpowiedzi.	58
7.4	Przykłady zdarzeń klasy AA zarejestrowane w kolejnych próbach.	59
7.5	Przykłady zdarzeń klasy AB/BA zarejestrowane w kolejnych próbach.	60
7.6	Przykłady zdarzeń klasy BB zarejestrowane w kolejnych próbach.	61
7.7	Porównanie pojedynczej odpowiedzi układu RLC dla klasy mieszanej AB zarejestrowanej przez kartę pomiarową (kolor niebieski) oraz klasą wygenerowaną w celu treningu klasyfikatora (kolor czerwony).	70
7.8	Przykłady wygenerowanych obrazów dla poszczególnych klas: (a) A, (b) AA, (c) AB, (d) BA, (e) B oraz (f) BB.	71
7.9	Przebieg procesu uczenia sieci VGG16.	74
7.10	Zestawienie wyników klasyfikacji po procesie uczenia sieci VGG16 dla układu RLC. . . .	75
7.11	Reprezentacja procesu trenowania sieci dla architektury VGG19.	76
7.12	Zestawienie wyników klasyfikacji po procesie uczenia sieci VGG19 dla układu RLC. . . .	77
7.13	Zestawienie wyników klasyfikacji po procesie testowania sieci VGG16 dla układu RLC na danych spoza zbioru treningowego.	79
7.14	Zestawienie wyników klasyfikacji po procesie testowania sieci VGG19 dla układu RLC na danych spoza zbioru treningowego.	80
8.1	Struktura materiału piezoelektrycznego [65].	83
8.2	Belka z laminatu FR-4 (90 mm × 25 mm × 1 mm) z zamocowanym aktuatorem MFC M-2814-P2 oraz punktem pomiarowym.	84
8.3	Uproszczony schemat doświadczenia z belką kompozytową.	86
8.4	Stanowisko pomiarowe z belką kompozytową.	86
8.5	Średnia wartość odpowiedzi belki kompozytowej na wymuszenia w zakresie $\eta_1 = 1 \text{ V}, \dots, \eta_{10} = 10 \text{ V}$, w chwili $t \in \langle 0, 180 \rangle \text{ ms}$	88
8.6	Porównanie pojedynczej odpowiedzi belki z piezoelektrykiem na wymuszenie 1 V oraz wpływ różnych czynników na kształt odpowiedzi.	89
8.7	Porównanie pojedynczej odpowiedzi belki z piezoelektrykiem na wymuszenie 10 V oraz wpływ różnych czynników na kształt odpowiedzi.	90
8.8	Przykłady wygenerowanych obrazów dla poszczególnych klas.	94

8.9	Reprezentacja procesu trenowania sieci VGG16 dla doświadczenia z belką kompozytową. . .	96
8.10	Macierz pomyłek uzyskana w procesie walidacji podczas treningu sieci VGG16 dla doświadczenia z belką kompozytową.	97
8.11	Zestawienie wyników klasyfikacji w procesie testowania sieci VGG16 dla doświadczenia z belką kompozytową.	98
8.12	Reprezentacja procesu trenowania sieci dla architektury VGG19 dla doświadczenia z belką kompozytową.	99
8.13	Zestawienie wyników klasyfikacji w procesie trenowania sieci VGG19 dla doświadczenia z belką kompozytową.	100
8.14	Zestawienie wyników klasyfikacji w procesie testu sieci VGG19 dla doświadczenia z belką kompozytową.	101
8.15	Zestawienie wyników klasyfikacji w procesie trenowania sieci VGG19 dla doświadczenia z belką kompozytową na dwukrotnie zwiększonym zbiorze treningowym i walidacyjnym. . .	102
8.16	Zestawienie wyników klasyfikacji w procesie testowania sieci VGG19 (trenowanym na zwiększonym zbiorze treningowym i walidacyjnym) dla doświadczenia z belką kompozytową.	103

Spis tabel

4.1	Zależności elektromechaniczne i ich analogie w układach elektrycznych	20
4.2	Parametry symulacji wykorzystane w doświadczeniu	23
5.1	Czas propagacji sygnału w zależności od wartości wymuszenia	30
5.2	Wartości tłumienia, częstotliwości oraz amplitudy wyznaczonych dla minimów lokalnych dla wymuszenia 200 mV	35
5.3	Wartości tłumienia, częstotliwości oraz amplitudy wyznaczone dla minimów lokalnych dla wymuszenia 100 mV	38
5.4	Wartości tłumienia, częstotliwości oraz amplitudy wyznaczone dla minimów lokalnych dla wymuszenia 300 mV	38
5.5	Porównanie parametrów modelu uzyskanych dla różnych funkcji błędu i poziomów wymuszenia	40
6.1	Zestawienie liczby warstw konwolucyjnych w architekturach VGG16 i VGG19	46
6.2	Porównanie liczby uczonych parametrów w architekturach VGG16 i VGG19 wykorzystanych dla klasyfikacji obrazu	51
6.3	Porównanie prawdopodobieństw klasyfikacji dla wybranych kategorii w VGG16 i VGG19	53
7.1	Parametry tłumienia b , częstości ω i amplitudy a wyznaczone na podstawie minimum funkcji błędu dla wymuszeń 100 mV i 200 mV przy różnych wartościach prawdopodobieństwa $p(\eta_i)$	56
7.2	Liczba pojedynczych wymuszeń układu RLC oraz ich udział procentowy w zarejestrowanej próbie (dla wymuszeń 100 mV, $p(100mV) = 0.5$ oraz 200 mV, $p(200mV) = 0.5$)	57
7.3	Liczba odpowiedzi złożonych układu RLC oraz ich udział procentowy w zarejestrowanej próbie (dla wymuszeń 100 mV i 200 mV, $p = 0.5$)	62
7.4	Wielkość impulsu i odpowiadające im wartości prawdopodobieństwa wystąpienia impulsu	63
7.5	Liczba pojedynczych wymuszeń układu RLC oraz ich udział procentowy w próbie, gdzie prawdopodobieństwo wystąpienia impulsów było równe $p(100mV) = p(200mV) = 0.5$	64
7.6	Liczba pojedynczych odpowiedzi układu RLC oraz ich udział procentowy w próbie, gdzie prawdopodobieństwo wystąpienia impulsów było równe $p(100mV) = 0.2$ oraz $p(200mV) = 0.8$	64
7.7	Liczba par odpowiedzi mieszanych w próbach oraz ich udział procentowy, gdzie prawdopodobieństwo wystąpienia impulsów było równe $p(100mV) = p(200mV) = 0.5$	65

7.8	Liczba par odpowiedzi mieszanych w próbach oraz ich udział procentowy, gdzie prawdopodobieństwo wystąpienia impulsów było równe $p(100mV) = 0.2$ oraz $p(200mV) = 0.8$. . .	67
7.9	Parametry procesu treningu sieci VGG16	73
7.10	Porównanie dokładności klasyfikacji dla sieci VGG16 i VGG19 w procesie uczenia i testów	81
8.1	Parametry doświadczenia wykorzystane w doświadczeniu z belką kompozytową z piezoceramiką MFC	87
8.2	Definicja klas w doświadczeniu z belką kompozytową z piezoceramiką MFC	91
8.3	Przypadki losowych serii impulsów dla doświadczenia z belką kompozytową	91
8.4	Przypadki wybrane do treningu i walidacji klasyfikatora dla belki kompozytowej	92
8.5	Przypadki testowe wykorzystane do testów działania klasyfikatora	93
8.6	Liczba obrazów wykorzystanych w procesie trenowania, walidacji i testowania klasyfikatorów dla poszczególnych klas	96
8.7	Porównanie dokładności klasyfikacji dla sieci VGG16 i VGG19 w procesie treningu i testów dla doświadczenia z belką kompozytową	103