



WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI
I INFORMATYKI

Imię i nazwisko studenta: Łukasz Pliszka
Nr albumu: 165570
Poziom kształcenia: Studia drugiego stopnia
Forma studiów: stacjonarne
Kierunek studiów: Elektronika i telekomunikacja
Specjalność: Komputerowe systemy elektroniczne

PRACA DYPLOMOWA MAGISTERSKA

Tytuł pracy w języku polskim: Badanie charakterystyk widmowych obiektów biologicznych w bliskiej podczerwieni

Tytuł pracy w języku angielskim: Investigation of spectral characteristics of biological objects in the near infrared

Opiekun pracy: dr inż. Maciej Wróbel

OŚWIADCZENIE dotyczące pracy dyplomowej zatytułowanej: Badanie charakterystyk widmowych obiektów biologicznych w bliskiej podczerwieni

Imię i nazwisko studenta: Łukasz Pliszka
Data i miejsce urodzenia: 21.12.1996, Świecie
Nr albumu: 165570

Wydział: Wydział Elektroniki, Telekomunikacji i Informatyki
Kierunek: Elektronika i telekomunikacja
Poziom kształcenia: drugi
Forma studiów: stacjonarne
Typ pracy: praca dyplomowa magisterska

Świadomy(a) odpowiedzialności karnej z tytułu naruszenia przepisów ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz. U. z 2019 r. poz. 1231, z późn. zm.) i konsekwencji dyscyplinarnych określonych w ustawie z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (t.j. Dz. U. z 2020 r. poz. 85, z późn. zm.),¹ a także odpowiedzialności cywilnoprawnej oświadczam, że przedkładana praca dyplomowa została opracowana przeze mnie samodzielnie.

Niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadaniem tytułu zawodowego.

Wszystkie informacje umieszczone w ww. pracy dyplomowej, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami zgodnie z art. 34 ustawy o prawie autorskim i prawach pokrewnych.

24.09.2024, Łukasz Pliszka

Data i podpis lub uwierzytelnienie w portalu uczelnianym Moja PG

**) Dokument został sporządzony w systemie teleinformatycznym, na podstawie §15 ust. 3b Rozporządzenia MNiSW z dnia 12 maja 2020 r. zmieniającego rozporządzenie w sprawie studiów (Dz.U. z 2020 r. poz. 853). Nie wymaga podpisu ani stempla.*

¹ Ustawa z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce:

Art. 312. ust. 3. W przypadku podejrzenia popełnienia przez studenta czynu, o którym mowa w art. 287 ust. 2 pkt 1–5, rektor niezwłocznie poleca przeprowadzenie postępowania wyjaśniającego.

Art. 312. ust. 4. Jeżeli w wyniku postępowania wyjaśniającego zebrany materiał potwierdza popełnienie czynu, o którym mowa w ust. 5, rektor wstrzymuje postępowanie o nadanie tytułu zawodowego do czasu wydania orzeczenia przez komisję dyscyplinarną oraz składa zawiadomienie o podejrzeniu popełnienia przestępstwa.

STRESZCZENIE

Niniejsza praca magisterska dotyczy zastosowania spektroskopii w zakresie bliskiej podczerwieni (NIRS) w różnych dziedzinach oraz analizy wyników pomiarów za pomocą zaawansowanych algorytmów. Głównym celem pracy jest zbadanie możliwości wykorzystania spektroskopii NIRS do analizy próbek biologicznych, a także opracowanie odpowiedniego układu pomiarowego.

W pracy przedstawiono wprowadzenie do tematyki spektroskopii, wraz z uwzględnieniem spektroskopii optycznej, innych metod spektralnych oraz dziedzin zastosowań NIR. Opisano także sprzęt i akcesoria niezbędne do wykonywania pomiarów, z uwzględnieniem źródeł światła, światłowodów, kolimatorów oraz najważniejszego elementu układu pomiarowego – spektrometru optycznego. W kolejnych rozdziałach skupiono się na wykonaniu pomiarów, w celu zapoznania się z używanym w pracy sprzętem, a następnie skonstruowano docelowy układ pomiarowy. Przedstawiono etapy przygotowania stanowiska pomiarowego, opracowanie oprogramowania do obsługi sprzętu oraz opisano sposoby przetwarzania i analizy danych pomiarowych.

W pracy przedstawiono wyniki pomiarów i ich analizę. Podsumowanie zawiera wnioski oraz przedstawienie dalszych możliwości rozwoju układu pomiarowego.

Słowa kluczowe:

Spektroskopia, NIRS, Metody spektralne, CNC, GRBL, PCA

Dziedzina nauki i techniki, zgodnie z wymogami OECD:

2. Nauki inżynierskie i techniczne. 2.2 Elektrotechnika, elektronika, inżynieria informatyczna

ABSTRACT

This master's thesis concerns the application of near-infrared spectroscopy (NIRS) in various fields and the analysis of measurement results using advanced algorithms. The main objective of the thesis is to investigate the possibilities of using NIRS spectroscopy for the analysis of biological samples, as well as to develop an appropriate measurement system.

The thesis presents an introduction to the subject of spectroscopy, including optical spectroscopy, other spectral methods and NIR application areas. The equipment and accessories necessary for performing measurements are also described, including light sources, optical fibers, collimators and the most important element of the measurement system - the optical spectrometer. The following chapters focus on performing measurements in order to familiarize yourself with the equipment used in the work, followed by construction of the measurement system. The stages of preparing the measurement station and developing software to operate the equipment are presented, as well as method for data processing and analysis.

The thesis presents the measurement results and their analysis. The summary contains conclusions and presents further possibilities for developing the measurement system.

Keywords:

Spectroscopy, NIRS, Spectral methods, CNC, GRBL, PCA

Field of science in accordance with OECD requirements:

2. Engineering and Technology, 2.2 Electrical engineering, electronic engineering, information engineering

SPIS TREŚCI

Streszczenie	3
Abstract.....	4
Spis treści	5
Wykaz ważniejszych oznaczeń i skrótów	8
1. Wstęp i cel pracy	9
1.1 Cel pracy	9
2.1 Spektroskopia.....	10
2.1.1 Energia fotonu i liczba falowa	11
2.1.2 Rodzaje metod spektralnych	11
2.1.3 Porównanie metod spektralnych	11
2.2 Zakres NIRS	12
2.3 Oprzyrządowanie.....	13
2.3.1 Układy spektrometrów NIR	13
2.3.2 Źródła światła.....	16
2.3.3 Detektory	16
2.3.4 Elementy toru optycznego	16
2.4 Metody pomiarowe	17
2.4.1 Absorbancja i transmitancja.....	17
2.4.2 Prawo Lamberta-Beera.....	18
2.4.3 Prawo addytywności absorpcji	18
2.4.4 Tryb transmisyjny i reflektancyjny.....	18
2.5 Dziedziny zastosowań NIRS	19
2.5.1 Branża spożywcza.....	19
2.5.2 Przemysł	21
2.5.3 Diagnostyka chorób w medycynie	21
2.5.4 Branża farmaceutyczna	23
3. Opis sprzętu i akcesoriów wykorzystywanych w pomiarach	24
3.1 Spektrometr NIR-1 firmy Wasatch.....	24
3.1.1 Zakres pracy	24
3.1.2 Zastosowany detektor.....	24
3.2 Źródło światła	26
3.3 Światłowody.....	26
4. Pomiar testowe układu	28
4.1 Przegląd wybranych do testów próbek płynnych	28
4.2 Pomiar próbek płynnych w trybie transmisyjnym w otwartej przestrzeni	28
4.2.1 Pomiar referencyjny (tylko źródło światła)	29
4.2.2 Pomiar z zamkniętym wyjściem spektrometru (Dark)	29
4.2.3 Pusta kuweta z polistyrenu	30

4.2.4	Próbka z wodą	31
4.2.5	Próbka z alkoholem izopropylowym	33
4.2.6	Próbka z DMSO	34
4.3	Pomiary próbek płynnych w uchwycie na kuwety	35
4.3.1	Pomiar referencyjny (tylko źródło światła)	36
4.3.2	Pomiar z zamkniętym wyjściem spektrometru (Dark)	36
4.3.3	Pusta kuweta plastikowa	37
4.3.4	Próbka z wodą	38
4.3.5	Próbka z alkoholem izopropylowym	39
4.3.6	Próbka z DMSO	40
4.4	Wpływ czasu integracji na pomiary w trybie dark	41
4.5	Wpływ czasu integracji na pomiary referencyjne	42
5.	Konstrukcja układu pomiarowego do pomiarów próbek w dwóch wymiarach	43
5.1	Określenie wymagań	43
5.2	Wybór sprzętu	44
5.3	Oprogramowanie układu pomiarowego	45
5.3.1	Oprogramowanie GRBL	45
5.3.2	Wybór języka programowania	45
5.3.3	Wybór i wykorzystanie dostępnych bibliotek	46
	Biblioteka do obsługi interfejsu graficznego	46
	Biblioteka do obsługi komunikacji z frezarką CNC	46
	Biblioteka do komunikacji ze spektrometrem	46
5.3.4	Interfejs graficzny programu - frezarka	46
5.3.5	Interfejs graficzny programu - spektrometr	48
5.3.6	Sterowanie frezarką CNC	50
5.3.7	Sterowanie spektrometrem NIR-1	51
5.3.8	Sterowanie wykonywaniem pomiarów	51
5.4	Projekt ruchomej części układu pomiarowego	52
5.5	Projekt uniwersalnego mocowania do elementów pomiarowych	53
5.6	Zmontowany układ pomiarowy	53
6.	Pomiary próbek biologicznych	56
6.1	Przegląd wybranych próbek biologicznych	56
6.2	Próbka 1 – cytryna	56
6.2.1	Pomiar środka cytryny	57
6.2.2	Pomiary krawędzi cytryny	57
6.2.3	Wnioski i porównania	58
6.3	Próbka 2 – boczek	59
6.3.1	Pomiar plastra boczku	60
6.3.2	Wnioski i porównania	61

7. Algorytmy i przetwarzanie danych pomiarowych	62
7.1 Algorytmy do przetwarzania danych pomiarowych	62
7.1.1 Przykłady metod przetwarzania danych	62
7.1.2 Analiza składowych głównych (PCA)	62
Standaryzacja zmiennych	63
Rozkład wartości własnych (EVD)	63
Redukcja wymiarów	63
Uogólniony wzór PCA	64
7.2 Wykorzystanie funkcji oprogramowania MATLAB	64
PCA	64
SNV	64
Wykreślanie wykresów	65
7.3 Przetwarzanie widm pomiarowych próbek cytryny	65
7.3.1 Próbką 1 – cytryna z jej rdzeniem	65
7.3.2 Próbką 2 – cytryna ze skórką	68
7.4 Próbką boczku	71
7.5 Wnioski	74
8. Podsumowanie	75
8.1 Napotkane problemy	75
8.1.1 Problem z synchronizacją pracy spektrometru i frezarki	75
8.1.2 Problem ze zbyt wysokim umieszczeniem głowicy pomiarowej	75
8.2 Możliwości dalszego rozwoju	75
8.2.1 Rozbudowa układu o możliwość pomiarów w trzech wymiarach (3D)	76
8.2.2 Udoskonalenie algorytmów przetwarzania danych	76
8.2.3 Rozwój oprogramowania do obsługi układu pomiarowego	76
8.3 Wnioski końcowe	76
Wykaz literatury	77
Dodatek A: Spis rysunków	80
Dodatek B: Spis tabel	82
Dodatek C: Kod programu pomiarowego	83
Plik main.py	83
Plik cnc_serial.py	83
Plik my_gui.py	83
Plik wasatch.py	91
Dodatek D: Kod programu MATLAB	96
Wykresy czasu integracji	96
Wykresy pojedynczych próbek	96
Wykresy i analiza PCA	97

WYKAZ WAŻNIEJSZYCH OZNACZEŃ I SKRÓTÓW

NIR – ang. Near Infrared – bliska podczerwień

PLS – ang. Partial Least Squares, metoda najmniejszych kwadratów

GRNN – ang. generalized regression neural network, sieć neuronowa uogólnionej regresji

UV – ang. ultraviolet, ultrafiolet

IR – ang. Infrared - podczerwień

DMSO – dimetylosulfotlenek

CNC – ang. computerized numerical control, komputerowe sterowanie urządzeniami

Python – język programowania

SNR – ang. aignal to noise ratio, stosunek sygnału do szumu

Model 3D – model w trzech wymiarach (X, Y, Z)

PCA – ang. Principal component analysis, analiza głównych składowych

SNV – ang. Standard Normal Variate, metoda normalizacji danych

EVD – ang. Eigenvalue decoposition, rozkład wartości własnych

API – ang. Application Programming Interface, interfejs programowania aplikacji

1. WSTĘP I CEL PRACY

Bliska podczerwień stanowi obszar spektrum elektromagnetycznego, który odgrywa kluczową rolę w analizie obiektów biologicznych ze względu na ich unikalne właściwości absorpcyjne w tym zakresie. do badania wykorzystuje się spektroskopię optyczną [1], wykorzystującą zakres optyczny – światło widzialne, ultrafioletowe oraz podczerwone. w niniejszej pracy skupiono się na zakresie optycznym bliskiej podczerwieni NIR (*ang. near-infrared*) – w zakresie od 900 do 1700 nm.

Ze względu na szybkość wykonywania pomiarów jak i fakt, że analiza nie wymaga niszczenia czy degradacji mierzonych próbek spektroskopia NIR znajduje zastosowanie w wielu różnorodnych dziedzinach. Metodę tę wykorzystuje się zarówno w medycynie, farmacji, przemyśle jak i w rolnictwie.

1.1 Cel pracy

Celem niniejszej pracy było zbadanie charakterystyk widmowych obiektów biologicznych w zakresie bliskiej podczerwieni. w tym celu konieczne było zbudowanie odpowiedniego układu, który będzie umożliwiał wykonywanie takich pomiarów.

Istotnym elementem pracy było zestawienie stabilnego stanowiska laboratoryjnego dla spektrometru NIR-1 firmy Wasatch Photonics, które umożliwiło przeprowadzenie powtarzalnych pomiarów widm optycznych próbek biologicznych oraz ich fantomów zarówno dla obiektów stałych, jak i próbek płynnych.

Do spektrometru, który jest bazą układu należało zaproponować układ do pomiarów transmisyjnych i/lub odbiciowych z możliwością wykonywania pomiarów w przestrzeni XYZ. do tego celu należało wykorzystać i zmodyfikować gotowe urządzenie umożliwiające ruch lub zbudować takie urządzenie.

Układ mógł być zbudowany w oparciu o światłowody lub w wolnej przestrzeni, a do realizacji celu pracy konieczne mogło być zaprojektowanie i wykonanie elementów w technologii druku 3D do mocowania poszczególnych części układu pomiarowego.

2. SPEKTROSKOPIA W BLISKIEJ PODCZERWIENI

W niniejszym rozdziale opisano i wyjaśniono na czym dokładnie polega spektroskopia. Skupiono się szerzej na spektroskopii optycznej, a także przedstawiono w skrócie wybrane rodzaje metod spektralnych. Ponadto szczegółowo przedstawiono zakres pomiarowy bliskiej podczerwieni, w którym wykonywane będą pomiary oraz analiza próbek biologicznych.

Rozdział ten opisuje również podstawowe konfiguracje spektrometrów oraz sondy pomiarowe. Przedstawia tryb transmisyjny, tryb odbiciowy (reflektancyjny) oraz światłowód. Dodatkowo w rozdziale przedstawiono dziedziny zastosowań spektroskopii NIR oraz ich przykładowe zastosowania.

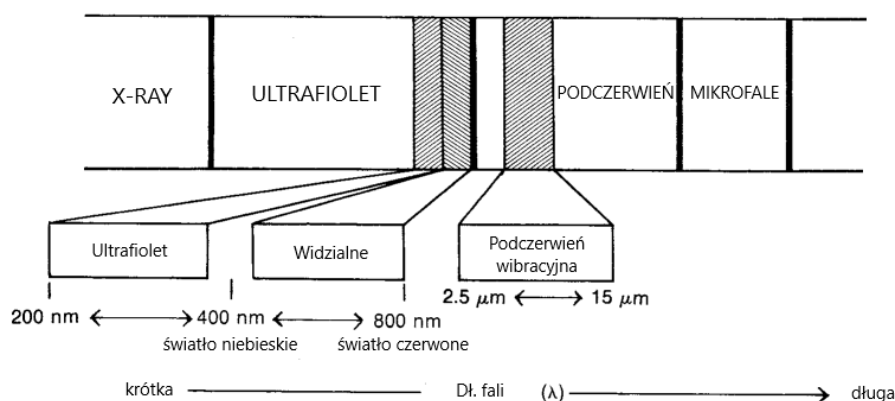
2.1 Spektroskopia

Spektroskopia [1] to dziedzina nauki, która zajmuje się badaniem i analizą widm promieniowania elektromagnetycznego. Promieniowanie elektromagnetyczne zostaje odbite lub pochłonięte o specyficznej długości fali przez badaną substancję. To, w jakim zakresie długości fali oraz jak intensywnie fala promieniowania zostanie odbita/pochłonięta zależy w szczególności od zawartości konkretnych pierwiastków lub związków chemicznych w danej substancji mierzonej.

Spektroskopia jest techniką, pozwalającą na pozyskanie wiedzy o strukturze atomowej i molekularnej, o stanach energetycznych oraz dynamice procesów chemicznych. Spektroskopia optyczna

Spektroskopia optyczna [1] skupia się na analizie promieniowania elektromagnetycznego w zakresie optycznym. Zakres optyczny obejmuje ultrafiolet (od 10 nm do 380 nm), światło widzialne (od 380 nm do 700 nm), oraz podczerwień (od 700 nm do 1 mm).

Na rysunku [Rysunek 1] przedstawiono część widma fali elektromagnetycznej z wyszczególnieniem zakresów długości fali, o których mowa wcześniej.



Rysunek 1. Część widma fali elektromagnetycznej, na podstawie [1]

Ponadto na rysunku [Rysunek 1] wskazano najbardziej użyteczne zakresy długości fali, wykorzystywane nie tylko w spektroskopii NIR, ale również w innych rodzajach metod spektralnych.

2.1.1 Energia fotonu i liczba falowa

Oprócz jednostki metrycznej, do określenia długości fali elektromagnetycznej λ (nm) wykorzystuje się liczbę falową $\tilde{\nu}$ (cm^{-1}). Poniżej przedstawiono wzór na konwersję długości fali na liczbę falową:

$$\tilde{\nu} = \frac{1}{\lambda} [\text{cm}^{-1}]$$

Liczba falowa jest odwrotnością długości fali, co umożliwia łatwiejsze porównanie różnych typów promieniowania. Ponadto warto zaznaczyć, że energia pojedynczego fotonu, który jest nośnikiem promieniowania elektromagnetycznego, zależy od długości fali zgodnie ze wzorem:

$$E = \frac{hc}{\lambda} [\text{J}]$$

Gdzie: h - stała Plancka, c to prędkość światła w próżni, λ - długość fali.

W ten sposób energia fotonu, liczba falowa oraz długość fali są ze sobą ściśle powiązane, co pozwala na dokładniejszą analizę widm i lepsze zrozumienie procesów zachodzących w substancjach badanych za pomocą spektroskopii.

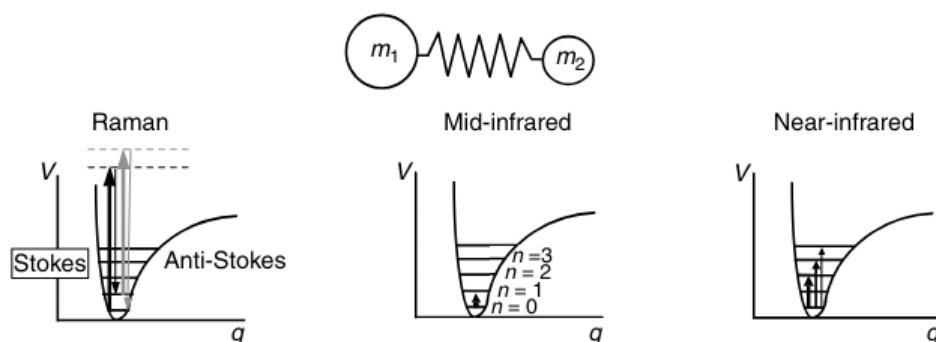
2.1.2 Rodzaje metod spektralnych

Do najczęściej wykorzystywanych metod spektralnych, oprócz spektroskopii NIR, należą:

- a) Spektroskopia MIR – z ang. Mid-Infrared Spectroscopy, spektroskopia w zakresie długości fali elektromagnetycznej 2500 nm – 25000 nm. Często wykorzystywana do identyfikacji i charakteryzacji związków chemicznych w różnych substancjach [2],
- b) Spektroskopia Ramana – z ang. Raman Spectroscopy, technika opierająca się na efekcie rozpraszania Ramanowskiego, gdzie światło, które oddziałuje z próbką, zmienia swoją częstotliwość w wyniku wymiany energii z cząstkami próbki [2],
- c) Spektroskopia UV-Vis – z ang. Ultra-Violet Visible Spectroscopy, wykorzystuje promieniowanie ultrafioletowe i widzialne w zakresie od około 200 do 800 nm. Jest stosowana do analizy absorpcji światła przez substancje, co umożliwia identyfikację i ilościową analizę związków chemicznych [2].
- d) Spektroskopia fluorescencyjna – z ang. Fluorescence spectroscopy, technika wykorzystująca emisję światła przez fluorofory po naświetleniu ich światłem o odpowiedniej długości fali. Jest używana w analizie strukturalnej, ilościowej analizie związków oraz w diagnostyce medycznej [2].

2.1.3 Porównanie metod spektralnych

Na rysunku [Rysunek 2] i tabeli [Tabela 1] porównano ze sobą spektrometrię Ramana, NIR oraz MIR [3].



Rysunek 2. Porównanie metod spektralnych - Ramana, MIR oraz NIR [3]

Tabela 1. Porównanie metod spektralnych

Technika	Raman	MIR	NIR
Rodzaj	Technika rozpraszania	Technika absorpcyjna	
Zakres	2500 nm – 200 μ m	2500 nm – 50 μ m	800 nm – 2500 nm
Źródło światła	Monochromatyczne	Polichromatyczne	
Przygotowanie próbki	Nie	Tak	Nie
Wielkość próbki	Mała, o niewielkiej grubości (μ m)		Duża (nawet do cm)
Typ światłowodu	Kwarcowy	Chalkogenkowy lub AgCl	Kwarcowy

Pomimo tego, że te trzy techniki znacznie się od siebie różnią, należą one do tej samej grupy – sygnały w widmach Ramana, MIR oraz NIR można obserwować dzięki zjawisku wibracji molekularnych [3]. Istotną różnicą jest to, że spektroskopia Ramana jest techniką wykorzystującą zjawisko rozpraszania, natomiast MIR i NIR opiera się na absorpcji promieniowania. Ponadto w technice Ramana wykorzystuje się monochromatyczne źródło światła z zakresu widzialnego NIR, natomiast w pozostałych technikach wykorzystywane są źródła polichromatyczne (o szerokim spektrum promieniowania), wykorzystujące między innymi wolfram.

W technice Ramana oraz NIR nie jest konieczne przygotowywanie próbek do pomiarów – pomiary mogą być wykonywane bezpośrednio na badanym obiekcie – bez niszczenia próbki. Ponadto spektroskopia NIR umożliwia wykonywanie badań na grubych próbkach, nawet powyżej centymetra, w przeciwieństwie do pozostałych technik, w których grubość badanego obiektu nie powinna przekraczać skali mikrometrów.

2.2 Zakres NIRS

Obszar IR jest bardzo szeroki energetycznie [4], dlatego został podzielony na trzy regiony:

- NIR – z ang. near infrared, obszar bliskiej podczerwieni (800 - 2500 nm),
- IR / MIR – z ang. infrared / mid infrared, obszar średniej podczerwieni (2500 – 25000 nm),
- FIR – z ang. far infrared, obszar dalekiej podczerwieni (25000 – 1000000 nm).

Spektroskopia NIR, o której w głównej mierze mowa w niniejszym rozdziale, zawiera się w zakresie, w którym spodziewane są pasma powstające z przejść elektronowych, pasma wynikające z składowych harmonicznych oraz kombinacji normalnych trybów drgań [4]. Spektroskopia NIR jest spektroskopią elektronową oraz oscylacyjną. Obejmuje absorpcję, emisję, rozpraszanie, odbicie, a także rozproszone odbicie światła. Region NIR także można podzielić na mniejsze podobszary:

- Region I (800 – 1200 nm),
- Region II (1200 – 1800 nm),
- Region III (1800 – 2500 nm).

Przy czym należy zaznaczyć, że granice tych obszarów nie są ściśle określone [4].

Region I wykazuje wysoką przezroczystość, ze względu na słabą energię pasm pojawiających się w tym regionie. z tego względu region ten ma zastosowania w biomedycynie oraz rolnictwie. Ze względu na swoją przezroczystość dla tkanek jest często nazywany „oknem materiałów biologicznych” [4].

Region II to region dla pierwszej i drugiej składowej harmonicznej rozciągających się drgań oraz różnych typów modów kombinowanych [4].

Region III zawiera głównie pasma przypisywane modom kombinowanym, za wyjątkiem drugiej składowej harmonicznej rozciągającego się trybu drgań. Wykazuje on gorszą przezroczystość w porównaniu do pozostałych regionów [4].

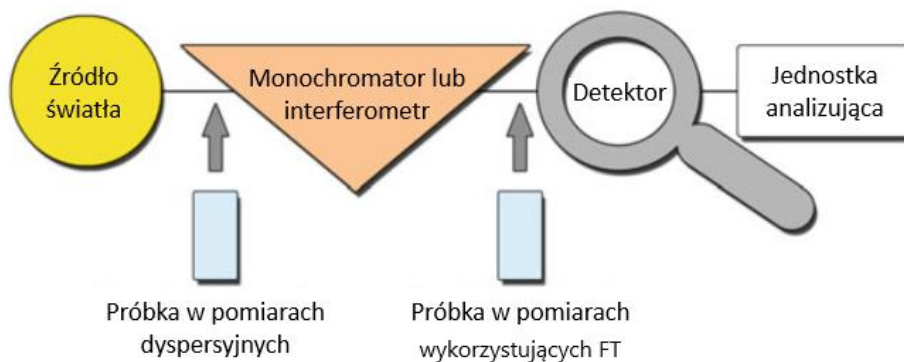
2.3 Oprzyrządowanie

2.3.1 Układy spektrometrów NIR

Typowy układ pomiarowy wykorzystujący spektrometr NIR składa się z kilku podstawowych elementów – źródła promieniowania, selektora długości fali lub interferometru oraz detektora. Składowe elementy są ze sobą połączone za pomocą optyki [4]. Można wyróżnić dwa ogólne rodzaje spektrometrów:

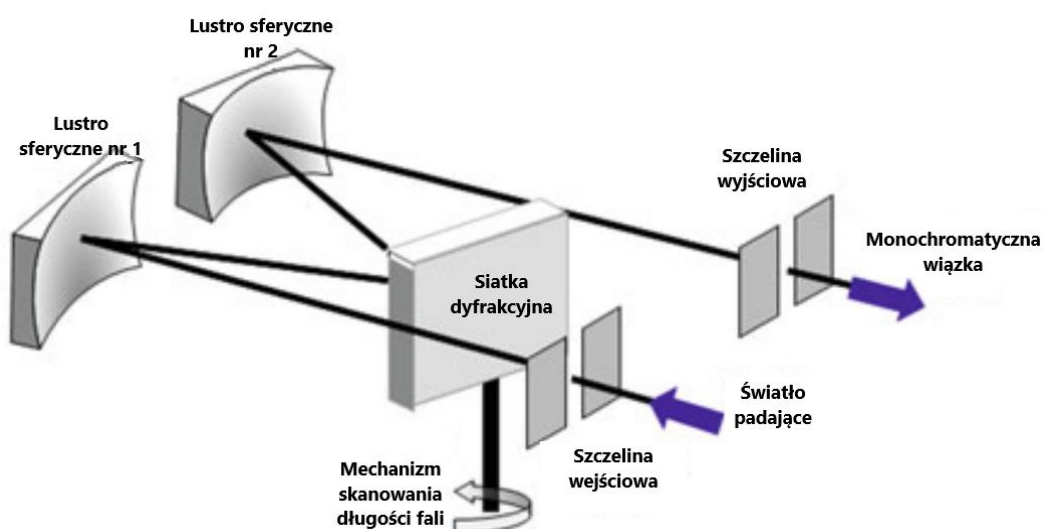
- dyspersyjne, wykorzystujące dyspersję długości fali przez pryzmat lub siatkę dyfrakcyjną,
- interferometryczne wykorzystujące transformację Fouriera (FT).

Na rysunku [Rysunek 3] przedstawiono schemat blokowy układu pomiarowego.



Rysunek 3. Schemat blokowy układu pomiarowego na podstawie [4].

W pierwszym rodzaju spektrometru wykorzystywany jest monochromator - selektor długości fali, który transmituje tylko wybrane, wąskie zakresy długości fali, które docierają do detektora w danym momencie. Podstawowym i często wykorzystywanym układem spektrometru dyspersyjnego jest układ z monochromatorem typu Czerny-Turner. Układ pokazano na rysunku [Rysunek 4].



Rysunek 4. Schemat monochromatora typu "Czerny-Turner". Na podstawie [4]

Głównymi elementami monochromatora są dwa lustra sferyczne oraz płaska siatka dyfrakcyjna. Lustra służą do skupiania wiązki światła. Lustro nr 1 kolimuje wiązkę na siatkę dyfrakcyjną, a lustro nr 2 skupia rozproszone światło z siatki i kieruje je do szczeliny wyjściowej. w zależności od kąta obrotu siatki dyfrakcyjnej względem luster i wiązki można zmieniać długość fali wiązki wyjściowej. Długość fali można obliczyć za pomocą wzoru:

$$\lambda = \frac{2}{N} * \cos\gamma * \sin\theta * 10^6 \text{ [nm]}$$

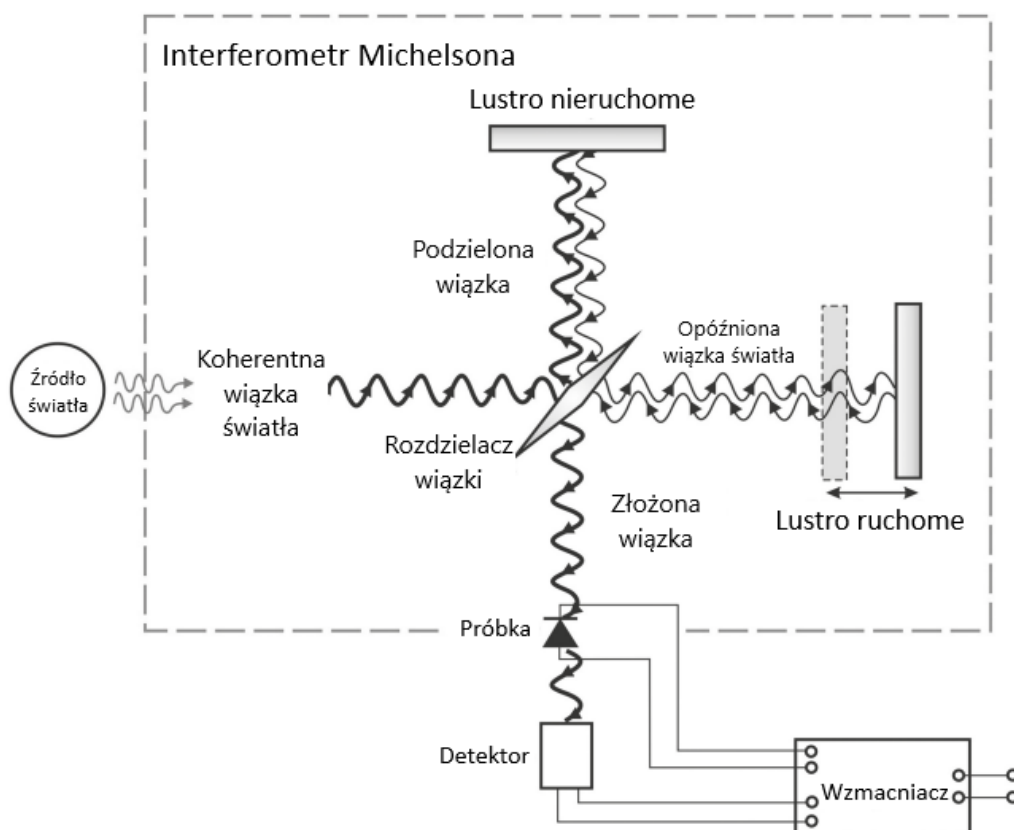
Gdzie: λ - długość fali, N – gęstość rowków siatki dyfrakcyjnej, γ - kąt odchylenia od osi środkowej, θ – kąt obrotu siatki dyfrakcyjnej.

W układach pomiarowych ze spektrometrem dyspersyjnym stosuje się dwie metody wykorzystujące monochromator:

- monochromator przed próbką - monochromator umieszczony jest za źródłem światła i selekcjonuje długość fali wiązki. Wiązka przechodzi przez próbkę, a za próbką, umieszczony jest pojedynczy detektor, który mierzy natężenie światła po przejściu przez próbkę,
- monochromator za próbką - próbka jest oświetlana szerokopasmowym źródłem światła, a monochromator umieszczony jest za próbką. Po przejściu przez próbkę, światło jest rozszczepiane przez siatkę dyfrakcyjną, a następnie trafia do detektora z ruchomą siatką dyfrakcyjną lub detektora liniowego, umożliwiającego pomiar wielu długości fali jednocześnie.

Drugi rodzaj spektrometru, zamiast klasycznego monochromatora do selekcji długości fali, wykorzystuje interferometr. Umożliwia to jednocześnie padanie wszystkich długości fali na detektor. Ruch zwierciadła powoduje zmianę natężenia sygnału na detektorze, a widmo uzyskiwane jest za pomocą transformacji Fouriera.

Najczęściej wykorzystywanym interferometrem jest interferometr Michelsona, którego schemat przedstawiono na rysunku [Rysunek 5].



Rysunek 5. Schemat interferometru Michelsona. Opracowano na podstawie [6]

Podstawowymi elementami układu jest stałe i ruchome lustro, na które kierowane są wiązki światła, przechodzące przez rozdzielacz wiązki. Rozdzielacz rozdziela wiązkę źródła światła na dwie. Jedna z wiązek służy jako wiązka odniesienia i trafia na nieruchome lustro a następnie do detektora. Natomiast druga odbija się od ruchomego lustra, wraca na dzielnik, a następnie jest skierowana w stronę detektora, gdzie zachodzi interferencja fali z wiązką odniesienia. Natężenie na detektorze w funkcji ruchu lustra tworzy interferogram, z którego można zrekonstruować widmo za pomocą transformaty Fouriera [6].

2.3.2 Źródła światła

W układach pomiarowych z wykorzystaniem spektrometru nie ma sprecyzowanego typu źródła światła. Źródło światła jest zależne od specyfiki układu pomiarowego – przede wszystkim od zakresu długości fali, dla jakich konstruuje się dany układ pomiarowy [7].

W przypadku pomiarów dla szerokiego zakresu długości fali w zakresie VIS i NIR jako źródło światła często stosuje się żarówki wolframowo-halogenowe, ze względu na ich szerokie widmo oraz stabilność temperaturową [4]. Dla innych zakresów, np. dla UV preferowane są źródła deuterowe, a dla dalekiej podczerwieni np. węgiel krzemu (SiC).

Do badań wymagających pomiaru wyłącznie dla określonej długości fali, stosuje się silne, monochromatyczne źródła, takie jak: lasery, np. półprzewodnikowe, na ciele stałym bądź lasery gazowe, a także diody elektroluminescencyjne [7].

2.3.3 Detektory

W podstawowych układach pomiarowych często zastosowanie znajdują detektory krzemowe. Krzem umożliwia budowę tanich detektorów zarówno jedno (1D) jak i dwuwymiarowych (2D). Dodatkowo detektory krzemowe charakteryzują się niskim zużyciem energii, co dodatkowo zwiększa ich atrakcyjność przy prostych układach pomiarowych. Niestety detektory krzemowe mają dwie istotne wady. Charakteryzują się dość wysokim stosunkiem sygnału do szumu (SNR) i czułością. Ponadto są ograniczone do działania w wąskim zakresie długości fali, ze względu na odcięcie przy ok. 1000 nm.

Dlatego, gdy wymagany jest szerszy zakres widmowy i/lub lepszy stosunek sygnału do szumu, stosuje się detektory o wyższej wydajności. w zależności od długości fali, odpowiedni wybór detektora jest kluczowy: detektory krzemowe są bardziej odpowiednie dla zakresu UV-Vis, natomiast dla zakresu IR lepiej sprawdzają się detektory z materiałów takich jak InGaAs, InGaAsP, a nawet germanowe (Ge). Najnowocześniejsze detektory, takie jak InGaAs, charakteryzują się doskonałym poziomem SNR oraz czułością. Jednak wszystkie te detektory wymagają stabilizacji temperaturowej za pomocą aktywnego chłodzenia, ponieważ ich szumy zmieniają się w zależności od temperatury [4].

2.3.4 Elementy toru optycznego

Elementy optyczne wykorzystywane w pomiarach w bliskiej podczerwieni, takie jak soczewki, lustra, kolimatory czy światłowody, stanowiące części składowe toru optycznego,

powinny charakteryzować się przezroczystością w zakresie badanych długości fali. Szczególnie ważne jest, aby elementy toru optycznego charakteryzowały się jak najmniejszą zawartością wodorotlenków (-OH), ze względu na ich absorpcję przy długościach fali 1400 nm i 1900 nm, mieszczących się w zakresie NIR [4].

2.4 Metody pomiarowe

Spektroskopia NIR może być stosowana do analizy różnych rodzajów próbek, między innymi ciał stałych, zawiesin, cieczy, proszków oraz gazów. Często kluczowe znaczenie mają takie parametry jak wybór metody wstępnej obróbki badanych substancji czy wybór warunków pomiaru. Ponadto konieczne może być badanie dużej ilości próbek w niewielkim czasie.

Spektroskopia NIR, poza wyborem odpowiednich metod i warunków pomiarowych, opiera się na fundamentalnych zasadach fizyki optycznej, takich jak absorbanca, prawo addytywności absorbancji oraz prawo Lamberta-Beera. Zrozumienie tych pojęć jest konieczne, ponieważ opisują one sposób, w jaki światło oddziałuje z próbką i jak mierzone jest pochłanianie promieniowania przez różne materiały.

2.4.1 Absorbancja i transmitancja

Absorbancja to wielkość bezwymiarowa, określająca stosunek natężenia światła pochłoniętego do natężenia światła padającego [5]. Jest miarą ilości światła pochłoniętego przez daną substancję (roztwór), co czyni ją najbardziej praktyczną wielkością do mierzenia dla spektrometrii. Absorbancja jest określona wzorem:

$$A = \log_{10}\left(\frac{I_0}{I}\right) [j. um.]$$

Gdzie: I_0 – natężenie światła padającego, I – natężenie światła pochłoniętego.

Transmitancja to stosunek natężenia światła przechodzącego przez próbkę do natężenia światła padającego. Przekazuje informację w jakim stopniu światło padające jest przepuszczane przez badaną substancję. Również jest wielkością bezwymiarową, z możliwością wyrażenia w procentach, określoną wzorem:

$$T = \frac{I}{I_0} [j. um.]$$

Gdzie: I_0 – natężenie światła padającego, I – natężenie światła pochłoniętego.

Absorbancja i transmitancja są ze sobą ściśle powiązane, gdyż obie opisują zjawiska związane z przechodzeniem światła przez próbkę. Znając oba wzory możliwe jest wyprowadzenie wzoru na absorbancję, korzystając z transmitancji [3]:

$$A = \log_{10}\left(\frac{1}{T}\right) = -\log_{10} T$$

2.4.2 Prawo Lamberta-Beera

Prawo Lamberta-Beera opisuje związek między absorbancją a właściwościami optycznymi próbki – jej stężeniem oraz grubością (długością drogi optycznej). Zgodnie z tym prawem, absorbancja jest proporcjonalna do stężenia substancji oraz długości optycznej, którą przebywa światło:

$$A = \varepsilon * c * l$$

Gdzie: ε – współczynnik absorpcji, c – stężenie substancji [mol/dm³]

Prawo Lamberta-Beera stanowi podstawę do ilościowego pomiaru stężenia substancji w spektroskopii [3].

2.4.3 Prawo addytywności absorpcji

Prawo addytywności absorpcji mówi, że całkowita absorbancja próbki, która zawiera wiele różnych substancji absorbujących światło, jest sumą absorbancji poszczególnych składników. Każda substancja w próbce przyczynia się do absorbancji w sposób niezależny [3]. Prawo opisano wzorem:

$$A = A_1 + A_2 + \dots + A_n$$

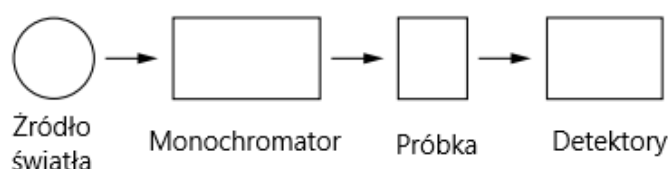
Gdzie: A_{1-n} – absorbancje poszczególnych substancji.

Prawo to jest szczególnie użyteczne w analizie złożonych mieszanin, ponieważ pozwala na rozdzielenie i analizę absorbancji różnych składników na podstawie pomiarów całkowitej absorbancji.

2.4.4 Tryb transmisyjny i reflektancyjny

Metody pomiaru w spektroskopii NIR można podzielić na dwie grupy – pomiary w trybie transmisyjnym oraz reflektancyjnym.

Metoda pomiaru w trybie transmisyjnym to metoda, w której wiązka źródła światła przechodzi przez badany obiekt, zanim trafi do detektora. Na rysunku [Rysunek 6] przedstawiono uproszczony schemat układu pomiarowego w trybie transmisyjnym.



Rysunek 6. Uproszczony schemat trybu transmisyjnego [3]

Układ składa się ze źródła światła, monochromatora, badanej próbki oraz detektora. Metoda może być stosowana tylko do materiałów przepuszczalnych światło, między innymi do cieczy, roztworów, zawiesin czy cienkich włókien i gazów.

Metoda pomiaru w trybie reflektancyjnym (odbiciowym) to metoda, w której źródło światła odbija się od powierzchni badanej próbki, zanim trafi do detektora. Na rysunku [Rysunek 7] przedstawiono uproszczony schemat układu pomiarowego w trybie reflektancyjnym (odbiciowym).



Rysunek 7. Uproszczony schemat trybu reflektancyjnego [3]

Układ składa się z tych samych elementów co układ trybu transmisyjnego, z tą różnicą, że wiązka światła nie przechodzi przez próbkę, a jest od niej odbijana, zanim trafi do detektora. Metoda może być stosowana do pomiarów próbek nieprzezroczystych, dużych, grubych, takich jak całe owoce, tabletki, czy ludzkie palce. Metodę można także stosować do próbek wymienionych dla trybu transmisyjnego.

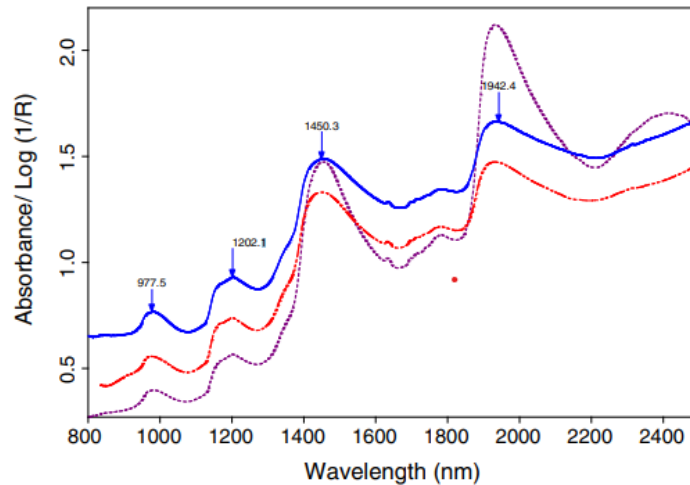
Zarówno tryb transmisyjny jak i reflektancyjny wymagają wykonania pomiaru odniesienia (pomiaru tła) przed wykonaniem właściwych pomiarów próbek. Dzięki pomiarowi intensywności światła tła, możliwe jest określenie absorpcji badanej próbki [4].

2.5 Dziedziny zastosowań NIRS

Spektrometria w zakresie NIRS ma szerokie zastosowanie w wielu branżach, między innymi w przemyśle, medycynie, farmaceutyce oraz branży spożywczej. Poniżej przedstawiono kilka przykładów wykorzystywania spektroskopii w tych dziedzinach.

2.5.1 Branża spożywcza

Pierwszym z przykładów użycia spektrometrii NIR jest analiza jakości owoców cytrusowych. Ze względu na rosnącą świadomość konsumentów i zapotrzebowaniu na owoce o gwarantowanej jakości producenci zmuszeni są do szukania coraz to nowszych technik oceny owoców cytrusowych [8]. Preferowane są przy tym nieniszczące techniki analizy jakości, ze względu na generowanie mniejszej ilości odpadów, a także umożliwienie wykonywania pomiarów tego samego produktu wielokrotnie. Na rysunku Rysunek 8. autorzy przedstawili widma absorpcji dla pomarańczy odmiany „Walencja”. Przedstawione widma różnią się od siebie ze względu na metodę pomiaru, ale wszystkie widma wykazują szczyty absorpcji na zbliżonych długościach fali.



Rysunek 8. Typowe widma absorpcji pomarańczy "Walencja" [8]

Widoczne na rysunku Rysunek 8. piki, oznaczone na długościach fal 977.5, 1202.1, 1450.3 oraz 1942.4 [nm], wskazują na pasma absorpcji dla wody której pasma mieszczą się w okolicach odpowiednio: 970, 1200, 1450, 1950 [nm]. Obserwowanie tych pasm wiąże się z tym, że pomarańcze w większości składają się właśnie z wody. do szczegółowej interpretacji pomiarów w artykule [8] wykorzystano algorytm regresji cząstkowych najmniejszych kwadratów (PLS), co było konieczne do opracowania modeli predykcyjnych i stworzenia narzędzia do analizy jakości owoców.

Natomiast w artykule [9] spektrometrię NIRS wykorzystano do oceny składu chemicznego mięsa jagnięcego. w badaniach przeprowadzono porównanie metody referencyjnej z badaniem za pomocą urządzenia FoodScan [10], przeznaczonego do badania produktów żywnościowych, wykorzystującego NIRS.



Rysunek 9. Zdjęcie urządzenia FoodScan™ 2 [10]

Wyniki uzyskane przez autorów wskazały, że spektrometria bliskiej podczerwieni daje porównywalne wyniki oznaczeń składników chemicznych w stosunku do metod referencyjnych, a urządzenie FoodScan może być wykorzystywane do analizy poziomu tłuszczu oraz wody w mięsie jagnięcym.

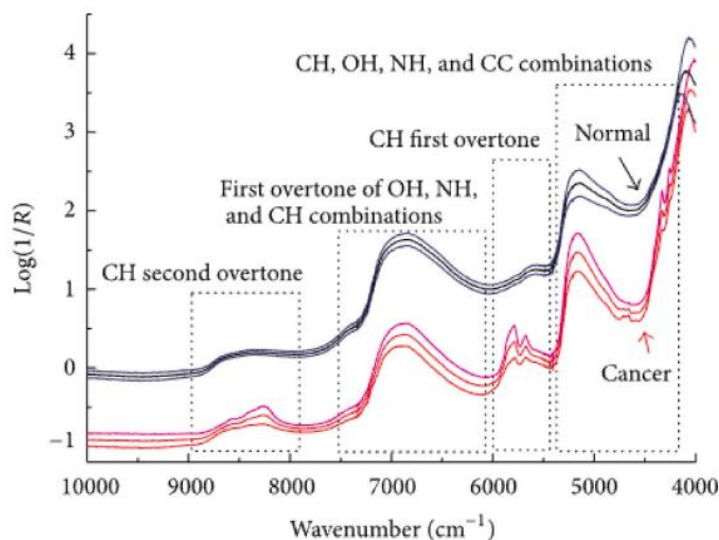
2.5.2 Przemysł

W przemyśle celulozowo-papierniczym kluczowe jest szybkie i rzetelne charakteryzowanie surowców [11] – między innymi drewna i pulpy. Spektrometria NIR w połączeniu z analizą wielowymiarową otwiera możliwość przewidywania mieszanki wiórów drewnianych z różnych gatunków drzew, co zbadali i wykazali autorzy artykułu [11]. Spektroskopię NIR wykorzystano także do badań właściwości pulpy drewnianej, które porównano później z 17 tradycyjnie mierzonymi własnościami pulpy. Na podstawie otrzymanych wyników osiągnięto ogólną zdolność predykcji na poziomie 61%. Model dobrze przewidywał niektóre właściwości, takie jak gęstość i grubość, ale miał trudności z innymi, na przykład sztywnością na rozciąganie i wytrzymałością na rozrywanie. Wynik ten podkreśla potencjał spektroskopii NIR w przemyśle celulozowo-papierniczym do szybkiej i nieinwazyjnej analizy surowców, co może przyczynić się do optymalizacji procesów produkcyjnych i podniesienia jakości końcowego produktu.

2.5.3 Diagnostyka chorób w medycynie

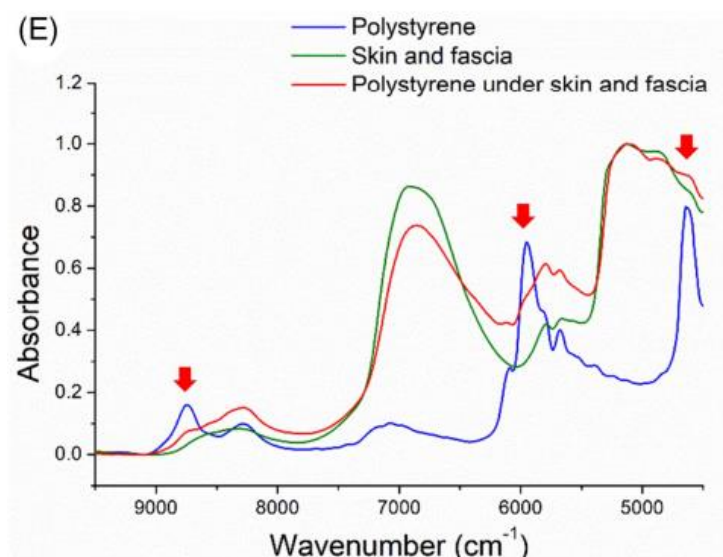
Wczesna diagnostyka i wykrycie choroby, w szczególności nowotworowej, ma pozytywny wpływ na wyniki leczenia i zwiększa szansę na całkowite wyzdrowienie chorego,

jednocześnie zmniejszając tym samym ryzyko śmierci [12]. Spektroskopia NIR pozwala szybko i bezinwazyjnie ocenić właściwości optyczne tkanek na poziomie mikronaczyń. Badając absorpcję, rozpraszanie i stężenie tlenu, wody i lipidów w tkankach pozwala ujawnić różnice między zdrową tkanką a guzem nowotworowym. Uzupełnia tym samym drogie metody obrazowania, takie jak rezonans magnetyczny czy tomografię komputerową. Na rysunku [Rysunek 10]. pokazano widma NIR zdrowych (kolor czarny) oraz nowotworowych (kolor czerwony) próbek kontrolnych, ze wskazaniem głównych składników pochłaniających, takich jak woda, lipidy czy tlen.



Rysunek 10. Widma absorpcji próbek kontrolnych [12]

Jak wynika z artykułu [13], spektroskopia w zakresie NIR może być również wykorzystywana do nieinwazyjnej oceny kości. Podstawową metodą badania kości jest badanie za pomocą promieniowania rentgenowskiego. Niesie ono za sobą konsekwencje, w postaci promieniowania jonizującego [14], mogącego wpływać negatywnie na tkanki organizmów żywych, dlatego zaleca się ograniczyć do minimum oddziaływanie na tego typu promieniowanie. Spektroskopia NIR jest nieinwazyjna, a badania mogą być wykonywane bez jakiegokolwiek negatywnego wpływu na badany organizm. w artykule [13] autorzy znaleźli wiele korelacji między parametrami spektralnymi związanymi z wodą, białkiem i tłuszczem a standardowymi parametrami kości uzyskanymi za pomocą pomiarów rentgenowskich, badając dłoń osoby chorej na wrodzoną łamliwość kości. Na rysunku Rysunek 11. autorzy przedstawili widma NIR polistyrenu (kolor niebieski), skóry i powięzi (kolor zielony) oraz polistyrenu leżącego pod skórą i powięzią (kolor czerwony). Dzięki temu udowodnili, że promieniowanie NIR może przenikać przez skórę i powięź, ponieważ wykryto sygnały z polistyrenu leżącego pod nimi.



Rysunek 11. Widma NIR polistyrenu oraz skóry i powięzi [13]

2.5.4 Branża farmaceutyczna

W artykule [15] autorzy przedstawiają wykorzystanie spektroskopii w zakresie NIR w branży farmaceutycznej. w badaniach zastosowano NIR do szybkiego określania fizycznych parametrów proszków leków ziołowych. Wykorzystano algorytm regresji cząstkowych najmniejszych kwadratów (PLS) [16] oraz sieci neuronowe uogólnionej regresji (GRNN) do transformacji fizycznych odcisków palców ziołowych proszków, ponieważ wyniki wykazały, że algorytmy te zapewniały najlepszą zdolność predykcyjną [15]. Użycie NIR w badaniach umożliwiło szybką i nieinwazyjną ocenę jakości surowców farmaceutycznych, co jest szczególnie cenne w kontroli jakości i procesie wytwarzania leków. Metoda pomiarów była szybka, ponieważ wymagała minimalnego przygotowania próbek do badań i umożliwiała bezpośredni pomiar na powierzchni próbek.

Mieszanie aktywnych składników leków z substancjami pomocniczymi jest krytycznym etapem wytwarzania leków i suplementów diety [17], ze względu na konieczność produkcji leku z jednorodną i stałą ilością aktywnych substancji w każdej tabletkę. Klasyczne metody, służące do oceny jakości wyprodukowanych tabletek są destrukcyjne, czasochłonne i kosztowne, a ponadto wymagają użycia rozpuszczalników. Spektroskopia NIR pozwala na nieinwazyjną, tanią i szybką ocenę jednorodności mieszaniny proszkowej, ponadto pozwala na określenie całej zawartości tabletki, a nie tylko składnika aktywnego. Ponadto w artykule [17] wskazano również inne zastosowania spektroskopii NIR w farmacji przy produkcji tabletek. Spektroskopię możemy wykorzystywać do kontroli procesu suszenia mieszaniny proszkowej, aby zachować jej odpowiednią wilgotność. Również w końcowym etapie produkcji leków możemy wykorzystać spektroskopię NIR, między innymi do badania grubości powłoki w tabletkach powlekanych, a także do nadzoru procesu pakowania tabletek w blistry.

3. OPIS SPRZĘTU I AKCESORIÓW WYKORZYSTYWANYCH W POMIARACH

Niniejszy rozdział zawiera opis sprzętu oraz akcesoriów, które wykorzystano do zestawienia układu pomiarowego. w rozdziale przedstawiono kluczowe elementy układu, między innymi wykorzystywany spektrometr. Opisano jego najważniejsze parametry, kluczowe w późniejszych pomiarach. Dodatkowo w rozdziale zaprezentowano zastosowane źródło światła, kolimatory, światłowody oraz kuwety na biologiczne próbki i płytę montażową.

3.1 Spektrometr NIR-1 firmy Wasatch

Spektrometr NIR-1 [Rysunek 12], produkowany przez firmę Wasatch, jest centralnym elementem układu pomiarowego. Poniżej przedstawiono kluczowe parametry tego urządzenia, które decydują o jego funkcjonalności i możliwościach pomiarowych.



Rysunek 12. Spektrometr NIR-1 firmy Wasatch [18]

3.1.1 Zakres pracy

Zakres pracy spektrometru jest jednym z kluczowych parametrów decydujących o jego zastosowaniach i możliwościach pomiarowych. Spektrometr NIR-1 charakteryzuje się zakresem pracy na poziomie 900 – 1700 nm, czyli w obszarze bliskiej podczerwieni NIR.

3.1.2 Zastosowany detektor

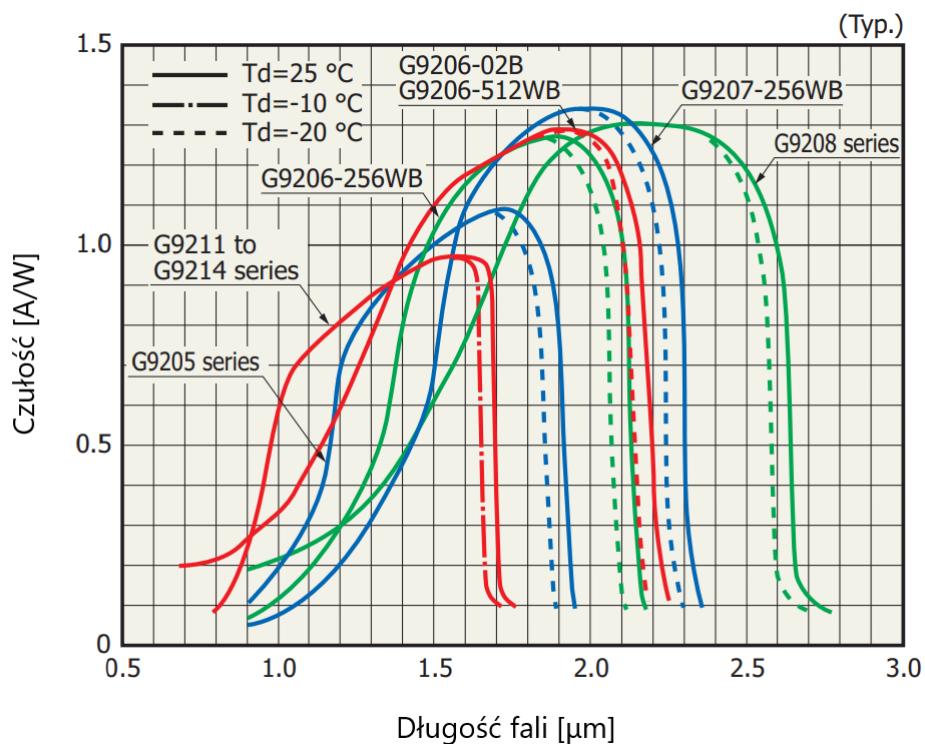
W Spektrometrze NIR-1 [18] zastosowano detektor z arsenku galowo-indowego (InGaAs), chłodzony za pomocą ogniwa termoelektrycznego (Peltiera). Model detektora to

G9214-512S firmy Hamamatsu, a jego szczegółowe parametry przedstawiono w tabeli [Tabela 2].

Tabela 2. Parametry detektora w spektrometrze NIR-1

Parametr	Wartość
Rozdzielczość	5 nm
Zakres spektralny	900 – 1700 nm
Temperatura pracy	-15 °C
Stabilność temperaturowa	±0.1 °C
Ilość aktywnych pikseli	512
Rozmiar piksela	25 x 500 µm
Czułość średnia	0.8 A/W
Czułość maksymalna	0.97 A/W
Zakres dynamiki	16.666
SNR	2400:1
Czas integracji	1 ms - 60 s
Częstotliwość próbkowania	500 Hz

Jednym z ważniejszych parametrów spektrometru jest jego czułość. Maksymalna czułość dla tego detektora wynosi 0.97 A/W dla długości fali około 1550 nm, co oznacza, że jest on najbardziej wydajny w tym zakresie spektralnym. Wykres czułości przedstawiono na rysunku [Rysunek 13].



Rysunek 13. Wykres czułości detektora G9214-512SA

Na długości fal bliskich krańcom zakresu spektralnego (900 i 1700 nm) czułość spada, co jest typowe dla detektorów półprzewodnikowych. Wydajność detektora spada również w obszarze widzialnym (poniżej 900 nm), przez co jego głównym zastosowaniem są pomiary w bliskiej podczerwieni [19].

Czułość detektora wyraża się również przez wydajność kwantową (QE), czyli stosunek liczby zaabsorbowanych fotonów do liczby wygenerowanych elektronów. Dla detektora InGaAs wydajność kwantowa może wynosić nawet 80-90% w optymalnym zakresie długości fal, co oznacza, że przekształca on większość zaabsorbowanych fotonów w sygnał elektryczny.

Ponadto czułość detektora jest również zależna od temperatury pracy. Chłodzenie za pomocą ogniwa Peltiera pozwala na obniżenie temperatury do -15°C , co redukuje szumy pochodzące z termicznej aktywności wewnętrznej detektora, gdyż szumy detektora zależą od jego temperatury pracy. Niższe zakłócenia poprawiają stosunek sygnału do szumu (SNR), który w przypadku tego detektora wynosi 2400:1. Wyższy SNR oznacza, że detektor lepiej radzi sobie z rejestrowaniem słabych sygnałów na tle szumów, co jest kluczowe dla precyzyjnych pomiarów.

Kolejnym kluczowym parametrem detektora jest czas integracji, w jakim może on zbierać pomiary. Szeroki zakres możliwości ustawienia czasu integracji umożliwia zebranie przez detektor wystarczająco dużą ilość światła, aby uzyskać dokładne spektrum. Możliwość wydłużenia czasu integracji jest szczególnie ważne dla próbek o niskiej transmisji lub słabej absorpcji. Dodatkowo wydłużenie czasu integracji może pozwolić na lepszą średnią wartość sygnału, co z kolei może poprawić stosunek sygnału do szumu (SNR).

3.2 Źródło światła

Do pomiarów wykorzystano źródło światła BPS101 [20] firmy BWTEK. Źródło wykorzystuje żarówkę wolframowo-halogenową, charakteryzującą się długą żywotnością. Ponadto źródło charakteryzuje się niewielkimi rozmiarami. w tabeli [Tabela 3] przedstawiono najważniejsze parametry urządzenia.

Tabela 3. Podstawowe parametry źródła światła BPS101

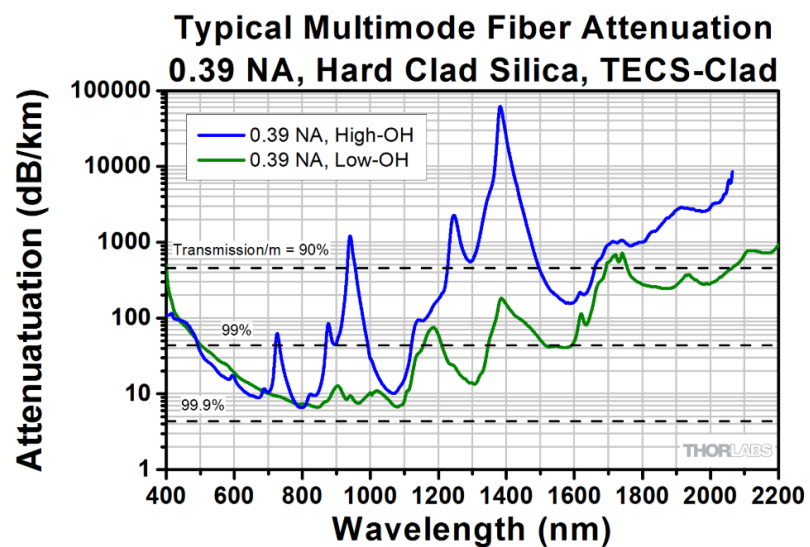
Parametr	Wartość
Źródło światła	Żarówka wolframowo - halogenowa
Zakres widmowy	350 nm – 2600 nm
Temperatura barwowa	2800 K
Zasilanie	6 V DC, <1.1 A
Żywotność	10 000 godzin

Źródło światła charakteryzuje się szerokim zakresem widmowym i jest wystarczające do wykonania pomiarów niniejszej pracy.

3.3 Światłowody

Do wykonania pomiarów wykorzystano światłowody wielomodowe FT400EMT [22] firmy THORLABS.

Światłowod [22] wykonany jest z czystego krzemu. Charakteryzuje się niskim tłumieniem w zakresie NIR, wykorzystywanym w pomiarach. Na rysunku [Rysunek 14] kolorem zielonym przedstawiono charakterystykę tłumienia światłowodu w zależności od długości fali.



Rysunek 14. Charakterystyka tłumienia światłowodu FT400EMT

4. POMIARY TESTOWE UKŁADU

W niniejszym rozdziale przeprowadzono testowe pomiary za pomocą spektrometru, zanim przystąpiono do zestawienia środowiska do pomiarów właściwych próbek biologicznych. do testowych pomiarów wybrano kilka próbek płynnych i wykonano pomiary w trybie transmisyjnym. Badania przeprowadzono w otwartej przestrzeni oraz w uchwycie na kuwety.

Dodatkowo zbadano wpływ czasu integracji na pomiary w trybie dark (zamkniętego wyjścia spektrometru) oraz w trybie light (pomiary ze źródłem światła, bez umieszczonych próbek).

4.1 Przegląd wybranych do testów próbek płynnych

Do wykonania testowych pomiarów w trybie transmisyjnym wykorzystano trzy substancje płynne oraz puste kuwety polistyrenowe. Materiały wybrano ze względu na ich łatwą dostępność.

Poniżej przedstawiono wybrane związki chemiczne:

- **Woda** – wybrana ze względu na swoją ogólną dostępność i absorpcje w paśmie NIR,
- **Alkohol izopropylowy** (izopropanol) – organiczny związek chemiczny z grupy alkoholi, wykorzystywany często jako łagodny rozpuszczalnik i środek dezynfekujący,
- **DMSO** (dimetylosulfotlenek) – organiczny związek chemiczny często stosowany w badaniach naukowych, jako rozpuszczalnik, a także jako lek, przez swoje właściwości przeciwzapalne i przeciwbólowe.

4.2 Pomiary próbek płynnych w trybie transmisyjnym w otwartej przestrzeni

Pomiar absorpcji w trybie transmisyjnym wymaga wykonania trzech oddzielnych pomiarów: pomiaru próbki (I), pomiaru tła (I_d) oraz pomiaru referencyjnego (I_0). Każdy z tych pomiarów jest widmem zależnym od długości fali (λ), więc w praktyce mamy do czynienia z wektorami danych, a nie pojedynczymi wartościami. Oznacza to, że wszystkie obliczenia są wykonywane dla każdego punktu spektralnego oddzielnie.

Pomiar próbki ($I(\lambda)$) to zliczenia fotonów, które przechodzą przez próbkę umieszczoną w kuwecie. Światło przechodzi przez substancję, co powoduje częściową absorpcję przez badany materiał. Dla każdej długości fali zarejestrowano, ile światła dotarło do detektora po przejściu przez próbkę

Pomiar tła ($I_d(\lambda)$) to zliczenia fotonów, które są rejestrowane, gdy żadne światło nie trafia na detektor. Pomiar ten jest używany do kompensacji szumów detektora i ewentualnych zakłóceń termicznych, które mogą występować niezależnie od obecności próbki czy źródła światła.

Pomiar referencyjny ($I_0(\lambda)$) to zliczenia fotonów, które docierają do detektora, gdy przez kuwetę przechodzi samo światło z lampy, bez próbki i kuwety. Ten pomiar jest wykonywany

w celu ustalenia maksymalnej ilości światła, która może dotrzeć do detektora dla każdej długości fali, bez wpływu absorpcji badanej substancji.

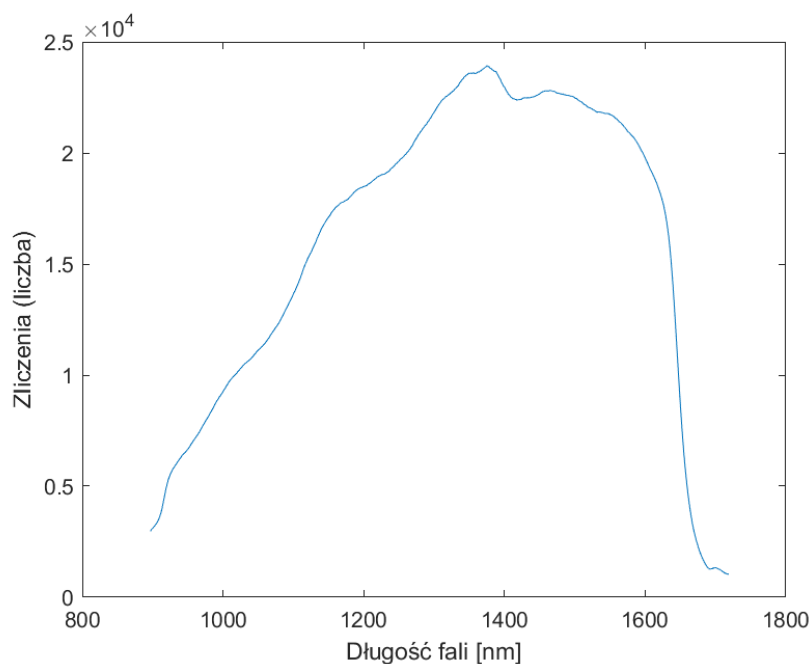
Pomiary próbek płynnych wykonano umieszczając kuwetę (pustą lub z substancją) przed spektrometrem, a następnie oświetlając ją za pomocą źródła światła [20]. Pomiary wykonano bez uśredniania, z czasem integracji równym 1 ms. Dodatkowo dla pomiarów próbek i pustej kuwety obliczono procentową transmisję, korzystając ze wzoru:

$$Transmisja [\%] = \frac{Raw - Dark}{Reference}$$

Gdzie: Raw – dane pomiarowe ze spektrometru (zliczenia), Dark – pomiar w trybie dark (zliczenia), Reference – pomiar w trybie light (zliczenia)

4.2.1 Pomiar referencyjny (tylko źródło światła)

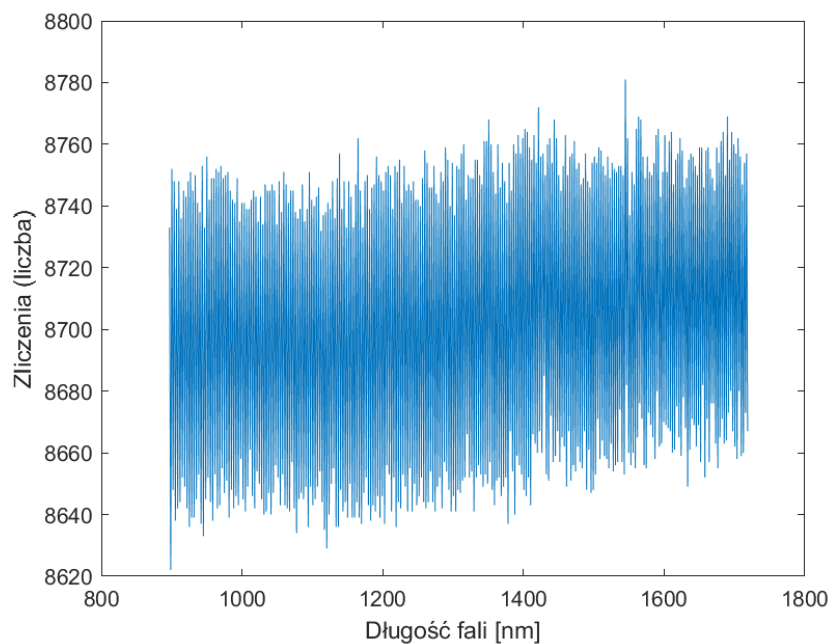
Pomiar referencyjny wykonano umieszczając źródło światła przed spektrometrem tak, aby wiązka światła padała na detektor. Widmo referencyjne przedstawiono na rysunku [Rysunek 15].



Rysunek 15. Wykres widma referencyjnego

4.2.2 Pomiar z zamkniętym wyjściem spektrometru (Dark)

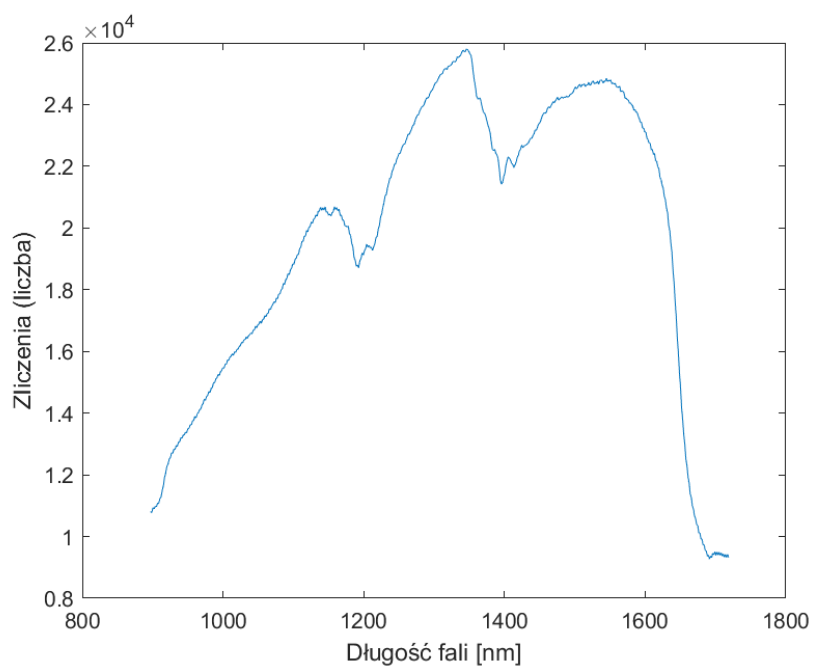
Pomiar w trybie dark wykonano zamykając wyjście spektrometru dołączoną do zestawu zaślepką, uniemożliwiając dopływ światła z zewnątrz. Widmo przedstawiono na rysunku [Rysunek 16].



Rysunek 16. Wykres widma dark

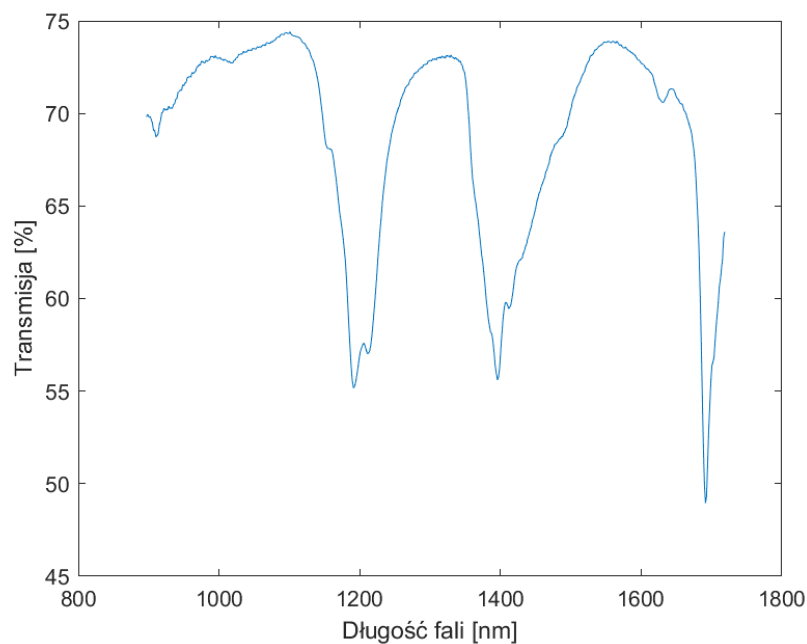
4.2.3 Pusta kuweta z polistyrenu

Widmo z pomiaru pustej kuwety z polistyrenu przedstawiono na rysunku [Rysunek 17].



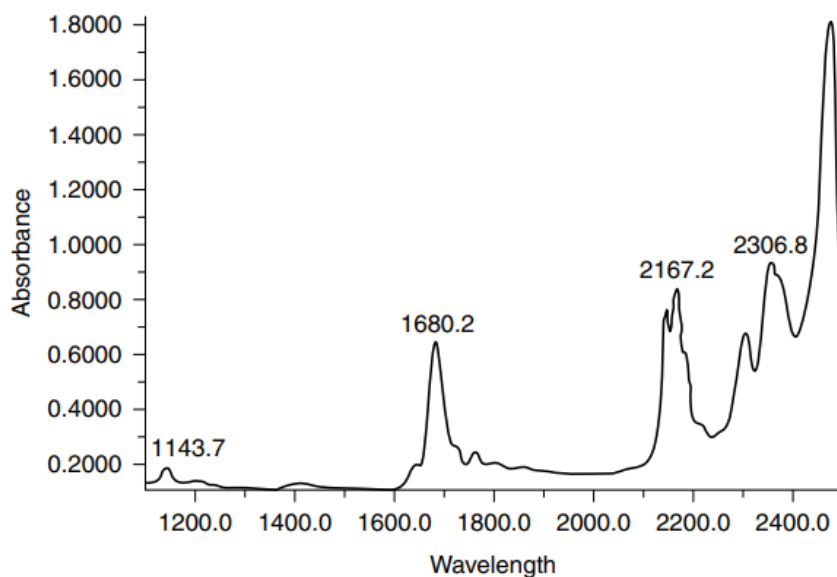
Rysunek 17. Wykres widma pustej kuwety z polistyrenu

Na rysunku [Rysunek 18] przedstawiono wykres procentowej transmisji pustej kuwety.



Rysunek 18. Wykres transmisji dla pustej kuwety z polistyrenu

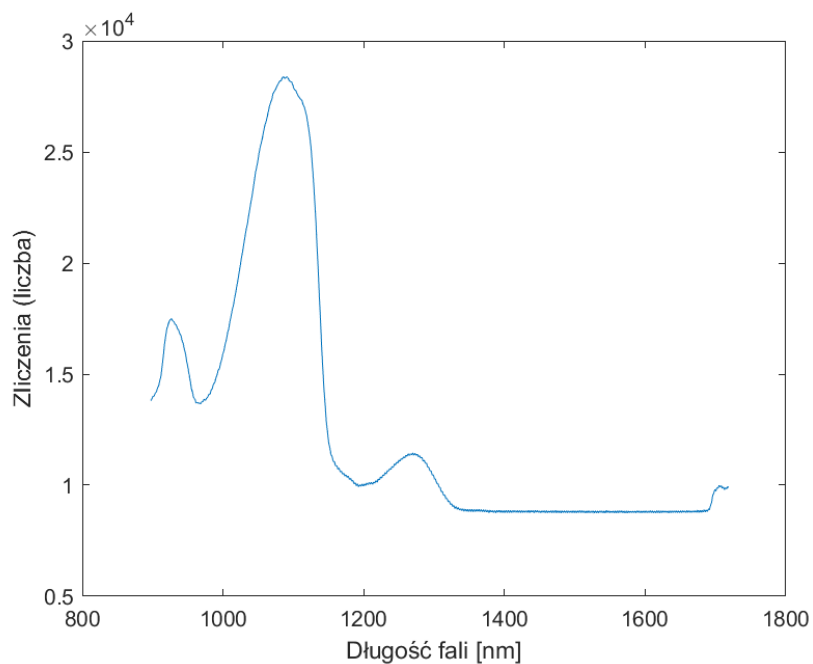
Obserwowane obniżenie transmisji w okolicach długości fali 1150 nm oraz 1700 nm było spodziewane, ponieważ polistyren charakteryzuje się czterema pasmami absorpcji: w okolicach 1143 nm, 1680 nm, 2167 nm oraz 2307 nm [3]. Pozostałych dwóch pasm nie zaobserwowano ze względu na ograniczenia detektora w spektrometrze. Na rysunku [Rysunek 19] przedstawiono wykres absorbancji dla polistyrenu, z zaznaczonymi maksimami absorpcji w funkcji długości fali.



Rysunek 19. Wykres absorbancji dla polistyrenu [3]

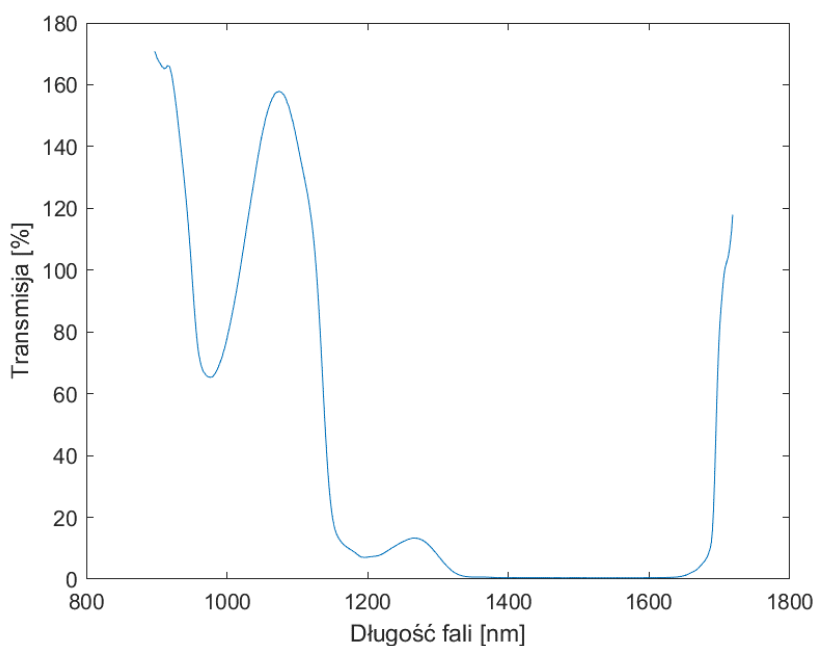
4.2.4 Próbką z wodą

Widmo z pomiaru kuwety z wodą przedstawiono na rysunku [Rysunek 20].



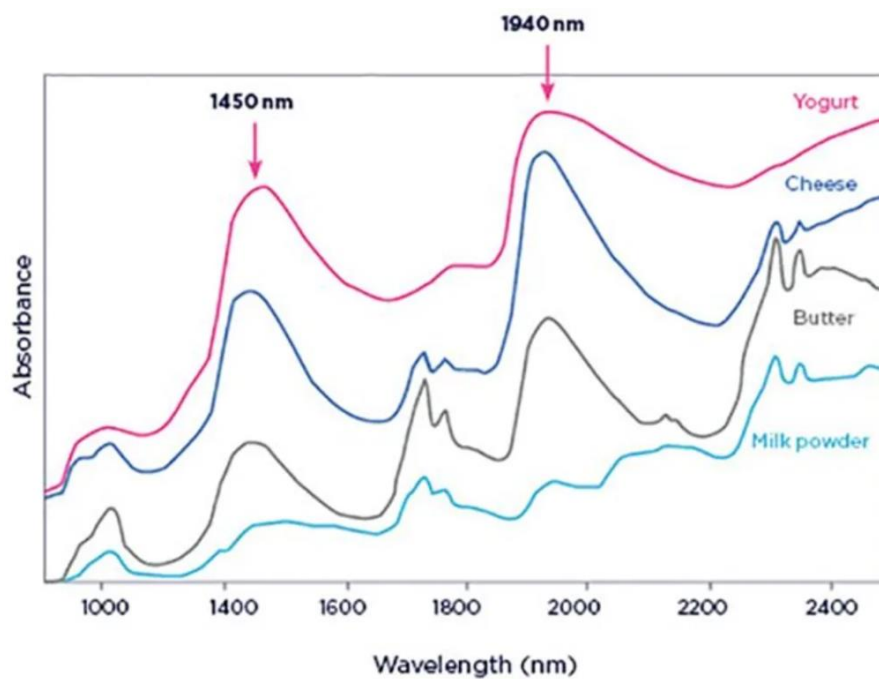
Rysunek 20. Wykres widma próbki z wodą

Na rysunku [Rysunek 21] przedstawiono wykres procentowej transmisji próbki z wodą.



Rysunek 21. Wykres procentowej transmisji dla próbki z wodą

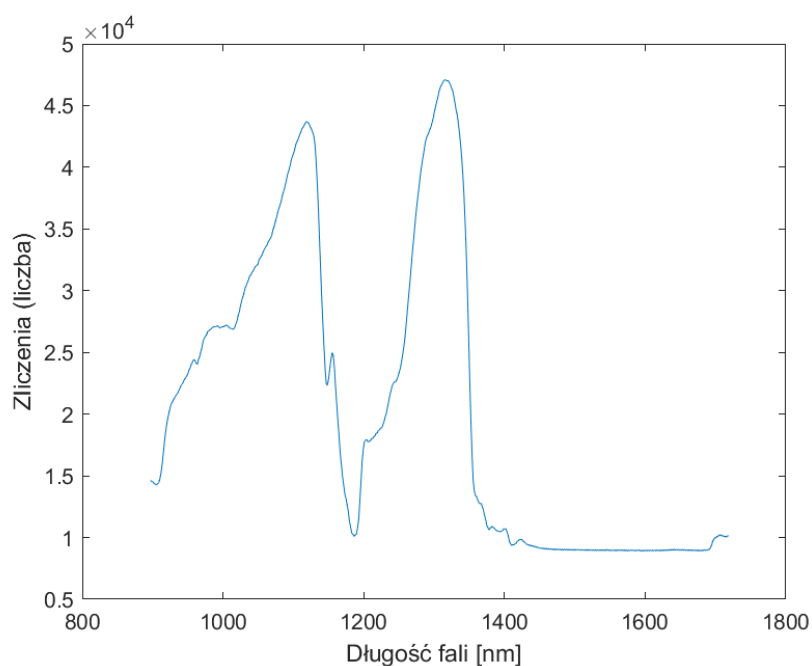
Obserwowane znaczne obniżenie transmisji w zakresie 1400 – 1700 nm było spodziewane, ponieważ woda charakteryzuje się silną absorpcją w bliskiej podczerwieni, w szczególności w okolicach 1450 nm oraz 1940 nm [23]. Drugiego zakresu nie zaobserwowano ze względu na ograniczone pasmo detektora w spektrometrze. Na rysunku [Rysunek 22] przedstawiono wykres absorpcji w funkcji długości fali dla wybranych produktów mlecznych, z zaznaczeniem pasm absorpcyjnych dla wody [23].



Rysunek 22. Wykres absorpcyjny dla wybranych produktów mlecznych [23]

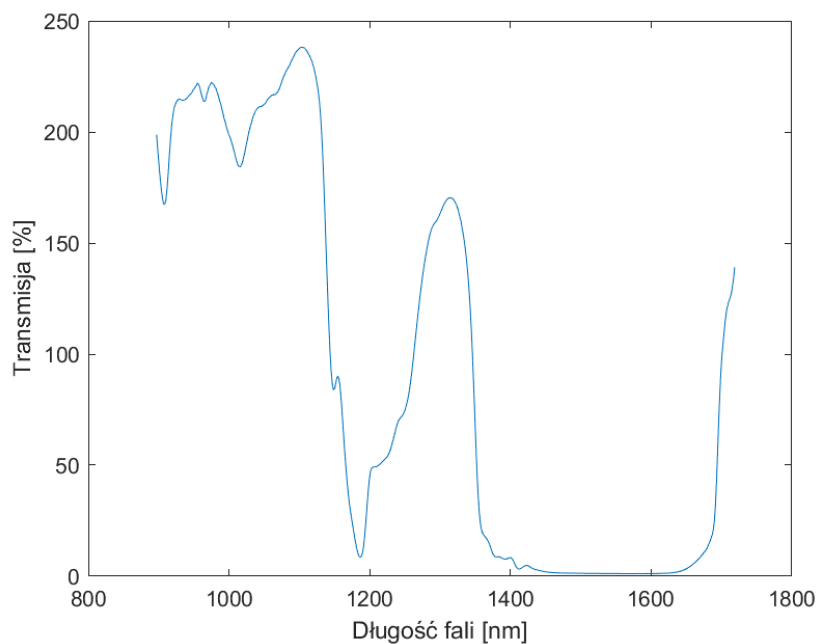
4.2.5 Próbkę z alkoholem izopropylowym

Widmo z pomiaru kulety z wodą przedstawiono na rysunku [Rysunek 23].



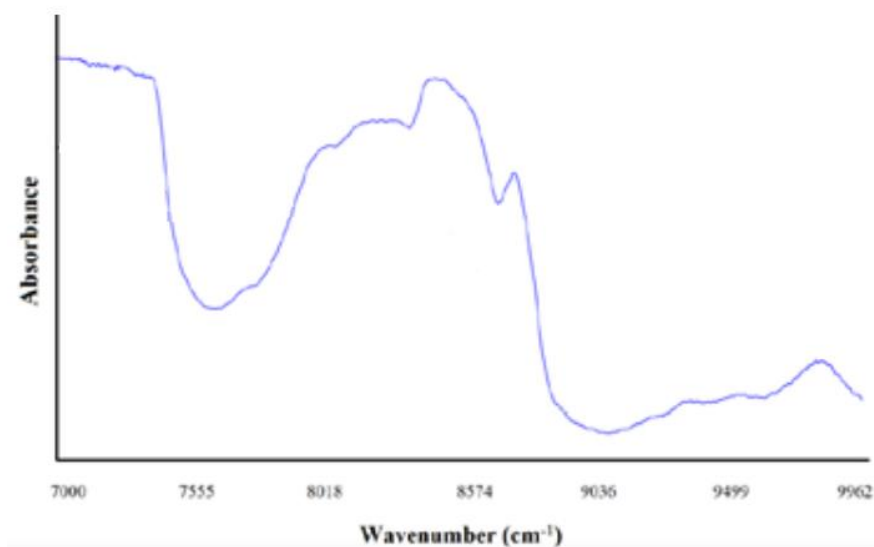
Rysunek 23. Wykres widma próbki alkoholu izopropylowego

Na rysunku [Rysunek 24] przedstawiono wykres procentowej transmisji próbki z alkoholem izopropylowym.



Rysunek 24. Wykres procentowej transmisji dla alkoholu izopropylowego

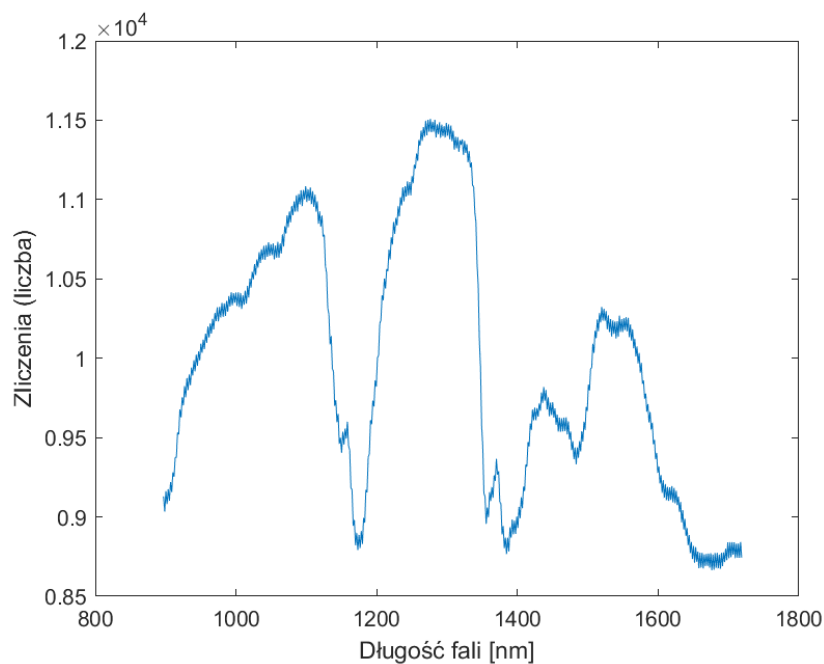
Obserwowane znaczne obniżenie transmisji w zakresach ok. 1200 nm oraz 1400 nm – 1680 nm było spodziewane, ponieważ izopropanol charakteryzuje się silną absorpcją w tych zakresach. Na rysunku [Rysunek 25] przedstawiono wykres absorbancji dla



Rysunek 25. Wykres absorbancji dla alkoholu izopropylowego [24]

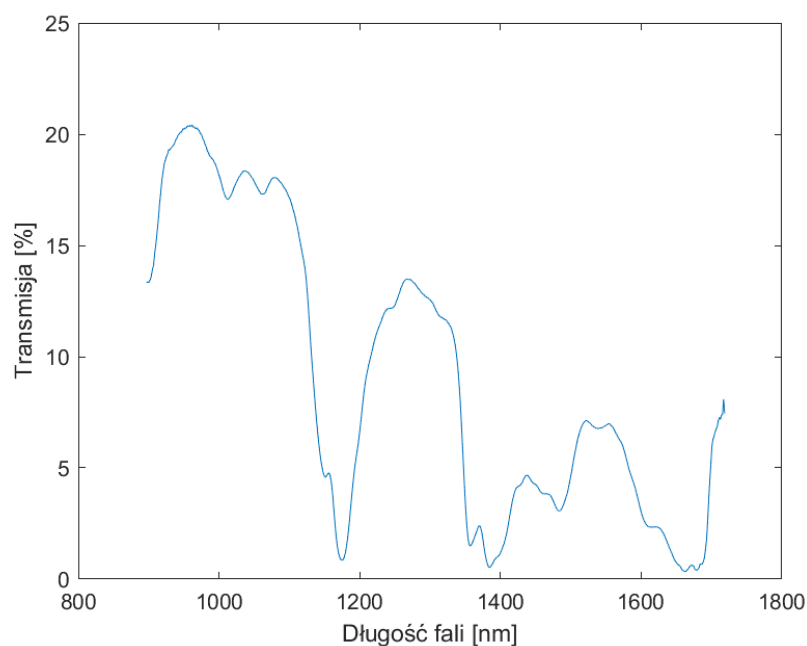
4.2.6 Próbką z DMSO

Widmo z pomiaru kuwety z wodą przedstawiono na rysunku [Rysunek 26].



Rysunek 26. Wykres widma próbki dimetylosulfotlenku (DMSO)

Na rysunku [Rysunek 27] przedstawiono wykres procentowej transmisji próbki z DMSO.



Rysunek 27. Wykres procentowej transmisji dla dimetylosulfotlenku (DMSO)

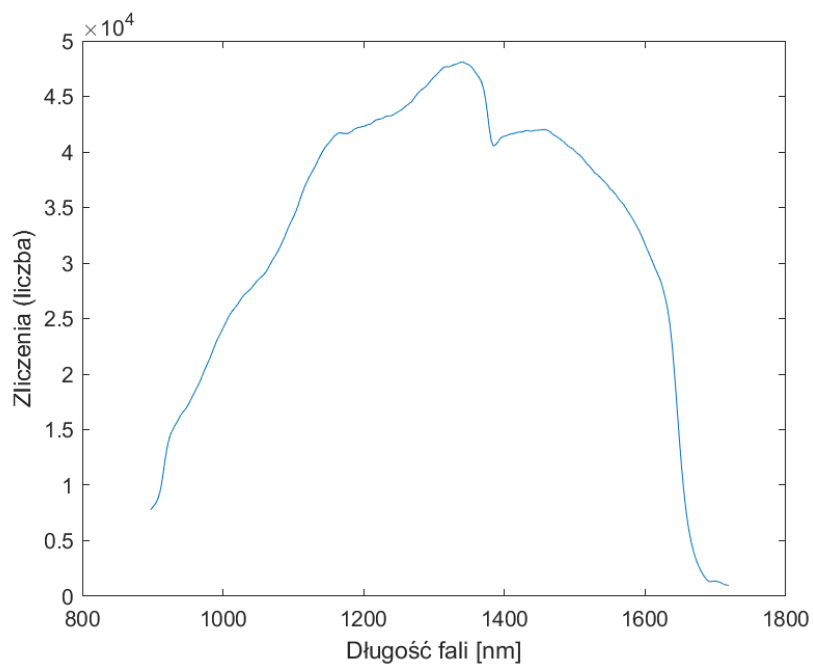
4.3 Pomiary próbek płynnych w uchwycie na kuwety

Pomiary próbek wykonano umieszczając kuwetę (pustą lub z substancją) w uchwycie na kuwety SQ-ALL [21]. Następnie uchwyt połączono ze źródłem światła [20] oraz spektrometrem za pomocą światłowodów [22]. Pomiary wykonano bez uśredniania, z czasem

integracji równym 1 ms. Dodatkowo dla pomiarów próbek i pustej kuwety obliczono procentową transmisję.

4.3.1 Pomiar referencyjny (tylko źródło światła)

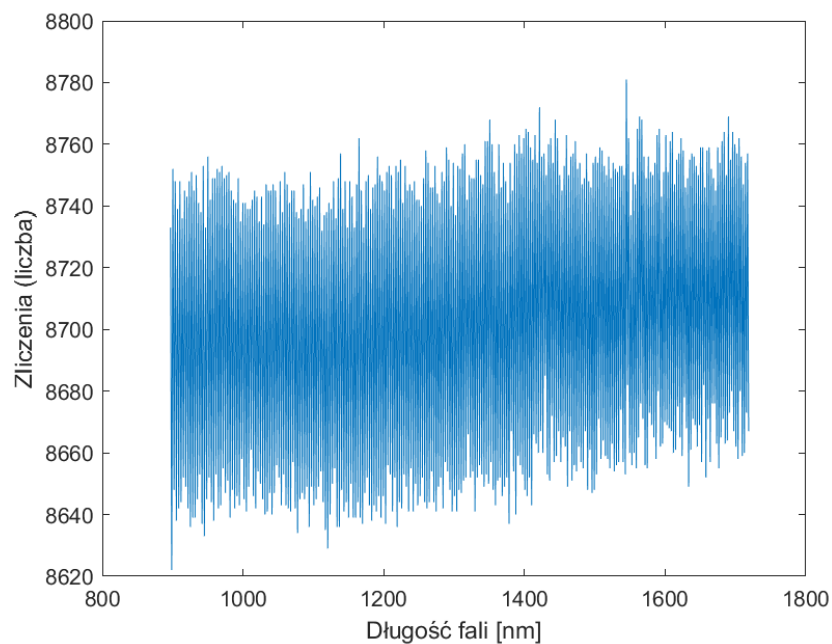
Pomiar referencyjny wykonano bez umieszczonej kuwety w uchwycie. Pomiar przedstawiono na rysunku [Rysunek 28].



Rysunek 28. Wykres widma referencyjnego

4.3.2 Pomiar z zamkniętym wyjściem spektrometru (Dark)

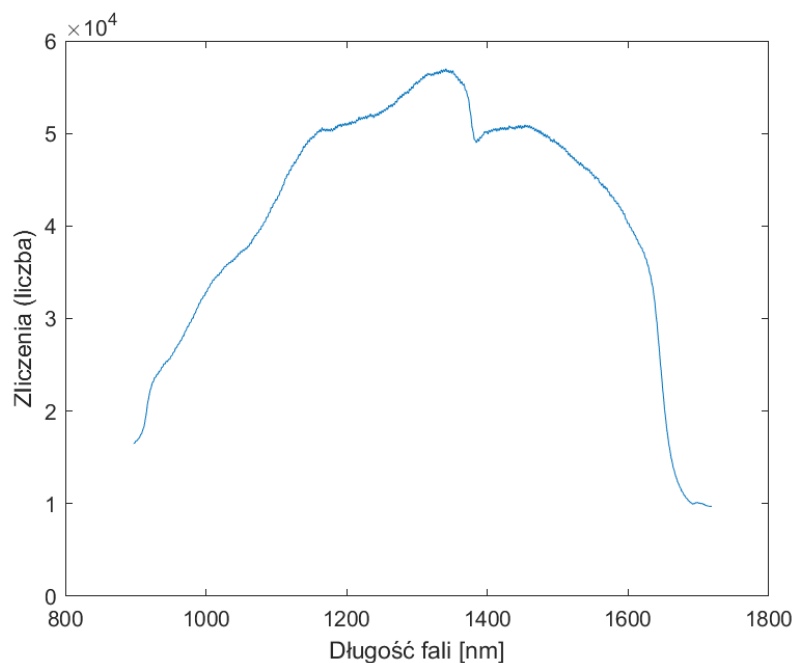
Pomiar trybu dark wykonano analogicznie jak w rozdziale [4.2.2]. Wyniki pomiarów przedstawiono na rysunku [Rysunek 29].



Rysunek 29. Wykres widma dark

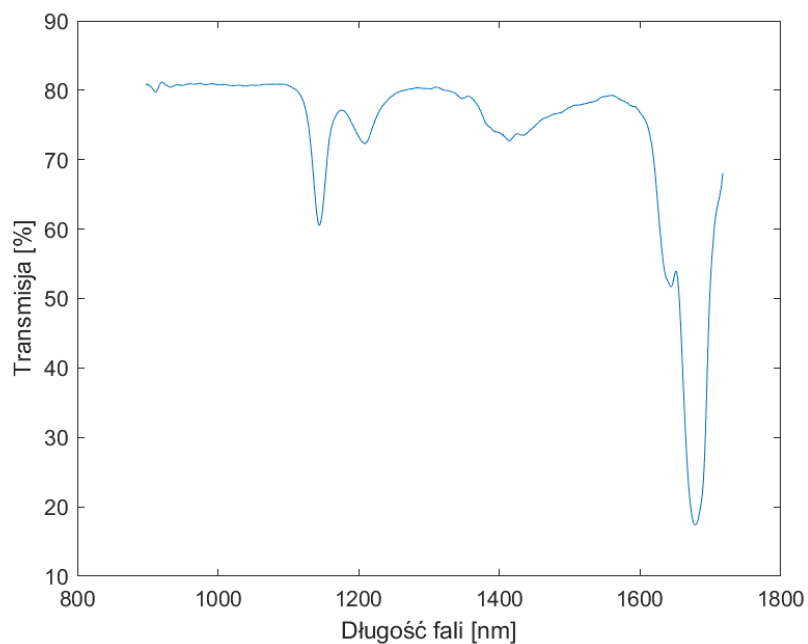
4.3.3 Pusta kuweta plastikowa

Widmo z pomiaru pustej kuwety umieszczonej w uchwycie przedstawiono na rysunku [Rysunek 30].



Rysunek 30. Wykres widma pustej kuwety z polistyrenu

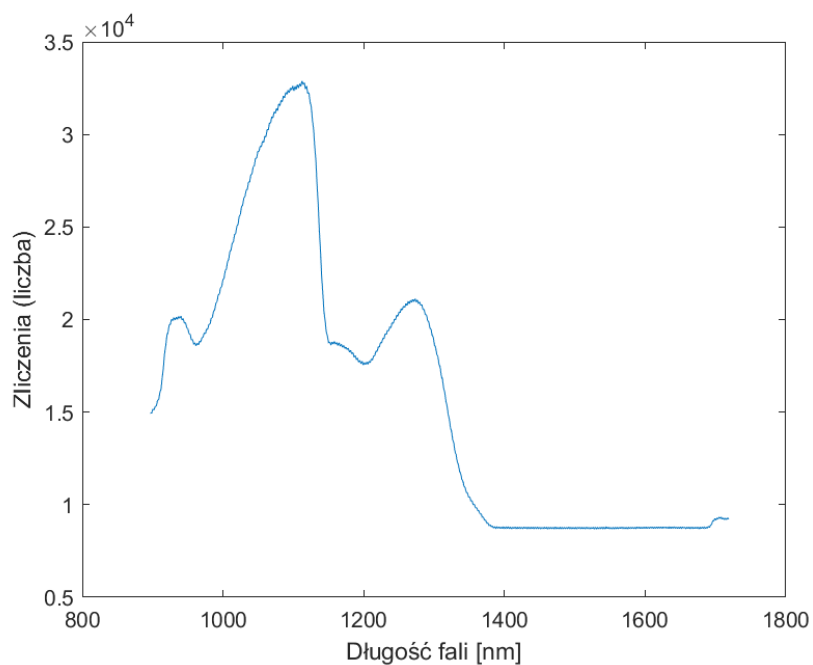
Na rysunku [Rysunek 31] przedstawiono wykres procentowej transmisji pustej kuwety.



Rysunek 31. Wykres procentowej transmisji dla pustej kuwety z polistyrenu

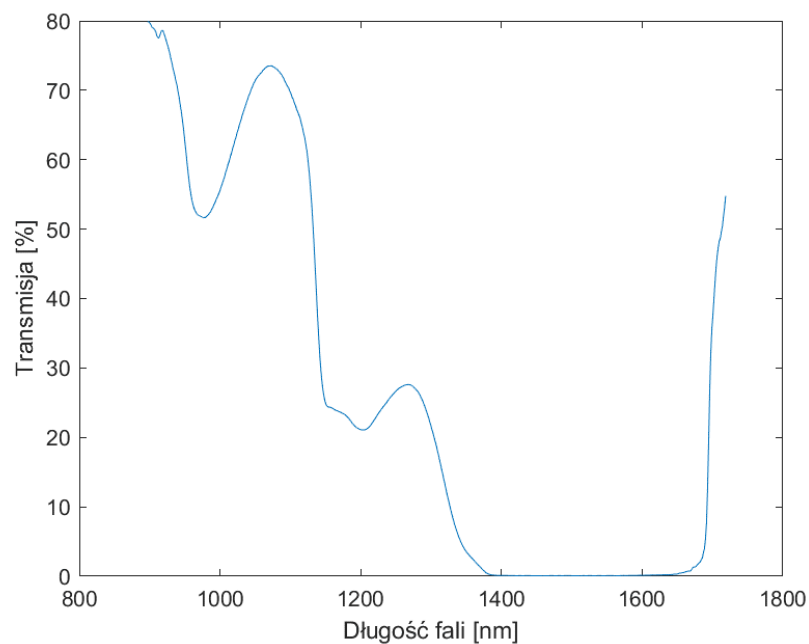
4.3.4 Próbką z wodą

Widmo z pomiaru kuwety z wodą umieszczonej w uchwycie przedstawiono na rysunku [Rysunek 32].



Rysunek 32. Wykres widma próbki z wodą

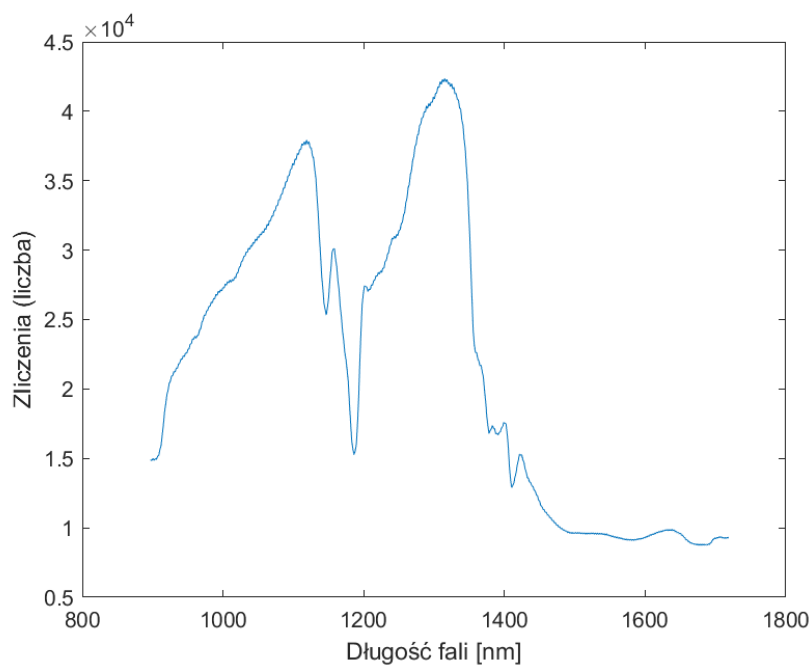
Na rysunku [Rysunek 33] przedstawiono wykres procentowej transmisji próbki z wodą.



Rysunek 33. Wykres procentowej transmisji dla próbki z wodą

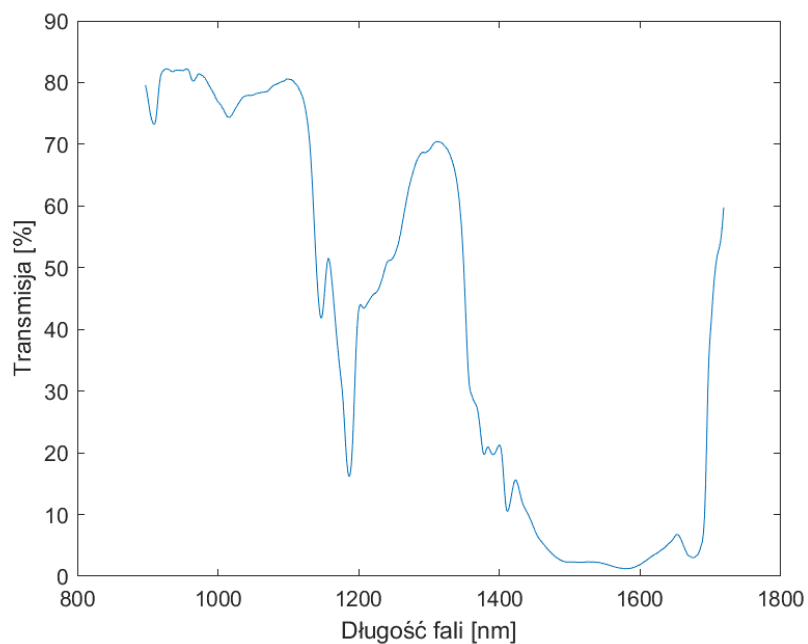
4.3.5 *Próbka z alkoholem izopropylowym*

Widmo z pomiaru kulety z alkoholem izopropylowym umieszczonej w uchwycie przedstawiono na rysunku [Rysunek 34].



Rysunek 34. Wykres widma próbki z alkoholem izopropylowym

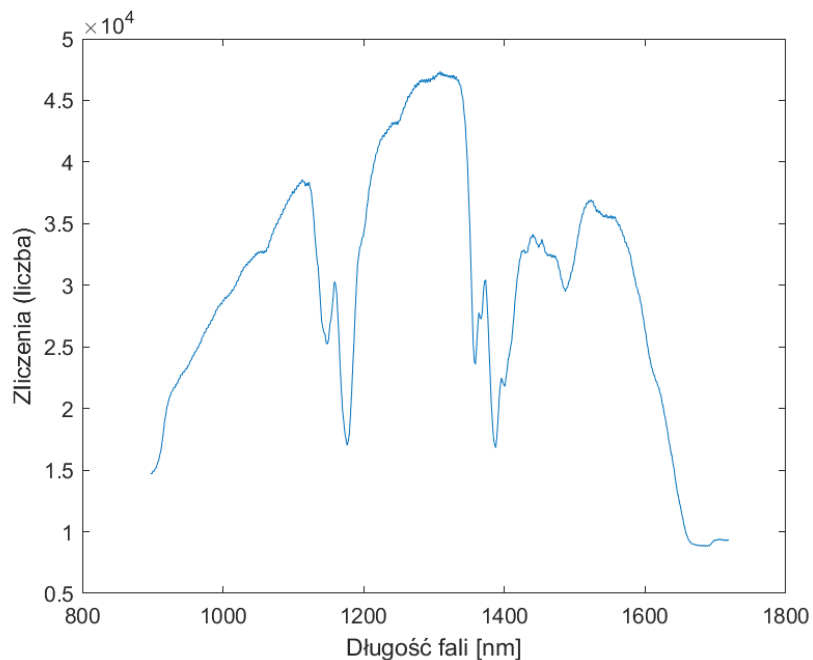
Na rysunku [Rysunek 35] przedstawiono wykres procentowej transmisji próbki z alkoholem izopropylowym.



Rysunek 35. Wykres procentowej transmisji dla próbki z alkoholem izopropylowym

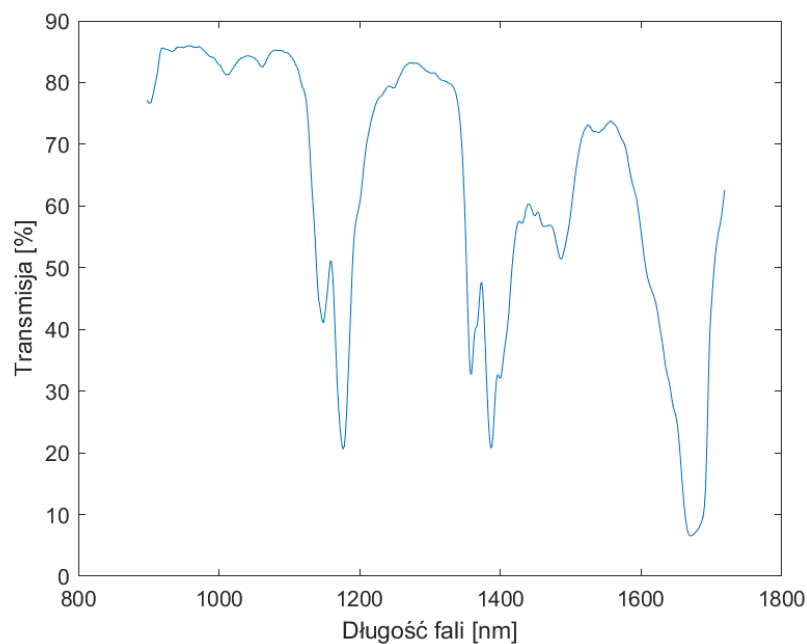
4.3.6 Próbka z DMSO

Widmo z pomiaru kuwety z DMSO umieszczonej w uchwycie przedstawiono na rysunku [Rysunek 36].



Rysunek 36. Wykres widma próbki dimetylosulfotlenku (DMSO)

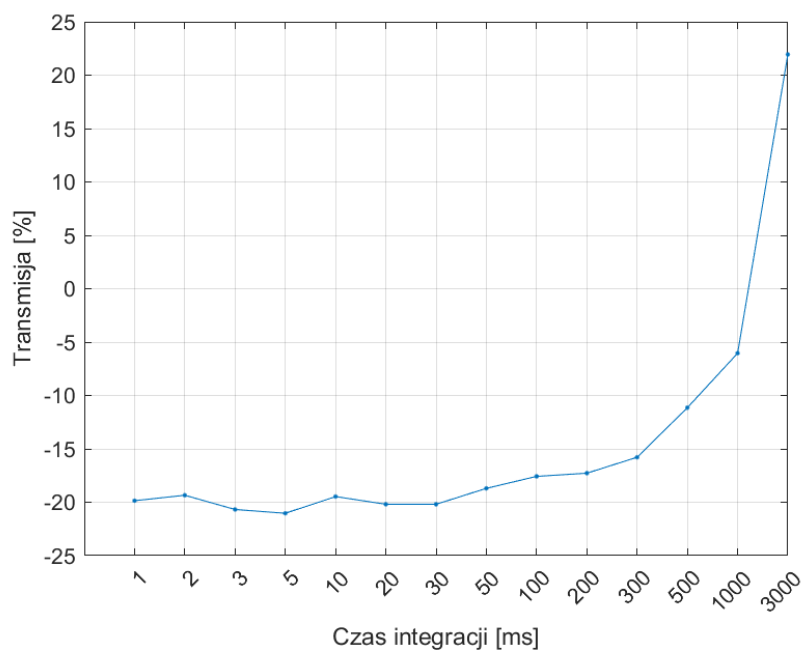
Na rysunku [Rysunek 37] przedstawiono wykres procentowej transmisji próbki z DMSO.



Rysunek 37. Wykres procentowej transmisji dla dimetylosulfotlenku (DMSO)

4.4 Wpływ czasu integracji na pomiary w trybie dark

Wpływ czasu integracji na pomiary w trybie dark pomierzono zamykając wejście spektrometru dołączoną do niego zaślepką, a następnie wykonano szereg pomiarów, zmieniając ustawienia czasu integracji na spektrometrze. Następnie wykreślono procentową transmisję dla losowo wybranej długości fali. Wyniki zaprezentowano na rysunku [Rysunek 38].

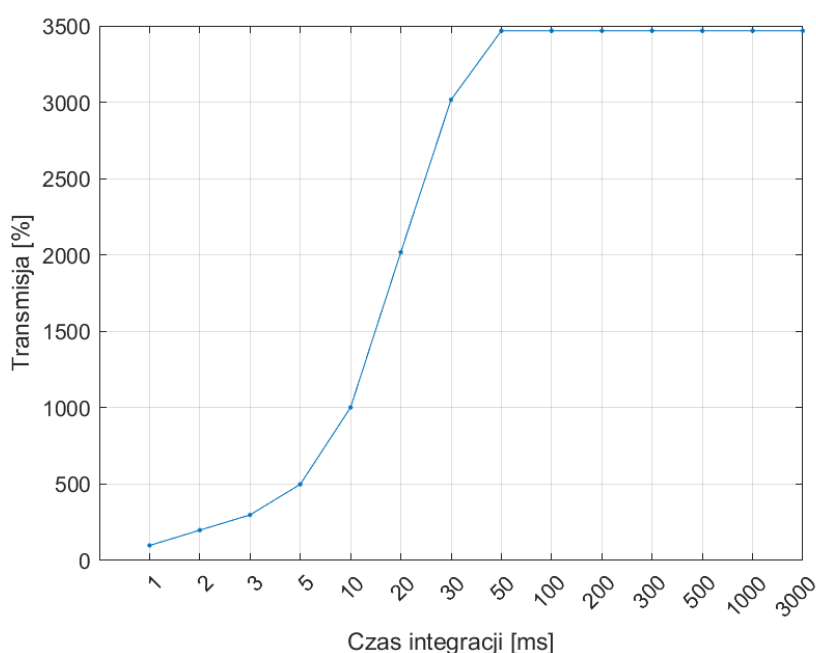


Rysunek 38. Transmisja [%] w funkcji czasu integracji dla pomiaru w trybie dark

Z pomiarów wynika, że czas integracji ma wpływ na pomiary w trybie dark. Wraz ze wzrostem czasu integracji rośnie procentowa transmisja. Czas integracji w pomiarach dark ma kluczowy wpływ na jakość i dokładność danych pomiarowych. Odpowiedni dobór czasu integracji ma istotny wpływ na minimalizację szumów i poprawę precyzji pomiarów.

4.5 Wpływ czasu integracji na pomiary referencyjne

Wpływ czasu integracji na pomiary referencyjne pomierzono wyciągając kuetę z uchwytu, a następnie wykonując szereg pomiarów, zmieniając ustawienia czasu integracji na spektrometrze. Następnie wykreślono procentową transmisję dla losowo wybranej długości fali. Wyniki zaprezentowano na rysunku [Rysunek 39].



Rysunek 39. Transmisja [%] w funkcji czasu integracji dla pomiaru referencyjnego

Z pomiarów wynika, że czas integracji ma wpływ również dla pomiarów referencyjnych. Analogicznie jak w pomiarach trybu dark, wraz ze wzrostem czasu integracji rośnie także procentowa transmisja. Zaobserwowano także, że od czasu integracji równego 50 ms transmisja osiągnęła swoje maksimum, ze względu na nasycenie się detektora w spektrometrze.

Dlatego w przypadku pomiarów referencyjnych ważne jest prawidłowe ustawienie czasu integracji tak, aby detektor nie osiągał swoich maksymalnych wartości, ponieważ może się to wiązać z częściową utratą danych przy pomiarach próbek biologicznych. Ponadto czas integracji nie powinien być zbyt mały. Przy zbyt krótkim czasie integracji detektor może mieć zbyt mało czasu na zebranie sygnału, co może prowadzić do niskiej czułości pomiarów a także do zmniejszenia poziomu sygnału do szumu (SNR).

5. KONSTRUKCJA UKŁADU POMIAROWEGO DO POMIARÓW PRÓBEK W DWÓCH WYMIARACH

Wcześniejsze pomiary próbek biologicznych wykonywano w zamontowanych na stałe uchwytach, bez możliwości regulacji dokładnego miejsca, dla którego wykonywany został pomiar. Nieruchomy uchwyt zastosowano, by zwiększyć jak najbardziej dokładność i precyzję wykonywanych pomiarów. Układ pomiarowy był odpowiedni do badania próbek jednorodnych, gdyż takie substancje zachowują jednorodną strukturę w całej swojej objętości. Problem pojawia się w momencie, gdy próbka biologiczna miałaby niejednorodną strukturę, co oznacza, że w zależności od położenia sondy pomiarowej (czoła światłowodu) od konkretnego miejsca, w którym badano próbkę, skład próbki różniłby się znacznie bądź nieznacznie od wcześniej badanego obszaru.

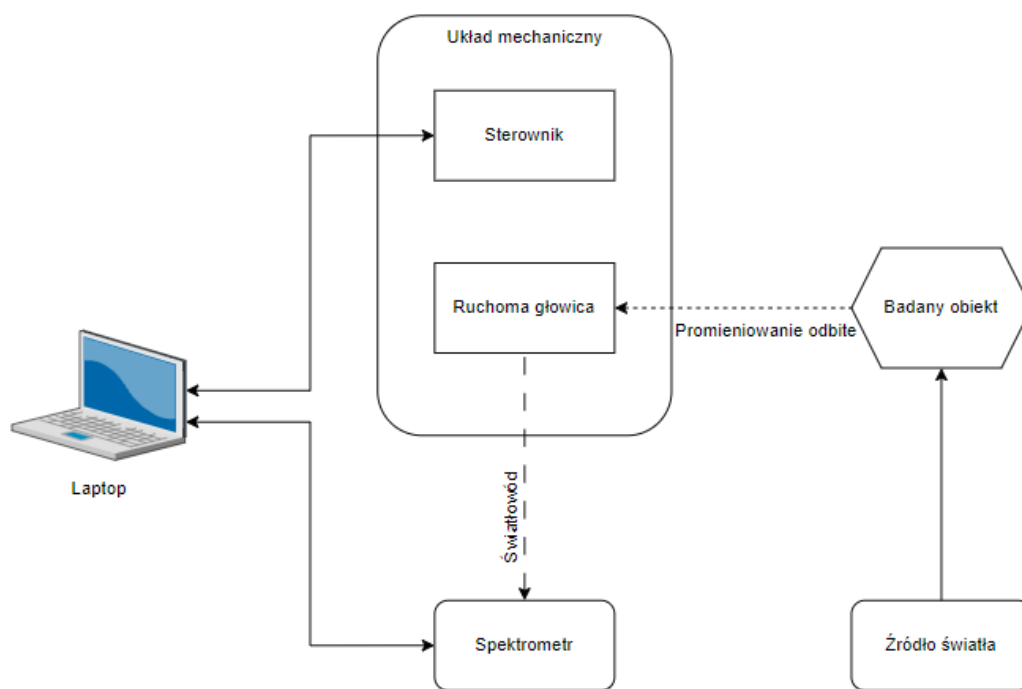
W niniejszym rozdziale przedstawiono sposób wykonywania pomiarów za pomocą światłowodu, który umieszczony został w specjalnie przygotowanym uchwycie na frezarce. Ponadto w rozdziale określono wymagania jakie układ pomiarowy powinien spełniać oraz wybrano odpowiedni sprzęt. Kluczowym elementem rozdziału jest oprogramowanie wybranej frezarki CNC oraz spektrometru, a także przygotowanie konstrukcji układu pomiarowego.

5.1 Określenie wymagań

Do pomiarów niejednorodnych próbek biologicznych potrzebne będzie skonstruowanie lub wykorzystanie gotowego rozwiązania, umożliwiającego poruszanie sondą pomiarową (czołem światłowodu) w przynajmniej dwóch kierunkach. Dodatkowo, aby pomiary były powtarzalne, konieczne jest odpowiednie sterowanie urządzeniem tak, aby możliwy był powrót do miejsc, w których wcześniej wykonywane zostały pomiary.

Ponadto urządzenie powinno umożliwiać konfigurację rozdzielczości pomiarów, by możliwe było dostosowanie kroku, co jaki wykonywać ma się pomiar oraz określać obszar, w obrębie którego sonda pomiarowa ma się poruszać.

Rysunek [Rysunek 40] przedstawia schemat ideowy docelowego układu pomiarowego.



Rysunek 40. Schemat docelowego układu pomiarowego

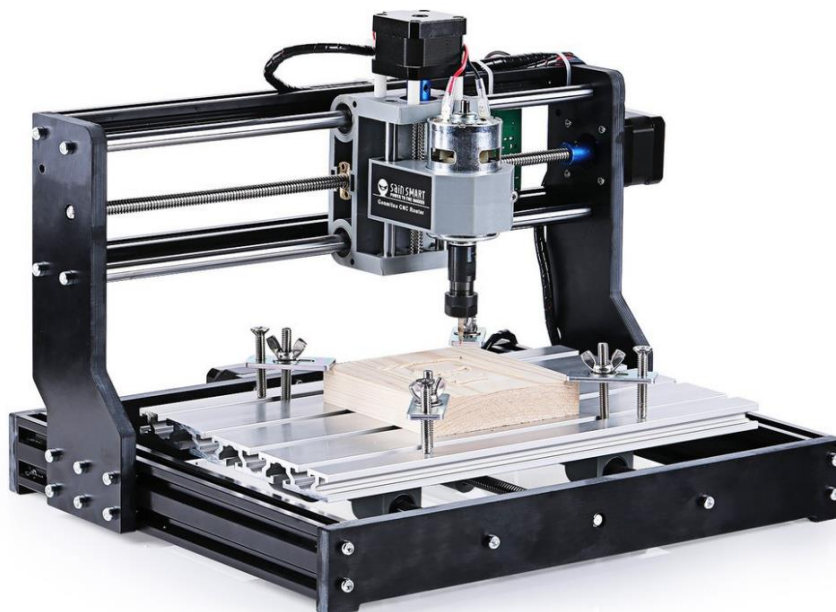
5.2 Wybór sprzętu

Urządzeniem, które może spełniać założenia projektowe jest stołowa frezarka CNC. Frezarka [25] jest to narzędzie precyzyjne, służące zazwyczaj do obróbki metalu bądź innych materiałów. Obróbka polega na skrawaniu materiału za pomocą poruszającego się ruchem obrotowym ostrza – frezu. Frezarka CNC [26] – ang. Computerized Numerical Control oznacza, że urządzeniem można sterować za pomocą komputera.

Frezarkę, aby była spełniała postawione wcześniej wymagania należy dostosować – do stanowiska pomiarowego nie jest potrzebny frez. Zamiast tego należy znaleźć rozwiązanie, które umożliwi zamontowanie czoła światłowodu, w taki sposób, aby móc poruszać głowicą frezarki.

Skorzystanie z gotowego elementu jakim jest frezarka skróci czas potrzebny do zestawienia stanowiska pomiarowego. Niezbędne będzie jedynie przygotowanie mocowania oraz oprogramowanie urządzenia w taki sposób, aby umożliwiała przeprowadzanie założonych pomiarów i współpracowało z wykorzystywanym spektrometrem.

Na sprzęt, który spełnia wymagania, wybrano frezarkę CNC Genmitsu 3018, przedstawioną na rysunku [Rysunek 41]. Wybrano ją ze względu na jej bliską dostępność w laboratorium oraz ze względu na łatwość przygotowania oprogramowania do jej obsługi.



Rysunek 41. Frezarka CNC Genmitsu 3018 [27]

Frezarka charakteryzuje się obszarem roboczym o wielkości 300 mm x 180 mm, co jest wystarczające do zastosowań w niniejszej pracy. Ponadto, sprzęt posiada zintegrowany sterownik, który umożliwia łatwe sterowanie urządzeniem z poziomu komputera PC.

5.3 Oprogramowanie układu pomiarowego

5.3.1 Oprogramowanie GRBL

Wybrana frezarka charakteryzuje się wykorzystywaniem darmowego oprogramowania, jakim jest GRBL [28]. GRBL jest to oprogramowanie, które umożliwia kontrolę frezarki za pomocą ustandaryzowanych komend. Komendy służą między innymi do poruszania silnikami frezarki w każdej z osi, do ustawiania parametrów frezarki, a także do konfiguracji szybkości poruszania się oraz kroku, o jaki ma się poruszyć głowica w danym kierunku. Wymienione komendy można wysyłać za pomocą portu szeregowego i kabla USB, podłączonego do komputera [29].

5.3.2 Wybór języka programowania

Aby móc sterować urządzeniem, należy przygotować program, który umożliwi komunikację z frezarką oraz będzie posiadał wszystkie możliwości, jakie potrzebne są w projekcie.

Do przygotowania oprogramowania wykorzystano język programowania Python. Język ten jest jednym z popularniejszych języków programowania na świecie [30]. Jest znany ze swojej prostoty, czytelności oraz uniwersalności. Posiada bogate biblioteki, które umożliwiają np. stworzenie prostego graficznego interfejsu oraz łatwą komunikację za pomocą portu

szeregowego. Zastosowanie tego języka skróci czas tworzenia programu oraz pozwoli skupić się na istotniejszych elementach, jakie należy przygotować.

5.3.3 Wybór i wykorzystanie dostępnych bibliotek

W celu przygotowania oprogramowania konieczne było przeprowadzenie szczegółowego rozeznania w dostępnych rozwiązaniach programistycznych. Proces ten obejmował poszukiwanie odpowiednich bibliotek, które mogłyby wspierać kluczowe aspekty projektu, takie jak obsługa interfejsu graficznego, kontrola spektrometru oraz komunikacja z frezarką CNC przez port szeregowy.

Biblioteka do obsługi interfejsu graficznego

Do zaprojektowania interfejsu graficznego wykorzystano bibliotekę Tkinter [31], dostępną dla języka programowania Python. w bibliotece zawarto gotowe rozwiązania, które umożliwiły przygotowanie interfejsu graficznego, między innymi:

- pola do wprowadzania tekstu,
- przyciski,
- pola wyświetlające tekst,
- paski postępu,
- ramki ułatwiające podział interfejsu na przejrzyste sekcje,
- metody do komunikacji z systemem w celu zapisu/odczytu plików.

Biblioteka do obsługi komunikacji z frezarką CNC

Do obsługi komunikacji z frezarką CNC wykorzystano bibliotekę pySerial [32], dostępną dla języka Python. w bibliotece zawarto wszystkie narzędzia potrzebne do komunikacji, między innymi:

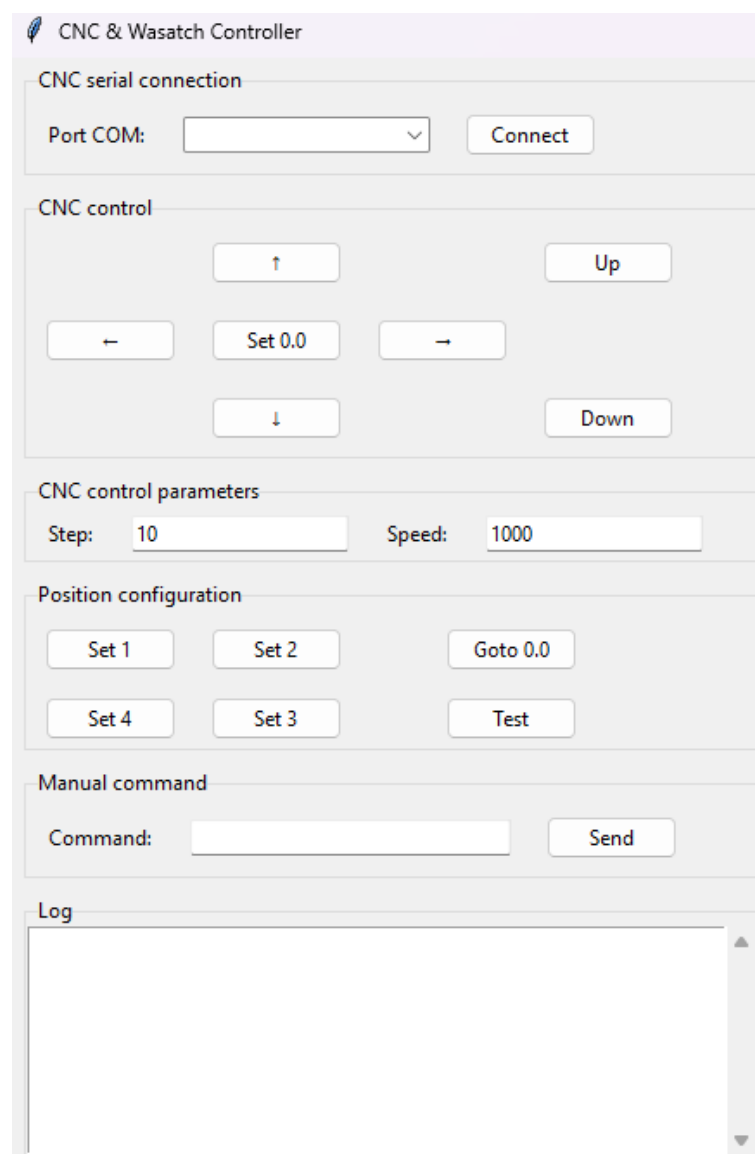
- wybór portu szeregowego do którego podłączono urządzenie,
- możliwość ustawienia parametrów połączenia, takich jak szybkość transmisji,
- możliwość odbierania i nadawania komunikatów z i do urządzenia.

Biblioteka do komunikacji ze spektrometrem

Do komunikacji ze spektrometrem NIR-1 wykorzystano API Wasatch.PY [33] przygotowane przez producenta spektrometru. w API zawarto wszelkie narzędzia potrzebne do komunikacji i sterowania urządzeniem. Ponadto w bibliotece producenta znalazł się także przykładowy kod, umożliwiający wykonanie wielu predefiniowanych funkcji oraz dobrze opisana dokumentacja, ułatwiająca zapoznanie się ze sprzętem i zestawienie układu pomiarowego.

5.3.4 Interfejs graficzny programu - frezarka

Po wybraniu frezarki i języka programowania przystąpiono do stworzenia prostego interfejsu graficznego, który spełni założenia projektowe. Na rysunku [Rysunek 42] przedstawiono przygotowany widok interfejsu.



Rysunek 42. Zrzut interfejsu graficznego programu do obsługi frezarki

W oknie programu znalazło się miejsce między innymi na ustawienia połączenia z frezarką – wybór portu szeregowego, pod jakim urządzenie będzie podłączone, a także możliwość połączenia/rozłączenia z urządzeniem.

Kluczowym elementem programu jest możliwość sterowania frezarką za pomocą przycisków, ustawienie ilości kroków, jakie wykonać ma głowica frezarki oraz prędkość, z jaką ma się poruszać.

W programie umieszczono również opcje ustawienia wierzchołków pola (przyciski Set 1-4), w obrębie którego poruszać ma się głowica przy automatycznych pomiarach, a także przycisk „Test”, który wykonuje ruch po ustawionych wierzchołkach, aby przed pomiarami potwierdzić poprawnie wybraną strefę.

Dodatkowo przewidziano możliwość wpisywania ręcznej komendy, w razie gdyby konieczne było wykonanie jakiejś niestandardowej komendy, nieobsługiwanej bezpośrednio w programie, ale obsługiwanej przez sterownik frezarki.

W dolnej części programu swoje miejsce znalazło przewijane pole tekstowe, w którym wypisywane będą wszystkie komendy, użyte do komunikacji z frezarką, aby w każdej chwili można było sprawdzić, co było wysyłane do frezarki.

Wybrano kilka istotnych fragmentów kodu, kluczowych do zaprojektowania interfejsu. Cały kod programu zawiera Dodatek C: Kod programu pomiarowego.

Do przygotowania podsekcji w każdej z części oprogramowania, aby zachować przejrzystość interfejsu graficznego wykorzystano kod:

```
# Left and right part of window
self.left_frame = ttk.PanedWindow(self.root)
self.left_frame.grid(row=0, column=0, sticky='n')

self.right_frame = ttk.PanedWindow(self.root)
self.right_frame.grid(row=0, column=1, sticky='n')
```

W kodzie zawarte są informacje o wierszu (*row*) oraz kolumnie (*column*), w jakich znajdować się mają ramki.

Do przycisków interfejsu graficznego wykorzystano kod:

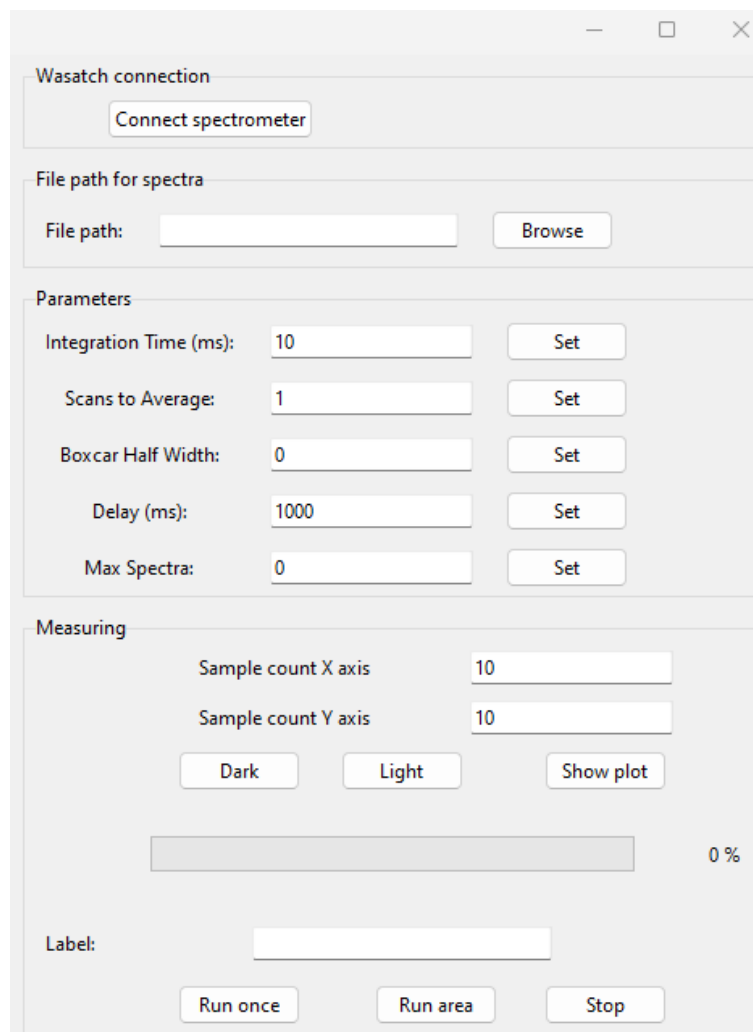
```
# Movement buttons
button = ttk.Button(self.control_frame, text='\u2190',
                    command=lambda d='\u2190': self.move(d)) # Left
button.grid(row=1, column=0, padx=10, pady=10)
```

W kodzie, analogicznie jak dla ramek, znajduje się informacja o położeniu przycisku. Dodatkowo do przycisku przypisano metodę uruchamianą podczas wciśnięcia przycisku (tu: *self.move(d)*) oraz tekst wyświetlany w interfejsie.

Po oprogramowaniu frezarki CNC i przetestowaniu działania wszystkich funkcjonalności, przystąpiono do przygotowania obsługi wykonywania pomiarów za pomocą spektrometru.

5.3.5 Interfejs graficzny programu - spektrometr

Kolejnym krokiem było zaprojektowanie interfejsu graficznego do części sterowania spektrometrem i konfiguracji pomiarów. Na rysunku [Rysunek 43] przedstawiono zrzut ekranu przygotowanego interfejsu.



Rysunek 43. Zrzut interfejsu graficznego programu do obsługi spektrometru

Okno programu podzielono na cztery części:

- Wasatch connection – służącą do konfiguracji połączenia z urządzeniem,
- File path for spectra = do ustawiania miejsca zapisu odczytywanych danych,
- Parameters – do ustawiania podstawowych parametrów spektrometru,
- Measuring – do wykonywania pomiarów w różnych trybach pracy.

W programie znalazło się miejsce na ustawienia najważniejszych parametrów spektrometru i wykonywanych pomiarów, takich jak czas integracji, ilość próbek używanych do uśredniania czy opóźnienie dla pomiarów w trybie ciągłym.

Ponadto program umożliwiać będzie wykonywanie pomiarów w kilku trybach. Oprócz pomiarów typu dark i light, program umożliwi pomiar pojedynczy oraz pomiary w trybie ciągłym, z możliwością ustawienia ilości próbek jakie spektrometr ma zebrać. Dodatkowo pomiar w trybie ciągłym został wyposażony w możliwość ustawienia ilości próbek dla osi X i Y, jeśli pomiary wykonywane będą dla dwuwymiarowej powierzchni.

Aby urozmaicić interfejs do oprogramowania dodano pasek postępu dla pomiarów w trybie dwuwymiarowym:

```

# Progress bar
self.progress_bar = ttk.Progressbar(
    self.wasatch_measure_frame, orient="horizontal",
    length=300, mode="determinate")
self.progress_bar.grid(row=3, column=0, columnspan=4,
                        padx=10, pady=5)

```

Pasek postępu został wprowadzony między innymi w celu informowania użytkownika o postępach w pomiarach. Ponadto dzięki jego wprowadzeniu, gdyby program z jakiegoś celu przestał funkcjonować (np. zawiesił się), użytkownik od razu zauważył brak postępu w pomiarach.

5.3.6 Sterowanie frezarką CNC

Sterowanie frezarką CNC w projekcie wymagało opracowania modułu odpowiedzialnego za zarządzanie połączeniem oraz komunikacją z urządzeniem za pomocą portu szeregowego. Główne funkcje tego modułu obejmują nawiązywanie połączenia z frezarką, wysyłanie poleceń GRBL oraz monitorowanie stanu urządzenia, w tym oczekiwanie na zakończenie ruchu narzędzia. Poniżej przedstawiono fragmenty kodu. Cały kod programu zawiera Dodatek C: Kod programu pomiarowego.

Do zestawienia połączenia z urządzeniem wykorzystano kod:

```

def connect_cnc(self, port):
    try:
        self.serial_port = serial.Serial(
            port, baudrate=115200, timeout=1)
        self.connected = True
        return "Connected to {}".format(port)
    except Exception as e:
        return "Error: {}".format(e)

```

Kod umożliwia połączenie z urządzeniem na wybranym za pomocą interfejsu graficznego porcie szeregowym, z ustawioną na stałe prędkością.

Dodatkowo kluczowym fragmentem modułu jest kod odpowiedzialny za wysyłanie komend do urządzenia:

```

def send_gcode(self, command):
    if self.serial_port and self.connected:
        self.serial_port.write((command + '\n').encode())
        return f"Command sent: {command}"
    return "Not connected to CNC"

```

Kod umożliwia wysyłanie do frezarki komend przygotowanych za pomocą części graficznej oprogramowania.

5.3.7 Sterowanie spektrometrem NIR-1

Do sterowania spektrometrem NIR-1 wykorzystano dedykowane oprogramowanie dostarczone przez producenta, które zapewnia pełną obsługę urządzenia i umożliwia przeprowadzanie pomiarów spektrometrycznych [33]. Oprogramowanie to jest dobrze zoptymalizowane pod kątem specyfikacji technicznych spektrometru i pozwala na precyzyjną kontrolę parametrów pracy, takich jak ustawienia długości fali, czas integracji, czy ilość zbieranych próbek.

Ponadto producent udostępnił szereg gotowych bibliotek programistycznych, które ułatwiają integrację spektrometru z własnym oprogramowaniem użytkownika. Biblioteki te zawierają niezbędne funkcje do nawiązywania połączenia i komunikacji ze spektrometrem za pomocą przewodu USB, co znacząco upraszcza proces implementacji obsługi urządzenia w środowisku projektowym. Dzięki temu nie było potrzeby pisania własnych, niskopoziomowych procedur komunikacji, co zredukowało czas potrzebny na integrację spektrometru z resztą układu pomiarowego.

5.3.8 Sterowanie wykonywaniem pomiarów

W projekcie kluczowym aspektem było zintegrowanie działania frezarki CNC i spektrometru w taki sposób, aby pomiar widmowy był wykonywany precyzyjnie w wyznaczonych punktach, po zakończeniu ruchu głowicy pomiarowej. Wymagało to synchronizacji obydwu urządzeń, tak aby spektrometr wykonywał pomiar tylko wtedy, gdy frezarka znajdowała się w odpowiedniej pozycji i zakończyła ruch. Synchronizacja ta była możliwa dzięki opracowaniu kodu, który zarządzał ruchem frezarki oraz automatycznie inicjował pomiar spektrometru.

Zaprojektowany kod pozwalał na płynną współpracę frezarki z układem pomiarowym spektrometru, zapewniając dokładność pomiaru i eliminując ewentualne zakłócenia wynikające z ruchu frezarki. Poniżej przedstawiono fragment funkcji *measure_and_move*.

```
# Move cnc
for i in range(self.samples_count_x):
    new_x = self.user_positions['1']['X'] + i * step_x
    for j in range(self.samples_count_y):
        # New position Y
        new_y = self.user_positions['1']['Y'] + j * step_y
        # Move cnc to new position
        move_command = f'G1 X-{new_x} Y-{new_y} F{self.get_speed()}'
        self.serial.send_gcode(move_command)
        self.waitForCNC()
        max_step = max(step_x, step_y)
        self.measureDelayFromSteps(max_step)
        current_measure += 1
        finished = self.wasatch.run("%2.4f, %2.4f" % (new_x, new_y))
        progress = int((current_measure / measure_count) * 100)
        self.update_progress(progress)
```

Fragment zawiera kod odpowiedzialny za obliczenie kroku o jaki przesunąć ma się głowica pomiarowa w obu osiach, instrukcję poruszania głowicą, oczekiwanie na zakończenie ruchu, instrukcję wykonania pomiaru przez spektrometr oraz aktualizację paska postępu. Cały kod programu zawiera Dodatek C: Kod programu pomiarowego.

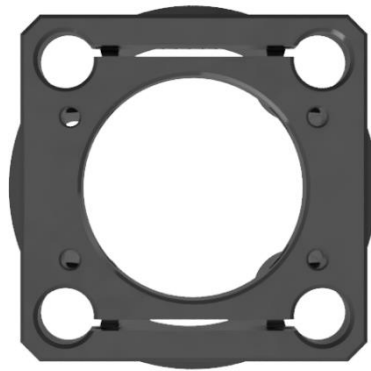
5.4 Projekt ruchomej części układu pomiarowego

Frezarka CNC domyślnie wyposażona jest w ruchomy silnik DC, do którego mocowany jest frez. do projektu układu pomiarowego frez i silnik należało zastąpić światłowodem. Przygotowano do tego celu adapter, umożliwiający zamocowanie światłowodu w uchwycie przeznaczonym na silnik. Rysunek [Rysunek 44] przedstawia model 3D adaptera.



Rysunek 44. Adapter mocujący światłowód do frezarki CNC

Model 3D przygotowano tak, aby mógł zostać wydrukowany na drukarce 3D oraz aby umożliwiał bezproblemowy montaż adaptera w uchwycie frezarki oraz światłowodu w adapterze. w modelu przewidziano także otwory na śruby montażowe do zablokowania głowicy światłowodu na konkretnej wysokości względem badanej próbki. Rysunek [Rysunek 45] przedstawia rzut modelu adaptera od spodu.



Rysunek 45. Adapter mocujący światłowód do frezarki CNC. Widok od dołu.

5.5 Projekt uniwersalnego mocowania do elementów pomiarowych

Aby zestawić układ pomiarowy, wykorzystano metalową płytę z gwintowanymi otworami na śruby mocujące. Na płycie umieszczono spektrometr, źródło światła oraz rozmieszczono światłowód tak, aby nie był nadmiernie zagięty. Przygotowano projekty dwóch elementów mocujących, umożliwiających stały montaż układu pomiarowego na metalowej płycie. Rysunek [Rysunek 46] przedstawia wymienione wcześniej modele 3D.



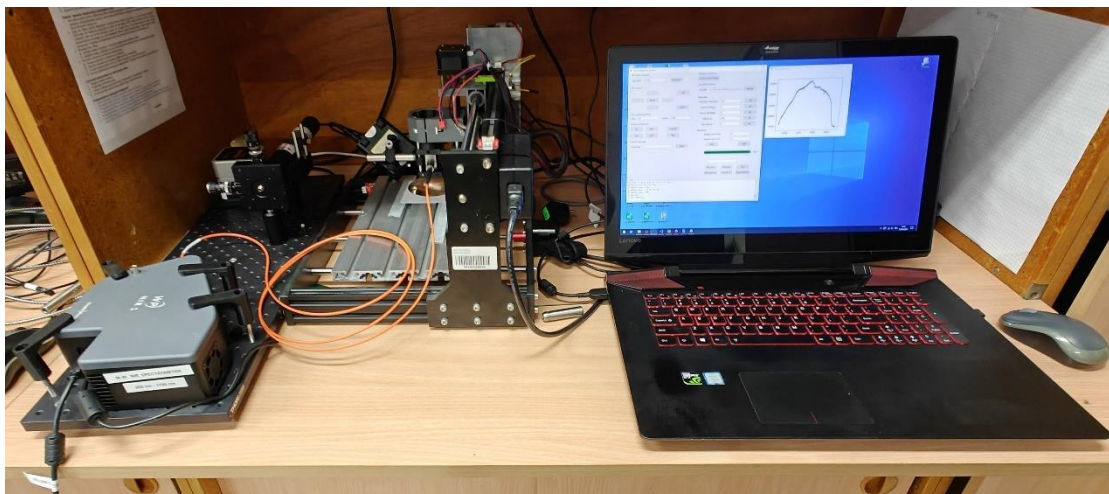
Rysunek 46. Mocowania do wysokich i niskich elementów

Modele przygotowano w dwóch wersjach – do montażu wysokich elementów oraz do montażu niskich elementów układu pomiarowego. w mocowaniach umieszczono otwory, umożliwiające przełożenie różnej długości śrub, dzięki czemu możliwa jest dodatkowa regulacja wysokości montowanych urządzeń. Śruby wkręcane w metalową płytę wraz z przygotowanym mocowaniem umożliwią zamocowanie urządzeń do płyty. Modele 3D przygotowano tak, aby można je było wydrukować na drukarce 3D, podobnie jak uchwyt frezarki CNC.

5.6 Zmontowany układ pomiarowy

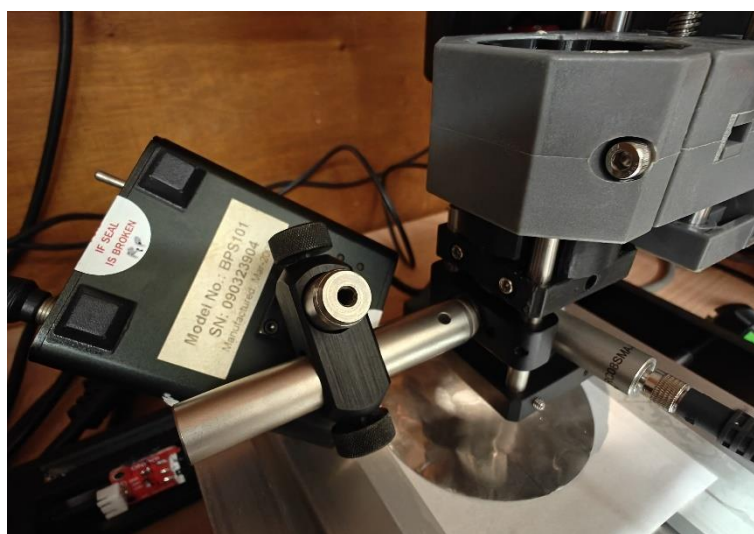
Układ pomiarowy został zmontowany zgodnie z założeniami projektowymi, a jego pełna konfiguracja została przedstawiona na rysunku [Rysunek 47]. Widoczny na nim system

obejmuje kluczowe elementy pomiarowe oraz komputer PC, na którym uruchomiono oprogramowanie sterujące układem pomiarowym.



Rysunek 47. Zmontowany układ pomiarowy wraz z laptopem

Warto również zwrócić uwagę na konstrukcję samego układu optycznego, który znajduje się na rysunku [Rysunek 48]. Przedstawiono na nim zamontowaną głowicę pomiarową wraz ze źródłem światła z wykorzystaniem zaprojektowanego i wydrukowanego na drukarce 3D uchwytu.



Rysunek 48. Zamontowana głowica pomiarowa wraz ze źródłem światła

Na rysunku [Rysunek 49] przedstawiono spektrometr zamontowany przy użyciu wcześniej zaprojektowanych i wykonanych na drukarce 3D uchwytów mocujących.



Rysunek 49. Zamontowany na płycie montażowej spektrometr NIR-1

Po zakończeniu montażu, przeprowadzono wstępne testy układu pomiarowego, które potwierdziły jego pełną funkcjonalność oraz gotowość do przeprowadzenia pomiarów.

6. POMIARY PRÓBEK BIOLOGICZNYCH

W niniejszym rozdziale wybrano konkretne próbki biologiczne, które wykorzystano do przeprowadzenia pomiarów. Omówiono wybrane próbki, a następnie dokonano pomiarów ich widm, z wykorzystaniem skonstruowanego wcześniej do tego celu układu pomiarowego.

6.1 Przegląd wybranych próbek biologicznych

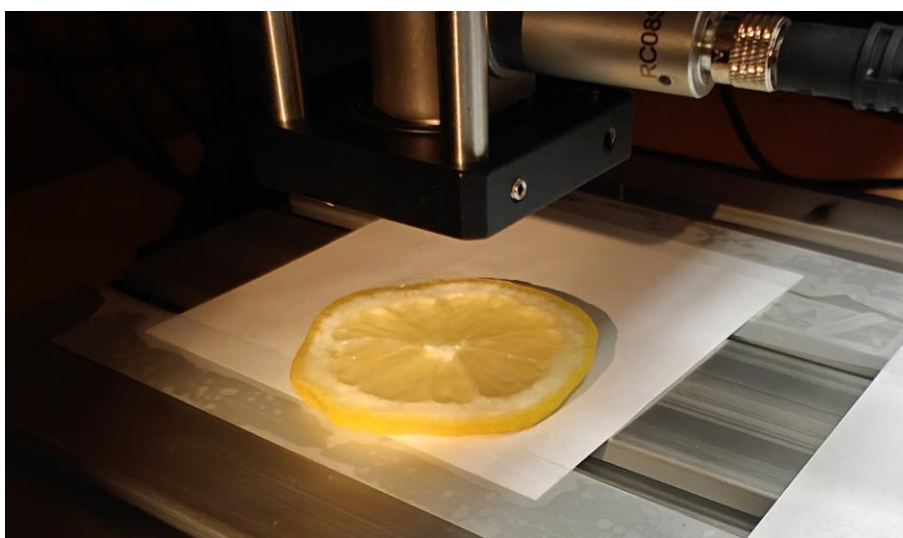
W ramach badania wybrano dwie próbki biologiczne do analizy spektroskopowej: cytrynę oraz boczek. Wybór tych próbek był motywowany ich różnorodnymi właściwościami chemicznymi i strukturami, które mogą dostarczyć cennych informacji na temat zastosowania spektroskopii w badaniu materiałów biologicznych.

Cytryna została wybrana ze względu na dużą zawartość wody w swoim składzie. Boczek z kolei, reprezentuje materiał biologiczny o innej charakterystyce chemicznej - jest bogaty w białka oraz lipidy.

Obie próbki zostały poddane pomiarom spektroskopowym w celu porównania ich widm oraz zrozumienia różnic w absorpcji promieniowania elektromagnetycznego, co jest kluczowe dla interpretacji wyników i dalszych badań nad właściwościami biologicznymi analizowanych materiałów.

6.2 Próbka 1 – cytryna

W ramach analizy spektroskopowej, cytryna została przygotowana i zmierzona w sposób umożliwiający dokładne badanie jej właściwości optycznych. Cytrynę pokrojono na plastry o grubości około 3 mm, a następnie umieszczono w skonstruowanym układzie pomiarowym. Głowicę pomiarową nakierowano na próbkę, poruszając głowicą frezarki za pomocą przygotowanego oprogramowania. Na rysunku [Rysunek 50] pokazano przygotowaną próbkę, umieszczoną w układzie pomiarowym, oświetloną za pomocą źródła światła.



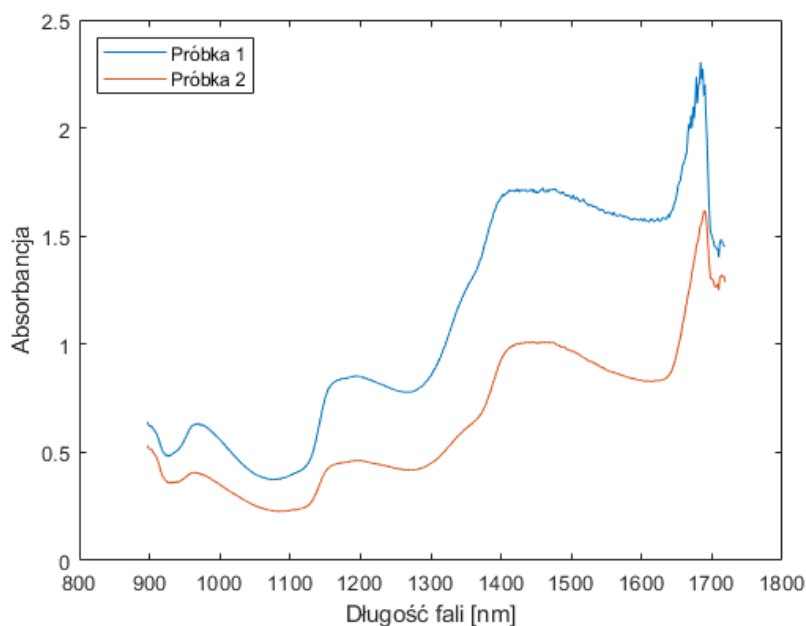
Rysunek 50. Próbka cytryny umieszczona w układzie pomiarowym

Pomiary zostały przeprowadzone w płaszczyźnie dwuwymiarowej na fragmencie o wymiarach 20 x 20 mm, z zagęszczeniem pomiarów co 1 mm. Czas integracji dla każdego pomiaru wynosił 1000 ms.

6.2.1 Pomiar środka cytryny

Środek cytryny został umieszczony w centrum powierzchni pomiarowej, co pozwoliło na uzyskanie reprezentatywnych danych dla całego obszaru.

Po zebraniu danych, wybrano dwa punkty z obszaru pomiarowego, aby wykreślić ich widma absorbancji. Rysunek [Rysunek 51] przedstawia te widma.



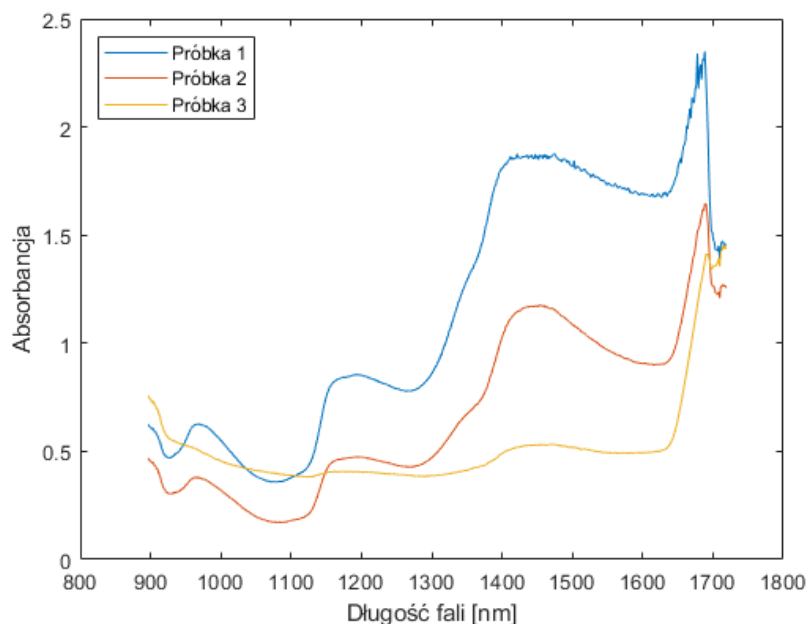
Rysunek 51. Absorbancja plastra cytryny dla dwóch próbek

Widmo opisane jako „Próbka 1” to widmo „rdzenia” cytryny, czyli jej centralnej części, natomiast widmo opisane jako „Próbka 2” to widmo części cytryny, zawierającej największą procentową ilość wody w strukturze.

Jak wynika z widm absorbancji, część cytryny zawierająca większą ilość wody i wartości mineralnych znacznie mniej absorbuje od rdzenia cytryny.

6.2.2 Pomiary krawędzi cytryny

Plaster cytryny umieszczono na powierzchni pomiarowej tak, aby jego skórka była w centralnej części. z tego względu wykonano pomiary częściowo dla plastra cytryny i częściowo dla pustej powierzchni. Po zebraniu danych wykreślono widma absorbancji dwóch punktów z obszaru pomiarowego. Rysunek [Rysunek 52] przedstawia te widma:



Rysunek 52. Absorbancja plastra cytryny dla dwóch próbek

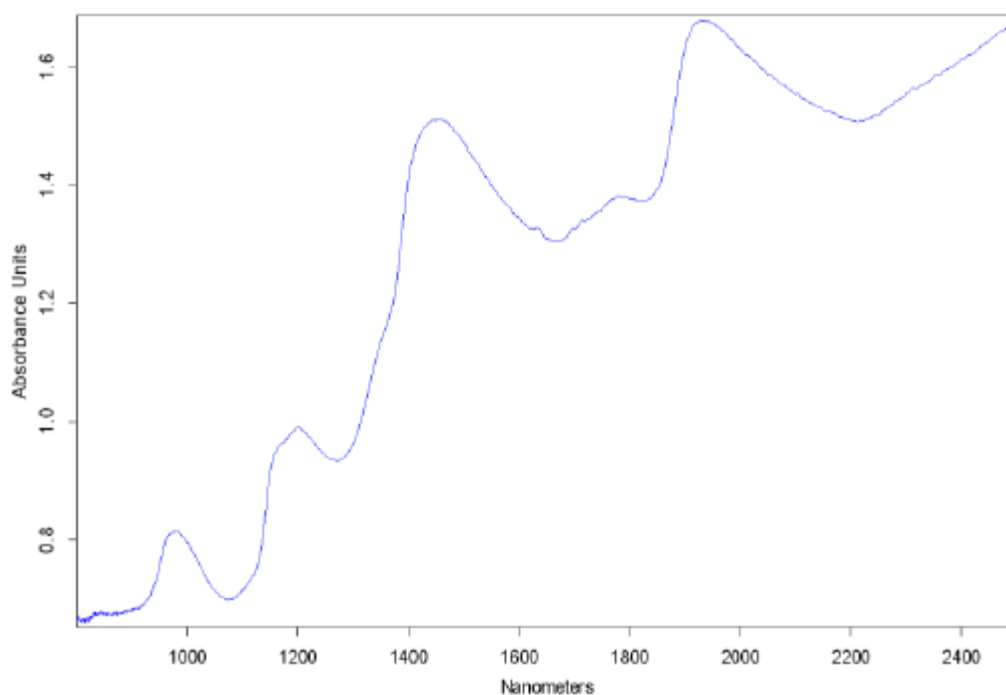
Widmo opisane jako „Próbka 1” to widmo skórki cytryny, natomiast widmo opisane jako „Próbka 2” to widmo części cytryny, zawierającej największą procentową ilość wody w strukturze. Dodatkowo wykreślono poglądowo widmo absorbancji dla części, w której nie była umieszczona próbka cytryny, tylko podkładka z aluminium – opisano ją jako „Próbka 3”.

Jak wynika z widm absorbancji, w tym przypadku również część cytryny zawierająca większą ilość wody i minerałów absorbuje znacznie słabiej od skórki cytryny. Natomiast absorbancja próbki, w której nie umieszczono plastra cytryny, tłumy w całym zakresie na zbliżonym poziomie. Wyjątkiem są miejsca skrajne, w których detektor spektrometru charakteryzuje się zmniejszoną czułością, dlatego nie będą analizowane.

6.2.3 Wnioski i porównania

Zarówno skórka, jak i rdzeń cytryny charakteryzują się wyższym poziomem absorbancji w porównaniu do pozostałych części owocu. Na wykresach widm absorbancji można zauważyć regiony, w których absorbancja osiąga lokalne maksima. Te maksima występują w kilku zakresach długości fali, tj. około 980 nm, 1200 nm oraz 1450 nm, które są charakterystyczne dla wody. Biorąc pod uwagę, że cytryna zawiera około 89% wody [34], obecność tych maksimów jest zgodna z oczekiwaniami i może świadczyć o poprawności i skuteczności przeprowadzonych pomiarów.

Ponadto autorzy artykułu [35] przeprowadzili pomiary widma absorbancji pomarańczy o odmianie „Walencja”, które przedstawiono na rysunku [Rysunek 53].

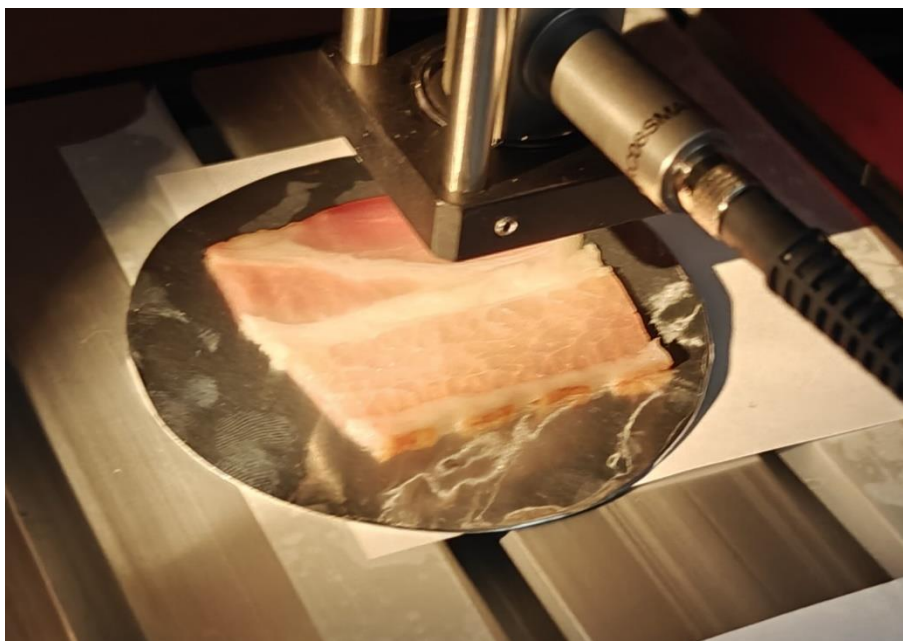


Rysunek 53. Widmo absorpcji pomarańczy „Walencia” [35]

Z wykresu widma absorpcji pomarańczy można odczytać podobne charakterystyczne maksima wartości absorpcji, jak w przypadku cytryny. Może to również potwierdzać poprawność pomiarów przeprowadzonych za pomocą skonstruowanego układu pomiarowego, ponieważ zarówno pomarańcza, jak i cytryna są cytrusami i posiadają zbliżone parametry. Dodatkowo, wykres widma absorpcji pomarańczy pokazuje maksimum absorpcji w okolicy 1980 nm, co również jest charakterystyczne dla wody. Jednakże, ze względu na to, że zakres ten wychodzi poza długości fal, które był w stanie zmierzyć detektor w spektrometrze, nie mogło on zostać porównany ani zmierzony.

6.3 Próbką 2 – boczek

W celu analizy spektroskopowej zakupiono boczek w plastrach o grubości około 1,5 mm. Następnie plaster przycięto i umieszczono w skonstruowanym układzie pomiarowym, analogicznie jak dla próbki cytryny. Na rysunku [Rysunek 54] przedstawiono próbkę w układzie pomiarowym.

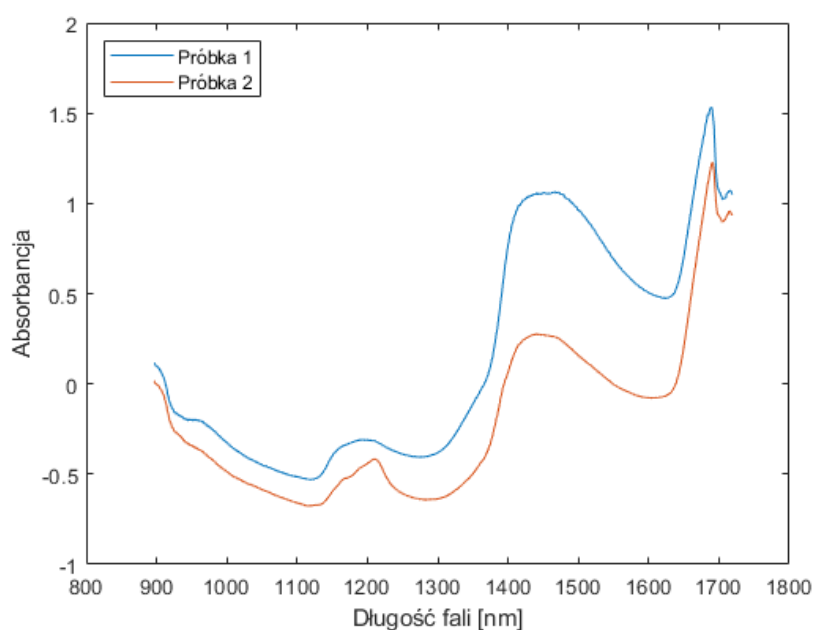


Rysunek 54. Próbka boczku umieszczona w układzie pomiarowym

Pomiary, analogicznie jak dla cytryny, zostały przeprowadzone w płaszczyźnie dwuwymiarowej na fragmencie o wymiarach 20 x 20 mm, z zagęszczeniem pomiarów co 1 mm. Czas integracji dla każdego pomiaru wynosił 200 ms, gdyż przy wyższych wartościach detektor ulegał nasyceniu.

6.3.1 Pomiar plastra boczku

Plaster boczku umieszczono w obszarze pomiarowym, a następnie wykonano szereg pomiarów. Na rysunku [Rysunek 55] przedstawiono wykres absorpcji dwóch próbek najbardziej różniących się od siebie.



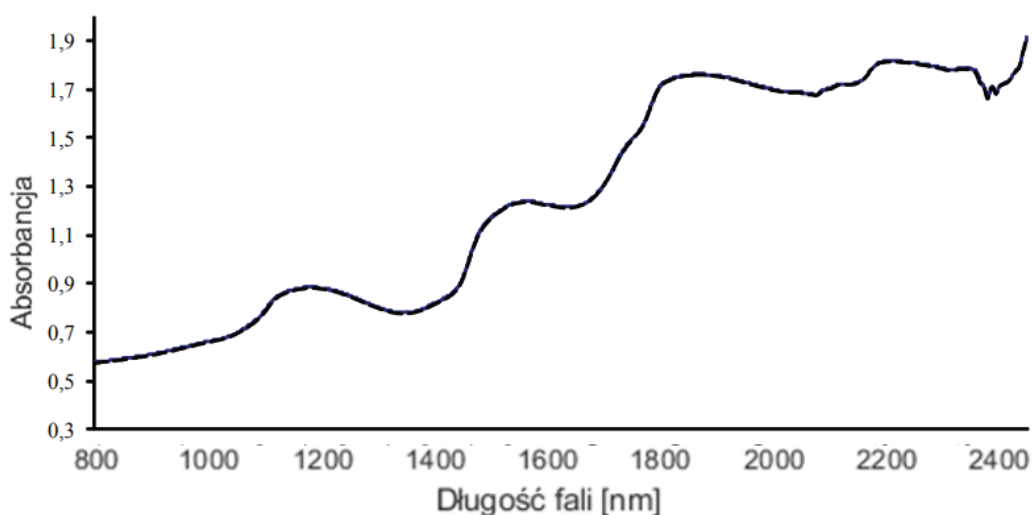
Rysunek 55. Absorbancja wybranych próbek pomiarowych boczku

Widmo absorbancji podpisane jako „Próbka 1” to część boczku zawierająca w swojej strukturze większą część białka, a mniejszą część lipidów. Próbka opisana jako „Próbka 2” to próbka zawierająca w swojej strukturze głównie lipidy.

6.3.2 Wnioski i porównania

Na podstawie widm absorbancji, próbka zawierająca większą ilość lipidów wykazuje mniejszą absorbancję w całym mierzonym zakresie. Na wykresach można zauważyć regiony, w których absorbancja osiąga lokalne maksima. Maksima te występują w kilku zakresach długości fali, tj. około 980 nm, 1210 nm, 1450 nm oraz 1700 nm, co jest charakterystyczne dla lipidów [36]. Biorąc pod uwagę, że boczek zawiera około 36% tłuszczu [37], obecność tych maksimów, podobnie jak w próbkach cytryny, jest zgodna z oczekiwaniami i świadczy o poprawności oraz skuteczności przeprowadzonych pomiarów.

Ponadto autor artykułu [38] przeprowadził badania na próbkach boczku, przedstawiając widmo absorbancji, które ukazano na rysunku [Rysunek 56].



Rysunek 56. Widmo absorbancji boczku [38]

Z widma absorbancji boczku można odczytać podobne charakterystyczne maksima absorbancji, jak w przypadku pomiarów wykonanych za pomocą skonstruowanego układu. Może to dodatkowo potwierdzać poprawność wykonanych pomiarów.

7. ALGORYTMY I PRZETWARZANIE DANYCH POMIAROWYCH

Współczesne technologie pomiarowe otwierają nowe możliwości w analizie próbek biologicznych. Uzyskiwane w ten sposób dane charakteryzują się wysoką złożonością, co wynika z obecności licznych i nakładających się na siebie sygnałów pochodzących od różnych składowych komponentów badanych próbek. Aby wydobyć z nich istotne informacje, niezbędne jest zastosowanie zaawansowanych algorytmów i metod przetwarzania danych.

W niniejszym rozdziale omówiono kluczowe aspekty związane z przetwarzaniem danych pomiarowych w kontekście spektrometrii NIR. Omówiono algorytmy, które odgrywają ważną rolę w analizie sygnałów spektrometrycznych. Ponadto skupiono się na przetwarzaniu widm pomiarowych oraz zastosowano analizę składowych głównych (PCA) na wcześniej wykonywanych pomiarach.

7.1 Algorytmy do przetwarzania danych pomiarowych

Zastosowanie algorytmów w przetwarzaniu danych pomiarowych jest kluczowe dla uzyskania wiarygodnych i użytecznych informacji z analizowanych próbek biologicznych. Efektywne przetwarzanie danych nie tylko zwiększa dokładność pomiarów, ale także pozwala na bardziej kompleksowe zrozumienie badanych materiałów, co ma istotne znaczenie zarówno w badaniach naukowych, jak i w zastosowaniach przemysłowych oraz medycznych.

7.1.1 Przykłady metod przetwarzania danych

Do jednych z najczęściej wykorzystywanych metod przetwarzania danych pomiarowych w spektroskopii NIR należą:

- a) Analiza składowych głównych (PCA) – metoda statystyczna, służąca do redukcji tabeli danych do jej podstawowych cech [39]. Metodę tą szerzej opisano w następnym rozdziale.
- b) Regresja metodą najmniejszych kwadratów cząstkowych (PLS) – technika uogólniająca i łącząca cechy z analizy PCA oraz regresji wielokrotnej. Przydatna do przewidywania zestawu zmiennych zależnych dla bardzo dużych zestawów zmiennych niezależnych [40],
- c) Filtracja Savitzky'ego-Golaya – filtr cyfrowy, spopularyzowany przez Abrahama Savitzky'ego i Marcela J. E. Golaya, umożliwiający wygładzanie widma sygnału. Dzięki zastosowaniu filtra można, bez znacznego wpływu na parametry użytkowe sygnału, zmniejszyć szum i zwiększyć jakość sygnału [42].

7.1.2 Analiza składowych głównych (PCA)

Analiza składowych głównych (PCA) – ang. Principal Component Analysis, to metoda statystyczna, stosowana do redukcji wymiarów danych. Umożliwia zmniejszenie ilości cech danych wielowymiarowych do mniejszej liczby nowych, niezależnych zmiennych, zwanych składowymi głównymi. Składowe te są kombinacjami oryginalnych składowych, co umożliwia

zachowanie maksymalnej ilości informacji, przy jednoczesnym uproszczeniu ilości danych [39]. Proces ten można podzielić na kilka etapów, a każdy z nich opisany jest odpowiednimi wzorami.

Standaryzacja zmiennych

Pierwszym i kluczowym krokiem jest standaryzacja zmiennych. Zmienne o większej skali mogą nieproporcjonalnie wpływać na wyniki analizy. Jeżeli jedna zmienna miałaby znacznie większy zakres wartości od innych, mogłaby dominować w analizie i zniekształcić wyniki analizy. Aby temu zapobiec, w analizie PCA stosuje się standaryzację zmiennych, polegającą na przekształcaniu zmiennych do skali bezwymiarowej [39].

Standaryzację zmiennych można wykonać za pomocą metody SNV (ang. Standard Normal Variate) [43]. Metoda umożliwia znormalizowanie widm. Polega na odjęciu każdego widma przez jego własną średnią i podzieleniu go przez jego odchylenie standardowe. Wynikiem tej operacji jest widmo, którego średnia jest równa 0 (lub bardzo bliska zeru, w zależności od precyzji obliczeń), a odchylenie standardowe jest równe 1. Metoda SNV jest skuteczna w eliminowaniu efektów skali i różnic w intensywności między próbkami, co pozwala na lepszą interpretację danych spektrometrycznych [43].

Rozkład wartości własnych (EVD)

Kolejnym krokiem w analizie PCA jest uproszczenie danych poprzez wybór mniejszej liczby wymiarów, które najlepiej oddają strukturę danych. Redukcja wymiarów jest realizowana za pomocą dekompozycji wartości własnych (EVD) macierzy kowariancji C . Wartości własne wskazują, jaka część wariancji jest wyjaśniana przez każdą ze składowych głównych, a suma wszystkich wartości własnych jest równa całkowitej wariancji danych [39]. Rozwiązujemy równanie dla wartości własnych i wektorów własnych:

$$Cv_i = \lambda_i v_i$$

Gdzie: λ_i – wartości własne, odpowiadające wariancji wyjaśnionej przez składowe główne, v_i – wektory własne, stanowiące nowe zmienne

Redukcja wymiarów

Następnym etapem jest skalowanie i interpretacja biplotu, który umożliwia wizualizację danych wielowymiarowych w dwóch wymiarach. Wartości własne wskazują, jaka część całkowitej wariancji danych jest wyjaśniona przez każdą składową główną. Aby uprościć dane, wybieramy tylko te składowe, które wyjaśniają największą część wariancji. Zwykle wybiera się te składowe, które mają największe wartości własne. Skumulowany procent wyjaśnionej wariancji dla k składowych obliczamy jako:

$$\text{Explained variance} = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^p \lambda_i} * 100\%$$

Gdzie: k – liczba wybranych składowych, p – całkowita liczba składowych (zmiennych)

Uogólniony wzór PCA

Analizę składowych głównych można również przedstawić w postaci modelu matematycznego. Uogólniony wzór na analizę składowych głównych opisuje, w jaki sposób dane wejściowe są dekomponowane na nowe zmienne (składowe główne). Model ten można zapisać jako [41]:

$$X = 1x'_{average} + T_a P'_a + E_a$$

Gdzie: X – oryginalne dane (surowe widma), 1 – wektor jednostkowy, $x'_{average}$ – wektor rzędowy będący średnią wszystkich próbek, T_a – macierz współczynników (score), P'_a – macierz ładunków (coeff), E_a – macierz reszt (danych których model nie wyjaśnia)

7.2 Wykorzystanie funkcji oprogramowania MATLAB

MATLAB oferuje szeroki zakres wbudowanych funkcji, które znacznie ułatwiają analizę danych, w tym takie operacje jak analiza głównych składowych (PCA), normalizacja zmiennego standardu (SNV) oraz wizualizacja wyników na wykresach. Poniżej opisano wykorzystanie wymienionych funkcji. Cały kod jaki wykorzystano przy przetwarzaniu pomiarów oraz wykreślaniu wykresów zawiera Dodatek D: Kod programu MATLAB

PCA

Analiza składowych głównych (PCA) w oprogramowaniu MATLAB jest możliwa dzięki dodatkowemu modułowi Statistics and Machine Learning Toolbox. Funkcję wykonującą analizę PCA zastosowano poprzez komendę:

$$[coeff, score, latent, tsquared, explained, mu] = pca(data)$$

Gdzie: $coeff$ - macierz współczynników składowych głównych, $score$ - macierz wyników składowych głównych, $latent$ - wartości własne, odpowiadające każdej składowej głównej, $tsquared$ - statystyki t-kwadratowe, $explained$ - procent wariancji wyjaśnionej przez każdą składową główną, mu - średnie wartości cech przed przeprowadzeniem PCA, $data$ – dane pomiarowe, użyte do analizy.

SNV

Normalizację SNV można zastosować poprzez użycie komendy:

$$snv_data = (data - \text{mean}(data, 2)) ./ \text{std}(data, 0, 2)$$

Gdzie: $\text{mean}()$ – funkcja odpowiedzialna za liczenie średniej, $\text{std}()$ – funkcja odpowiedzialna za obliczenie odchylenia standardowego, $data$ – dane do normalizacji

Wykreślanie wykresów

MATLAB jest również doskonałym narzędziem do wizualizacji danych. Funkcje takie jak *plot()*, *plotmatrix()* czy *scatter()*, pozwalają na generowanie wykresów 2D i 3D. Przykładowo, aby wykreślić wyniki analizy PCA, użyto następującej komendy:

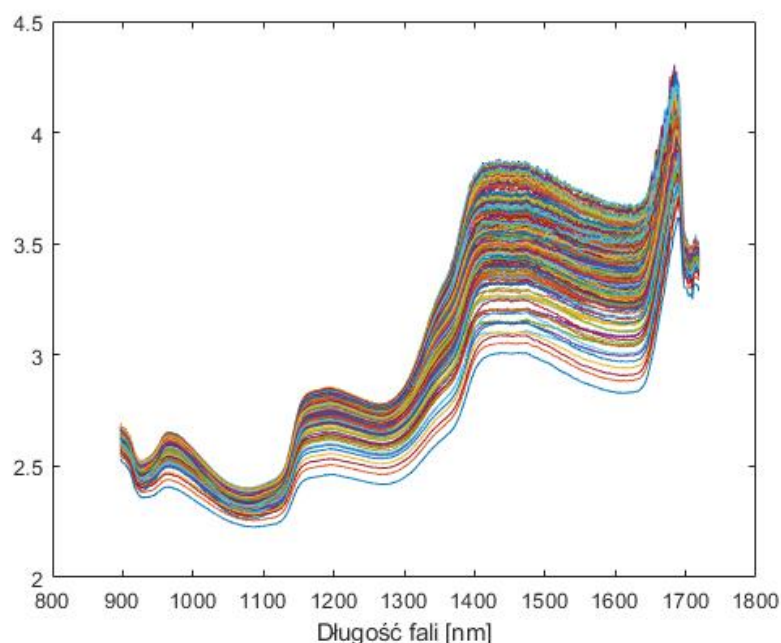
```
plotmatrix(score(:,1:pca_count))
```

Gdzie: *score* – składowe główne, *pca_count* – ilość składowych do wykreślenia.

7.3 Przetwarzanie widm pomiarowych próbek cytryny

7.3.1 Próbką 1 – cytryna z jej rdzeniem

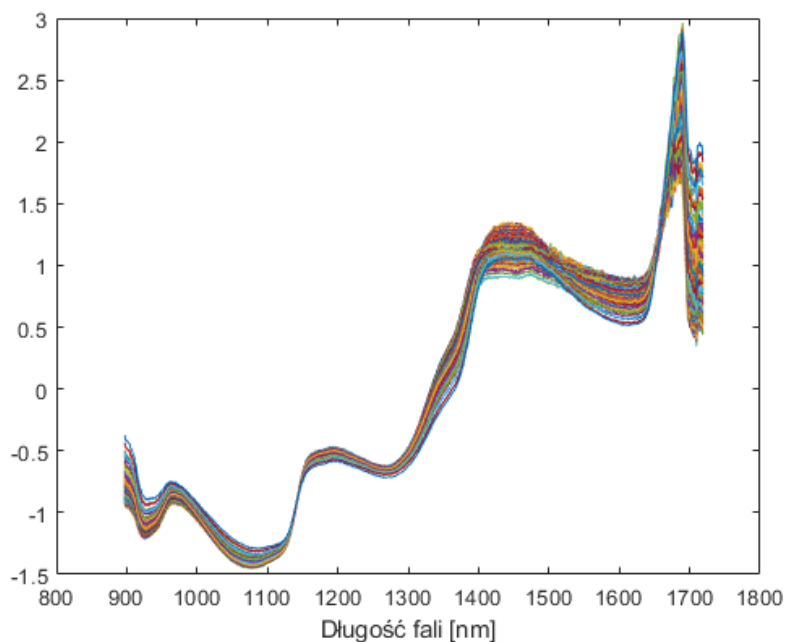
Dla próbek zebranych w rozdziale [6.2.1] za pomocą oprogramowania MATLAB obliczono absorbcję dla każdego pomiaru. Następnie wykreślono wszystkie widma absorpcji na rysunku [Rysunek 57].



Rysunek 57. Absorbancja próbek cytryny z rozdziału [6.2.1]

Widma te wykazały zróżnicowanie w poziomach absorpcji w zależności od długości fali, co może być bezpośrednio związane z różnicami w zawartości wody i innych składników chemicznych między poszczególnymi częściami cytryny - miąższu i rdzenia.

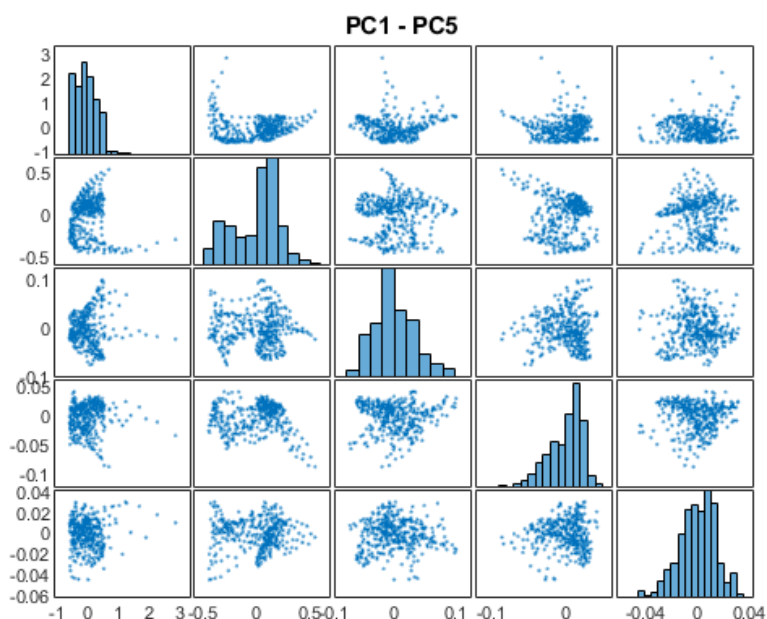
Następnie zastosowano metodę standardowej zmiennej normalnej (SNV) przy pomocy oprogramowania MATLAB, która pozwoliła na usunięcie przesunięć w danych i poprawę porównywalności widm. Znormalizowane widma przedstawiono na rysunku [Rysunek 58].



Rysunek 58. Znormalizowane widma próbek cytryny z rozdziału [6.2.1]

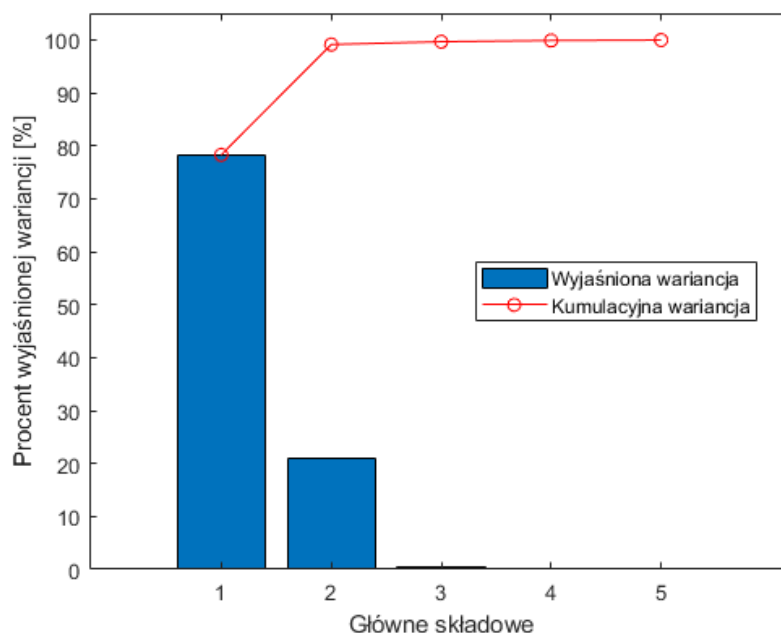
Znormalizowanie danych spowodowało wzmocnienie różnic między poszczególnymi próbkami. Umożliwiło to obserwację, w jakich zakresach widma najbardziej się od siebie różnią. Największe różnice można zaobserwować w pasmach absorpcji charakterystycznych dla wody, co może sugerować różnice w procentowej zawartości wody dla różnych próbek.

W kolejnym kroku dokonano analizy PCA, z użyciem wbudowanej w oprogramowanie MATLAB funkcji. Wyniki dla pięciu pierwszych składowych głównych wykreślono na rysunku [Rysunek 59].



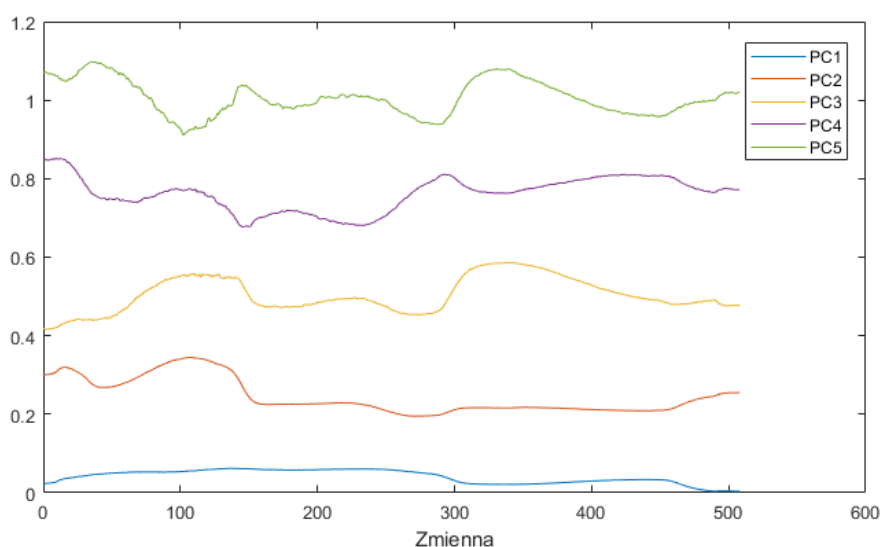
Rysunek 59. Wykres analizy PCA pierwszych pięciu składowych głównych (cytryna)

Analiza pokazała, że pierwsze pięć składowych głównych jest wyjaśnione przez odpowiednio: 78%, 21%, 0,51%, 0,23% oraz 0,08% wszystkich danych, co można odczytać z wektora *explained*, wygenerowanego przez funkcję oprogramowania MATLAB. Na rysunku [Rysunek 60] przedstawiono wektor *explained* dla pierwszych pięciu składowych głównych.



Rysunek 60. Procent wyjaśnionej wariancji dla cytryny

Na rysunku [Rysunek 61] przedstawiono wykres współczynników PCA (PC1-PC5). Wykres pozwala na analizę jak zmienne oryginalne są skorelowane z głównymi składowymi.

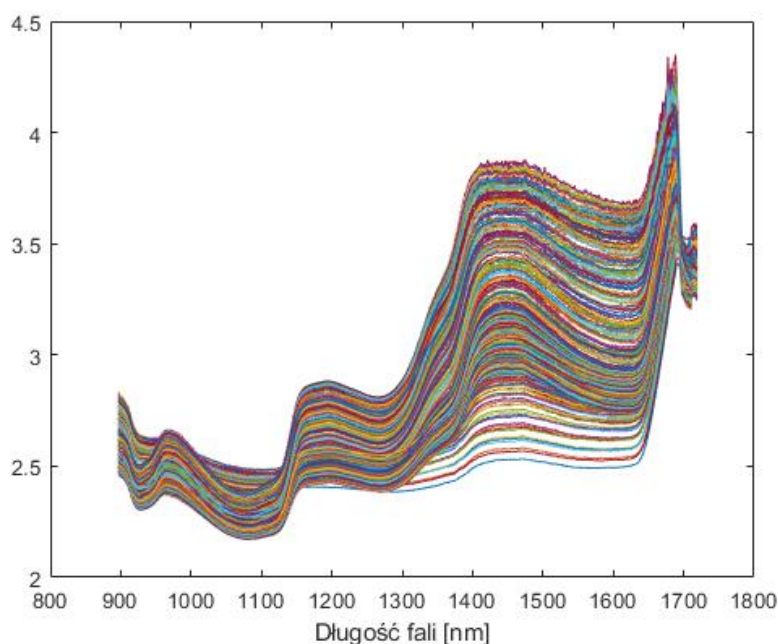


Rysunek 61. Współczynniki głównych składowych PC1-PC5 dla cytryny (środek).

Widma absorbancji dla próbki cytryny z rdzeniem wykazały zróżnicowanie w poziomach absorbancji, co można przypisać różnicom w zawartości wody i innych substancji chemicznych między miąższem a rdzeniem. Metoda SNV pomogła zniwelować przesunięcia w danych, zwiększając precyzję porównania widm. Największe różnice w znormalizowanych widmach dotyczą pasm absorpcji charakterystycznych dla wody, co sugeruje, że różne części cytryny mają odmienne proporcje zawartości wody. Analiza PCA ujawniła, że 78% wariancji danych zostało wyjaśnione przez pierwszą składową główną, co wskazuje, że większość różnic w widmach można przypisać jednej dominującej zmiennej. Dalsze składowe miały znacząco mniejszy wpływ, co może sugerować, że zmienność między poszczególnymi próbkami jest zdominowana przez zawartość wody lub innych podstawowych składników.

7.3.2 Próbką 2 – cytryna ze skórką

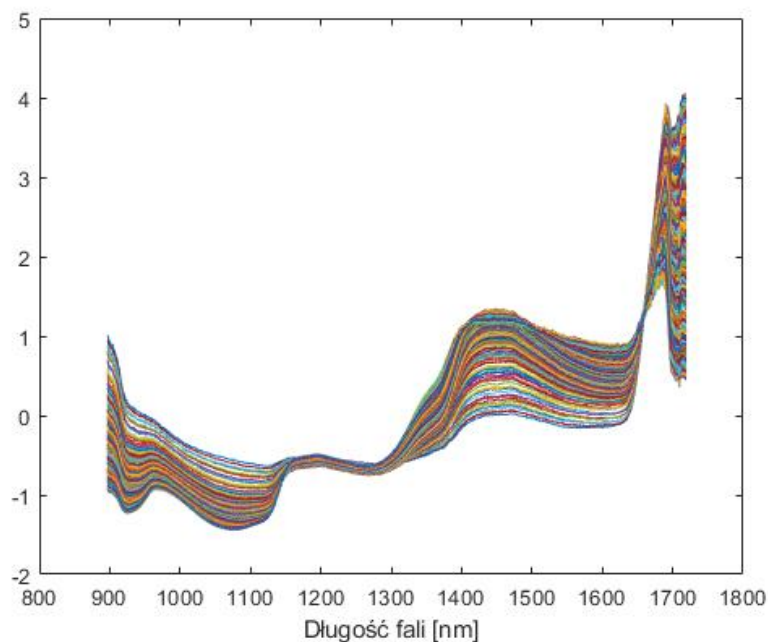
Analogicznie jak dla próbki cytryny ze rdzeniem, dla próbek cytryny ze skórką zebranych w rozdziale [6.2.2], obliczono absorbancję dla każdego pomiaru. Następnie wykreślono wszystkie widma absorbancji na rysunku [Rysunek 62].



Rysunek 62. Absorbancja próbek cytryny z rozdziału [6.2.2]

Widma te również wykazały zróżnicowanie w poziomach absorbancji, z tą różnicą, że można zaobserwować widma w których pomiary były wykonane dla obszaru, w którym nie umieszczono próbki cytryny.

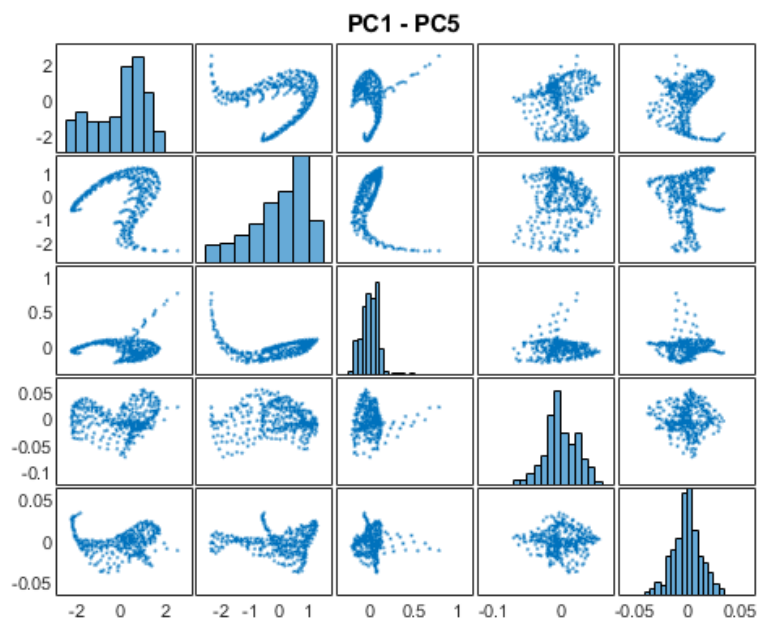
W kolejnym kroku zastosowano metodę standardowej zmiennej normalnej (SNV). Znormalizowane widma przedstawiono na rysunku [Rysunek 63].



Rysunek 63. Znormalizowane widma próbek cytryny z rozdziału [6.2.2]

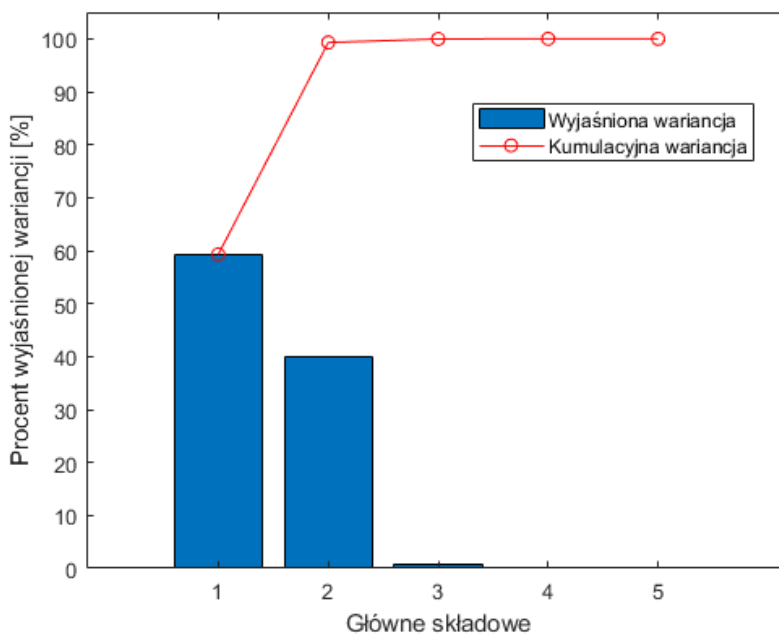
Znormalizowanie danych również spowodowało wzmocnienie różnic między poszczególnymi próbkami. Tym razem, ze względu na obecność próbek w których nie znajdowała się cytryna, różnice w znormalizowanych widmach są znacznie bardziej widoczne.

W kolejnym kroku dokonano analizy PCA. Wyniki dla pięciu pierwszych składowych głównych wykreślono na rysunku [Rysunek 64].



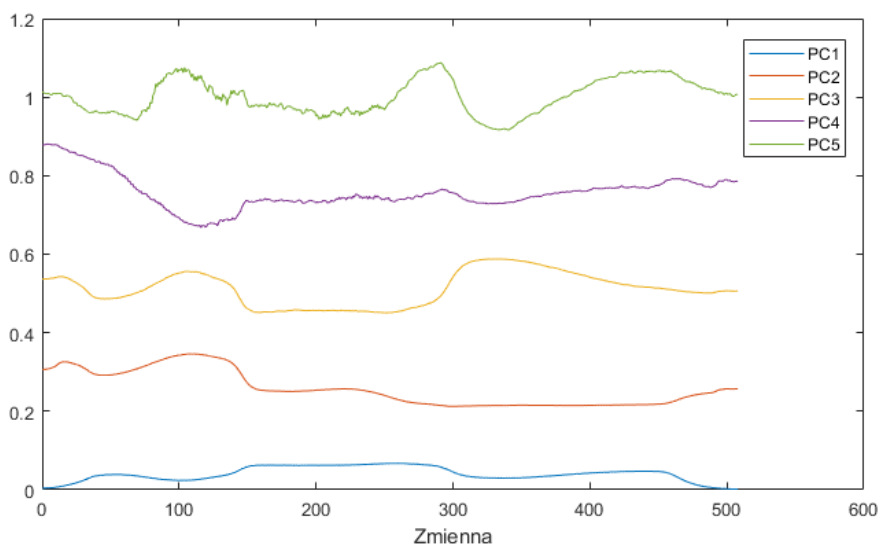
Rysunek 64. Wykres analizy PCA pierwszych pięciu składowych głównych (cytryna)

Analiza pokazała, że pierwsze pięć składowych głównych jest wyjaśnione przez odpowiednio: 59%, 40%, 0,66%, 0,03% oraz 0,01% wszystkich danych. Na rysunku [Rysunek 65] przedstawiono wektor *explained* dla pierwszych pięciu składowych głównych.



Rysunek 65. Procent wyjaśnionej wariancji dla cytryny

Na rysunku [Rysunek 66] przedstawiono wykres współczynników PCA (PC1-PC5).



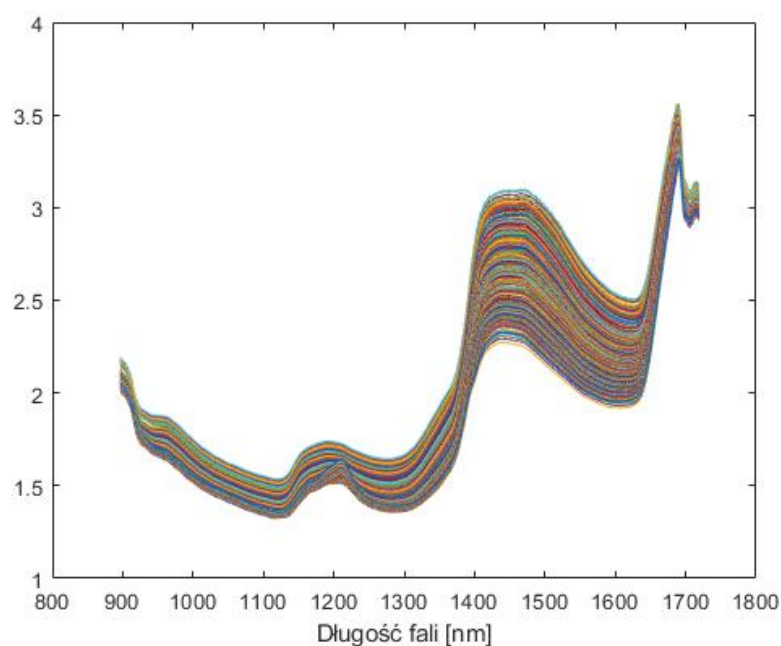
Rysunek 66. Współczynniki głównych składowych PC1-PC5 dla cytryny (krawędź).

Podobnie jak w przypadku próbki cytryny z rdzeniem, widma absorbancji dla cytryny ze skórką wykazały zróżnicowanie, jednak większe różnice zaobserwowano w próbkach, które nie zawierały cytryny. Po zastosowaniu metody SNV różnice w znormalizowanych widmach stały

się bardziej wyraźne. Obecność próbek bez cytryny znacząco wpłynęła na dane, a analiza PCA wykazała, że dwie pierwsze składowe wyjaśniają niemal całą wariancję (59% i 40%), co sugeruje, że te dwie składowe główne dominują w różnicach między próbkami, zwłaszcza w próbkach bez cytryny.

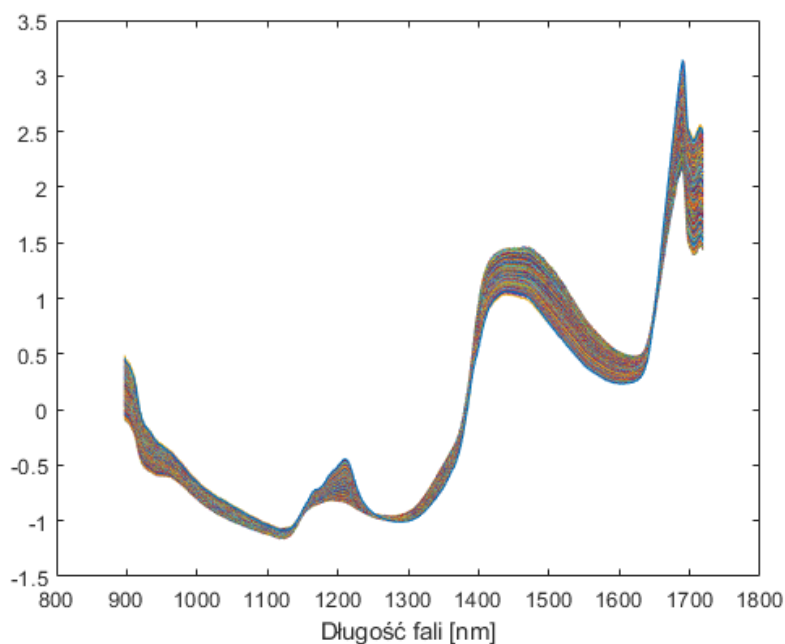
7.4 Próbką boczku

W przypadku próbki boczku postąpiono analogicznie – zebrano dane, obliczono absorbancję i wykreślono ją na rysunku [Rysunek 67].



Rysunek 67. Absorbancja próbek boczku z rozdziału [6.3.1]

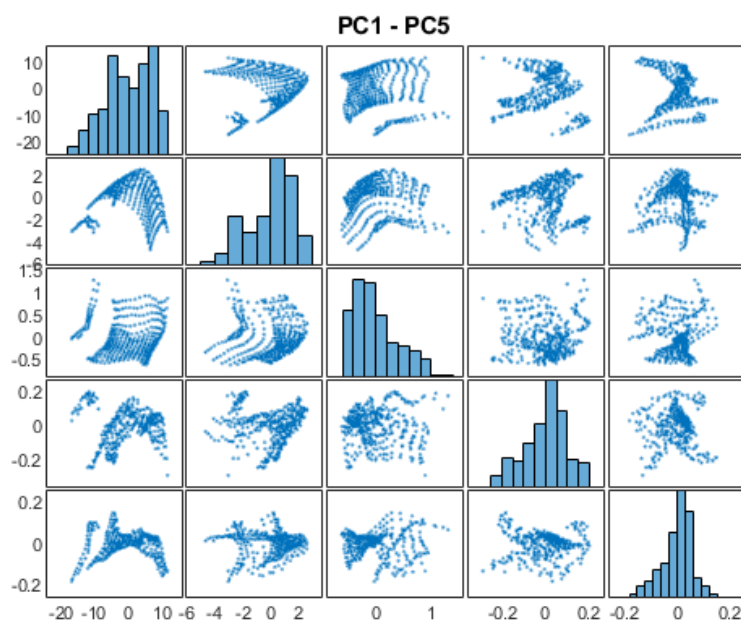
Następnie znormalizowano dane przy pomocy SNV i zaprezentowano na rysunku [Rysunek 68].



Rysunek 68. Znormalizowane widma boczków z rozdziału [6.3.1]

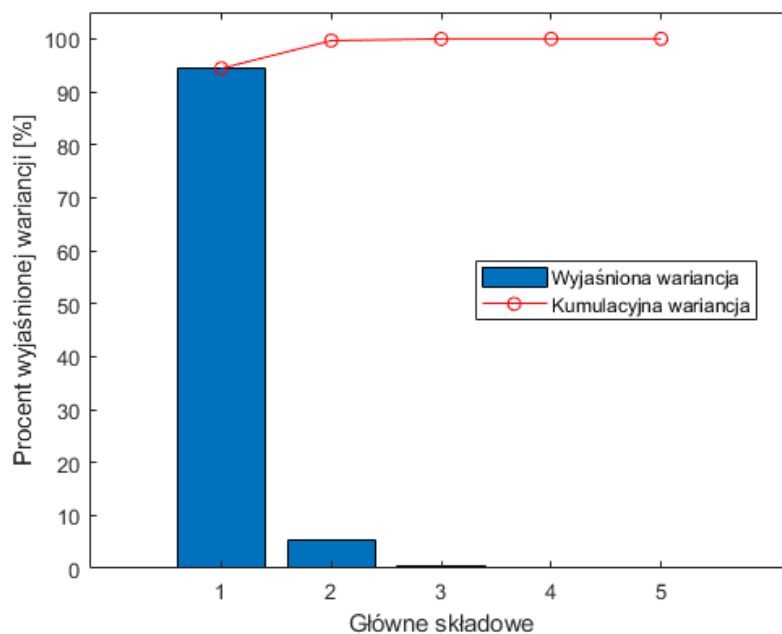
Znormalizowanie danych także w tym przypadku spowodowało wzmocnienie różnic między poszczególnymi próbkami, tym razem w pasmach charakterystycznych dla lipidów.

W kolejnym kroku dokonano analizy PCA, z użyciem wbudowanej w oprogramowanie MATLAB funkcji. Wyniki dla trzech pierwszych składowych głównych wykreślono na rysunku [Rysunek 69].



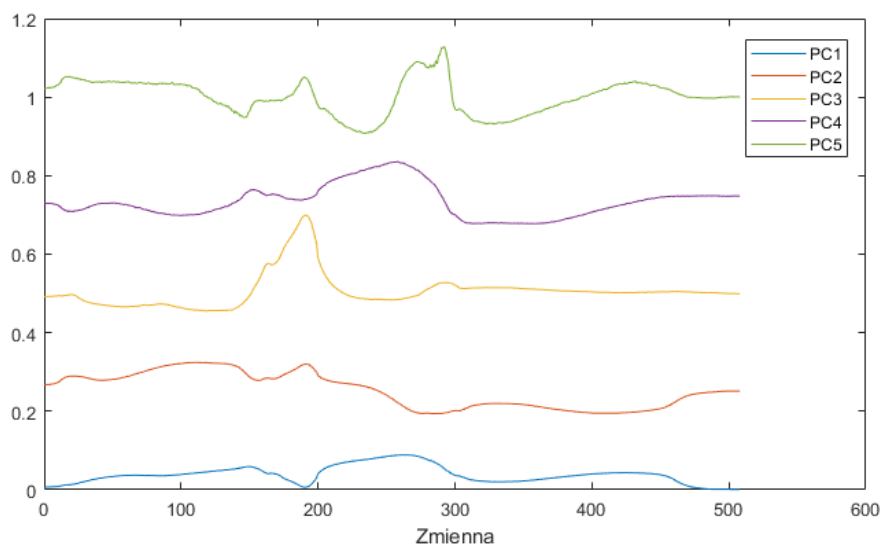
Rysunek 69. Wykres analizy PCA dla trzech składowych głównych (boczek)

Analiza pokazała, że pierwsze pięć składowych głównych jest wyjaśnione przez odpowiednio: 94,4%, 5,25%, 0,32%, 0,019% oraz 0,006% wszystkich danych. Na rysunku [Rysunek 65] przedstawiono wektor *explained* dla pierwszych pięciu składowych głównych.



Rysunek 70. Procent wyjaśnionej wariancji dla boczku

Na rysunku [Rysunek 71] przedstawiono wykres współczynników PCA (PC1-PC5).



Rysunek 71. Współczynniki głównych składowych PC1-PC5 dla boczku.

Widma absorbancji dla próbek boczku wykazały wyraźne różnice, głównie w zakresie pasm charakterystycznych dla lipidów, co wynika z obecności tłuszczu w boczku. Zastosowanie SNV pozwoliło na wzmocnienie tych różnic, umożliwiając dokładniejsze porównanie próbek.

Analiza PCA wskazała, że pierwsza składowa wyjaśnia aż 94,4% całkowitej wariancji, co może sugerować, że większość zmienności danych pochodzi z różnic w zawartości lipidów między poszczególnymi próbkami.

7.5 Wnioski

Przeprowadzone analizy widmowe próbek cytryny i boczku pozwoliły na wyciągnięcie istotnych wniosków na temat ich składu chemicznego i różnic strukturalnych. w przypadku cytryny, największe różnice dotyczyły zawartości wody, co potwierdziły zarówno widma absorpcyjności, jak i analiza PCA. w przypadku boczku kluczową zmienną była zawartość lipidów, co również znalazło odzwierciedlenie w analizie PCA, gdzie pierwsza składowa wyjaśniała większość wariancji.

Dzięki zastosowaniu metod SNV i PCA możliwe było uzyskanie bardziej precyzyjnych i porównywalnych wyników, co potwierdziło, że różnice w zawartości składników, takich jak woda czy lipidy, mają kluczowe znaczenie w analizach spektroskopowych próbek biologicznych.

8. PODSUMOWANIE

Celem niniejszej pracy było zbadanie charakterystyk widmowych obiektów biologicznych w zakresie bliskiej podczerwieni (NIR). Aby to osiągnąć, konieczne było zaprojektowanie i zbudowanie odpowiedniego układu pomiarowego, umożliwiającego precyzyjne i powtarzalne pomiary widm optycznych zarówno dla próbek stałych, jak i płynnych. Kluczowym elementem projektu było zestawienie stanowiska laboratoryjnego dla spektrometru NIR-1 firmy Wasatch Photonics, które stanowiło podstawę do przeprowadzenia pomiarów a także wykorzystanie frezarki CNC jako urządzenia mechanicznego, służącego do poruszania głowicą pomiarową o zadaną odległość, w celu umożliwienia wykonywania pomiarów dwuwymiarowych.

8.1 Napotkane problemy

W trakcie realizacji pracy napotkano na kilka wyzwań technicznych.

8.1.1 Problem z synchronizacją pracy spektrometru i frezarki

Jednym z głównych problemów była potrzeba synchronizacji pracy spektrometru z urządzeniem umożliwiającym ruch w przestrzeni XYZ - frezarką CNC. Problem ten dotyczył momentu uruchamiania pomiarów na spektrometrze, który musiał być ściśle powiązany z zakończeniem ruchu urządzenia. Niestety, frezarka nie dostarczała bezpośredniej informacji o zakończeniu ruchu, co uniemożliwiało natychmiastowe rozpoczęcie pomiaru. Rozwiązaniem tego problemu było dokładne pomierzenie czasu trwania ruchu dla różnych kroków urządzenia oraz zastosowanie interpolacji do przewidzenia momentu zakończenia ruchu. Dzięki temu możliwe było zsynchronizowanie pracy spektrometru z ruchem urządzenia, co zwiększyło precyzję uzyskiwanych wyników.

8.1.2 Problem ze zbyt wysokim umieszczeniem głowicy pomiarowej

Kolejnym istotnym problemem było początkowe zbyt wysokie umieszczenie głowicy pomiarowej względem badanych próbek. Taka konfiguracja prowadziła do rozmycia i niedokładności pomiarów, co znacząco obniżało ich jakość. Aby temu zaradzić, zaprojektowano specjalny uchwyt dla głowicy światłowodowej, który pozwolił na jej zamontowanie bliżej próbek. Zastosowanie technologii druku 3D umożliwiło szybkie i precyzyjne wykonanie elementów uchwyty, co skutecznie rozwiązało problem i pozwoliło na uzyskanie bardziej dokładnych wyników pomiarów.

8.2 Możliwości dalszego rozwoju

Mimo osiągnięcia zamierzonych celów, istnieje kilka obszarów, w których można rozwinąć i ulepszyć zaprojektowany układ pomiarowy.

8.2.1 Rozbudowa układu o możliwość pomiarów w trzech wymiarach (3D)

Aktualny układ pozwala na przeprowadzanie pomiarów jedynie w dwóch wymiarach (2D). Wprowadzenie pomiarów 3D umożliwiłoby bardziej kompleksową analizę próbek biologicznych, uwzględniając ich strukturę przestrzenną. Taka rozbudowa systemu mogłaby znacząco zwiększyć jego użyteczność w badaniach nad materiałami biologicznymi.

8.2.2 Udoskonalenie algorytmów przetwarzania danych

Obecne algorytmy przetwarzania danych spełniają podstawowe wymagania analityczne, jednak rozwinięcie ich o bardziej zaawansowane techniki, takie jak algorytmy uczenia maszynowego, mogłoby zwiększyć precyzję i szybkość analizy danych. Automatyczna klasyfikacja danych oraz wykrywanie anomalii mogłyby prowadzić do bardziej zaawansowanych wniosków naukowych.

8.2.3 Rozwój oprogramowania do obsługi układu pomiarowego

Rozbudowa oprogramowania zarządzającego całym układem pomiarowym pozwoliłaby na lepszą integrację poszczególnych elementów systemu. Włączenie algorytmów przetwarzania danych bezpośrednio do oprogramowania oraz wprowadzenie wizualizacji wyników pomiarów w czasie rzeczywistym mogłoby znacząco ułatwić pracę z systemem, poprawiając jednocześnie komfort obsługi i efektywność pomiarów.

8.3 Wnioski końcowe

Realizacja niniejszej pracy umożliwiła stworzenie efektywnego i precyzyjnego układu do pomiarów widmowych w zakresie bliskiej podczerwieni. Pomimo napotkanych problemów technicznych, udało się osiągnąć założone cele, a wprowadzone modyfikacje pozwoliły na uzyskanie powtarzalnych i wiarygodnych wyników. Istnieje jednak potencjał do dalszego rozwoju zaprojektowanego układu, szczególnie w kontekście rozbudowy funkcjonalności i ulepszenia algorytmów przetwarzania danych. Przedstawione propozycje mogą stanowić solidną podstawę do dalszych prac badawczych, a także do zastosowań przemysłowych, zwiększając tym samym wartość praktyczną opracowanego rozwiązania.

WYKAZ LITERATURY

1. Pavia, Kriz, Lampman, "Introduction to Spectroscopy", Thomson Learning (2001).
2. Valery Tuchin, "Handbook of Optical Biomedical Diagnostics", SPIE Press (2016).
3. Donald A. Burns, Emil W. Ciurczak, "Handbook of Near-Infrared Analysis", CRC Press (2008).
4. Y. Ozaki, C. Huck, S. Tsuchikawa, S. Engelsen, "Near-Infrared Spectroscopy Theory, Spectral Analysis, Instrumentation, and Applications", Springer (2021).
5. Valery Tuchin, Tissue optics, "Light scattering Methods and Instruments for Medical Diagnosis", SPIE Press (2007).
6. A. Rogalski, Z. Bielecki, "Detection of optical signals", CRC Press (2022).
7. H. W. Siesler, Y. Ozaki, S. Kawata, H. M. Heise, "Near-Infrared Spectroscopy. Principles, Instruments, Applications", WILEY-VCH (2002).
8. Magwaza, L.S., Opara, U.L., Nieuwoudt, H. et al. „NIR Spectroscopy Applications for Internal and External Quality Analysis of Citrus Fruit - A Review". Food Bioprocess Technol 5, 425–444 (2012).
9. Ślęzak, Czub, Świątek, Niżnikowski, Głowacz, „Wykorzystanie spektroskopii w bliskiej podczerwieni (NIRS) w ocenie składu chemicznego mięsa jagnięcego", Roczniki Naukowe Polskiego Towarzystwa Zootechnicznego 9(3), 69-76 (2013).
10. Karta katalogowa analizatora FoodScan <https://www.fossanalytics.com/pl-pl/products/foodscan-2-meat-analyser> [data dostępu: 11.01.2024r]
11. Antti, Henrik, Michael Sjöström and Lars Wallbäcks. "Multivariate calibration models using NIR spectroscopy on pulp and paper industrial applications." Journal of Chemometrics v10(5-6) 591-603 (1996)
12. Rui Vitorino, António S. Barros, Sofia Guedes, Douglas C. Caixeta, Robinson Sabino-Silva, Diagnostic and monitoring applications using near infrared (NIR) spectroscopy in cancer and other diseases, Photodiagnosis and Photodynamic Therapy, Vol. 42, , 103633 (2023)doi: 10.1016/j.pdpdt.2023.103633
13. Shanas, No'ad & Querido, William & Dumont, Alex & Yonko, Elizabeth & Carter, Erin & OK, Jina & Karchner, James & Barbe, Mary & Ali, Sayed & Patil, Chetan & Raggio, Cathleen & Pleshko, Nancy. Clinical Application of Near Infrared Fiber Optic Spectroscopy for Non-Invasive Bone Assessment. Journal of Biophotonics 13 10 e201960172 (2020). DOI:1002/jbio.201960172.
14. <https://encyklopedia.pwn.pl/haslo/promieniowanie-jonizujace;3962659.html> [data dostępu: 12.02.2024r.]
15. S. Wu, Sijun & Zhang, Xiaoyang & Zhou, Guoming & Wu, Jiaheng & Song, Wen & Zhang, Ying & Li, Zheng & Li, Wenlong. Physical fingerprint transformation of herbal medicines powders using near-infrared spectroscopy. Advanced Powder Technology 34104244. (2023). doi:10.1016/j.ap.2023.104244.

16. StatSoft (2006). Elektroniczny Podręcznik Statystyki PL, Krakow, WEB:
https://www.statsoft.pl/textbook/stathome_stat.html?https%3A%2F%2Fwww.statsoft.pl%2Ftextbook%2Fstpls.html [data dostępu: 12.02.2024r.]
17. Roggo, Pascal Chalus, Lene Maurer, Carmen Lema-Martinez, Aurélie Edmond, Nadine Jent, A review of near infrared spectroscopy and chemometrics in pharmaceutical technologies, Journal of Pharmaceutical and Biomedical Analysis, vol.44(3), , 683-700 (2007)
18. Karta katalogowa spektrometru: https://wasatchphotonics.com/wp-content/uploads/WP-PS_WPNIR-I_15Jan18.pdf [data dostępu: 13.03.2024r.]
19. Karta katalogowa detektora wykorzystywanego w spektrometrze:
https://www.hamamatsu.com/content/dam/hamamatsu-photonics/sites/documents/99_SALES_LIBRARY/ssd/g9211-256s_etc_kmir1011e.pdf [data dostępu: 16.03.2024r.]
20. Karta katalogowa źródła: https://bwtek.com/Download_datasheets/BPS101.pdf [data dostępu: 13.03.2024r.]
21. Karta katalogowa uchwytu na próbki płynne:
<https://www.optosirius.co.jp/OceanOptics/technical/MNL-1001%20Square%20One%20QSG%20Rev%20B.pdf> [data dostępu: 3.09.2024r.]
22. Karta katalogowa światłowodu:
<https://www.thorlabs.com/drawings/2a0bdc14fdf10abd-60775934-0094-153D-2BCBECE75FF4CE33/FT400EMT-SpecSheet.pdf> [data dostępu: 13.03.2024r.]
23. <https://www.abvista.com/news/a-guide-to-nir-understanding-nir-spectra> [data dostępu: 08.09.2024]
24. S. Dharani, T. Khuroo, S F. Barakh Ali, “Application of Near-infrared Spectroscopy and Multivariate Methods for the Estimation of Isopropyl Alcohol Content in Hand Sanitizer Formulation” Acta Pharm. Sci. Vol 60:(1), (2022) doi:10.23893/1307-2080.APS.6002
25. <https://encyklopedia.pwn.pl/haslo/frezarka;3902775.html> [data dostępu: 8.02.2024r.]
26. <https://encyklopedia.pwn.pl/haslo/obrabiarka;3949412.html> [data dostępu: 8.02.2024r.]
27. Karta katalogowa frezarki CNC: <https://docs.sainsmart.com/article/3jgtgas880-genmitsu-3018-pro-resources> [data dostępu: 8.02.2024r.]
28. Dokumentacja GRBL: <https://github.com/grbl/grbl> [data dostępu: 9.02.2024r.]
29. Dokumentacja GRBL: <https://bengler.no/grbl> [data dostępu: 9.02.2024r.]
30. Dokumentacja Python: <https://pypl.github.io/PYPL.html> [data dostępu: 12.02.2024r.]
31. Dokumentacja biblioteki Tk: <https://docs.python.org/3/library/tk.html> [data dostępu: 12.02.2024r.]

32. Dokumentacja biblioteki PySerial: <https://pyserial.readthedocs.io/en/latest/> [data dostępu: 12.02.2024r.]
33. Dokumentacja biblioteki Wasatch.PY:
<https://wasatchphotonics.com/api/Wasatch.PY/annotated.html> [data dostępu: 12.02.2024r.]
34. E. T. Bartholomew, "Internal Decline of Lemons. II. Growth Rate, Water Content, and Acidity of Lemons at Different Stages of Maturity." *American Journal of Botany* 10(3), 117–26 (1923) doi:10.2307/2435359.
35. Magwaza, Lembe & Opara, Umezuruike & Nieuwoudt, Hélène & Cronje, Paul. "Non-destructive quality assessment of 'Valencia' orange using FT-NIR spectroscopy", 15th International Conference on NIR Spectroscopy, Cape Town, South Africa (2011).
36. A. Marabi, Alejandro & Raemy, A. & Bauwens, I. & Burbidge, Adam & Wallach, Rony & Saguy, I. Sam. "Effect of fat content on the dissolution enthalpy and kinetics of a model food powder". *Journal of Food Engineering* 85(4) 518-527 (2008) 1016/j.jfoodeng.2007.08.012.
37. U.S. Department of Agriculture, Agricultural Research Service, Pork, cured, bacon, cooked, restaurant: <https://fdc.nal.usda.gov/fdc-app.html#/food-details/749420/nutrients> [data dostępu: 12.02.2024r.]
38. Christiaan Kapper, "Rapid prediction of pork quality: correlation of fresh meat measurements to pork water holding capacity and its technological quality." PhD Thesis, Wageningen University, Wageningen, NL (2012).
39. Greenacre, M., Groenen, P.J.F., Hastie, T. et al. Principal component analysis. *Nature Reviews Methods Primers* 2 100 (2022) doi:10.1038/s43586-022-00184-w.
40. Herve Abdi, Hervé & Williams, Lynne. Partial Least Squares Methods: Partial Least Squares Correlation and Partial Least Square Regression. *Methods in molecular biology* 930 549-79 (2013). 10.1007/978-1-62703-059-5_23.
41. Lars Norgaard, Rasmus Bro, Soren Engelsen, "Principal Component Analysis and Near Infrared Spectroscopy", A white paper from FOSS.
42. Neal B. Gallagher Savitzky-Golay Smoothing and Differentiation Filter (2020). 10.13140/RG.2.2.20339.50725.
43. M. Zeaiter, D. Rutledge, "Comprehensive Chemometrics", Elsevier B.V. (2009).

Dodatek A: Spis rysunków

Rysunek 1. Część widma fali elektromagnetycznej, na podstawie [1]	10
Rysunek 2. Porównanie metod spektralnych - Ramana, MIR oraz NIR [3]	12
Rysunek 3. Schemat blokowy układu pomiarowego na podstawie [4]	14
Rysunek 4. Schemat monochromatora typu "Czerny-Turner". Na podstawie [4]	14
Rysunek 5. Schemat interferometru Michelsona. Opracowano na podstawie [6]	15
Rysunek 6. Uproszczony schemat trybu transmisyjnego [3]	18
Rysunek 7. Uproszczony schemat trybu reflektancyjnego [3]	19
Rysunek 8. Typowe widma absorbancji pomarańczy "Walencja" [8]	20
Rysunek 9. Zdjęcie urządzenia FoodScan™ 2 [10]	21
Rysunek 10. Widma absorbancji próbek kontrolnych [12]	22
Rysunek 11. Widma NIR polistyrenu oraz skóry i powięzi [13]	23
Rysunek 12. Spektrometr NIR-1 firmy Wasatch [18]	24
Rysunek 13. Wykres czułości detektora G9214-512SA	25
Rysunek 14. Charakterystyka tłumienia światłowodu FT400EMT	27
Rysunek 15. Wykres widma referencyjnego	29
Rysunek 16. Wykres widma dark	30
Rysunek 17. Wykres widma pustej kuwety z polistyrenu	30
Rysunek 18. Wykres transmisji dla pustej kuwety z polistyrenu	31
Rysunek 19. Wykres absorbancji dla polistyrenu [3]	31
Rysunek 20. Wykres widma próbki z wodą	32
Rysunek 21. Wykres procentowej transmisji dla próbki z wodą	32
Rysunek 22. Wykres absorbancji dla wybranych produktów mlecznych [23]	33
Rysunek 23. Wykres widma próbki alkoholu izopropylowego	33
Rysunek 24. Wykres procentowej transmisji dla alkoholu izopropylowego	34
Rysunek 25. Wykres absorbancji dla alkoholu izopropylowego [24]	34
Rysunek 26. Wykres widma próbki dimetylosulfotlenku (DMSO)	35
Rysunek 27. Wykres procentowej transmisji dla dimetylosulfotlenku (DMSO)	35
Rysunek 28. Wykres widma referencyjnego	36
Rysunek 29. Wykres widma dark	37
Rysunek 30. Wykres widma pustej kuwety z polistyrenu	37
Rysunek 31. Wykres procentowej transmisji dla pustej kuwety z polistyrenu	38
Rysunek 32. Wykres widma próbki z wodą	38
Rysunek 33. Wykres procentowej transmisji dla próbki z wodą	39
Rysunek 34. Wykres widma próbki z alkoholem izopropylowym	39
Rysunek 35. Wykres procentowej transmisji dla próbki z alkoholem izopropylowym ..	40
Rysunek 36. Wykres widma próbki dimetylosulfotlenku (DMSO)	40
Rysunek 37. Wykres procentowej transmisji dla dimetylosulfotlenku (DMSO)	41
Rysunek 38. Transmisja [%] w funkcji czasu integracji dla pomiaru w trybie dark	41
Rysunek 39. Transmisja [%] w funkcji czasu integracji dla pomiaru referencyjnego ..	42

Rysunek 40. Schemat docelowego układu pomiarowego	44
Rysunek 41. Frezarka CNC Genmitsu 3018 [27]	45
Rysunek 42. Zrzut interfejsu graficznego programu do obsługi frezarki	47
Rysunek 43. Zrzut interfejsu graficznego programu do obsługi spektrometru	49
Rysunek 44. Adapter mocujący światłowód do frezarki CNC.....	52
Rysunek 45. Adapter mocujący światłowód do frezarki CNC. Widok od dołu.....	53
Rysunek 46. Mocowania do wysokich i niskich elementów.....	53
Rysunek 47. Zmontowany układ pomiarowy wraz z laptopem.....	54
Rysunek 48. Zamontowana głowica pomiarowa wraz ze źródłem światła	54
Rysunek 49. Zamontowany na płycie montażowej spektrometr NIR-1	55
Rysunek 50. Próbkę cytryny umieszczona w układzie pomiarowym	56
Rysunek 51. Absorbancja plastra cytryny dla dwóch próbek	57
Rysunek 52. Absorbancja plastra cytryny dla dwóch próbek	58
Rysunek 53. Widmo absorbancji pomarańczy „Walencia” [35]	59
Rysunek 54. Próbkę boczku umieszczona w układzie pomiarowym	60
Rysunek 55. Absorbancja wybranych próbek pomiarowych boczku.....	61
Rysunek 56. Widmo absorbancji boczku [38].....	61
Rysunek 57. Absorbancja próbek cytryny z rozdziału [6.2.1].....	65
Rysunek 58. Znormalizowane widma próbek cytryny z rozdziału [6.2.1]	66
Rysunek 59. Wykres analizy PCA pierwszych pięciu składowych głównych (cytryna)66	
Rysunek 60. Procent wyjaśnionej wariancji dla cytryny	67
Rysunek 61. Współczynniki głównych składowych PC1-PC5 dla cytryny (środek). ...	67
Rysunek 62. Absorbancja próbek cytryny z rozdziału [6.2.2].....	68
Rysunek 63. Znormalizowane widma próbek cytryny z rozdziału [6.2.2]	69
Rysunek 64. Wykres analizy PCA pierwszych pięciu składowych głównych (cytryna)69	
Rysunek 65. Procent wyjaśnionej wariancji dla cytryny	70
Rysunek 66. Współczynniki głównych składowych PC1-PC5 dla cytryny (krawędź)..	70
Rysunek 67. Absorbancja próbek boczku z rozdziału [6.3.1].....	71
Rysunek 68. Znormalizowane widma boczku z rozdziału [6.3.1]	72
Rysunek 69. Wykres analizy PCA dla trzech składowych głównych (boczek).....	72
Rysunek 70. Procent wyjaśnionej wariancji dla boczku	73
Rysunek 71. Współczynniki głównych składowych PC1-PC5 dla boczku.....	73

Dodatek B: Spis tabel

Tabela 1. Porównanie metod spektralnych	12
Tabela 2. Parametry detektora w spektrometrze NIR-1	25
Tabela 3. Podstawowe parametry źródła światła BPS101	26

Dodatek C: Kod programu pomiarowego

Plik main.py

```
import tkinter as tk
from gui.my_gui import MyGUI
from cnc.cnc_serial import CNCSerial
from nir1.wasatch import Wasatch
import sys

if __name__ == "__main__":
    root = tk.Tk()
    serial_connection = CNCSerial()
    wasatch = Wasatch(root, sys.argv)
    gui = MyGUI(root, serial_connection, wasatch)
    root.mainloop()
```

Plik cnc_serial.py

```
import serial
import serial.tools.list_ports
import time

class CNCSerial:
    def __init__(self):
        self.serial_port = None
        self.connected = False

    def connect_cnc(self, port):
        try:
            self.serial_port = serial.Serial(
                port, baudrate=115200, timeout=1)
            self.connected = True
            return "Connected to {}".format(port)
        except Exception as e:
            return "Error: {}".format(e)

    def disconnect_cnc(self):
        if self.serial_port:
            self.serial_port.close()
            self.connected = False
            return "Disconnected from CNC"
        return "Not connected to CNC"

    def send_gcode(self, command):
        if self.serial_port and self.connected:
            self.serial_port.write((command + '\n').encode())
            return f"Command sent: {command}"
        return "Not connected to CNC"

    def wait_for_ending_move(self):
        if self.serial_port and self.connected:
            self.serial_port.write(('?\n').encode())
            time.sleep(0.1)
            response = self.serial_port.read_until().decode().strip()
            if "Idle" in response:
                return True
        return False

    def list_serial_ports(self):
        return [port.device for port in serial.tools.list_ports.comports()]
```

Plik my_gui.py

```
import tkinter as tk
from tkinter import ttk, scrolledtext, filedialog
import time
import threading
import numpy as np

class MyGUI:
    def __init__(self, root, serial_connection, wasatch):
```

```

self.root = root
self.serial = serial_connection
self.wasatch = wasatch
self.root.title("CNC & Wasatch Controller")

# Actual X and Y position
self.current_position = {'X': 0, 'Y': 0}
self.user_positions = {'2': None, '3': None, '4': None}

self.integration_time = 10
self.scans_to_average = 1
self.boxcar_half_width = 0
self.delay_ms = 1000
self.max_spectra = 0

self.samples_count_x = 1
self.samples_count_y = 1

self.output_file = None

self.setup_ui()
self.running = False
self.measure_thread = threading.Thread()

def setup_ui(self):

    # Left and right part of window
    self.left_frame = ttk.PanedWindow(self.root)
    self.left_frame.grid(row=0, column=0, sticky='n')

    self.right_frame = ttk.PanedWindow(self.root)
    self.right_frame.grid(row=0, column=1, sticky='n')

    # Serial connection frame
    self.serial_connection_frame = ttk.LabelFrame(self.left_frame, text="CNC serial connection")
    self.serial_connection_frame.grid(row=0, column=0, padx=10, pady=5, sticky="ew")

    self.serial_port_label = ttk.Label(self.serial_connection_frame, text="Port COM:")
    self.serial_port_label.grid(row=0, column=0, padx=10, pady=10)

    self.serial_ports = self.serial.list_serial_ports()
    self.serial_port_combobox = ttk.Combobox(self.serial_connection_frame, values=self.serial_ports)
    self.serial_port_combobox.grid(row=0, column=1, padx=10, pady=10)

    self.connect_button = ttk.Button(self.serial_connection_frame, text="Connect",
command=self.toggle_connection)
    self.connect_button.grid(row=0, column=2, padx=10, pady=10)

    # Control frame for movement
    self.control_frame = ttk.LabelFrame(self.left_frame, text="CNC control")
    self.control_frame.grid(row=1, column=0, padx=10, pady=5, sticky="ew")

    # Movement buttons
    button = ttk.Button(self.control_frame, text='\u2190',
command=lambda d='\u2190': self.move(d)) # Left
    button.grid(row=1, column=0, padx=10, pady=10)

    button = ttk.Button(self.control_frame, text='\u2191', command=lambda d='\u2191': self.move(d)) # Up
    button.grid(row=0, column=1, padx=10, pady=10)

    button = ttk.Button(self.control_frame, text='\u2192', command=lambda d='\u2192': self.move(d)) # Right
    button.grid(row=1, column=2, padx=10, pady=10)

    button = ttk.Button(self.control_frame, text='\u2193', command=lambda d='\u2193': self.move(d)) # Down
    button.grid(row=2, column=1, padx=10, pady=10)

    # Z-axis control
    self.up_button = ttk.Button(self.control_frame, text="Up", command=lambda: self.move('Up'))
    self.up_button.grid(row=0, column=3, padx=10, pady=10)

    self.down_button = ttk.Button(self.control_frame, text="Down", command=lambda: self.move('Down'))
    self.down_button.grid(row=2, column=3, padx=10, pady=10)

    # Adding "0,0" Button to reset position

```

```

        self.zero_button = ttk.Button(self.control_frame, text="Set 0.0", command=lambda:
self.set_position_zero())
        self.zero_button.grid(row=1, column=1, padx=10, pady=10)

# Step and Speed frame
self.step_speed_frame = ttk.LabelFrame(self.left_frame, text="CNC control parameters")
self.step_speed_frame.grid(row=2, column=0, padx=10, pady=5, sticky="ew")

self.step_label = ttk.Label(self.step_speed_frame, text="Step:")
self.step_label.grid(row=0, column=0, padx=10, pady=5)
self.step_entry = ttk.Entry(self.step_speed_frame)
self.step_entry.grid(row=0, column=1, padx=10, pady=5)
self.step_entry.insert(tk.END, '10') # Default value

self.speed_label = ttk.Label(self.step_speed_frame, text="Speed:")
self.speed_label.grid(row=0, column=2, padx=10, pady=5)
self.speed_entry = ttk.Entry(self.step_speed_frame)
self.speed_entry.grid(row=0, column=3, padx=10, pady=5)
self.speed_entry.insert(tk.END, '1000') # Default value

# Position setting and test movement frame
self.position_frame = ttk.LabelFrame(self.left_frame, text="Position configuration")
self.position_frame.grid(row=3, column=0, padx=10, pady=5, sticky="ew")

# Position buttons
self.set_button_1 = ttk.Button(self.position_frame, text="Set 1", command=lambda: self.set_position(1))
self.set_button_1.grid(row=0, column=0, padx=10, pady=5)

self.set_button_2 = ttk.Button(self.position_frame, text="Set 2", command=lambda: self.set_position(2))
self.set_button_2.grid(row=0, column=1, padx=10, pady=5)

self.set_button_3 = ttk.Button(self.position_frame, text="Set 3", command=lambda: self.set_position(3))
self.set_button_3.grid(row=1, column=1, padx=10, pady=5)

self.set_button_4 = ttk.Button(self.position_frame, text="Set 4", command=lambda: self.set_position(4))
self.set_button_4.grid(row=1, column=0, padx=10, pady=5)

self.test_button = ttk.Button(self.position_frame, text="Test", command=self.test_positions)
self.test_button.grid(row=1, column=2, padx=50, pady=5)

# Adding "Init" Button to send initialization commands
self.init_button = ttk.Button(self.position_frame, text="Goto 0.0", command=self.move_to_zero)
self.init_button.grid(row=0, column=2, padx=50, pady=10)

# Manual command frame
self.manual_command_frame = ttk.LabelFrame(self.left_frame, text="Manual command")
self.manual_command_frame.grid(row=4, column=0, padx=10, pady=5, sticky="ew")

self.manual_command_label = ttk.Label(self.manual_command_frame, text="Command:")
self.manual_command_label.grid(row=0, column=0, padx=10, pady=10)

self.manual_command_entry = ttk.Entry(self.manual_command_frame, width=30)
self.manual_command_entry.grid(row=0, column=1, padx=10, pady=10)

self.manual_command_button = ttk.Button(self.manual_command_frame, text="Send",
command=self.send_manual_command)
self.manual_command_button.grid(row=0, column=2, padx=10, pady=10)

# Log frame
self.log_frame = ttk.LabelFrame(self.left_frame, text="Log")
self.log_frame.grid(row=5, column=0, padx=10, pady=5, sticky="ew")

self.log_text = scrolledtext.ScrolledText(self.log_frame, wrap=tk.WORD, height=8, width=50)
self.log_text.pack(fill=tk.BOTH, expand=True)

# Spectrometer connection frame
self.wasatch_connection_frame = ttk.LabelFrame(self.right_frame, text="Wasatch connection")
self.wasatch_connection_frame.grid(row=0, column=1, padx=10, pady=5, sticky="nsew")

self.wasatch_connect_button = ttk.Button(self.wasatch_connection_frame, text = "Connect
spectrometer", command=self.wasatch.connect)
self.wasatch_connect_button.grid(row=0, column=0, padx=50, pady=5)

# File path selection for saving spectra
self.file_path_frame = ttk.LabelFrame(self.right_frame, text="File path for spectra")

```

```

self.file_path_frame.grid(row=1, column=1, padx=10, pady=5, sticky="ew")

self.file_path_label = ttk.Label(self.file_path_frame, text="File path:")
self.file_path_label.grid(row=0, column=0, padx=10, pady=10)

self.file_path_entry = ttk.Entry(self.file_path_frame, width=30)
self.file_path_entry.grid(row=0, column=1, padx=10, pady=10)

self.file_path_button = ttk.Button(self.file_path_frame, text="Browse", command=self.browse_file_path)
self.file_path_button.grid(row=0, column=2, padx=10, pady=10)

# Wasatch parameters
self.wasatch_parameters_frame = ttk.LabelFrame(self.right_frame, text="Parameters")
self.wasatch_parameters_frame.grid(row=2, column=1, padx=10, pady=5, sticky="ew")

# Integration time control
self.integration_time_label = ttk.Label(self.wasatch_parameters_frame, text="Integration Time (ms):")
self.integration_time_label.grid(row=2, column=0, padx=10, pady=5)
self.integration_time_entry = ttk.Entry(self.wasatch_parameters_frame)
self.integration_time_entry.grid(row=2, column=1, padx=10, pady=5)
self.integration_time_entry.insert(tk.END, str(self.integration_time)) # Default value

self.set_integration_time_button = ttk.Button(self.wasatch_parameters_frame, text="Set",
command=self.set_integration_time)
self.set_integration_time_button.grid(row=2, column=2, padx=10, pady=5)

# Scans to average control
self.scans_to_average_label = ttk.Label(self.wasatch_parameters_frame, text="Scans to Average:")
self.scans_to_average_label.grid(row=3, column=0, padx=10, pady=5)
self.scans_to_average_entry = ttk.Entry(self.wasatch_parameters_frame)
self.scans_to_average_entry.grid(row=3, column=1, padx=10, pady=5)
self.scans_to_average_entry.insert(tk.END, str(self.scans_to_average)) # Default value

self.set_scans_to_average_button = ttk.Button(self.wasatch_parameters_frame, text="Set",
command=self.set_scans_to_average)
self.set_scans_to_average_button.grid(row=3, column=2, padx=10, pady=5)

# Boxcar half width control
self.boxcar_half_width_label = ttk.Label(self.wasatch_parameters_frame, text="Boxcar Half Width:")
self.boxcar_half_width_label.grid(row=4, column=0, padx=10, pady=5)
self.boxcar_half_width_entry = ttk.Entry(self.wasatch_parameters_frame)
self.boxcar_half_width_entry.grid(row=4, column=1, padx=10, pady=5)
self.boxcar_half_width_entry.insert(tk.END, str(self.boxcar_half_width)) # Default value

self.set_boxcar_half_width_button = ttk.Button(self.wasatch_parameters_frame, text="Set",
command=self.set_boxcar_half_width)
self.set_boxcar_half_width_button.grid(row=4, column=2, padx=10, pady=5)

# Delay ms control
self.delay_ms_label = ttk.Label(self.wasatch_parameters_frame, text="Delay (ms):")
self.delay_ms_label.grid(row=5, column=0, padx=10, pady=5)
self.delay_ms_entry = ttk.Entry(self.wasatch_parameters_frame)
self.delay_ms_entry.grid(row=5, column=1, padx=10, pady=5)
self.delay_ms_entry.insert(tk.END, str(self.delay_ms)) # Default value

self.set_delay_ms_button = ttk.Button(self.wasatch_parameters_frame, text="Set",
command=self.set_delay_ms)
self.set_delay_ms_button.grid(row=5, column=2, padx=10, pady=5)

# Max spectra control
self.max_spectra_label = ttk.Label(self.wasatch_parameters_frame, text="Max Spectra:")
self.max_spectra_label.grid(row=6, column=0, padx=10, pady=5)
self.max_spectra_entry = ttk.Entry(self.wasatch_parameters_frame)
self.max_spectra_entry.grid(row=6, column=1, padx=10, pady=5)
self.max_spectra_entry.insert(tk.END, str(self.max_spectra)) # Default value

self.set_max_spectra_button = ttk.Button(self.wasatch_parameters_frame, text="Set",
command=self.set_max_spectra)
self.set_max_spectra_button.grid(row=6, column=2, padx=10, pady=5)

# Measuring frame
self.wasatch_measure_frame = ttk.LabelFrame(self.right_frame, text="Measuring")
self.wasatch_measure_frame.grid(row=3, column=1, padx=10, pady=5, sticky="nsew")

```

```

axis")
    self.wasatch_samples_countX_label = ttk.Label(self.wasatch_measure_frame, text="Sample count X
axis")
    self.wasatch_samples_countX_label.grid(row=0, column=1, padx=50, pady=5)
    self.wasatch_samples_countX_entry = ttk.Entry(self.wasatch_measure_frame)
    self.wasatch_samples_countX_entry.grid(row=0, column=2, padx=10, pady=5)
    self.wasatch_samples_countX_entry.insert(tk.END, '10') # Default value

    self.wasatch_samples_countY_label = ttk.Label(self.wasatch_measure_frame, text="Sample count Y
axis")
    self.wasatch_samples_countY_label.grid(row=1, column=1, padx=50, pady=5)
    self.wasatch_samples_countY_entry = ttk.Entry(self.wasatch_measure_frame)
    self.wasatch_samples_countY_entry.grid(row=1, column=2, padx=10, pady=5)
    self.wasatch_samples_countY_entry.insert(tk.END, '10') # Default value

    self.wasatch_dark_button = ttk.Button(self.wasatch_measure_frame, text = "Dark",
command=self.run_dark)
    self.wasatch_dark_button.grid(row=2, column=0, columnspan=2, padx=5, pady=5)

    self.wasatch_light_button = ttk.Button(self.wasatch_measure_frame, text = "Light",
command=self.run_light)
    self.wasatch_light_button.grid(row=2, column=1, columnspan=2, padx=5, pady=5)

    self.toggle_plot_button = ttk.Button(self.wasatch_measure_frame, text="Show plot",
command=self.toggle_plot)
    self.toggle_plot_button.grid(row=2, column=2, columnspan=2, padx=5, pady=5)

    # Progress bar
    self.progress_bar = ttk.Progressbar(self.wasatch_measure_frame, orient="horizontal",
length=300, mode="determinate")
    self.progress_bar.grid(row=3, column=0, columnspan=4, padx=10, pady=5)

    # Progress bar label
    self.progress_label = ttk.Label(self.wasatch_measure_frame, text="0 %", anchor="center")
    self.progress_label.grid(row=3, column=3, padx=10, pady=25)

    self.run_once_description = ttk.Label(self.wasatch_measure_frame, text="Label:")
    self.run_once_description.grid(row=4, column=0, columnspan=1, padx=10, pady=10)

    self.run_once_description = ttk.Entry(self.wasatch_measure_frame, width=30)
    self.run_once_description.grid(row=4, column=1, columnspan=2, padx=10, pady=10)

    self.wasatch_run_button = ttk.Button(self.wasatch_measure_frame, text = "Run once",
command=self.start_measurement_once)
    self.wasatch_run_button.grid(row=5, column=0, columnspan=2, padx=5, pady=5)

    self.wasatch_run_button = ttk.Button(self.wasatch_measure_frame, text = "Run area",
command=self.start_measurement)
    self.wasatch_run_button.grid(row=5, column=1, columnspan=3, padx=5, pady=5)

    self.wasatch_stop_button = ttk.Button(self.wasatch_measure_frame, text = "Stop",
command=self.stop_measurement)
    self.wasatch_stop_button.grid(row=5, column=2, columnspan=2, padx=5, pady=5)

def toggle_plot(self):
    if self.toggle_plot_button["text"] == "Show plot":
        self.toggle_plot_button.config(text="Hide plot")
    else:
        self.toggle_plot_button.config(text="Show plot")
    self.wasatch.toggle_plot()

def update_progress(self, progress):
    self.progress_bar["value"] = progress
    self.progress_label["text"] = str(progress) + ' %'
    self.root.update_idletasks()

def set_integration_time(self):
    integration_time = int(self.integration_time_entry.get())
    self.wasatch.set_integration_time(integration_time)

def set_scans_to_average(self):
    scans_to_average = int(self.scans_to_average_entry.get())
    self.wasatch.set_scans_to_average(scans_to_average)

def set_boxcar_half_width(self):
    boxcar_half_width = int(self.boxcar_half_width_entry.get())

```

```

        self.wasatch.set_boxcar_half_width(boxcar_half_width)

def set_delay_ms(self):
    delay_ms = int(self.delay_ms_entry.get())
    self.wasatch.set_delay_ms(delay_ms)

def set_max_spectra(self):
    max_spectra = int(self.max_spectra_entry.get())
    self.wasatch.set_max_spectra(max_spectra)

def browse_file_path(self):
    file_path = filedialog.asksaveasfilename(defaultextension=".csv", filetypes=[("CSV files", "*.csv"), ("All
files", "*.*")])
    if file_path:
        self.file_path_entry.delete(0, tk.END)
        self.file_path_entry.insert(0, file_path)
        self.wasatch.set_output_file_path(file_path)

def toggle_connection(self):
    if self.serial.connected:
        log_message = self.serial.disconnect_cnc()
        self.connect_button.config(text="Connect")
    else:
        port = self.serial_port_combobox.get()
        log_message = self.serial.connect_cnc(port)
        self.connect_button.config(text="Disconnect")
    self.log(log_message)

def log(self, message):
    self.log_text.insert(tk.END, message + "\n")
    self.log_text.see(tk.END)

def move(self, direction):
    step = float(self.get_step())
    speed = self.get_speed()

    if direction == '\u2191': # Front (Y+)
        self.current_position['Y'] -= step
        command = f'G1 Y{step} F{speed}'
    elif direction == '\u2193': # Rear (Y-)
        self.current_position['Y'] += step
        command = f'G1 Y-{step} F{speed}'
    elif direction == '\u2190': # Left (X-)
        self.current_position['X'] -= step
        command = f'G1 X{step} F{speed}'
    elif direction == '\u2192': # Right (X+)
        self.current_position['X'] += step
        command = f'G1 X-{step} F{speed}'
    elif direction == 'Up': # Up (Z+)
        command = f'G1 Z{step} F{speed}'
    elif direction == 'Down': # Down (Z-)
        command = f'G1 Z-{step} F{speed}'
    else:
        command = ""

    if command:
        log_message = self.serial.send_gcode('G91')
        log_message = self.serial.send_gcode(command)
        self.log(log_message)
        log_message = self.serial.send_gcode('G90')

def get_step(self):
    return self.step_entry.get()

def get_speed(self):
    return self.speed_entry.get()

def send_manual_command(self):
    command = self.manual_command_entry.get()
    log_message = self.serial.send_gcode(command)
    self.log(log_message)

def set_position(self, position_number):
    self.user_positions[str(position_number)] = self.current_position.copy()
    self.log(f"Position {position_number} set to X:{self.current_position['X']}, Y:{self.current_position['Y']}.")

```



```

def test_positions(self):
    # Move to the initial position 0,0 first
    self.serial.send_gcode('G90')

    for pos_number in range(1, 5):
        position = self.user_positions[str(pos_number)]
        if position is not None:
            command = f"G1 X-{position['X']} Y{-position['Y']} F1000"
            self.serial.send_gcode(command)
            self.log(f"Moving to position {pos_number}: X={position['X']}, Y={position['Y']}")
            time.sleep(int(self.get_step())/10)
        # Move back to position 1
        position_1 = self.user_positions['1']
        if position_1 is not None:
            command = f"G1 X-{position_1['X']} Y{position_1['Y']} F1000"
            self.serial.send_gcode(command)
            self.log(f"Returning to position 1: X={position_1['X']}, Y={position_1['Y']}")
            time.sleep(1)
        else:
            self.log("Position 1 is not selected")

def set_position_zero(self):
    self.current_position = {'X': 0, 'Y': 0}
    self.serial.send_gcode('G92 X0 Y0 Z0')
    self.log("Current position set as 0,0.")

def move_to_zero(self):
    init_commands = [
        'G90',
        'G1 X0 Y0 Z0',
        'G91',
    ]
    for cmd in init_commands:
        self.log(self.serial.send_gcode(cmd))
        # time.sleep(1)

    self.log("Complete.")

def start_measurement(self):
    self.stop_measurement()

    if not self.running:
        self.running = True
        # New thread for measuring
        if not self.measure_thread or not self.measure_thread.is_alive():
            self.measure_thread = threading.Thread(target=self.measure_and_move)
            self.measure_thread.start() # Run thread

def start_measurement_once(self):
    run_once_string = self.run_once_description.get()
    if run_once_string == "":
        run_once_string = 'single measure'
    self.wasatch.init_file()
    self.log(run_once_string)
    finished = self.wasatch.run(run_once_string)
    self.wasatch.close_file()

    if finished == False:
        self.running = False
        self.log("Stopped. Measure from wasatch.py return False.")
        return

def stop_measurement(self):
    self.running = False

def measure_and_move(self):
    self.update_progress(0)

    self.samples_count_x = int(self.wasatch_samples_countX_entry.get())
    self.samples_count_y = int(self.wasatch_samples_countY_entry.get())

    step_x = (self.user_positions['2']['X'] - self.user_positions['1']['X']) / (self.samples_count_x - 1)
    step_y = (self.user_positions['4']['Y'] - self.user_positions['1']['Y']) / (self.samples_count_y - 1)

```

```

        # Turn cnc into start point (0,0)
        self.serial.send_gcode('G90')
        move_command = f'G1 X-{self.user_positions["1"]["X"]} Y{self.user_positions["1"]["Y"]}
F{self.get_speed()}'
        self.serial.send_gcode(move_command)
        self.log("Moving to start position")
        self.waitForCNC()

        current_measure = 0
        measure_count = self.samples_count_x * self.samples_count_y

        self.wasatch.init_file()

        while self.running:
            # Move cnc
            for i in range(self.samples_count_x):
                new_x = self.user_positions['1']['X'] + i * step_x
                isChangedX = True
                for j in range(self.samples_count_y):
                    if not self.running:
                        break
                    # New position Y
                    new_y = self.user_positions['1']['Y'] + j * step_y
                    # Move cnc to new position
                    move_command = f'G1 X-{new_x} Y-{new_y} F{self.get_speed()}'
                    self.serial.send_gcode(move_command)
                    self.log(f"Moving to position X: {new_x}, Y: {new_y}")
                    self.waitForCNC()
                    max_step = max(step_x, step_y)
                    self.measureDelayFromSteps(max_step)
                    if(isChangedX):
                        self.measureDelayFromSteps(step_x*self.samples_count_x)
                        isChangedX=False
                    current_measure += 1
                    self.log(f"Measure {current_measure} out of {measure_count}.")
                    finished = self.wasatch.run("%2.4f, %2.4f" % (new_x, new_y))
                    progress = int((current_measure / measure_count) * 100)

                    self.update_progress(progress)
                    if not finished:
                        self.running = False
                        self.log("Stopped. Measure from wasatch.py returned False.")
                        break

                if not self.running:
                    self.log("Stopped.")
                    break

            self.running = False

    def waitForCNC(self):
        while not self.serial.wait_for_ending_move():
            pass

    def measureDelayFromSteps(self, step):
        wait_time_ms = self.interpolate_time(step) + 100
        time.sleep(wait_time_ms/1000)

    def interpolate_time(self, step):
        steps = np.array([1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 100, 120, 150])
        times = np.array([500, 1150, 1560, 2000, 2700, 3300, 3800, 4400, 4900, 5500, 6600, 7800, 9700])
        if step <= steps[0]:
            return times[0]
        else:
            return np.interp(step, steps, times)

    def run_dark(self):
        self.wasatch.init_file()
        self.log(f"Dark measure")
        finished = self.wasatch.run("dark")
        self.wasatch.close_file()

        if finished == False:
            self.running = False
            self.log("Stopped. Measure from wasatch.py return False.")

```

```

        return

    def run_light(self):
        self.wasatch.init_file()
        self.log(f"Light measure")
        finished = self.wasatch.run("light")
        self.wasatch.close_file()

    if finished == False:
        self.running = False
        self.log("Stopped. Measure from wasatch.py return False.")
        return

```

Plik wasatch.py

```

import os
import re
import sys
import time
import numpy
import signal
import psutil
import logging
import datetime
import argparse
import tkinter as tk
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from matplotlib.figure import Figure
import wasatch
from wasatch import utils
from wasatch import applog
from wasatch.WasatchBus import WasatchBus
from wasatch.OceanDevice import OceanDevice
from wasatch.WasatchDevice import WasatchDevice
from wasatch.WasatchDeviceWrapper import WasatchDeviceWrapper
from wasatch.RealUSBDevice import RealUSBDevice
import logging

log = logging.getLogger(__name__)

class Wasatch:
    def __init__(self, root, argv=None):
        self.bus = None
        self.device = None
        self.logger = None
        self.outfile = None
        self.type = "default"
        self.light_spectrum = None
        self.args = self.parse_args(argv)
        self.logger = applog.MainLogger(self.args.log_level)
        print("Wasatch.PY version %s", wasatch.__version__)
        self.root = root

        # Create tkinter window for plot
        self.graph_window = tk.Toplevel(self.root)
        self.graph_window.title("Measurement plot")
        self.fig = Figure(figsize=(5, 4), dpi=100)
        self.ax = self.fig.add_subplot(111)
        self.canvas = FigureCanvasTkAgg(self.fig, master=self.graph_window)
        self.canvas.draw()
        self.canvas.get_tk_widget().pack(side=tk.TOP, fill=tk.BOTH, expand=1)
        self.graph_window.withdraw()

    def set_logger_handler(self, logger_handler):
        self.logger.addHandler(logger_handler)

    def parse_args(self, argv):
        parser = argparse.ArgumentParser(description="Simple demo to acquire spectra from command-line interface")
        parser.add_argument("--log-level", type=str, default="INFO", help="logging level [DEBUG,INFO,WARNING,ERROR,CRITICAL]")
        parser.add_argument("--integration-time-ms", type=int, default=10, help="integration time (ms, default 10)")

```

```

1)"))
        parser.add_argument("--scans-to-average", type=int, default=1, help="scans to average (default
        parser.add_argument("--boxcar-half-width", type=int, default=0, help="boxcar half-width (default 0)")
        parser.add_argument("--delay-ms", type=int, default=1000, help="delay between integrations
(ms, default 1000)")
        parser.add_argument("--outfile", type=str, default=None, help="output filename (e.g.
path/to/spectra.csv)")
        parser.add_argument("--max", type=int, default=0, help="max spectra to acquire (default
0, unlimited)")
        parser.add_argument("--non-blocking", action="store_true", help="non-blocking USB interface
(WasatchDeviceWrapper instead of WasatchDevice)")
        parser.add_argument("--ascii-art", action="store_true", help="graph spectra in ASCII")
        parser.add_argument("--version", action="store_true", help="display Wasatch.PY version and
exit")

```

```

# parse argv into dict
args = parser.parse_args(argv[1:])
if args.version:
    print("Wasatch.PY %s" % wasatch.__version__)
    sys.exit(0)

# normalize log level
args.log_level = args.log_level.upper()
if not re.match("^(DEBUG|INFO|ERROR|WARNING|CRITICAL)$", args.log_level):
    print("Invalid log level: %s (defaulting to INFO)" % args.log_level)
    args.log_level = "INFO"

return args

def connect(self):
    """ If the current device is disconnected, and there is a new device,
    attempt to connect to it. """

    if self.device is not None:
        return

    if self.bus is None:
        print("instantiating WasatchBus")
        self.bus = WasatchBus(use_sim = False)

    if not self.bus.device_ids:
        print("No Wasatch USB spectrometers found.")
        return

    device_id = self.bus.device_ids[0]
    print("connect: trying to connect to %s", device_id)
    device_id.device_type = RealUSBDevice(device_id)

    if self.args.non_blocking:
        print("instantiating WasatchDeviceWrapper (non-blocking)")
        device = WasatchDeviceWrapper(
            device_id = device_id,
            log_queue = self.logger.log_queue,
            log_level = self.args.log_level)
    else:
        print("instantiating WasatchDevice (blocking)")
        if device_id.vid == 0x24aa:
            device = WasatchDevice(device_id)
        else:
            device = OceanDevice(device_id)

    ok = device.connect()
    if not ok:
        print("connect: can't connect to %s", device_id)
        return

    print("connect: device connected")

    self.device = device
    self.reading_count = 0

    return device

def run(self, type):
    self.type = type

```

```

if self.device is None:
    print("Not connected to spectrometer")
    return False

# apply initial settings
self.device.change_setting("integration_time_ms", self.args.integration_time_ms)
self.device.change_setting("scans_to_average", self.args.scans_to_average)
self.device.change_setting("detector_tec_enable", True)

start_time = datetime.datetime.now()
self.attempt_reading()
end_time = datetime.datetime.now()

# compute how much longer we should wait before the next reading
reading_time_ms = int((end_time - start_time).microseconds / 1000)
sleep_ms = self.args.delay_ms - reading_time_ms
if sleep_ms > 0:
    print("sleeping %d ms (%d ms already passed)", sleep_ms, reading_time_ms)
    try:
        time.sleep(float(sleep_ms) / 1000)
    except:
        pass
    return True

def attempt_reading(self):
    try:
        reading_response = self.acquire_reading()
    except Exception as exc:
        print("attempt_reading caught exception", exc_info=1)
        return

    if isinstance(reading_response.data, bool):
        if reading_response.data:
            print("received poison-pill, exiting")
            return
        else:
            print("no reading available")
            return

    if reading_response.data.failure:
        return

    self.process_reading(reading_response.data)

def acquire_reading(self):
    while True:
        reading = self.device.acquire_data()
        if reading is None:
            print("waiting on next reading")
        else:
            return reading

def process_reading(self, reading):
    if self.args.scans_to_average > 1 and not reading.averaged:
        return

    self.reading_count += 1

    if self.args.boxcar_half_width > 0:
        spectrum = utils.apply_boxcar(reading.spectrum, self.args.boxcar_half_width)
    else:
        spectrum = reading.spectrum

    if self.args.ascii_art:
        print("\n".join(wasatch.utils.ascii_spectrum(spectrum, rows=20, cols=80,
x_axis=self.device.settings.wavelengths, x_unit="nm")))
    else:
        spectrum_min = numpy.amin(spectrum)
        spectrum_max = numpy.amax(spectrum)
        spectrum_avg = numpy.mean(spectrum)
        spectrum_std = numpy.std(spectrum)
        size_in_bytes = psutil.Process(os.getpid()).memory_info().rss

```

```

        print("Reading: %10d Detector: %5.2f degC Min: %8.2f Max: %8.2f Avg: %8.2f StdDev:
%8.2f Memory: %11d" % (
        self.reading_count,
        reading.detector_temperature_degC,
        spectrum_min,
        spectrum_max,
        spectrum_avg,
        spectrum_std,
        size_in_bytes))
        print("%s", str(reading))

    # if self.type == "light":
    #     self.light_spectrum = spectrum
    # elif self.type != "dark" and self.light_spectrum is not None:
    #     # spectrum -= self.light_spectrum
    #     spectrum = numpy.subtract( spectrum,self.light_spectrum)

    # print(spectrum)

    if self.outfile:
        self.outfile.write("%s;%s;%5.2f;%s\n" % (self.type,
        datetime.datetime.now(),
        reading.detector_temperature_degC,
        ",".join(format(x, ".2f") for x in spectrum)))

    self.draw_graph(spectrum)
    return

def draw_graph(self, spectrum):
    # Clear previous plot
    self.ax.clear()
    self.ax.plot(self.device.settings.wavelengths, spectrum)
    self.canvas.draw()

def set_output_file_path(self, outfile_path):
    self.args.outfile = outfile_path

    if self.outfile:
        try:
            self.outfile.close()
        except Exception as e:
            print("Error closing previous outfile: %s", str(e))

    try:
        file_exists = os.path.isfile(outfile_path)
        file_has_data = file_exists and os.path.getsize(outfile_path) > 0

        if file_has_data:
            self.outfile = open(outfile_path, "a")
        else:
            self.outfile = open(outfile_path, "w")
            self.outfile.write("type,time,temp,%s\n" % ",".join(format(x, ".2f") for x in
self.device.settings.wavelengths))

        print('Filepath set to: %s', outfile_path)
    except Exception as e:
        print("Error initializing %s: %s", outfile_path, str(e))
        self.outfile = None

def set_integration_time(self, integration_time_ms):
    self.args.integration_time_ms = integration_time_ms
    print('Inetration time set to %i ms', integration_time_ms)

def set_scans_to_average(self, scans_to_average):
    self.args.scans_to_average = scans_to_average
    print('Scans to average set to %i', scans_to_average)

def set_boxcar_half_width(self, boxcar_half_width):
    self.args.boxcar_half_width = boxcar_half_width
    print('Boxcar half width set to %i', boxcar_half_width)

def set_delay_ms(self, delay_ms):
    self.args.delay_ms = delay_ms
    print('Delay set to %i ms', delay_ms)

```

```

def set_max_spectra(self, max_spectra):
    self.args.max = max_spectra
    print('Max spectra set to %i', max_spectra)

def init_file(self):
    if self.args.outfile:
        try:
            file_exists = os.path.isfile(self.args.outfile)
            file_has_data = file_exists and os.path.getsize(self.args.outfile) > 0

            if file_has_data:
                self.outfile = open(self.args.outfile, "a")
            else:
                self.outfile = open(self.args.outfile, "w")
                self.outfile.write("type;time;temp;%s\n" % " ".join(format(x, ".2f") for x in
self.device.settings.wavelengths))

        except Exception as e:
            print(f"Error initializing {self.args.outfile}: {e}")
            self.outfile = None

def close_file(self):
    if self.args.outfile:
        self.outfile.close()

def init_file_without_header(self):
    if self.args.outfile:
        try:
            self.outfile = open(self.args.outfile, "w")
        except:
            print("Error initializing %s", self.args.outfile)
            self.outfile = None

def toggle_plot(self):
    if self.graph_window.winfo_ismapped():
        self.graph_window.withdraw() # Hide plot
    else:
        self.graph_window.deiconify() # Show plot

def signal_handler(signal, frame):
    print('\nInterrupted by Ctrl-C...shutting down', end=' ')
    clean_shutdown()

def clean_shutdown():
    print("Exiting")
    if demo:
        if demo.args and demo.args.non_blocking and demo.device:
            print("closing background thread")
            demo.device.disconnect()

        if demo.logger:
            print("closing logger")
            print(None)
            demo.logger.close()
            time.sleep(1)
            applog.explicit_log_close()

        if demo.outfile:
            print("closing outfile")
            demo.outfile.close()
    sys.exit()

demo = None
if __name__ == "__main__":
    signal.signal(signal.SIGINT, signal_handler)

    demo = Wasatch(sys.argv)
    if demo.connect():
        print("Press Control-Break to interrupt...")
        demo.run()

    clean_shutdown()

```

Dodatek D:Kod programu MATLAB

Wykresy czasu integracji

```
% Inicjalizacja tablicy do przechowywania danych
all_data = [];

% Tablica z numerami plików
file_numbers = [1, 2, 3, 5, 10, 20, 30, 50, 100, 200, 300, 500, 1000, 3000];

% Wybieramy 14 plików z tablicy file_numbers
files_to_use = file_numbers(1:14);

for i = 1:numel(files_to_use)
    % Tworzenie nazwy pliku
    filename = ['integration_time_dark/', num2str(files_to_use(i)), '.csv'];
    % Wczytanie danych z pliku CSV
    data = readmatrix(filename);
    % Wybór odpowiednich kolumn
    start_column = 3;
    column_step = 4; % Krok pomiędzy kolejnymi kolumnami
    num_columns = 30; % Łączna liczba wybranych kolumn
    selected_columns = start_column : column_step : start_column + (num_columns - 1) * column_step; %
Indeksy wybranych kolumn
    selected_data = data(243, selected_columns); % Wybór danych z wiersza 230 i wybranych kolumn
    all_data = [all_data; selected_data]; % Dodanie nowych danych do tablicy
end

% Obliczenie średniej ze wszystkich wczytanych danych
average_data = mean(all_data); % Obliczenie średniej
averages = zeros(numel(files_to_use), num_columns); % Tworzenie macierzy na średnie dla każdego
zestawu danych
for i = 1:numel(files_to_use)
    % ... (wczytanie danych z pliku i wybór odpowiednich kolumn, jak wcześniej)

    % Obliczenie średniej z aktualnego zestawu danych i zapisanie jej
    averages(i, :) = mean(all_data(i:numel(files_to_use):end, :)); % Obliczenie średniej dla danego zestawu
danych
end
figure(1)
plot(averages(1:14), '-');

xlabel('Czas integracji [ms]');
ylabel('Transmisja [%]');
xticks(1:num_columns);
xticklabels(string(file_numbers));

grid on;
hold off;
```

Wykresy pojedynczych próbek

```
% Nazwa pliku CSV
% filename = 'citrus/cytryna_krawedz_integr_1000ms.csv';
% filename = 'citrus/cytryna_integr_1000ms.csv';
% dark_filename = 'citrus/dark_integr_1000ms_.csv';
% light_filename = 'citrus/light_int_1000ms_.csv';
% plot_file = 'cytryna_probki';

filename = 'bacon/boczek_flaga_int_200ms_10x10_step20x20.csv';
% filename = 'bacon/boczek_int_200ms_20x20_step20x20.csv';
dark_filename = 'bacon/dark.csv';
light_filename = 'bacon/light_200ms.csv';
plot_file = 'boczek_probki';

% Wczytanie całego pliku CSV
all_data = readmatrix(filename);
% Pominięcie pierwszego wiersza i pierwszych trzech kolumn
data = all_data(2:end, 4:end);
x_range = all_data(1, 4:end);

% Wczytanie całego pliku CSV dla dark
```



```

all_dark = readmatrix(dark_filename);
% Pomińcie pierwszą wiersz i pierwszych trzech kolumn
dark = all_dark(2, 4:end);

% Wczytanie całego pliku CSV dla light
all_light = readmatrix(light_filename);
% Pomińcie pierwszą wiersz i pierwszych trzech kolumn
light = all_light(2:end, 4:end);

% Obliczenie transmisji jako procentu
transmission = 100*(data - dark) ./ (light);

% Obliczenie absorbancji na podstawie transmisji
absorbance = -log10(transmission / 100);

% Wybór próbek do wykreślenia
index1 = 19;
index2 = 395;
% index3 = 1;

% Wykres absorbancji dla wybranych próbek
figure(1)
plot(x_range, absorbance(index1,:), '-', 'DisplayName', 'Próbka 1');
hold on;
plot(x_range, absorbance(index2,:), '-', 'DisplayName', 'Próbka 2');
% plot(x_range, absorbance(index3,:), '-', 'DisplayName', 'Próbka 3');
% Dodanie legendy i opisów osi
legend('show', 'Location', 'northwest');
xlabel('Długość fali [nm]');
ylabel('Absorbancja');
hold off;

% Wykres transmisji dla wybranych próbek
figure(2)
plot(x_range, transmission(index1,:), '-', 'DisplayName', 'Próbka 1');
hold on;
plot(x_range, transmission(index2,:), '-', 'DisplayName', 'Próbka 2');
% Dodanie legendy i opisów osi
legend show;
xlabel('Długość fali [nm]');
ylabel('Transmisja [%]');
hold off;

```

Wykresy i analiza PCA

```

% Nazwa pliku CSV
% filename = 'bacon/boczek_flaga_int_200ms_10x10_step20x20.csv';
% filename = 'bacon/boczek_int_200ms_20x20_step20x20.csv';
dark_filename = 'bacon/dark.csv';
light_filename = 'bacon/light_200ms.csv';
filename = 'citrus/cytryna_integr_1000ms.csv';
% filename = 'citrus/cytryna_krawedz_integr_1000ms.csv';
% dark_filename = 'citrus/dark_integr_1000ms_.csv';
% light_filename = 'citrus/light_int_1000ms_.csv';

% Wczytanie całego pliku CSV
all_data = readmatrix(filename);
% Pomińcie pierwszą wiersz i pierwszych trzech kolumn
data = all_data(2:end, 4:end);
x_range = all_data(1, 4:end);

% Wczytanie całego pliku CSV dla dark
all_dark = readmatrix(dark_filename);
% Pomińcie pierwszą wiersz i pierwszych trzech kolumn
dark = all_dark(2, 4:end);

% Wczytanie całego pliku CSV dla light
all_light = readmatrix(light_filename);
% Pomińcie pierwszą wiersz i pierwszych trzech kolumn
light = all_light(2:end, 4:end);

% Indeksy wierszy, które chcemy wykreślić
rows_to_plot = [10, 100, 200, 400];

% Przesunięcia dla każdej serii danych (można dostosować wartości)

```

```

offsets = [0, 10000, 20000, 30000];

% Wykresy dla poszczególnych wierszy z dodanym przesunięciem
figure(1)
plot(x_range, data(rows_to_plot(1), :) + offsets(1), '-', 'DisplayName', 'Próbka 10');
hold on;
plot(x_range, data(rows_to_plot(2), :) + offsets(2), '-', 'DisplayName', 'Próbka 100');
plot(x_range, data(rows_to_plot(3), :) + offsets(3), '-', 'DisplayName', 'Próbka 200');
plot(x_range, data(rows_to_plot(4), :) + offsets(4), '-', 'DisplayName', 'Próbka 400');

% Dodanie legendy i opisów osi
legend show;
xlabel('Długość fali [nm]');
hold off;

% Odjęcie wartości dark i podzielenie przez light
data = (data - dark) ./ (light);

% Przygotowanie danych SNV
data_snv = snv(data);

% Wykresy dla poszczególnych wierszy SNV z dodanym przesunięciem
figure(2)
plot(x_range, data_snv(rows_to_plot(1), :) + offsets(1)/10000, '-', 'DisplayName', 'Próbka 10');
hold on;
plot(x_range, data_snv(rows_to_plot(2), :) + offsets(2)/10000, '-', 'DisplayName', 'Próbka 100');
plot(x_range, data_snv(rows_to_plot(3), :) + offsets(3)/10000, '-', 'DisplayName', 'Próbka 200');
plot(x_range, data_snv(rows_to_plot(4), :) + offsets(4)/10000, '-', 'DisplayName', 'Próbka 400');

% Dodanie legendy i opisów osi
legend show;
xlabel('Długość fali [nm]');
hold off;

% Przygotowanie danych do PCA
[PCAccoeff,PCAScore,PCALatent,PCAtsquared,PCAexplained,PCAmu] = pca(data);

% Wyświetlenie składowych PC
% Ilość składowych do wykresu
pca_count = 5;

figure(3)

plotmatrix(PCAScore(:,1:pca_count))
label = sprintf('PC1 - PC%d', pca_count);
title(label);

% Indeksy składowych do wykresów
pca_index1 = 1;
pca_index2 = 2;

% Wyświetlenie wag
figure(4)
scatter(PCAScore(:,pca_index1),PCAScore(:,pca_index2))
label = sprintf('PC%d/PC%d', pca_index1, pca_index2);
title(label);

% Wyświetlanie PCAexplained
figure(5)
bar(PCAexplained(1:5));
hold on;
plot(cumsum(PCAexplained(1:5)), '-o', 'Color', 'r'); % Linia kumulacyjna
ylim([0 105]);
xlabel('Główne składowe');
ylabel('Procent wyjaśnionej wariancji [%]');
legend('Wyjaśniona wariancja', 'Kumulacyjna wariancja', 'Location', 'best');
hold off;

```