

Politechnika Wrocławska

Wydział Elektroniki, Fotoniki i Mikrosystemów

KIERUNEK: Automatyka i Robotyka (AIR)

PRACA DYPLOMOWA INŻYNIERSKA

TYTUŁ PRACY:
Analiza i rozpoznawanie dźwięków na
potrzeby interakcji człowiek-robot

AUTOR:
Olga Meder

PROMOTOR:
Dr inż. Robert Muszyński,
Katedra Cybernetyki i Robotyki

*Wszystkim, którzy wspiera-
ją moje robotyczne i muzyczne
przygody*

Streszczenie

Niniejsza praca opisuje budowę systemu służącego do identyfikacji wysokości zagranych dźwięków i porównywania jej do wartości wzorcowej. Celem pracy było zapoznanie się z dostępnymi rozwiązaniami z zakresu przetwarzania sygnałów dźwiękowych oraz implementacja programu do analizy ewentualnych błędów podczas gry na instrumentach muzycznych w środowisku MATLAB. Dla wygody użytkownika, zaprojektowano aplikację graficzną przedstawiającą funkcjonalności systemu. W ramach testów, przeprowadzono badania na gitarze klasycznej, pianinie elektrycznym oraz podczas śpiewania. Wyznaczano częstotliwości podstawowe zagranych dźwięków oraz porównano je z wgranymi wartościami wzorcowymi. Zaproponowany system w przyszłości może służyć jako robot społeczny w roli nauczyciela muzyki lub stanowić podstawę do opracowania bardziej zaawansowanych struktur do analizy dźwięku.

Słowa kluczowe: częstotliwość dźwięku, przetwarzanie sygnałów, FFT, MATLAB

Abstract

This paper entails an in-depth description and an extensive evaluation of a system aiming for an identification of the sound pitch of a played melody and its comparison with the built-in melody pattern. The purpose of this study is to provide a brief overview of the current solutions available in the field of sound signal processing, as well as to implement an analysis program in order to detect possible errors while playing musical instruments in the MATLAB environment. Additionally, in order to simplify the user's experience, an application visually displaying the functionalities of the system has been developed. Furthermore, several tests were carried out on a classical guitar, electric piano, and while singing as part of the system's final examination. Respectively, the fundamental frequencies of the played sounds have been identified and compared with the uploaded reference values. In the future, this system may be implemented as a social robot acting in the role of a music teacher or may serve as the basis for developing more advanced sound analysis structures.

Keywords: sound frequency, signal processing, FFT, MATLAB

Spis treści

1	Wstęp	3
2	Aplikacje komercyjne	5
2.1	Note Recognizer	6
2.2	Spek Acoustic Spectrum Analyser	6
2.3	Porównanie aplikacji	7
2.3.1	Podobieństwo	7
2.3.2	Różnice	8
3	Rozpoznawanie dźwięków	9
3.1	Podział nagrania na dźwięki	11
3.1.1	Metoda ręczna	11
3.1.2	Metoda automatyczna	11
3.2	Rozpoznanie częstotliwości podstawowej	14
3.2.1	Transformacja Fouriera	14
4	Struktura systemu	16
4.1	Tryb analizy zapisanych utworów	16
4.2	Tryb pracy na żywo	17
5	Implementacja	19
5.1	Rejestrowanie dźwięków	19
5.2	Dzielenie utworów na pojedyncze dźwięki	20
5.3	Określenie częstotliwości za pomocą FFT	22
5.4	Porównanie gry z wzorcem	23
5.5	Interfejs graficzny aplikacji	24
6	Testy	26
6.1	Działanie systemu	26
6.2	Pianino	30
6.3	Gitara	31
6.4	Śpiew	32
6.5	Wnioski	34
7	Podsumowanie	35
	Literatura	37
	Spis tabel	40
	Spis rysunków	41
A	Częstotliwości tonów podstawowych i ich harmonicznyc	42

Rozdział 1

Wstęp

Mikołaj Kubicki znany szerzej jako *Meek, Oh Why?* [Wike] w swoim utworze zatytułowanym „*Jestem dźwiękiem i słowem*” opisuje otaczające go dźwięki jako nierozłączny element egzystencji człowieka [Mee]. Przedstawia siebie jako zbiór napotkanych przez całe życie słów i dźwięków oraz pokazuje ich wpływ na codzienne funkcjonowanie. Niniejsza praca powstała w wyniku podobnych rozważań i próby analizowania muzyki oraz napotkanych dźwięków pod kątem technicznym. Biorąc pod uwagę, jak ważnym elementem funkcjonowania jest dźwięk, oraz chcąc połączyć ze sobą dwie pasje: muzyki i robotyki, stworzono wizję systemu służącego do identyfikacji wysokości zagranych dźwięków i porównywania jej do wartości wzorcowej.

Zaproponowany system może posłużyć jako część implementacyjna robota społecznego występującego w roli nauczyciela muzyki i pomóc przy nauce gry na instrumentach. Robot ten mógłby przejąć kłopotliwe zadanie osądzania ucznia lub pomagać dyrygentom prowadzić orkiestrę. Miałby zastosowanie w każdym procesie nauki muzyki. Mógłby nawet nucić zadane melodie i przeprowadzać rozgrzewki wokalne lub zapisywać i analizować próby zespołów muzycznych. Przydałby się również przy rozpoznawaniu dźwięków podczas tworzenia własnych kompozycji, co znacząco ułatwiłoby pracę muzykom podczas tworzenia własnych utworów.

Podstawowymi cechami dźwięku są: wysokość, głośność, czas trwania i barwa [Wikb]. W systemie skupiono się wyłącznie na rozpoznaniu wysokości oraz określeniu czasu trwania dźwięku. Dodatkowym ograniczeniem jest okrojony przedział mierzonej częstotliwości. Wynika to z faktu, że nie ma potrzeby sprawdzania wartości poniżej 10 Hz oraz powyżej 1kHz, ponieważ przedział wysokości dźwięków badanych instrumentów mieści się w określonych wartościach [labb].

Praca przedstawia znalezione, dostępne rozwiązania i aplikacje komercyjne, na podstawie których opracowano strukturę systemu. Opisuje również przydatną teorię muzyki i wprowadza pojęcia, niezbędne do zrozumienia przetwarzania sygnałów dźwiękowych. Zdefiniowany został również problem podziału całego utworu na pojedyncze dźwięki oraz rozpoznanie ich częstotliwości podstawowych. W pracy znajduje się opis analizy fourierowskiej [Zie05, Mar] – narzędzia matematycznego, wykorzystywanego podczas transformacji dziedzinę czasu w dziedzinę częstotliwości.

Podczas pracy systemu wydzielono dwa tryby: tryb analizy wcześniej zapisanych utworów oraz tryb pracy na żywo. Zdecydowano się na taki podział, ponieważ mając

mechanizm do rozpoznawania dźwięków, warto wykorzystać go nie tylko do grania w rzeczywistości, ale również do analizowania błędów z wcześniej zapisanych utworów. Nie zawsze zachodzi potrzeba analizowania całego utworu. Rozdzielając strukturę systemu na dwa osobne tryby, wprowadza się dodatkową funkcjonalność i użytkownik może podjąć decyzję, czy chce uzyskać informację na temat wcześniej zagranej utworu (lub jego fragmentu), czy też w czasie rzeczywistym otrzymywać informację zwrotną o każdym, pojedynczym dźwięku.

W pracy opisano implementację systemu oraz przeprowadzone testy. Skupiono się na 3 różnych instrumentach: pianinie, gitarze oraz śpiewie *. Zdecydowano się na taki wybór, ponieważ są to najpopularniejsze i szeroko dostępne instrumenty muzyczne. Nie wyklucza się dalszej pracy nad rozwojem systemu i dodaniem innych instrumentów do analizy.

Układ pracy jest następujący. W rozdziale 2 zostały opisane aplikacje komercyjne. Wśród wybranych aplikacji, szczegółowo opisano i porównano dwie z nich. Rozdział 3 jest wstępem teoretycznym wprowadzającym niezbędne pojęcia i opisy metod wykorzystywanych w strukturze. Rozdział 4 przedstawia strukturę systemu podczas pracy w dwóch trybach: analizy wcześniej zapisanych utworów oraz pracy na żywo. W rozdziale 5 została opisana implementacja systemu w środowisku MA-TLAB. Rozdział 6 przedstawia testy i wyniki przeprowadzonych badań. Rozdział 7 jest podsumowaniem pracy i opisem planów na rozbudowę systemu.

*śpiew traktujemy jako instrument

Rozdział 2

Aplikacje komercyjne

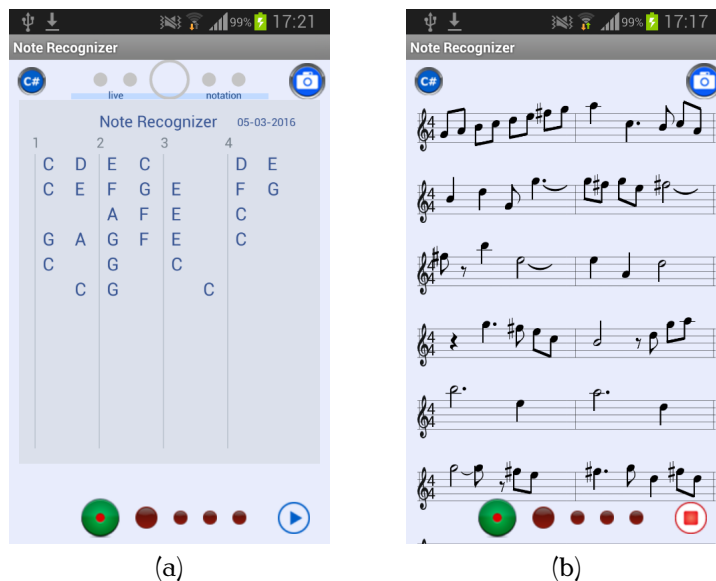
Aplikacje rozpoznające wysokość zagranych lub śpiewanych dźwięków znacznie ułatwiają pracę z muzyką i cieszą się coraz większą popularnością. Pojawiają się rozwiązania, które nie tylko przekazują informacje na temat wysokości dźwięku, ale od razu przetwarzają je na zapis nutowy lub konwertują do wygodnych dla użytkownika formatów danych. W wyborze odpowiedniej aplikacji, ważnym aspektem jest dokładność rozpoznawania dźwięków oraz łatwość i wygoda użytkowania.

Choć na rynku znajduje się już dużo rozwiązań powiązanych z problematyką rozpoznawania i analizy dźwięków, takich jak analizatory widma [[laba](#)], detektory dźwięku [[mf](#)], czy programy do rozpoznawania utworów [[Sha](#)], to wciąż brakuje aplikacji do przetwarzania całych, nagranych utworów i porównywania ich ze wzorcem.

Pod względem działania systemu, najbardziej zbliżone jest urządzenie zwane dalej stroikiem (*ang. tuner*) [[Set05](#)]. Stroik służy do dostrajania instrumentów (najczęściej strunowych) poprzez wykrywanie rozbieżności pomiędzy wartościami wzorcowymi a zagranyymi. Urządzenie lub aplikacja informuje muzyka, czy zagrany dźwięk jest za wysoki, za niski, czy poprawny. Stroik nie dostarcza jednak funkcji wgrywania swojego wzorca. Bazuje na wbudowanych na stałe konkretnych wartościach wzorcowych. Wśród aplikacji realizujących funkcję strojenia można wymienić:

- Instrument Tuner online [[the](#)],
- DaTuner 3.105 [[LLC](#)],
- Pro guitar tuner (tylko do gitary) [[Pro](#)].

Na rynku komercyjnym znajduje się dużo aplikacji polegających na przetwarzaniu dźwięków m.in. Music Tutor [[mus](#)], SoundHound [[Sou](#)], Note Recognizer [[Pla](#)], Guitar Tuner [[um](#)] czy Spek Acoustic [[spe](#)]. Różnią się one dostępnością źródeł, interfejsem graficznym czy zapisem otrzymanych wyników. Z dostępnych aplikacji, zdecydowano się na porównanie i szczegółową analizę dwóch: Note Recognizer oraz Spek Acoustic Spectrum Analyser. Obie aplikacje mają wysoką skuteczność działania. Różnią się jednak funkcjonalnościami i podejściem do przetwarzania sygnałów dźwiękowych.



Rysunek 2.1 Aplikacja Note Recognizer a) zapis dźwięków w postaci nazw zwyczajowych b) zapis dźwięków w postaci notacji muzycznej [Pla]

2.1 Note Recognizer

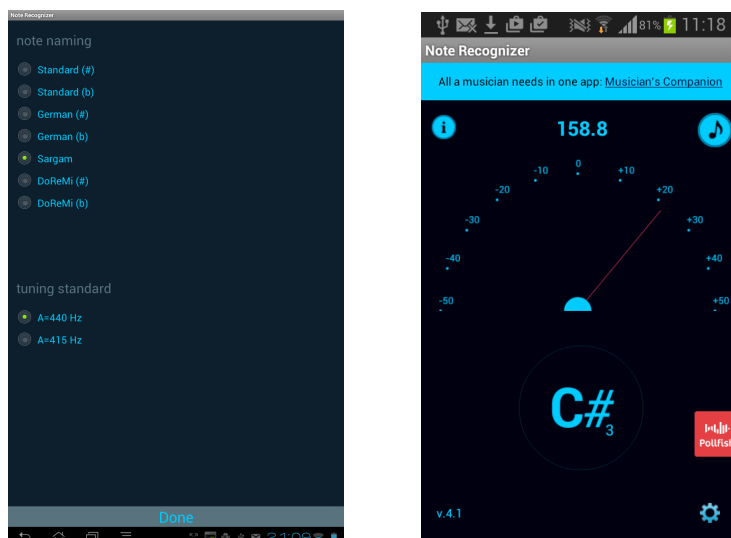
Aplikacja Note Recognizer [Pla] jest obecnie jedną z najpopularniejszych i najlepszych dostępnych na rynku aplikacji rozpoznających wysokości dźwięków. Jest aplikacją płatną, jednak twórcy udostępniają wersję demo, w której niektóre funkcje są niedostępne. Note Recognizer zapisuje wyniki w postaci notacji muzycznej, jak i w postaci nazw zwyczajowych dźwięków (rysunek 2.1).

Aplikacja umożliwia wybór różnych standardów notacji, dwa rodzaje strojenia, transpozycji* i kilka innych funkcji, dotyczących teorii muzyki (zobacz rysunek 2.2). Pozwala rejestrować dźwięki oraz zatrzymywać i wznawiać nagranie. W wersji płatnej istnieje także możliwość eksportowania plików MIDI. W głównym oknie aplikacji zamieszczony jest stroik w postaci wskaźnika sygnalizującego odchylenie od wartości szukanej częstotliwości. Podczas przeprowadzonych testów, aplikacja w większości dobrze rozpoznała wysokość dźwięków.

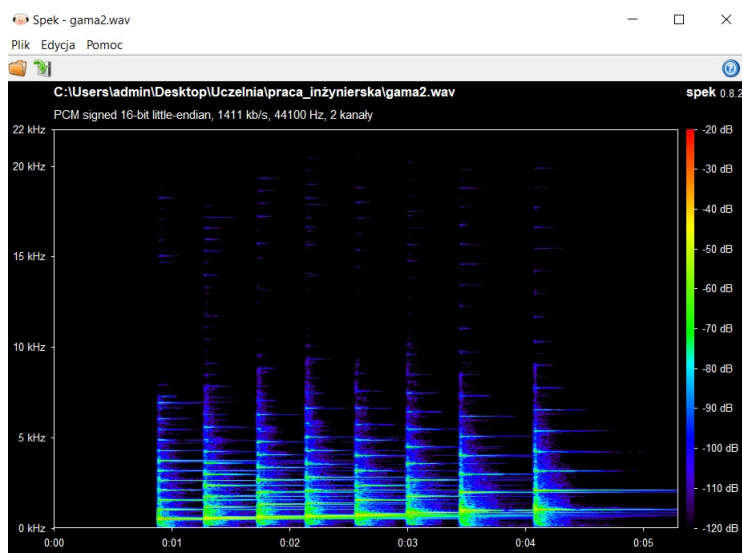
2.2 Spek Acoustic Spectrum Analyser

Aplikacja Speak Acoustic Spectrum Analyser [spe] podchodzi inaczej do tematu analizowania dźwięku. Jej główną funkcjonalnością jest wyznaczanie spektrogramu plików audio. Posiada prosty interfejs graficzny (zobacz rysunek 2.3). Obsługuje większość formatów audio. Ma automatyczną regulację gęstości widma. Skierowana jest jednak do odbiorców bardziej zainteresowanych przetwarzaniem sygnałów niż rozpoznawaniem dźwięków. Dużo sprawniej analizuje całe utwory niż dźwięki zarejestrowane za pomocą urządzenia, na którym jest zainstalowana. Jest bezpłatną

*Różne standardy notacji, strojenia i transpozycja wychodzą poza ramy tej pracy i nie będą szerzej omawiane



Rysunek 2.2 Interfejs graficzny aplikacji Note Recognizer [Pla]



Rysunek 2.3 Spek Acoustic Spectrum Analyser

aplikacją open source'ową, napisaną w języku C++. Kod został udostępniony przez twórcę na platformie GitHub [ale].

2.3 Porównanie aplikacji

2.3.1 Podobieństwo

Obie aplikacje przeznaczone są do analizy i przetwarzania dźwięków. Są wyposażone w funkcjonalności ułatwiające pracę podczas tworzenia muzyki. Ich interfejsy graficzne są proste i intuicyjne. Pod względem jakości i dokładności, wyróżniają się na tle innych aplikacji komercyjnych. Obie aplikacje są wciąż rozbudowywane.

Twórcy dostarczają kolejne wersje, z dodatkowymi udogodnieniami i funkcjonalnościami.

2.3.2 Różnice

Każda z opisanych wyżej aplikacji skierowana jest do innych odbiorców. Note Recognizer jest przeznaczona głównie dla muzyków, którzy potrzebują w jak najszybszy sposób zapisać utworzoną melodię lub rozpoznać wysokości zagranych dźwięków. Dostarcza dużo funkcjonalności dotyczących teorii muzyki, lecz niektóre z nich są płatne. Spek Acoustic to aplikacja całkowicie bezpłatna. Sprawdza się bardziej przy technicznym podejściu do przetwarzania sygnałów. Główną funkcjonalnością jest analiza spektrogramu. W stosunku do Note Recognizer dostarcza możliwość analizowania całych utworów, jak i pojedynczych dźwięków. Jest również aplikacją open source'ową i w przeciwieństwie do Note Recognizer istnieje dostęp do dokumentacji technicznej i możliwość analizy wykorzystanych funkcji.

Rozdział 3

Rozpoznawanie dźwięków

Dźwięk jest ciągłym sygnałem rozchodzącym się w ośrodku sprężystym, w postaci drgań. Podstawowymi cechami dźwięku są [Wikb]:

- wysokość,
- głośność,
- czas trwania,
- barwa.

Wysokość dźwięku określamy przez podanie jego częstotliwości podstawowej. Głośność jest określona przez amplitudę drgań. Czas trwania definiuje okres, w którym dźwięk wybrzmiewa, natomiast barwa dźwięku jest zależna od widma i wpływa na jego subiektywny odbiór. W dźwięku można wyróżnić częstotliwość podstawową i jej harmoniczne (zobacz rysunek 3.1). Taki szereg harmoniczny to szereg składowych, których kolejne częstotliwości są wielokrotnościami częstotliwości podstawowej.

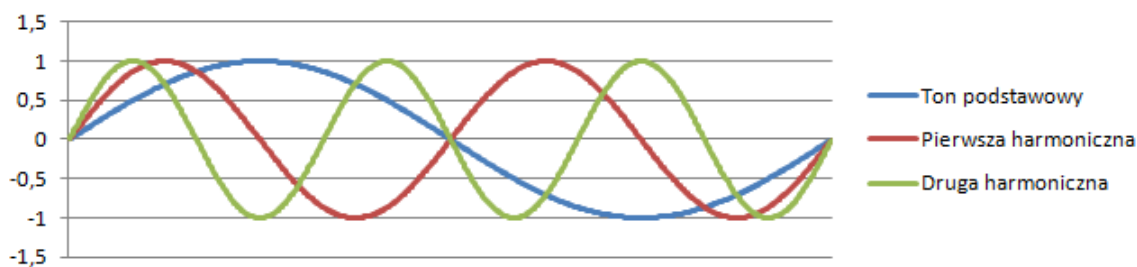
Utwór muzyczny składa się z dźwięków granych w określonym tempie i tworzących jego linie melodyczne [Wikh]. Linią melodyczną nazywamy zbiór dźwięków o określonej wysokości i czasie trwania. Wymienione cechy utworu pozwala opisać notacja muzyczna lub zapis nutowy [Muz]. Do określenia tempa wykorzystujemy miarę BPM (*ang. Beats Per Minute*), czyli liczbę uderzeń lub ćwierćnut na minutę [Wika]. Im wyższe BPM, tym większe tempo utworu.

W zapisie nutowym podstawową jednostką czasu trwania dźwięku jest cała nuta (zobacz rysunek 3.2). Całą nutę możemy podzielić na dwie półnuty. Kolejno półnuty dzielimy na ćwierćnuty, te następnie na ósemki, szesnastki itd.

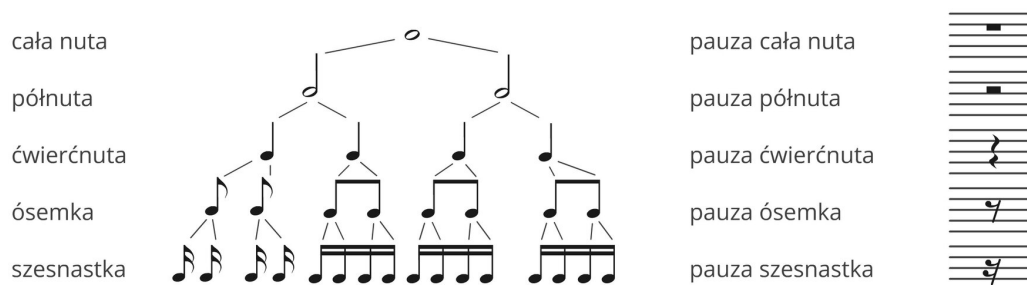
W notacji muzycznej najmniejszą jednostką odległości pomiędzy dwoma dźwiękami jest półton* [Hab68]. Potocznie, częściej stosuje się pojęcie tonu, czyli odległości odpowiadającej różnicy dwóch półtonów. Podstawą do określenia wysokości poszczególnych półtonów jest częstotliwość 440 Hz, która umownie została nazwana dźwiękiem A4 (zobacz rysunek A.1). Zależności pomiędzy odległościami kolejnych półtonów określa zależność

$$f = 440 \cdot 2^{n/12} \text{ Hz}, \quad (3.1)$$

*tak przyjmuje się w notacji europejskiej, oczywiście można wyróżnić mniejsze odległości dźwięków, jednak są rzadko stosowane.



Rysunek 3.1 Składowe harmoniczne dźwięku [CDu]



Rysunek 3.2 Notacja muzyczna [Edu]

gdzie n to odległość/liczba półtonów od wysokości A_4 , f – częstotliwość szukanego dźwięku. Przykładowo, częstotliwość dźwięku E_5 , którego odległość od A_4 to 7 półtonów, wynosi

$$f_{E_5} = 440 \cdot 2^{7/12} = 659.26 \text{ Hz.} \quad (3.2)$$

Czysty dźwięk o zadanej częstotliwości stanowi przebieg sinusoidalny. Przebiegi sinusoidalne są przebiegami okresowymi i cechują się pewną powtarzalnością. W muzyce, powtarzającym się interwałem jest oktawa. Składa się ona z 12 półtonów, zaczynając od dźwięku C [labb, Stu] (zobacz rysunek A.1). Odległość pomiędzy kolejnymi oktawami, oznacza dwukrotność częstotliwości podstawowej

$$C_2 = 2 \cdot C_1 = 32.7 \cdot 2 = 65.41 \text{ Hz.} \quad (3.3)$$

W podstawowej wersji, rozpoznawanie dźwięku polega na określeniu jego czasu trwania oraz wyznaczeniu częstotliwości jego tonu podstawowego. Aby rozpoznać dźwięk z wykorzystaniem dostępnych dziś technologii komputerowych, należy go przekształcić do postaci cyfrowej (dyskretnej) w procesie jego próbkowania. Próbkowanie to proces pomiaru sygnału w określonych punktach czasu [HiF]. Częstość próbkowania wpływa na dokładność odzwierciedlenia sygnału ciągłego. Przy dobrze częstotliwości, należy pamiętać o uwzględnieniu twierdzenia Kotelnikowa-Shannona [Kon, Sza03]:

Twierdzenie 1 Częstotliwość próbkowania F_s musi być większa niż dwukrotność najwyższej składowej częstotliwości w mierzonym sygnale.

Częstotliwość próbkowania powinna być co najmniej dwa razy większa od największej częstotliwości składowej obecnej w próbkowanym sygnale. Inaczej, podczas odwrotnego procesu odtworzenia oryginalnego sygnału, pojawią się błędy (zjawisko to nazywane jest aliasingiem). Najczęściej spotykaną częstotliwością próbkowania jest częstotliwość 44000 [Hz]. Używa się jej między innymi do nagrywania płyt CD.

Sygnały dyskretne można przetwarzać i poddawać analizie widmowej. Poniżej przedstawiono opis teoretyczny procesu dzielenia całego utworu na pojedyncze dźwięki oraz rozpoznawania ich częstotliwości podstawowych. Opisano metody podziału utworów oraz metody detekcji sygnałów. Przedstawiono również matematyczną definicję transformacji Fouriera.

3.1 Podział nagrania na dźwięki

Z perspektywy dalszych działań, podział na osobne dźwięki jest niezwykle istotny. Jako, że nie da się prawidłowo odczytać poszczególnych częstotliwości dźwięków, poddając analizie częstotliwościowej cały utwór (zobacz rysunek 3.3), należy go podzielić na osobne dźwięki. Zapewnia to odczyt częstotliwości tonu podstawowego danego dźwięku oraz jego harmonicznym. Do podziału nagrania na pojedyncze dźwięki można podejść na dwa sposoby: wyznaczyć ręcznie początek i koniec każdego dźwięku lub określić jego czas trwania w sposób automatyczny.

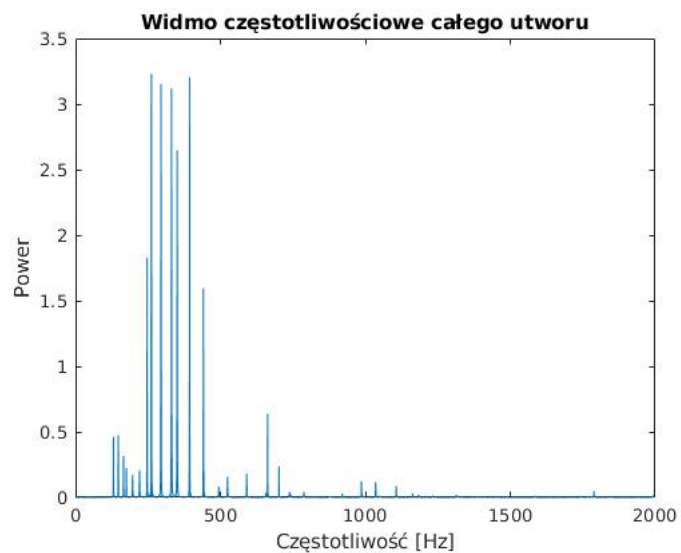
3.1.1 Metoda ręczna

Metoda ręczna polega na manualnym wyznaczaniu początków i końców dźwięków w utworze. Z wykresu przebiegu amplitudowego całego utworu, odczytuje i zapisuje się indeksy próbek rozpoczęcia oraz indeksy próbek zakończenia pojedynczych dźwięków, co pokazano na rysunku 3.4. Na koniec następuje podział utworu na fragmenty z pojedynczymi dźwiękami.

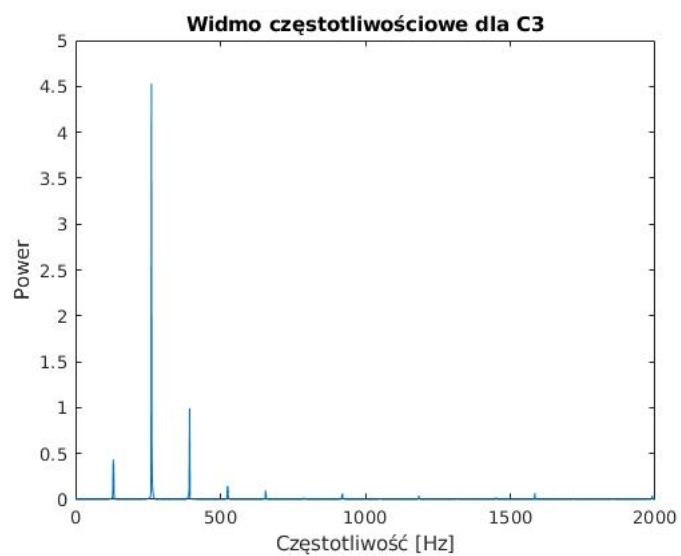
3.1.2 Metoda automatyczna

Automatyczne rozpoznawanie czasu trwania dźwięku jest bardziej skomplikowanym procesem. Należy zwrócić uwagę nie tylko na sam podział utworu na pojedyncze dźwięki, ale pamiętać również o automatycznym zapisie danych. W tym procesie problem stanowi głośność utworu, czyli amplituda drgań. Podczas cichej i szybkiej gry występuje problem z brakiem znaczących skoków amplitud oraz brakiem wyraźnych przerw pomiędzy kolejnymi dźwiękami. Do rozwiązania tego problemu wykorzystuje się metody częstotliwościowe i metody czasowe.

Podczas przetwarzania sygnałów dźwiękowych z reguły na początku operuje się przebiegiem amplitudowym w dziedzinie czasu. Metody częstotliwościowe polegają na analizie widma częstotliwościowego i wykrywaniu znaczących zmian wartości częstotliwości. Ponieważ przejście z dziedziny czasu na dziedzinę częstotliwości wymaga dużej mocy obliczeniowej, należy odpowiednio dobrać częstość badania widma częstotliwościowego. Przy zbyt dużej częstości proces zachodzi powoli i zajmuje zbyt dużo czasu. Dobranie za małej częstości grozi jednak niedokładnością pomiaru, czasami na tyle dużą, że uzyskiwane wyniki są całkowicie błędne. Dzieje się tak,

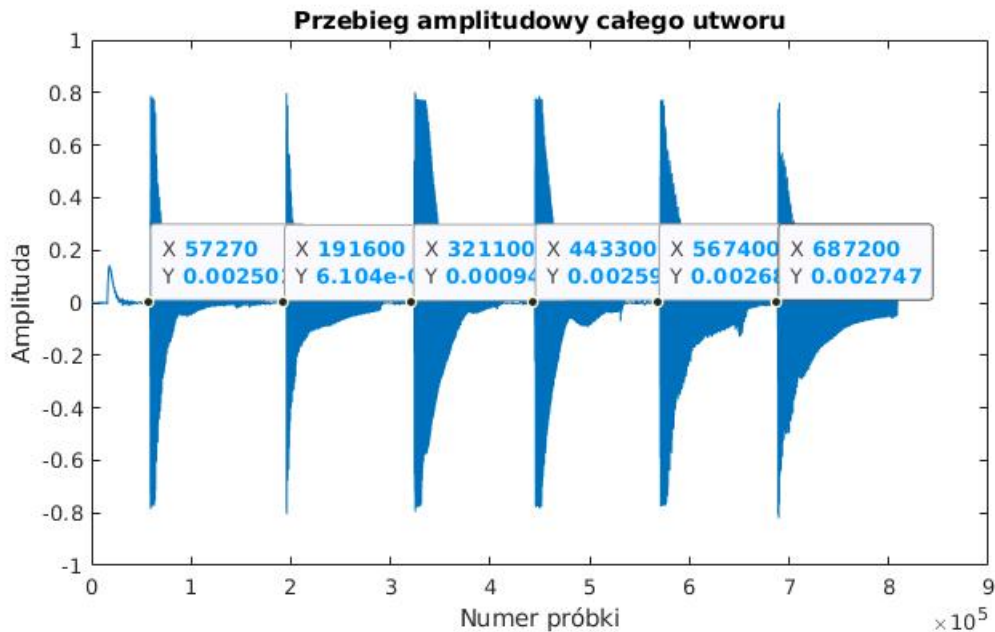


(a)



(b)

Rysunek 3.3 Porównanie widma całego utworu, do pojedynczego dźwięku. a) widmo częstotliwościowe całego utworu, b) widmo częstotliwościowe pojedynczego dźwięku



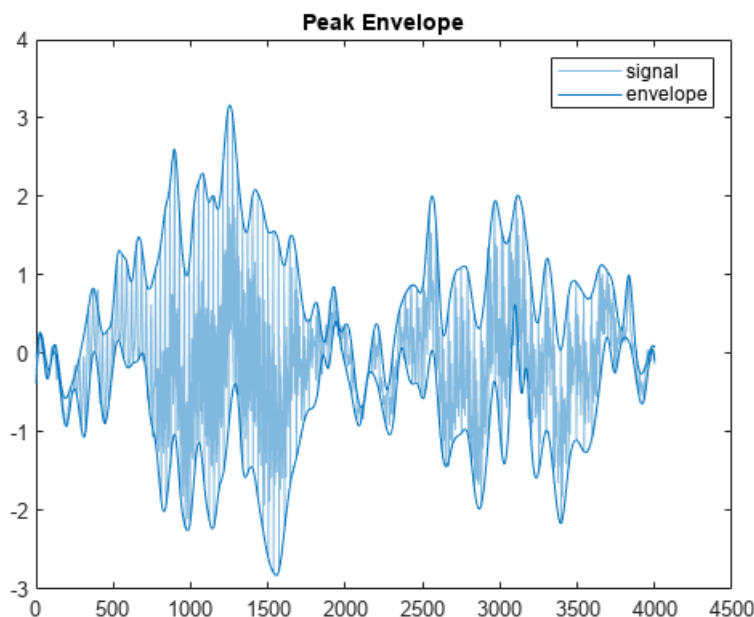
Rysunek 3.4 Ręczne wyznaczanie czasu trwania dźwięków

ponieważ za mała liczba pomiarów nie pozwala na zarejestrowanie zmian częstotliwości. Należy więc ustawić odpowiedni próg, który określa, czy różnica pomiędzy wartościami częstotliwości jest wystarczająca, żeby uznać ją za zmianę i rozpoznać jako początek następnego dźwięku.

Metody czasowe polegają na bezpośredniej analizie przebiegów amplitudowych, dlatego są szybszym rozwiązaniem niż badanie widm częstotliwościowych [nie]. Nie wymagają tak dużej mocy obliczeniowej i nie są tak wrażliwe na małe zmiany. Poniżej krótko opisano dwie wybrane metody: metoda detekcji obwiedni oraz metoda detekcji progowej.

Detekcja obwiedni. Obwiednia oscylującego sygnału to krzywa wyznaczająca jego ekstrema. Jej wartością jest chwilowa amplituda analizowanego sygnału [Rad, Wikf]. Funkcja obwiedni może być funkcją dowolnej zmiennej. Podczas przetwarzania sygnałów cyfrowych, najważniejszą cechą obwiedni jest jej wygląd w dziedzinie czasu. Detektor obwiedni generuje granicę, w której znajduje się sygnał, odzwierciedlając kształt obwiedni, co pokazuje rysunek 3.5. Pozwala to w sztuczny sposób zapamiętywać i odwzorowywać kształty sygnałów oraz brzmienia dźwięków np. w syntezatorach.

Obwiednię wykorzystuje się również do rozpoznawania czasu trwania dźwięków. Przebieg amplitudowy pojedynczego dźwięku ma charakterystyczny kształt, rozpoczynający się bardzo wysoką amplitudą i gasnącym sygnałem w fazie końcowej. Należy rozpoznać znaczącą zmianę wartości amplitudy i ustawić ją jako początek dźwięku. Rozpoznany kształt obwiedni pozwala ograniczyć przebieg amplitudowy i podzielić cały utwór na pojedyncze dźwięki.



Rysunek 3.5 Obwiednia (envelope) sygnału [MATd]

Detekcja progowa. Detekcja progowa polega na wykrywaniu przejść przebiegu czasowego przez konkretną wartość progową [nie]. Wartości poniżej progu są odrzucane, wartości powyżej progu, uznawane za szukany sygnał. Istotna jest chwila przejścia sygnału przez wartość progową. Zbocze narastające badanego sygnału w chwili przejścia oznacza rozpoczęcie, zbocze opadające – zakończenie sygnału.

Poprawna detekcja wymaga dobrania odpowiedniej wartości progowej. Zbyt niska wartość progu powoduje, że nie eliminujemy niepożądanych wartości, za wysoka – nie pozwala na wykrycie całego zakresu szukanego sygnału. Duża zmiana wartości amplitudy przebiegu sinusoidalnego na początku każdego dźwięku ułatwia rozpoznanie początku dźwięku.

3.2 Rozpoznanie częstotliwości podstawowej

Dźwięki klasyfikuje się ze względu na częstotliwość ich tonu podstawowego, tj. wysokość dźwięku. Ich nazwy zwyczajowe składają się z dwóch komponentów: nazwy tonu podstawowego oraz numeru oktawy. W celu rozpoznania częstotliwości tonu podstawowego należy przekształcić przebieg amplitudowy dźwięku do widma częstotliwościowego. Umożliwia to analiza fourierowska.

3.2.1 Transformacja Fouriera

Do analizy częstotliwościowej sygnałów wykorzystuje się transformację Fouriera [Zie05, Lyo10]. Przekształca ona funkcję (z definicji ciągłą [Mar]) z dziedziny czasu w inną funkcję dziedziny częstotliwości. W wyniku powstaje transformata Fouriera.

Prosta i odwrotna transformacja Fouriera wygląda następująco †:

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft} dt, \quad (3.4)$$

$$x(t) = \int_{-\infty}^{+\infty} X(f)e^{j2\pi ft} df, \quad (3.5)$$

gdzie $X(f)$ jest widmem Fouriera sygnału $x(t)$ [Zie05], które określa zależność pomiędzy częstotliwościami, i to jednostka urojona ($i^2 = -1$), a f to konkretna częstotliwość. Widmo można otrzymać poprzez całkowanie iloczynu sygnału $x(t)$ ze składowymi harmonicznymi, opisanymi za pomocą funkcji eksponencjalnych $e^{-2\pi ft}$. Prosta transformacja Fouriera służy do rozkładu funkcji bazowej $x(t)$ na różne częstotliwości. Odwrotna transformacja Fouriera scala częstotliwości składowych harmonicznymi do funkcji bazowej.

Jako że transformacja Fouriera z definicji opisuje analizę funkcji ciągłej, żeby przekształcać sygnały dyskretne, które wykorzystuje się w przetwarzaniu dźwięków, należy wprowadzić dyskretną transformację Fouriera (DFT):

$$X(kf_0) = \frac{1}{N} \sum_{n=0}^{N-1} x(n\Delta t)e^{-j2\pi(k/N)n}, \quad (3.6)$$

$$X(n\Delta t) = \sum_{k=0}^{N-1} x(kf_0)e^{j2\pi(k/N)n}, \quad (3.7)$$

gdzie N to liczba wszystkich próbek, k – indeks częstotliwości, n – indeks czasu a f_0 określa iloraz częstotliwości próbkowania sygnału przez liczbę próbek [Zie05].

Szereg Fouriera to szereg rozkładający funkcje okresowe na sumy funkcji trygonometrycznych [Wikg]. Należy zmodyfikować wzór na prostą, dyskretną transformację Fouriera 3.6 tak, żeby móc wyznaczyć współczynniki szeregu Fouriera.

$$c_k = \sum_{k=0}^{T-1} x(kf_0)e^{-j2\pi(k/T)}, \quad (3.8)$$

gdzie T to okres próbkowania sygnału, c_k – współczynniki szeregu Fouriera. Do przetwarzania sygnałów często wykorzystuje się szybką transformację Fouriera (*ang. Fast Fourier Transform – FFT*) [Lyo10] w celu zmniejszenia złożoności obliczeniowej. Jeden z najprostszych algorytmów, radix-2 [Chw] polega na minimalizacji obliczeń współczynników c_k , aby przyspieszyć proces DFT. Algorytm wykorzystuje okresowość funkcji sinusoidalnych i grupuje składniki na parzyste i nieparzyste. Metodą dziel i zwyciężaj zmniejsza dwukrotnie liczbę wyznaczanych współczynników i znacząco skraca czas pracy algorytmu.

†zakłada się istnienie tych całek

Rozdział 4

Struktura systemu

Struktura systemu bazuje na wybranych funkcjonalnościach z różnych aplikacji komercyjnych (zobacz rozdział 2), takich jak: rozpoznawanie wysokości dźwięku, strojenie instrumentów, zapis tonów podstawowych jako nazw zwyczajowych, analizowanie całych utworów czy dodanie swojego własnego wzorca. Zasada działania systemu opisanego poniżej przypomina działanie stroika [Set05]. Zaprojektowany system pozwala na pracę w dwóch trybach: analizy wcześniej zapisanych utworów i pracy na żywo. Pierwszy tryb pozwala użytkownikowi systemu uzyskać informacje na temat ewentualnych różnic pomiędzy wgranym wzorcem a nagraniem. Dzieląc nagrany utwór na pojedyncze dźwięki, identyfikuje wysokości każdego z nich i porównuje z wartością wzorcową. Wynikiem końcowym jest jednorazowa odpowiedź na temat poprawności gry lub wskazanie ewentualnych błędów. Drugi tryb został skonstruowany do pracy na żywo. Zapętłony system wywołuje po kolei bloki, które umożliwiają identyfikację wysokości pojedynczego dźwięku i szukają różnic między nim a wgranym wzorcem. Tryb pracy na żywo różni się od analizy całego utworu tym, że użytkownik dostaje wiadomość zwrotną na temat każdego, pojedynczego dźwięku w czasie rzeczywistym. Po uzyskaniu informacji zwrotnej, użytkownik może zagrać następny dźwięk.

4.1 Tryb analizy zapisanych utworów

Strukturę systemu w wersji do analizy wcześniej zapisanych utworów, zamieszczono na rysunku 4.1. W systemie można wyróżnić 4 etapy: odczytanie informacji z nagrania, podział utworu na pojedyncze dźwięki, identyfikacja wysokości dźwięków oraz porównanie zapisanego utworu z wzorcem.

Na wejście pierwszego bloku podaje się wcześniej zapisany utwór w postaci pliku z rozszerzeniem .mp3 lub .wav. Drugi blok odczytuje podstawowe informacje, takie jak częstotliwość próbkowania, długość czas trwania całego utworu oraz amplitudy drgań przebiegów sinusoidalnych.

Kolejny blok służy do rozpoznania początków i końców dźwięków. W strukturze wykorzystano metodę automatyczną czasową, bazującą na detekcji progowej, z ruchomym progiem (podrozdział 3.1.2). O wysokości progu decyduje średnia z całej amplitudy nagrania. Takie rozwiązanie jest odporne na błędy detekcji i powoduje ustawienie prawidłowej wartości progowej, dzięki której zostaje wykryty cały za-



Rysunek 4.1 Schemat struktury systemu do analizy zapisanych utworów

kres szukanego sygnału. Wykorzystując automatyczny zapis danych, następny blok dzieli cały utwór na pojedyncze dźwięki.

Etap rozpoznawania wysokości pojedynczych dźwięków podzielony jest na 3 bloki: nałożenie szybkiej transformaty Fouriera (*ang. FFT- Fast Fourier Transform*) (podrozdział 3.2.1), określenie podstawowej częstotliwości dźwięku i jego harmonicznych, oraz zidentyfikowanie jego wysokości i przypisanie mu wartości.

W wyniku pierwszej operacji otrzymuje się widmo częstotliwościowe. Z uzyskanego widma odczytuje się częstotliwość tonu podstawowego, jako najmniejszą składową częstotliwościową dźwięku. Kolejne prążki widma określają składowe harmoniczne. W trzecim bloku przypisuje się nazwę zwyczajową do zidentyfikowanej wysokości tonu podstawowego. Przystawia się także harmoniczne dźwięku, aby uzyskać większą pewność.

Kończącym etapem jest porównanie dźwięku z wzorcem i przekazanie użytkownikowi wyniku końcowego, w postaci pliku wynikowego z wyszczególnionymi ewentualnymi błędami.

4.2 Tryb pracy na żywo

Rysunek 4.2 pokazuje strukturę systemu w wersji do pracy na żywo. W tym trybie można wyróżnić 3 etapy: odczytanie informacji na temat zagranej, pojedynczego dźwięku, identyfikacja jego wysokości oraz porównanie z wzorcem. System działa w pętli, która kończy się po ostatnim, zagrany dźwięku. Odpowiedź dla użytkownika zwracana jest w czasie rzeczywistym po każdym obiegu pętli.



Rysunek 4.2 Schemat struktury systemu na żywo

Wejście pierwszego bloku jest skonfigurowane z kartą dźwiękową urządzenia, na którym działa system. Użytkownik nagrywa pojedynczy dźwięk, który automatycznie jest w systemie zapisywany. Drugi blok odczytuje amplitudę drgań tego dźwięku. Tryb drugi różni się od poprzedniego tym, że od samego początku w jednej iteracji pętli analizowany jest tylko jeden dźwięk, zatem nie ma potrzeby rozpoznawania jego początku oraz końca.

Na drugim etapie następuje rozpoznanie wysokości dźwięku i przypisanie mu odpowiedniej wartości częstotliwości. Schemat działania jest taki sam, jak w przypadku trybu do analizy wcześniej zapisanego utworu (zobacz schemat działania 4.1), czyli wykorzystuje się FFT do wykreślenia widma częstotliwościowego i określa częstotliwości tonu podstawowego.

Ostatnim etapem jest porównanie zagranego dźwięku z wzorcem oraz określenie ewentualnego błędu. Po wykonaniu etapu końcowego, system wraca do pierwszego bloku, w którym użytkownik nagrywa kolejny dźwięk.

Rozdział 5

Implementacja

Do implementacji systemu wykorzystano środowisko MATLAB [Mata]. MATLAB to platforma programistyczna, która służy do wykonywania obliczeń, tworzenia skryptów, wykresów, modeli i symulacji, aplikacji etc. Posiada bogaty zbiór specjalistycznych funkcji, kolekcji i toolboksów, czyli bibliotek dostarczających narzędzia o szerokim zastosowaniu naukowym. MATLAB umożliwia przetwarzanie i analizę sygnałów, między innymi dźwięków i obrazów [Rud]. Właściciel środowiska MATLAB, firma MathWorks, udostępnia dokumentację, przykłady użycia funkcji wraz z instruktażami i podręcznikami do nauki, oraz dostarcza system pomocy i forum dostępne dla użytkowników środowiska MATLAB.

W środowisku MATLAB przetwarzanie sygnałów dźwiękowych jest możliwe dzięki dodatkowym bibliotekom i kolekcjom, dostępnym na stronie firmy MathWorks [Mata]. Do implementacji systemu użyto dwóch toolboksów: *Signal Processing Toolbox* i *Audio Toolbox*, oraz trzech kolekcji: *Matlab Audio Analysis Collection*, *Midi Tools Collection* i *Real-Time Audio Processor Collection*.

Konstrukcję programu można podzielić na 4 główne części:

- nagrywanie,
- rozpoznanie początków i końców dźwięków,
- identyfikację ich wysokości,
- porównanie z wzorcem.

Poniżej pokrótce rozpisano rozwiązanie implementacyjne każdej z nich.

5.1 Rejestrowanie dźwięków

Signal Processing Toolbox dostarcza szereg funkcji służących do przetwarzania i analizowania plików audio. W systemie wykorzystano dwie podstawowe funkcje do nagrywania sygnałów dźwiękowych i odczytywania z nich informacji: `audiorecorder` i `audioread`. Poniżej zostały przedstawione i opisane przykłady wykorzystania funkcji w programie.

Fragment kodu służący do rejestrowania dźwięków ma postać:

```
1 Fs = 44100; %częstotliwość próbkowania nagrywanego sygnału
2 noc = 1; %liczba kanałów (1-mono, 2-stereo)
3 nob = 16; %liczba bitów
4 time = 5; %długość trwania czynności nagrywania
5 ObjR1 = audiorecorder(Fs, nob, noc);
6 record(ObjR1);
7 for i = 1:time
8     pause(1);
9 end
10 stop(ObjR1);
11
12 %recordblocking(ObjR1, 5);
```

W trybie pracy na żywo, wejście systemu jest skonfigurowane z kartą dźwiękową urządzenia na którym pracuje. Mikrofon wbudowany w urządzenie odbiera analogowy sygnał dźwiękowy i za pomocą przetwornika A/C przekształca go na sygnał cyfrowy. W pamięci zostaje zarejestrowany dźwięk, który przez podłączoną kartę dźwiękową jest przekazywany do systemu. Funkcja `audiorecorder` tworzy obiekt z podanymi właściwościami: częstotliwością próbkowania, liczbą kanałów audio oraz liczbą bitów przypadającą na jedną próbkę [MATb]. Przekazanie sygnału z zewnętrznego urządzenia do obiektu następuje podczas wywołania funkcji `record`. Zaletą wykorzystania obiektu `audiorecorder` jest możliwość skorzystania z dostarczonych metod, takich jak rejestrowanie, wznawianie czy zatrzymywanie nagrania.

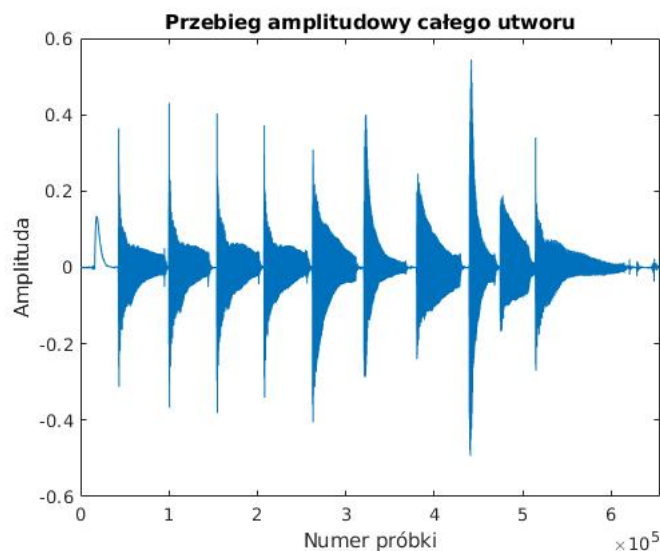
Kolejny fragment realizujący odczytywanie danych z nagrania wygląda następująco:

```
1 [y, Fs] = audioread('omrs2.wav'); %plik .wav lub .mp3
2 y = y(:,1);
3 info = audioinfo('omrs2.wav')
```

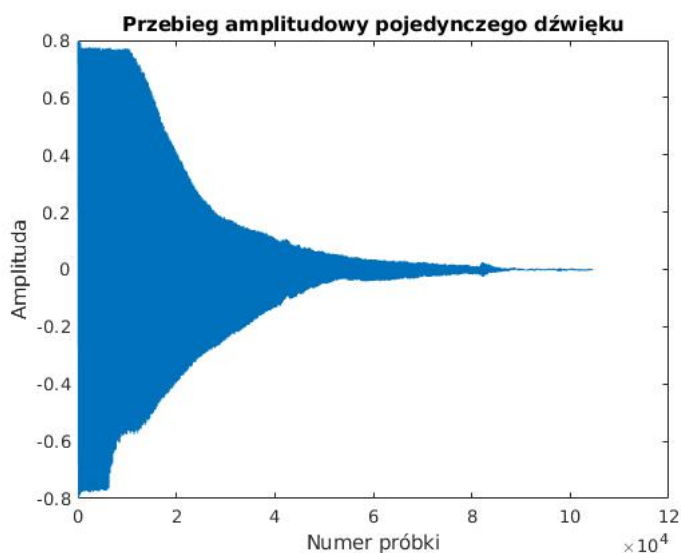
W trybie analizy wcześniej zapisanych utworów do przetwarzania zarejestrowanego nagrania wykorzystano funkcję `audioread`. Na wejście systemu podawany jest plik audio z rozszerzeniem `.wav` lub `.mp3`. Funkcja zwraca macierz z przebiegiem amplitudowym `y` oraz częstotliwością próbkowania `Fs`. Dla potrzeb dalszej analizy sygnał dwukanałowy (jeśli taki został nagrany) zostaje zredukowany do sygnału jednokanałowego. Aby odczytać czas trwania utworu lub porównać uzyskaną częstotliwość próbkowania z zapisaną użyto funkcji `audioinfo`, która dostarcza zbiór informacji uzyskanych z podanego nagrania.

5.2 Dzielenie utworów na pojedyncze dźwięki

W trybie na żywo nie wydzielono osobnej części na rozpoznanie początku i końca dźwięku. Wynika to z faktu, że podczas pojedynczego obiegu pętli analizie poddawany jest tylko jeden dźwięk. Opisana w podrozdziale 5.1 funkcja `record` zostaje



Rysunek 5.1 Przebieg całego utworu



Rysunek 5.2 Przebieg amplitudowy pojedynczego dźwięku

wywoływana po otwarciu okna czasowego. Po upływie określonego czasu, w którym użytkownik powinien zagrać dźwięk, okno jest zamykane i następuje dalsza analiza. Warunkiem prawidłowej pracy systemu jest zagranie przez użytkownika następnego dźwięku dopiero po wyświetleniu przez program informacji zwrotnej o wysokości aktualnego.

Inna sytuacja ma miejsce podczas analizy zagranej wcześniej i zarejestrowanego utworu. Rozpoznanie początku i końca dźwięków jest kluczowym elementem wpływającym na poprawność działania programu. W systemie wykorzystano podział automatyczny, w wyniku którego z przebiegu amplitudowego całego utworu (zobacz rysunek 5.1) otrzymano przebiegi każdego pojedynczego dźwięku (zobacz rysunek 5.2).

Proces podziału utworu rozpoczyna się wyznaczeniem ruchomej wartości pro-

gowej, która zależy od średniej amplitudy całego nagranych przebiegu. Następnym krokiem jest wywołanie pętli, w której zostają wyznaczone i zapisywane indeksy próbek, uznawanych za rozpoczęcie dźwięku. Przesuwając małe okna czasowe po całym utworze, system sprawdza różnice w wartościach amplitudy. Maksymalna zmiana w danym oknie czasowym sygnalizuje rozpoczęcie dźwięku, pod warunkiem że amplituda przekracza wartość progową. W dalszych fragmentach proces jest powtarzany do momentu znalezienia początku następnego dźwięku. Koniec analizowanego dźwięku poprzedza początek następnego. Aby mieć pewność, że nie zapisujemy dwóch dźwięków jednocześnie, od indeksu końcowego odejmuje się niewielką dodatnią wartość, przesuwając koniec o kilka próbek. Dzięki temu koniec poprzedniego i początek następnego dźwięku nie nakładają się na siebie. Ostatnim elementem procesu jest podział całego nagrania na pojedyncze dźwięki z wykorzystaniem znalezionych indeksów.

5.3 Określenie częstotliwości za pomocą FFT

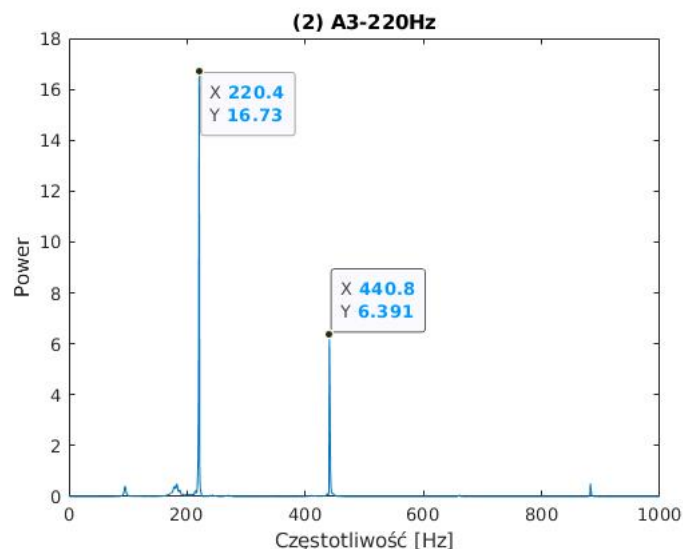
Poniższy fragment kodu przedstawia zastosowanie funkcji `fft`.

```
1 ft = fft(y);
2 n = length(y);
3 Fs= 48000;
4 f = (0:n-1)*(Fs/n);
5 power = abs(ft).^2/n;
```

Funkcja `fft` [MATc] zwraca dyskretną transformatę Fouriera (DFT) sygnału podanego na wejście, wykorzystując algorytm FFT opisany w podrozdziale 3.2.1 do transformacji Fouriera. Do wykreślenia widma, oprócz funkcji `fft`], należy obliczyć wektor częstotliwości. Jest to stosunek częstotliwości próbkowania do długości sygnału wejściowego. Częstotliwość próbkowania powinna być co najmniej dwa razy większa od największej częstotliwości składowej obecnej w próbkowanym sygnale [Kon]. Moc widma uzyskuje się poprzez nałożenie na wyjście algorytmu FFT wartości bezwzględnej. Aby wykreślić widmo częstotliwościowe i wyznaczyć wartość częstotliwości w środowisku MATLAB, wykorzystuje się funkcję `plot`, w sposób jak poniżej.

```
1 figure
2 plot(f,power);
```

Efekt działania funkcji `plot` przedstawia rysunek 5.3. Szukana częstotliwość tonu podstawowego jest najniższą częstotliwością składową dźwięku (nie bierzemy pod uwagę szumów poniżej wartości progowej). W widmie częstotliwościowym widoczne są również składowe harmoniczne rozpoznawanego dźwięku. W metodzie automatycznej, wartości częstotliwości wyświetlonych prążków zapisywane są do osobnej macierzy, która jest niezbędna do dalszej analizy.



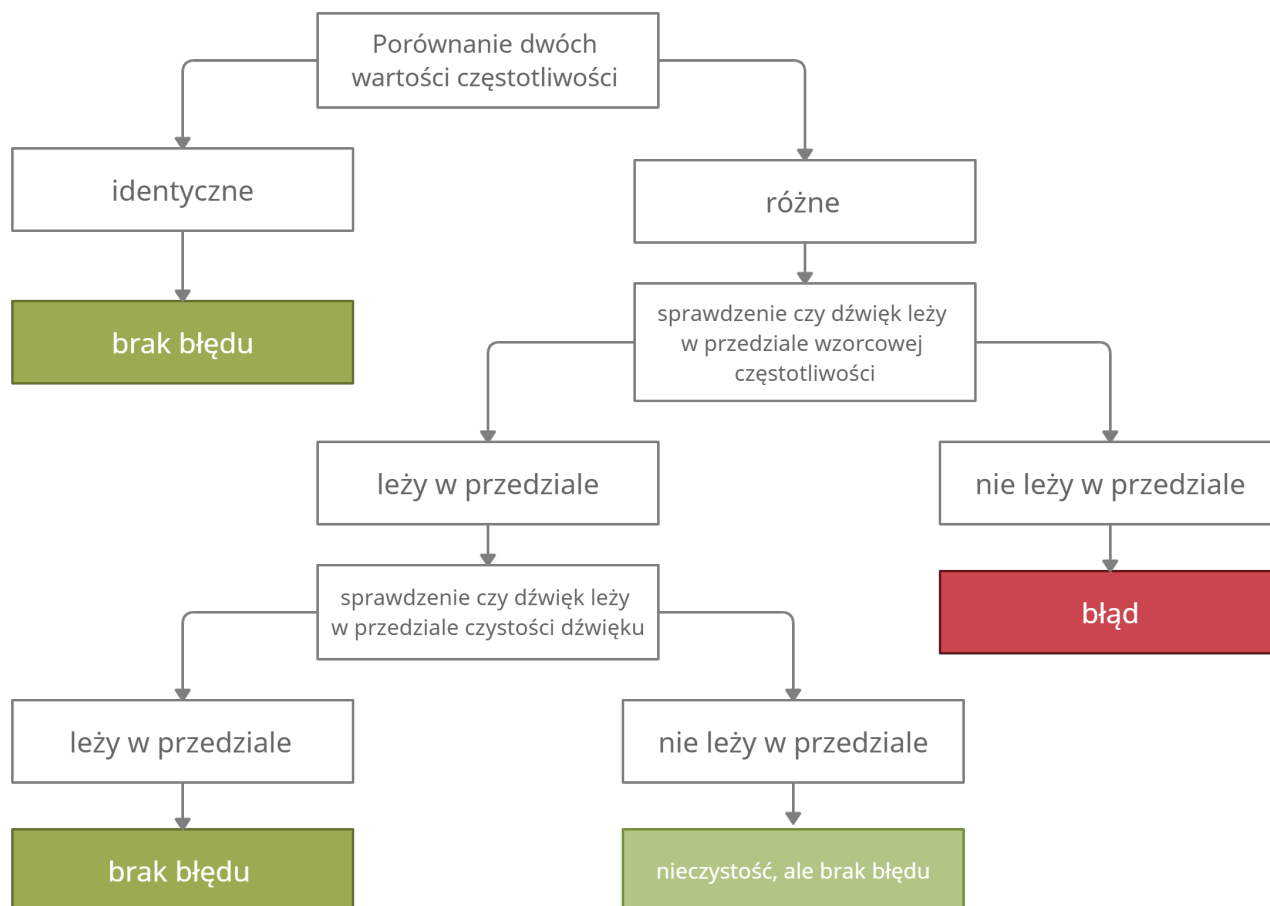
Rysunek 5.3 Wykres widma przedstawiający efekt działania funkcji plot

5.4 Porównanie gry z wzorcem

W ostatnim etapie porównuje się rozpoznane częstotliwości z częstotliwościami wzorcowymi. Wzorzec jest zapisywany w pliku MIDI z rozszerzeniem .mid [Wikid]. Interfejs do wgrzywania utworów, bazujący na kolekcji *midi Tools Collection*, zamienia plik binarny MIDI na plik z częstotliwościami poszczególnych dźwięków. Wymogiem do prawidłowego działania systemu jest wzorzec, który musi być linią melodyczną danego instrumentu. Użytkownik nie może wgrać utworu polifonicznego, z wieloma ścieżkami i instrumentami.

Porównanie gry z wzorcem następuje na poziomie częstotliwościowym. Algorytm postępowania został przedstawiony na rysunku 5.4. W momencie wykrycia dwóch identycznych* częstotliwości, system uznaje to za brak błędu. Jeśli częstotliwość zagrana znajduje się w przedziale częstotliwości wzorcowej oraz w przedziale czystości dźwięku, system również traktuje to jako brak błędu. W przypadku częstotliwości leżącej poza przedziałem czystości, ale znajdującej się w przedziale wzorcowej częstotliwości, system może wskazać, że dźwięk jest nieczysty, jednak poprawnie zagrany. Jeśli rozpoznana częstotliwość nie należy do przedziału wzorcowej częstotliwości, system zwraca informację o błędnie zgranym dźwięku.

*Za identyczne uznawane są częstotliwości, których różnica mniejsza jest niż 1 Hz



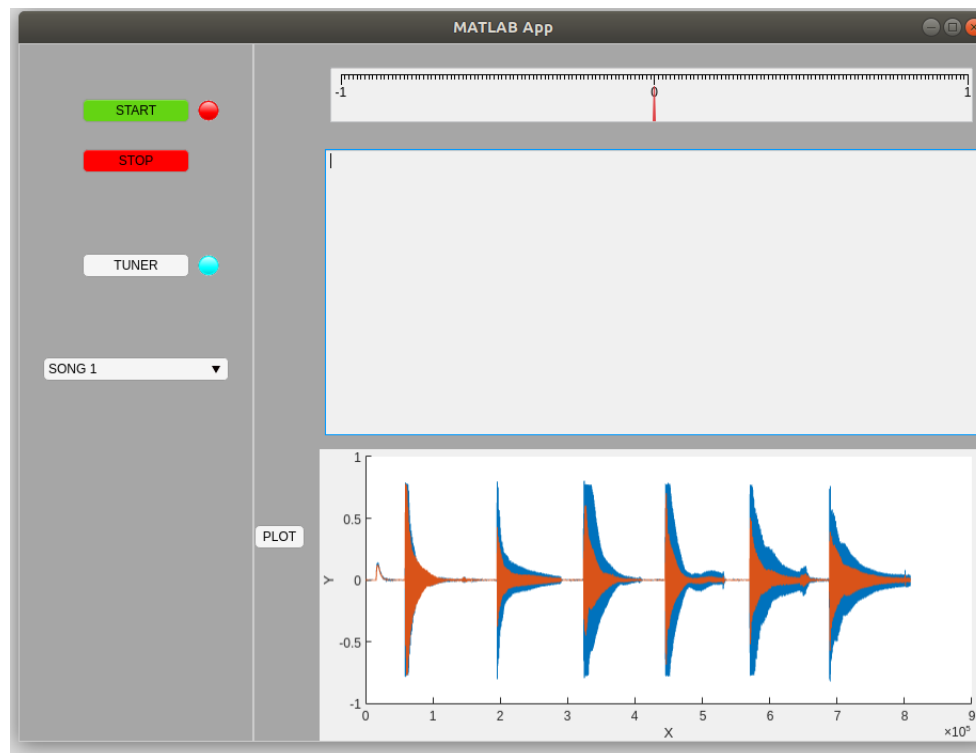
Rysunek 5.4 Algorytm porównania częstotliwości

5.5 Interfejs graficzny aplikacji

Dla wygody użytkownika zaprojektowano interfejs graficzny aplikacji, który umożliwia sprawne posługiwanie się systemem. Do zaprojektowania okienkowej struktury wykorzystano dedykowane narzędzie w środowisku MATLAB, jakim jest App Designer Matlab [MATE]. Umożliwia ono dodawanie wykresów, przycisków, pól tekstowych czy graficznych wskaźników. Połączenie aplikacji ze skrypcem MATLAB realizowane jest za pomocą tzw. *callbacks*, czyli wywołań zwrotnych. Działanie aplikacji bazuje na obiektowych cechach języka MATLAB. W wygenerowanym skrypcie można tworzyć i wywoływać metody oraz definiować klasy. Użytkownik App Designer ma uprawnienia tylko do fragmentów generowanego kodu. Jest to odgórne zabezpieczenie przed utratą danych. Podczas tworzenia aplikacji można korzystać z dwóch widoków: *Design View* i *Code View*. Każda zmiana w polu projektowym zostaje automatycznie zapisana w kodzie.

Użytkownik systemu posiada dostęp do następujących funkcjonalności:

- wybór wzorcowego utworu,
- rozpoczęcie i zakończenie nagrywania,
- wyświetlanie ścieżki do zagrania i ścieżki zagranej,



Rysunek 5.5 Wygląd interfejsu graficznego aplikacji

- wyświetlanie amplitudy nagranych dźwięków,
- strojenie.

Pokazany na rysunku 5.5 interfejs został podzielony na dwa pola: pierwsze z przyciskiem rozpoczęcia nagrywania i diodą wskazującą aktywny tryb nagrywania, przyciskiem zakończenia nagrywania, przyciskiem służącym do uruchamiania stroika (również z sygnalizującą diodą) oraz pasek wyboru wzorca. Drugie pole zawiera ścieżkę z wzorcowymi dźwiękami, ścieżkę z dźwiękami nagranyymi oraz wskaźnik, który określa odchylenia od prawidłowych wartości częstotliwości dla lepszej wizualizacji ewentualnych błędów. W dolnej części został wyświetlony przebieg amplitudy nagranych utworu.

Rozdział 6

Testy

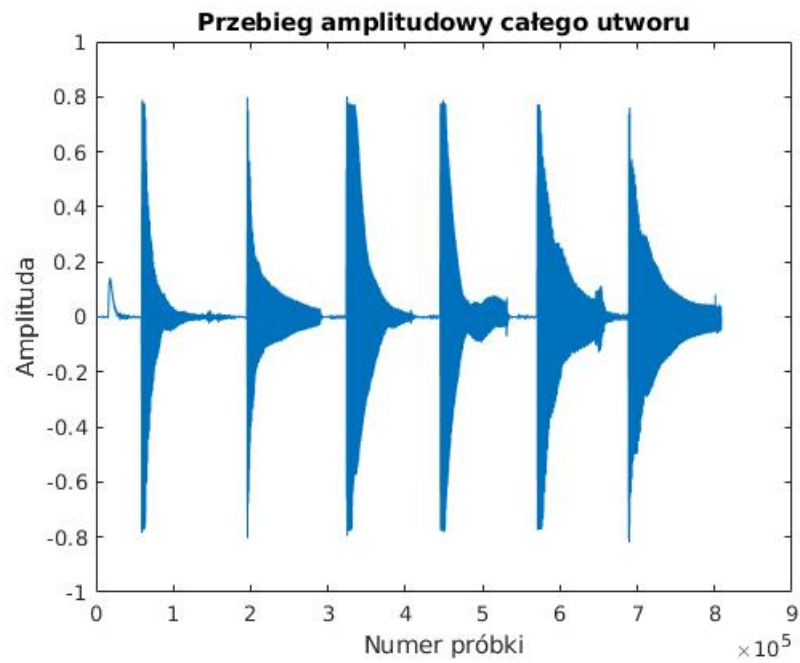
Na potrzeby badań wyróżniono 3 różne instrumenty: pianino, gitarę oraz śpiew. Dla każdego instrumentu wykonano 3 różne testy sprawdzające zachowanie systemu, w wyniku których otrzymano widmo częstotliwościowe i odczytano częstotliwości podstawowe nagranych dźwięków. Do badań wykorzystano samodzielnie zarejestrowane utwory. Podstawową melodią wzorcową wykorzystaną w testach, jest fragment polskiej pieśni wojskowej "O mój rozmarynie" [Wike]. Poniżej, na bazie przykładowego testu, opisano szczegółowo zachowanie systemu bez etapu porównującego nagrany częstotliwość z częstotliwością wzorcową. Dalej przedstawiono wyniki wybranych testów, spośród wszystkich przeprowadzonych badań.

6.1 Działanie systemu

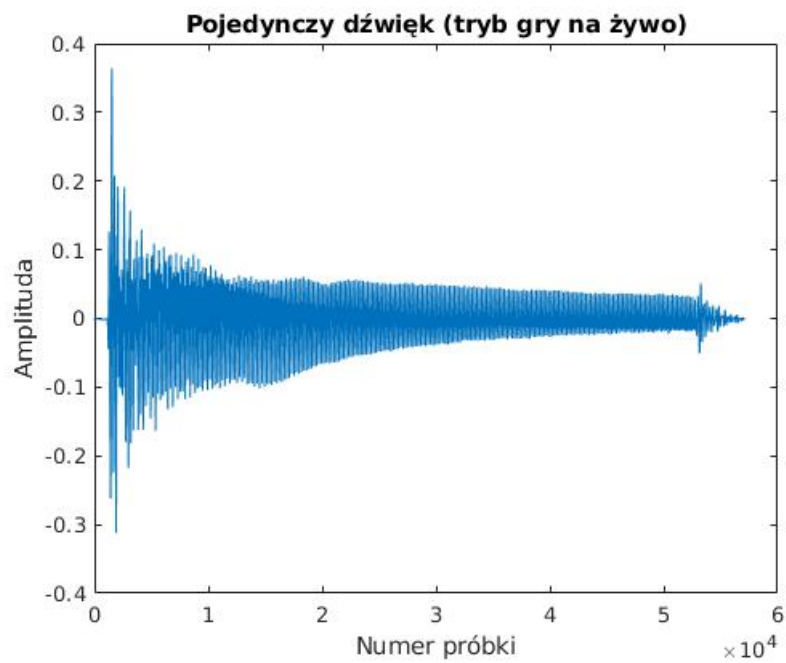
Nagrywanie Na wejście systemu podano wcześniej zapisany utwór w postaci pliku z rozszerzeniem .mp3 (lub .wav), albo korzystając z wyżej opisanej funkcji `audio-recorder`, zarejestrowano nagrany dźwięk w czasie rzeczywistym. Aby sprawdzić poprawność wykonanych operacji, wyświetlono przebiegi amplitudowe nagrania *PusteStrunyGitara.mp3* (zobacz rysunek 6.1) oraz pojedynczego dźwięku (zobacz rysunek 6.2).

Podział utworu na dźwięki Do podziału wcześniej zapisanego utworu na pojedyncze dźwięki wykorzystano metodę automatyczną i metodę ręczną. Rezultaty automatycznego podziału przedstawia rysunek 6.3. Podział ręczny tego samego utworu znajduje się na rysunku 6.4. Na potrzeby dalszych badań, podział utworu na pojedyncze dźwięki zrealizowano metodą ręczną.

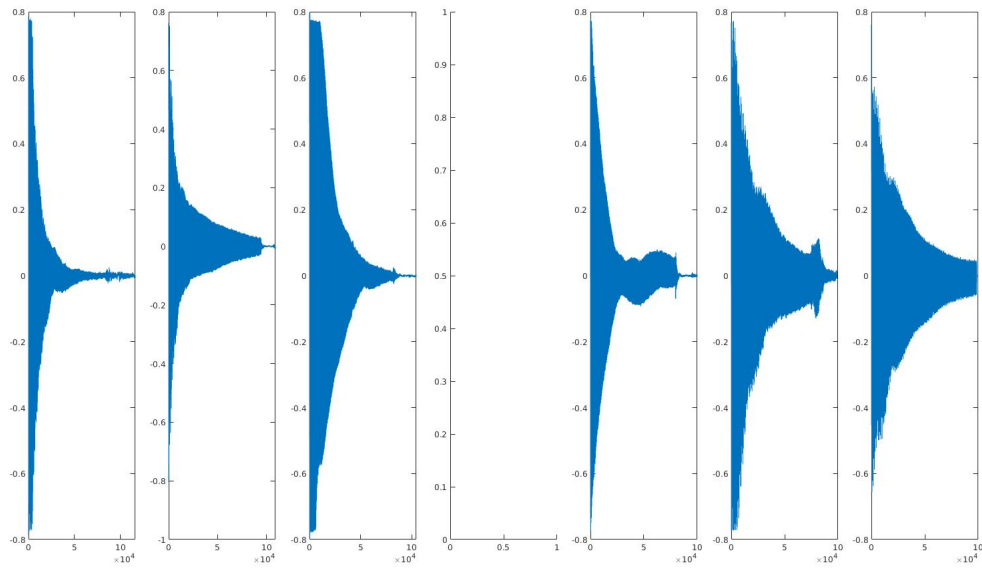
Rozpoznanie częstotliwości podstawowej Wywołanie funkcji `fft` w środowisku MATLAB powoduje zwrócenie dyskretnej transformacji Fouriera, dzięki której wykonywane jest przejście z dziedziny czasu do dziedziny częstotliwości. Za pomocą funkcji `plot` wyświetlone zostało widmo częstotliwościowe, z którego odczytano składowe częstotliwości dźwięku. Rezultaty wykonanych operacji przedstawia rysunek 6.5.



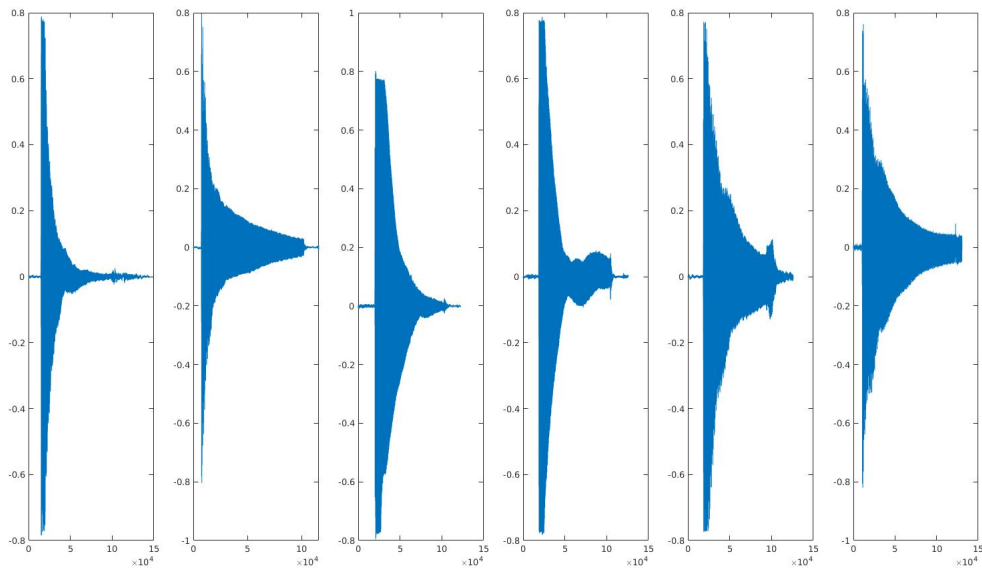
Rysunek 6.1 Przebieg amplitudowy całego utworu (test)



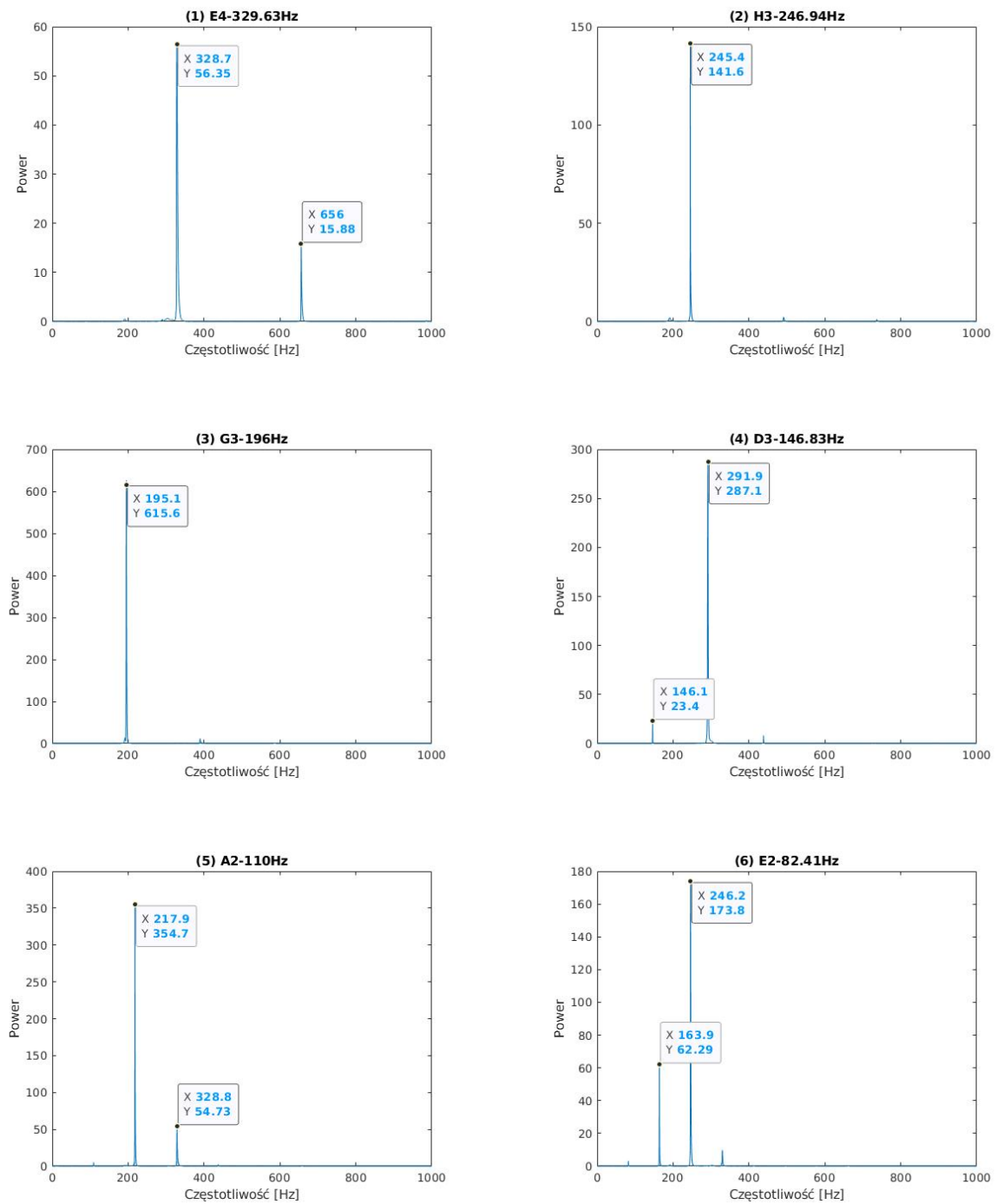
Rysunek 6.2 Przebieg amplitudowy pojedynczego dźwięku (test)



Rysunek 6.3 Podział utworu na pojedyncze dźwięki za pomocą metody automatycznej



Rysunek 6.4 Podział utworu na pojedyncze dźwięki za pomocą metody ręcznej



Rysunek 6.5 Widma częstotliwościowe pojedynczych dźwięków

6.2 Pianino

W przeprowadzonych badaniach wykorzystano klawiaturę MIDI Alesis V49 z oprogramowaniem DAW – Ableton Live 11 Suite. Analizowane utwory nagrano za pomocą sampli Grand Piano i wyeksportowano do plików .wav. Zrealizowano trzy nagrania:

- gamę leżącą w trzeciej oktawie,
- fragment "O mój rozmarynie" w wersji szybkiej bez przerw pomiędzy dźwiękami,
- fragment "O mój rozmarynie" w wersji wolnej z wyraźnymi przerwami pomiędzy dźwiękami.

W tabelach 6.1 – 6.5 przedstawiono wyniki testów. W tabeli 6.6 podsumowano je i określono skuteczność systemu. Litera n stojąca przy nazwie zwyczajowej dźwięku oznacza nieczystość zagranego dźwięku, jednak nie jest uznawana jako błąd. System poprawnie wykrył ilość dźwięków oraz w większości przypadków ich wysokości. Błędnie jednak odczytał oktawę, w której leżą. Błąd wynika z nieprawidłowego określenia jednej ze składowej harmonicznnej jako tonu podstawowego. Częstotliwość tonu podstawowego została odczytana jako szum, ponieważ miała za małą moc.

Tabela 6.1 Wyniki rozpoznanych częstotliwości gamy zagranej na pianinie

wzorzec [Hz]	130.81	146.83	164.81	174.61	196	220	246.94	261.63
wzorzec nazwa	C3	D3	E3	F3	G3	A3	H3	C4
gra [Hz]	262	294.6	330.8	350.2	393.3	441	247.1	261.6
gra nazwa	C4	D4	E4	F4	G4n	A4	H3	C4

Tabela 6.2 Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej szybko na pianinie (część 1)

wzorzec [Hz]	220	220	220	220	293.66
wzorzec nazwa	A3	A3	A3	A3	D4
gra [Hz]	441.3	442.1	220.3	441	294.3
gra nazwa	A4n	A4n	A3	A4	D4

Tabela 6.3 Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej szybko na pianinie (część 2)

wzorzec [Hz]	261.63	246.94	261.63	246.94	220
wzorzec nazwa	C4	H3	C4	H3	A3
gra [Hz]	261.6	246.8	262.1	246.2	441.2
gra nazwa	C4	H3	C4	H3	A4n

Tabela 6.4 Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej wolno na pianinie (część 1)

wzorzec [Hz]	220	220	220	220	293.66
wzorzec nazwa	A3	A3	A3	A3	D4
gra [Hz]	441.3	441.7	441.7	441.7	294.2
gra nazwa	A4n	A4n	A4n	A4n	D4

Tabela 6.5 Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej wolno na pianinie (część 2)

wzorzec [Hz]	261.63	246.94	261.63	246.94	220
wzorzec nazwa	C4	H3	C4	H3	A3
gra [Hz]	261.3	246.9	261.4	247.1	441.5
gra nazwa	C4	H3	C4	H3	A4n

Tabela 6.6 Podsumowanie wyników badań dla pianinia

Nazwa utworu	liczba dźwięk.	liczba wykrytych dźwięk.	skuteczność	skuteczność [%]
gama C3	8	8	2 z 8	25
O mój rozmarynie szybko	10	10	6 z 10	60
O mój rozmarynie wolno	10	10	5 z 10	50

6.3 Gitara

W przeprowadzonych badaniach wykorzystano gitarę klasyczną *Yamaha C-40* oraz gitarę elektroakustyczną *Fender Malibu Player Midnight Satin*. Do zarejestrowania nagranych dźwięków wykorzystano dyktafon w telefonie. Zrealizowano trzy nagrania:

- sześć pustych strun,
- fragment "O mój rozmarynie" w wersji szybkiej bez przerw pomiędzy dźwiękami,
- fragment "O mój rozmarynie" w wersji wolnej z wyraźnymi przerwami pomiędzy dźwiękami.

W tabelach 6.7 – 6.11 przedstawiono wyniki testów. W tabeli 6.12 podsumowano je i określono skuteczność systemu. Litera n stojąca przy nazwie zwyczajowej dźwięku oznacza nieczystość zagranej dźwięku, jednak nie jest uznawana jako błąd. System poprawnie wykrył ilość dźwięków oraz rozpoznał większość ich wysokości poprawnie. Najgorsze wyniki uzyskano przy nagraniu pustych strun.

Tabela 6.7 Wyniki rozpoznanych częstotliwości pustych strun zagrzanych na gitarze

wzorzec [Hz]	329.63	246.94	196	146.83	110	82.41
wzorzec nazwa	E4	H3	G3	D3	A2	E2
gra [Hz]	328.7	245.4	195.1	291.9	217.9	246.2
gra nazwa	E4	H3n	G3	D4	A3n	H3

Tabela 6.8 Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej szybko na gitarze (część 1)

wzorzec [Hz]	220	220	220	220	293.66
wzorzec nazwa	A3	A3	A3	A3	D4
gra [Hz]	220.5	220.4	219.5	219.7	296
gra nazwa	A3	A3	A3	A3	D4n

Tabela 6.9 Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej szybko na gitarze (część 2)

wzorzec [Hz]	261.63	246.94	261.63	246.94	220
wzorzec nazwa	C4	H3	C4	H3	A3
gra [Hz]	263	495.2	263	247.2	220
gra nazwa	C4n	H4	C4n	H3	A3

Tabela 6.10 Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej wolno na gitarze (część 1)

wzorzec [Hz]	220	220	220	220	293.66
wzorzec nazwa	A3	A3	A3	A3	D4
gra [Hz]	220.6	219.8	220.3	220.1	296
gra nazwa	A3	A3	A3	A3	D4n

Tabela 6.11 Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej wolno na gitarze (część 2)

wzorzec [Hz]	261.63	246.94	261.63	246.94	220
wzorzec nazwa	C4	H3	C4	H3	A3
gra [Hz]	262.9	247.4	262.6	247.2	220.2
gra nazwa	C4n	H3	C4	H3	A3

6.4 Śpiew

W przeprowadzonych badaniach wykorzystano dyktafon w telefonie. Zrealizowano trzy nagrania:

- pięć pierwszych tonów gamy leżącej w czwartej oktawie,
- fragment "O mój rozmarynie" w wersji szybkiej bez przerw pomiędzy dźwiękami,
- fragment "O mój rozmarynie" w wersji wolnej z wyraźnymi przerwami pomiędzy dźwiękami.

Tabela 6.12 Podsumowanie wyników badań dla gitary

nazwa utworu	liczba dźwięk.	liczba wykrytych dźwięk.	skuteczność	skuteczność [%]
puste struny	6	6	3 z 6	50
O mój rozmarynie szybko	10	10	9 z 10	90
O mój rozmarynie wolno	10	10	10 z 10	100

W tabelach 6.13 – 6.17 przedstawiono wyniki testów. W tabeli 6.18 podsumowano je i określono skuteczność systemu. Litera n stojąca przy nazwie zwyczajowej dźwięku oznacza nieczystość zagrane dźwięku, jednak nie jest uznawana jako błąd. W dwóch testach skuteczność systemu wyniosła 100%, jednak dźwięki w większości przypadków były nieczyste. System błędnie rozpoznał ilość dźwięków w nagraniu fragmentu "O mój rozmarynie". Stało się tak, ponieważ w przebiegu amplitudowym podczas śpiewania nie wykryto charakterystycznych przerw w nagraniu. Przez to niektóre dźwięki nie rozpoznano, a utwór został źle podzielony.

Tabela 6.13 Wyniki rozpoznanych częstotliwości fragmentu zaśpiewanej gamy

wzorzec [Hz]	261.63	293.66	329.63	349.23	392
wzorzec nazwa	C ₄	D ₄	E ₄	F ₄	G ₄
gra [Hz]	264.8	293.7	329.3	352.3	397.5
gra nazwa	C ₄ n	D ₄	E ₄	F ₄ n	G ₄ n

Tabela 6.14 Wyniki rozpoznanych częstotliwości "O mój rozmarynie" szybko zaśpiewanej (część 1)

wzorzec [Hz]	220	220	220	220	293.66
wzorzec nazwa	A ₃	A ₃	A ₃	A ₃	D ₄
gra [Hz]	218.7	222.6	223.9	263.5	250.3
gra nazwa	A ₃ n	A ₃ n	A ₃ n	C ₄ n	H ₃ n

Tabela 6.15 Wyniki rozpoznanych częstotliwości "O mój rozmarynie" szybko zaśpiewanej (część 2)

wzorzec [Hz]	261.63	246.94	261.63	246.94	220
wzorzec nazwa	C ₄	H ₃	C ₄	H ₃	A ₃
gra [Hz]	261	218.6			
gra nazwa	C ₄	A ₃ n			

Tabela 6.16 Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej wolno zaśiewanej (część 1)

wzorzec [Hz]	220	220	220	220	293.66
wzorzec nazwa	A3	A3	A3	A3	D4
gra [Hz]	217.1	221.9	221.5	221.5	297.4
gra nazwa	A3n	A3n	A3n	A3n	D4n

Tabela 6.17 Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej wolno zaśiewanej (część 2)

wzorzec [Hz]	261.63	246.94	261.63	246.94	220
wzorzec nazwa	C4	H3	C4	H3	A3
gra [Hz]	263.8	249.8	266.1	251.2	223.9
gra nazwa	C4n	H3n	C4n	H3n	A3n

Tabela 6.18 Podsumowanie wyników badań dla śpiewu

nazwa utworu	liczba dźwięków	liczba wykrytych dźwięków	skuteczność	skuteczność [%]
gama C3	5	5	5 z 5	100
O mój rozmarynie szybko	10	7	4 z 10	40
O mój rozmarynie wolno	10	11	10 z 10	100

6.5 Wnioski

Przeprowadzone badania sprawdzały poprawność działania systemu. Wynikiem końcowym jest porównanie wartości częstotliwości każdego zagranej dźwięku oraz określenie skuteczności systemu. Najniższą skuteczność otrzymano podczas badań przeprowadzonych na pianinie. Biorąc pod uwagę jednak tylko same nazwy tonu podstawowego (bez określenia oktawy w której leży ton) skuteczność systemu byłaby bliska 100%. System najlepiej poradził sobie z fragmentem "O mój rozmarynie" zagrany lub śpiewany powoli. Szybkość gry wpływa znacząco na wykrywanie początku i końca dźwięków. Jedyne błędy wykrycia ilości dźwięków zaobserwowano podczas badania śpiewu.

Rozdział 7

Podsumowanie

Głównym celem pracy była budowa systemu służącego do analizowania i rozpoznawania dźwięków, który w przyszłości będzie można zaimplementować w robocie społecznym, pełniącym funkcję nauczyciela gry na instrumentach. System miał identyfikować wysokości zagranych dźwięków oraz porównywać je ze wzorcowymi wartościami.

Poddano analizie rynek dostępnych rozwiązań i wybrano aplikacje komercyjne, które swoim działaniem przypominają planowany system. Porównano dwie aplikacje, wyróżniające się na tle innych. Nie znaleziono jednak rozwiązania, które w pełni oddaje zamierzony cel.

Zebrano dostępne informacje na temat przetwarzania sygnałów dźwiękowych. Zdefiniowano pojęcie dźwięku, przytoczono fragmenty teorii muzyki, których zrozumienie jest niezbędne do dalszej analizy. Opisano od strony teoretycznej metody użyte w systemie, przy czym szczególnie skupiono się na problemie podziału całego utworu na pojedyncze dźwięki oraz rozpoznaniu ich częstotliwości podstawowej. Przytoczono narzędzia matematyczne potrzebne do przetwarzania sygnałów dyskretnych.

Następnie opisano schemat struktury działania planowanego systemu. Wyróżniono dwa tryby pracy systemu: tryb analizy wcześniej zapisanych utworów oraz tryb pracy na żywo. Opisano wejście i wyjście każdego bloku schematu struktury.

W części implementacyjnej przedstawiono środowisko MATLAB, w którym napisano program do identyfikacji wysokości dźwięków. Opisano funkcje użyte do implementacji, ich argumenty i zwracane wartości. Przedstawiono sposób wyświetlania danych, takich jak przebiegi amplitudowe zarejestrowanych dźwięków czy ich widma częstotliwościowe, oraz schemat przyrównywania dwóch wartości częstotliwości do siebie i wyszukiwania ewentualnych błędów w zagranych utworach. Dla wygody użytkownika, zaprojektowano interfejs graficzny aplikacji przedstawiający funkcjonalności systemu.

Przedstawione rozwiązania pozwalają na budowę systemu służącego do analizowania oraz rozpoznawania wysokości dźwięków, co było głównym celem pracy.

W ramach testów przeprowadzono badania na pianinie, gitarze oraz podczas śpiewania. Zaprezentowano zachowanie systemu na etapach nagrywania, podziału utworu na dźwięki oraz rozpoznania częstotliwości podstawowej. Wyznaczano wysokości zagranych dźwięków oraz porównano je z wartościami wzorcowymi fragmentów wybranych utworów. W wyniku otrzymano widma częstotliwościowe każdego

pojedynczego dźwięku oraz spis odczytanych wartości częstotliwości tonów podstawowych.

Z dostępnych funkcji systemu można uzyskać informację na temat wysokości dźwięków i samemu ją porównać, jednak automatyczne działanie systemu nie daje satysfakcjonujących efektów. Metoda automatyczna podziału utworu na pojedyncze dźwięki nie zawsze prawidłowo rozpoznaje początek i koniec zagranych dźwięków. Stworzony interfejs do dodawania wzorcowych plików MIDI nie działa poprawnie. Dlatego na tym etapie nie jest możliwa automatyczna analiza ewentualnych błędów popełnionych podczas gry na instrumencie. Należy poprawić również działanie aplikacji graficznej oraz odpowiednio połączyć funkcjonalności ze stworzonym interfejsem graficznym.

Funkcjonalny system do rozpoznawania dźwięków może stanowić podstawę do opracowania bardziej zaawansowanych struktur do analizy sygnałów dźwiękowych. Pracę można traktować jako zbiór dostępnych, przeanalizowanych i przebadanych metod rozpoznawania wysokości dźwięków.

W przyszłości należy skupić się na poprawie i rozbudowie metod automatycznych rozpoznawania początku i końca dźwięku oraz porównania dwóch wartości częstotliwości, w celu weryfikacji ewentualnych różnic pomiędzy wartością zagranych a wartością wzorcową. Dodatkowym kierunkiem pracy nad systemem może być analiza utworów polifonicznych, zagranych na więcej niż jednym instrumencie. Na tym etapie efekt końcowy pracy jest zadowalający. W dalszej perspektywie, można przetestować strukturę systemu na innych instrumentach, a jako wynik końcowy stworzyć zapis nutowy zamiast nazw zwyczajowych.

Literatura

- [ale] alexkay. Spek Acoustic Spectrum Analyser GitHub. <https://github.com/alexkay/spek>.
- [CDu] CDur.pl. Składowe harmoniczne dźwięku. http://www.cdur.pl/Artykuly/5,Fizyczne_podstawy_dzwieku_i_tercjowej_budowy_akordow.
- [Chw] Tomasz Chwiej. Algorytm radix-2. <http://galaxy.agh.edu.pl/~chwiej/mn/wyk/fft.pdf>.
- [Edu] Zintegrowana Platforma Edukacyjna. Notacja muzyczna. <https://zpe.gov.pl/a/wartosci-rytmiczne-nut-i-pauz---wprowadzenie-podstawowych-elementow-notacji-muzycznej>.
- [Hab68] Jerzy Habela, *Słowniczek muzyczny*. Warszawa: PWM, 1968.
- [HiF] HiFi.pl. Próbkowanie sygnału. <https://www.hifi.pl/sloownik/probkowanie.php>.
- [Kon] Jarosław Konieczny. Podstawy próbkowania sygnałów. https://home.agh.edu.pl/~koniejar/LVlinki/Wykl_probkowanie.pdf.
- [laba] Muzyczne laboratorium. Analizator widma. <https://muzycznelaboratorium.pl/8-czestotliwosci-nut>.
- [labb] Muzyczne laboratorium. Częstotliwość nut i oktawa. <https://muzycznelaboratorium.pl/8-czestotliwosci-nut>.
- [LLC] Prometheus Interactive LLC. Datuner 3.105. <https://www.appbrain.com/app/datuner-tuner-metronome/com.bork.dsp.datuna>.
- [Lyo10] Richard G. Lyons, *Understanding Digital Signal Processing*. Pearson, 2010.
- [Mar] Michał Marchewczyk. Transformacja Fouriera. https://home.agh.edu.pl/~zobmat/2020/II_mich_mar/transformacje.html.
- [Mata] MathWorks. Środowisko MATLAB. <https://www.mathworks.com/products/matlab.html>.
- [MATb] MATLAB. Funkcja audiorecorder. https://www.mathworks.com/help/matlab/ref/audiorecorder.html?s_tid=srchtitle_audiorecorder_1.
- [MATc] MATLAB. Funkcja fft. <https://www.mathworks.com/help/matlab/ref/fft.html>.
- [MATd] MATLAB. Obwiednia sygnału. <https://www.mathworks.com/help/signal/ref/envelope.html>.

- [MATE] MATLABlog. App designer. <http://matlablog.ont.com.pl/matlab-app-designer/>.
- [Mee] Oh Why? Meek. Jestem dźwiękiem i słowem. <https://www.youtube.com/watch?v=nWfiJZ2CDnU>.
- [mt] Online mic test. Pitch detector. <https://www.onlinemictest.com/tuners/pitch-detector/>.
- [mus] musictutorapp. Music tutor. <https://musictutorapp.com/musictutor>.
- [Muz] Muzyczny.pl. Zapis nutowy. <https://muzyczny.pl/portal/zapis-nutowy/>.
- [nie] Autor nieznany. Algorytmy detekcji częstotliwości podstawowej. <https://sound.eti.pg.gda.pl/student/akmuz/03-f0.pdf>.
- [Pla] Google Play. Note recognizer. <https://play.google.com/store/apps/details?id=nl.robertloeberdevelopment&hl=pl&gl=US>.
- [Pro] ProGuitar. Pro guitar tuner. <https://www.proguitar.com/guitar-tuner>.
- [Rad] Karol Radecki. Modulacje analogowe v. 2.2, materiały do wykładu sygnały i modulacje Politechnika Warszawska. <https://www.ire.pw.edu.pl/~ksnopek/SYMSE/modAM.pdf>.
- [Rud] Zbigniew Rudnicki. Co to jest MATLAB? http://www.kkiem.agh.edu.pl/sites/all/old_kkiem/dydakt/matlab/1matlab.htm.
- [Set05] William A. Sethares, *Tuning, Timbre, Spectrum, Scale*. Springer Science & Business Media, 2005.
- [Sha] Shazam. Shazam. <https://www.shazam.com/home>.
- [Sou] SoundHound. Soundhound. <https://www.soundhound.com/soundHound>.
- [spe] spek.cc. Aplikacja spek acoustic spectrum analyser. <http://spek.cc/>.
- [Stu] Estrada Studio. Podstawy akustyki. <https://estradaistudio.pl/porady/29419-troche-podstaw-akustyki>.
- [Sza03] Jerzy Szabatin, *Przetwarzanie sygnałów*. Wydawnictwa Komunikacji i Łączności WKŁ, 2003.
- [the] theonlinemetronome. Instrument tuner online. <https://theonlinemetronome.com/instrument-tuner>.
- [um] Pick up music. Guitar tuner. <https://www.pickupmusic.com/tools/guitar-tuner>.
- [Wika] Wikipedia. BPM. https://pl.wikipedia.org/wiki/Uderzenia_na_minut%C4%99.
- [Wikb] Wikipedia. Cechy dźwięku. [https://pl.wikipedia.org/wiki/D%C5%BAwi%C4%99k_\(muzyka\)](https://pl.wikipedia.org/wiki/D%C5%BAwi%C4%99k_(muzyka)).
- [Wikc] Wikipedia. Meek, oh why? https://pl.wikipedia.org/wiki/Meek,_Oh_Why%3F.
- [Wikd] Wikipedia. Midi. <https://pl.wikipedia.org/wiki/MIDI>.
- [Wike] Wikipedia. O mój rozmarynie. https://pl.wikipedia.org/wiki/O_%C3%B3j_rozmarynie.

- [Wikf] Wikipedia. Obwiednia sygnału. https://pl.wikipedia.org/wiki/Obwiednia_sygna%C5%82u.
- [Wikg] Wikipedia. Szereg fouriera. https://pl.wikipedia.org/wiki/Szereg_Fouriera.
- [Wikh] Wikipedia. Utwór muzyczny. https://pl.wikipedia.org/wiki/Utw%C3%B3r_muzyczny.
- [Zie05] Tomasz P. Zieliński, *Cyfrowe przetwarzanie sygnałów od teorii do zastosowań*. Wydawnictwa Komunikacji i Łączności sp. z o.o Warszawa, 2005.

Spis tabel

6.1	Wyniki rozpoznanych częstotliwości gamy zagranej na pianinie . . .	30
6.2	Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej szybko na pianinie (część 1)	30
6.3	Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej szybko na pianinie (część 2)	30
6.4	Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej wolno na pianinie (część 1)	31
6.5	Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej wolno na pianinie (część 2)	31
6.6	Podsumowanie wyników badań dla pianinia	31
6.7	Wyniki rozpoznanych częstotliwości pustych strun zagrzanych na gitarze	32
6.8	Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej szybko na gitarze (część 1)	32
6.9	Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej szybko na gitarze (część 2)	32
6.10	Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej wolno na gitarze (część 1)	32
6.11	Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej wolno na gitarze (część 2)	32
6.12	Podsumowanie wyników badań dla gitary	33
6.13	Wyniki rozpoznanych częstotliwości fragmentu zaśpiewanej gamy .	33
6.14	Wyniki rozpoznanych częstotliwości "O mój rozmarynie" szybko zaśpiewanej (część 1)	33
6.15	Wyniki rozpoznanych częstotliwości "O mój rozmarynie" szybko zaśpiewanej (część 2)	33
6.16	Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej wolno zaśpiewanej (część 1)	34
6.17	Wyniki rozpoznanych częstotliwości "O mój rozmarynie" zagranej wolno zaśpiewanej (część 2)	34
6.18	Podsumowanie wyników badań dla śpiewu	34

Spis rysunków

2.1	Aplikacja Note Recognizer	6
2.2	Interfejs graficzny aplikacji Note Recognizer	7
2.3	Spek Acoustic Spectrum Analyser	7
3.1	Składowe harmoniczne dźwięku	10
3.2	Notacja muzyczna	10
3.3	Porównanie widma całego utworu, do pojedynczego dźwięku. a) widmo częstotliwościowe całego utworu, b) widmo częstotliwościowe pojedynczego dźwięku	12
3.4	Ręczne wyznaczanie czasu trwania dźwięków	13
3.5	Obwiednia (envelope) sygnału	14
4.1	Schemat struktury systemu do analizy zapisanych utworów	17
4.2	Schemat struktury systemu na żywo	18
5.1	Przebieg całego utworu	21
5.2	Przebieg amplitudowy pojedynczego dźwięku	21
5.3	Wykres widma przedstawiający efekt działania funkcji plot	23
5.4	Algorytm porównania częstotliwości	24
5.5	Wygląd interfejsu graficznego aplikacji	25
6.1	Przebieg amplitudowy całego utworu (test)	27
6.2	Przebieg amplitudowy pojedynczego dźwięku (test)	27
6.3	Podział utworu na pojedyncze dźwięki za pomocą metody automatycznej	28
6.4	Podział utworu na pojedyncze dźwięki za pomocą metody ręcznej	28
6.5	Widma częstotliwościowe pojedynczych dźwięków	29
A.1	Częstotliwości tonów podstawowych i ich harmonicznch	43

Dodatek A

Częstotliwości tonów podstawowych i ich harmonicznyc

nuta	ton podstawowy [Hz]	pierwsza harmoniczna [Hz]	druga harmoniczna [Hz]	trzecia harmoniczna [Hz]
A ₀	27.5	55	82.5	110
B ₀	29.14	58.27	87.41	116.54
H ₀	30.87	61.74	92.6	123.47
C ₁	32.7	65.41	98.11	130.81
C# ₁	34.65	69.3	103.94	138.59
D ₁	36.71	73.42	110.12	146.83
D# ₁	38.89	77.78	116.67	155.56
E ₁	41.2	82.41	123.61	164.81
F ₁	43.65	87.31	130.96	174.61
F# ₁	46.25	92.5	138.75	185
G ₁	49	98	147	196
G# ₁	51.91	103.83	155.74	207.65
A ₁	55	110	165	220
B ₁	58.27	116.54	174.81	233.08
H ₁	61.74	123.47	185.21	246.94
C ₂	65.41	130.81	196.22	261.63
C# ₂	69.3	138.59	207.89	277.18
D ₂	73.42	146.83	220.25	293.66
D# ₂	77.78	155.56	233.35	311.13
E ₂	82.41	164.81	247.22	329.63
F ₂	87.31	174.61	261.92	349.23
F# ₂	92.5	185	277.5	369.99
G ₂	98	196	294	392
G# ₂	103.83	207.65	311.48	415.3
A ₂	110	220	330	440
B ₂	116.54	233.08	349.62	466.16
H ₂	123.47	246.94	370.41	493.88
C ₃	130.81	261.63	392.44	523.25
C# ₃	138.59	277.18	415.77	554.37
D ₃	146.83	293.66	440.5	587.33
D# ₃	155.56	311.13	466.69	622.25

Rysunek A.1 Częstotliwości tonów podstawowych i ich harmonicznyc