POLITECHNIKA WROCŁAWSKA

WYDZIAŁ ELEKTRONIKI

1. KIERUNEK: Informatyka
2. SPECJALNOŚĆ: Inżynieria internetowa
3. PRACA DYPLOMOWA
4. INŻYNIERSKA

Aplikacja mobilna umożliwiająca modulowanie głosu ludzkiego

Mobile Application for Human Voice Modulation

AUTOR:  
Piotr Sotor

AUTOR:

Imię i nazwisko



PROWADZĄCY PRACĘ:

* + - * 1. dr inż. Jacek Mazurkiewicz, jednostka

1. OCENA PRACY:
2. WROCŁAW 2015

[1 Wstęp 4](#_Toc436783022)

[2 Cel i zakres pracy 5](#_Toc436783023)

[3 Przegląd istniejących aplikacji o podobnych funkcjonalnościach 6](#_Toc436783024)

[3.1 Odtwarzacze multimediów 6](#_Toc436783025)

[3.2 Aplikacje przeznaczenia rozrywkowego 7](#_Toc436783026)

[4 Opis procesu modulacji dźwięku 8](#_Toc436783027)

[5 Przegląd zagadnień modulacji dźwięku 10](#_Toc436783028)

[5.1 Charakterystyki sygnału mowy 10](#_Toc436783029)

[5.2 Stacjonarność sygnału 10](#_Toc436783030)

[5.3 Twierdzenie Kotielnikowa-Shannona 10](#_Toc436783031)

[5.4 Funkcje okna 10](#_Toc436783032)

[5.5 Porównanie kilku wybranych 10](#_Toc436783033)

[5.6 Cross-fading 10](#_Toc436783034)

[5.7 Transformata Fouriera 10](#_Toc436783035)

[5.7.1 Szybka transformata Fouriera (FFT) 10](#_Toc436783036)

[5.7.2 Odwrotna transformata Fouriera 10](#_Toc436783037)

[6 Aplikacja realizująca modulowanie głosu 11](#_Toc436783038)

[6.1 Zdefiniowane modulacje 11](#_Toc436783039)

[6.1.1 Filtr o konfigurowalnej przepustowości pasm częstotliwości 11](#_Toc436783040)

[6.1.2 Skalowanie sygnału w dziedzinie częstotliwości 12](#_Toc436783041)

[6.1.3 Rozmycie charakterystyk częstotliwościowych 12](#_Toc436783042)

[6.2 Zrealizowana funkcja okna 13](#_Toc436783043)

[6.3 Zaimplementowane metody cross-fadingu 15](#_Toc436783044)

[6.3.1 Liniowy 16](#_Toc436783045)

[6.3.2 Sinusowo – cosinusowy 17](#_Toc436783046)

[6.4 Opis programu 17](#_Toc436783047)

[6.4.1 Założenia projektowe 17](#_Toc436783048)

[6.4.1.1 Założenia funkcjonalne 17](#_Toc436783049)

[6.4.1.2 Założenia niefunkcjonalne 17](#_Toc436783050)

[6.4.2 Struktura projektu 17](#_Toc436783051)

[6.4.3 Biblioteki zewnętrzne 17](#_Toc436783052)

[6.4.4 Analiza wydajnościowa 18](#_Toc436783053)

[6.5 Instrukcja obsługi 18](#_Toc436783054)

[6.5.1 Opis interfejsu 18](#_Toc436783055)

[6.5.2 Pliki konfiguracyjne 18](#_Toc436783056)

[6.6 Przykłady działania 18](#_Toc436783057)

[7 Wnioski i podsumowanie 19](#_Toc436783058)

# Wstęp

Przełom tysiącleci niewątpliwie można opatrzyć wieloma etykietami znaczących z punktu widzenia znanej ludzkości historii punktów zwrotnych lub (r)ewolucji. Wymieniać je można długo – od rewolucji obyczajowej, przez rewolucję komputerową, na rewolucji Internetowej kończąc. Niespotykany dotąd w historii szybki postęp w różnych dziedzinach techniki umożliwił konstrukcję czystych ekologicznie samochodów, spoglądanie w zakątki kosmosu dzięki orbitalnym teleskopom, oraz globalną komunikację z niemalże dowolnego miejsca na Ziemi. Dzięki postępowi i miniaturyzacji elektroniki ludzie dosłownie otaczają się komputerami, które codziennie noszą w kieszeniach swoich spodni lub czeluściach torebki czy to w formie kart płatniczych czy telefonów komórkowych. W ciągu zaledwie dwóch dekad urządzenie, które jeszcze w latach dziewięćdziesiątych XX wieku było gadżetem dla biznesmenów, stało się już niemal przedmiotem pierwszej potrzeby. Ludzie wykształcili w sobie wewnętrzny przymus pozostawania w ciągłym kontakcie między sobą, a sama komunikacja przybiera coraz to wytworniejsze formy, choć nie można powiedzieć by starsze były wypierane przez nowe. Na ogół nowe formy uzupełniają przestrzenie jeszcze nie zagospodarowane przez poprzedników. Można więc powiedzieć, że z całą pewnością podstawowa funkcja telefonu komórkowego – połączenie telefoniczne – będzie integralną częścią globalnej komunikacji jeszcze przez dłuższy czas.

Wraz z rozpowszechnieniem telefonii komórkowej wiele firm i organizacji zdecydowało się przenieść część niezbędnych interakcji z klientem z kontaktu twarzą w twarz do rozmowy telefonicznej. Nie jest to regułą, ale wiele z nich zastrzega sobie prawo do nagrywania prowadzonych połączeń. Wszystko po to, by w przypadku wątpliwości ze strony klienta firma posiadała niezbity dowód na zawarcie z pracownikiem umowy słownej, która jest równie wiążąca co umowa pisemna.

Dla nikogo nie będzie zaskoczeniem stwierdzenie, że jednym z najbardziej wartościowych surowców w obecnych czasach jest informacja. *Informacja* jest jednak terminem bardzo ogólnym. Jest nią tak imię, nazwisko, jak i lokalizacja najczęściej odwiedzanych sklepów, czy dane cechy personalne takie jak skan siatkówki, odcisk palca lub dokładna charakterystyka mowy, nawyki wymowy. Zwłaszcza te ostatnie mogą stać się przedmiotem nielegalnego wykorzystania i zmanipulowania, biorąc pod uwagę częste przechowywanie nagrań firm bez wykorzystania odpowiednich zabezpieczeń.

Nasuwa się więc pytanie. Czy jedyną reakcją na informację o prawdopodobnym nagrywaniu rozmowy musi być wybór pomiędzy natychmiastowym zakończeniem połączenia, a zaryzykowaniem, że nic złego się z tym nagraniem w przyszłości nie stanie? A co, jeśli zostanie dodana do dylematu również trzecia możliwość: zniekształcenie własnego głosu w taki sposób, by był zrozumiały, ale uniemożliwiał jednoznaczne zidentyfikowanie rozmówcy?

Poniższa praca jest dopiero pierwszym krokiem ku zaoferowaniu użytkownikowi telefonu komórkowego świetlanej trzeciej opcji wyboru. Przedstawia efekty różnych zabiegów modulujących dźwięk mowy, oraz oszacowuje zdolność współczesnego sprzętu do wykonania złożonej modulacji na dźwięku wysokiej jakości w czasie rzeczywistym, który jest niezbędny z punktu widzenia prowadzenia rozmowy telefonicznej.

# Ce­­­­l i za­­­kres pracy

[co zostało zrealizowane, jakie przyjęto założenia, przewodnik po pracy, co i gdzie po kolei jest omawiane (akapit o każdym z rozdziałów, króciutkie ich streszczenie, będzie sporo tekstu :D)]

# Przegląd istniejących aplikacji o podobnych funkcjonalnościach

Najpopularniejszym źródłem aplikacji dedykowanych systemowi Android jest sklep internetowy Google Play. Duża liczba dostępnych na nim aplikacji jest jedną z przyczyn popularności samego systemu Android. Z racji jego popularności przegląd aplikacji został oparty o zbiór dostępny właśnie na Google Play.

Dostępne aplikacje umożlwiające modulowanie głosu można podzielić na poniżej opisane dwie kategorie.

## Odtwarzacze multimediów

Szeroka oferta sklepu internetowego zawiera wiele proponowanych aplikacji służących odtwarzaniu muzyki zapisanej z różnych formatach i jakościach. Oprócz odtwarzania i organizowania plików w listy odtwarzania (ang.p*laylists*) część bardziej zaawansowanych umożliwia modulacje odtwarzanego dźwięku. Najczęstszym stosowanym zabiegiem jest zastosowanie korektora (ang. *equalizer*). Jest niczym innym niż filtrem tłumiącym lub wzmacniającym określone pasma widma odtwarzanego dźwięku. Uzyskanie podobnego efektu jest jednym z celów poniższej pracy.

Odtwarzanie szeregu plików dźwiękowych wymusza wystąpienie sytuacji rozpoczęcia odtwarzania następnego w momencie zakończenia poprzedniego. W niektórych aplikacja umożliwia się ustawienie płynnego przechodzenia pomiędzy utworami poprzez stopniowe wyciszanie kończącego się i zgłośnienie kolejnego. Zabieg ten nazywany jest cross-fadingiem i został również wykorzystany w poniższej pracy.

Poniżej przedstawiam przykładową aplikację oferującą powyżej opisane funkcjonalności.

**Neutron Music Player**

Oferowane funkcjonalności:

* Obsługa plików w formatach: MP3, WMA, 3GP, FLAC, WAVE i inne
* Organizacji plików w listy utworów, płynne przejście pomiędzy utworami
* Graficzny, konfigurowalny korektor
* Filtry szumów, hałasu statycznego
* Crossfeed poprawiający jakość dźwięku stereo

## Aplikacje przeznaczenia rozrywkowego

[nagraj – zmien – odtworz, lub voip clienty]

Choć odtwarzanie multimediów również można uznać za funkcjonalność przeznaczenia rozrywkowego to jednak nie należy rozpatrywać tych aplikacji w tej samej kategorii. W przypadku odtwarzaczy zabiegi modulacyjne dźwięku mają na celu jedynie wzmocnienie wrażeń słuchowych użytkownika, ale nie są główną funkcjonalnością aplikacji. Programy sklasyfikowane tutaj jako rozrywkowe stawiają modulacje dźwięku jako główną funkcjonalność.

Najczęściej przedstawiają taki sam scenariusz użycia. Umożliwiają nagranie dźwięku lub wybranie pliku z pamięci trwałej, nałożenie wybranego efektu modulującego na nagranie i odtworzenie nagrania. Powyższy scenariusz został wykorzystany w poniższej pracy. Część aplikacji tej kategorii ponadto integrują w sobie funkcjonalności klienta sieci VoIP (*Voice over Internet Protocol*) co pozwala na prowadzenie rozmów poprzez łącze internetowe wraz z wykonaniem modulacji głosu rozmówcy. Nawiązanie połączenia z innym użytkownikiem często wiąże się z dodatkowymi kosztami opłacenia serwera sieci VoIP określonego przez dostawcę aplikacji. Z reguły nie umożliwiają łączenia się z innymi poprzez inne serwery niż zdefiniowane przez producenta.

Jakieś przykładowe podać

# Opis procesu modulacji dźwięku

Modulacja dźwięku jest niczym innym niż procesem przetwarzania pewnego sygnału. Z tego powodu wykorzystuje się do tego wiele metod powszechnie wykorzystywanych w bardzo różnych dziedzinach technicznych jak na przykład telekomunikacja lub sejsmografii.   
Z punktu widzenia wykorzystywanych algorytmów nie ma różnicy czy przetwarzane są próbki przebiegu fali akustycznej czy sejsmicznej.

Aby przeprowadzić modulację dźwięku niezbędne jest uzyskanie zapisu dźwięku w postaci fali. Zrealizowane w ten sposób nagranie przedstawia rzeczywisty sygnał ciągły w postaci listy wartości wynikającej z procesu próbkowania sygnału i kwantyzacji próbek. Parametrem próbkowania sygnału jest częstotliwość próbkowania, która wyznacza liczbę próbek przypadającą na sekundę nagrania, a parametrem kwantyzacji jest jej rozdzielczość, czyli dokładność odwzorowania realnej wartości na dyskretną rzeczywistość maszyny. Różnica pomiędzy wynikiem kwantyzacji, a rzeczywistą wartością sygnału w danej próbce nosi nazwę *błędu kwantyzacji*. W poniższej pracy umożliwia się nagrywanie i odtwarzanie dźwięku z częstotliwościami próbkowania 8000 Hz, 16000 Hz i 44100 Hz, oraz stosuje się zapis próbek z rozdzielczością 16 bitową, co oznacza, że próbka reprezentowana jest jedną z wartości.

Praca nad sygnałami rzeczywistymi, a nie syntetycznie uzyskanymi w podejściu teoretycznym, wiąże się z koniecznością wzięcia pod uwagę kilku kwestii. Pierwszą rzeczą jest zawsze występujący szum lub zniekształcenia. Można zakwalifikować do nich zarówno odgłosy otoczenia jak i niedoskonałość urządzenia rejestrującego. Rozróżnić je można pod kątem deterministyczności występowania oraz statyczności przebiegu. Tak więc szumy urządzenia, który często ma charakter *szumu białego* można postrzegać jako zakłócenie deterministyczne (ponieważ występuje ciągle) i statyczne (ponieważ nie zmienia się w czasie). Dokładnym przeciwieństwem będzie odgłos przejeżdżającego samochodu lub osoby rozmawiające w tle.

Kolejną cechą sygnałów rzeczywistych, a w szczególności fal akustycznych, jest ich niedeterministyczność, zmienność w czasie i nieokresowość. Zastosowanie transformaty Fouriera na takim sygnale daje w rezultacie wynik nie mający z rzeczywistym sygnałem wiele wspólnego. Dzieje się tak z powodu własności samej transformaty, która zakłada okresowość sygnału. Ponieważ wykorzystanie transformaty Fouriera jest kluczowym elementem w zakresie przetwarzania sygnałów niezbędne jest wykonanie pewnych operacji, by rezultaty transformaty były możliwie wiarygodne. Pierwszym krokiem jest podział długiego sygnału na krótkie fragmenty, dalej nazywane *oknami czasowymi*. Prowadzi to do uzyskania stosunkowo niezmiennego przebiegu sygnału, który można już uznać za sygnał stacjonarny. W przypadku przetwarzania dźwięku przyjmuje się za optymalną długość okna ciąg próbek reprezentujący nagranie o długości *10 – 20 ms*. Dokładna długość zależy od użytej częstotliwości próbkowania sygnału, ponieważ dąży się do tego, by wybrana liczba próbek miała postać , tj. była potęgą liczby 2. Jest to warunek konieczny by w przetwarzaniu wykorzystać szybką transformatę Fouriera. Dokładna liczba próbek może zostać wyliczona z następującego wzoru:

W przypadku zapisu kilku kanałów przy użyciu jednego pliku należy w tym momencie zadbać o rozdzielenie poszczególnych kanałów. Próbki sygnału dla różnych kanałów zapisywane są naprzemiennie po sobie,, a przetwarzanie konkretnych kanałów powinno zostać wykonane niezależnie.

Podział sygnału na okna czasowe rozwiązuje problem stacjonarności sygnału. Pozorną okresowość sygnału uzyskuje się poprzez zastosowanie *funkcji okna.* W rezultacie jej zastosowania przebieg początku sygnału upodabnia się do przebiegu końca sygnału. Przebieg *funkcji okna* wymaga zadbania o to, by kolejne *okna czasowe* nakładały się na siebie. W przeciwnym razie odwzorowanie sygnału w modulacji byłoby przekłamane.

Po wykonaniu *funkcji okna* na fragmencie sygnału należy wykorzystać transformatę Fouriera. W jej wyniku następuje przejście z postrzegania sygnału w dziedzinie czasu na dziedzinę częstotliwości. W rezultacie otrzymuje się informację o amplitudzie danej częstotliwości składowej oraz o fazie danej fali. W kontekście przetwarzania dźwięku nie wykorzystuje się w żadnym stopniu przesunięcia w fazie poszczególnych częstotliwości składowych. Przetwarzanie sygnału w dziedzinie częstotliwości jest kluczowe dla modulacji.

Przeprowadzenie modulacji w znacznym stopniu ogranicza się do odpowiedniej zmiany wartości widma sygnału uzyskanego dzięki transformacie. W niniejszej pracy zaproponowano 3 modulacje: skalowanie częstotliwości, filtr przepustowy oraz rozmycie charakterystyk widma.

Po przetworzeniu widma danego *okna czasowego* należy zastosować na nim odwrotną transformatę Fouriera. Jest to zabieg odwrotny do transformaty Fouriera, a więc na podstawie zapisu sygnału w formie widma uzyskuje przebieg sygnału w dziedzinie czasu.

Ostatnim krokiem całego procesu jest połączenie nakładających się fragmentów *okien czasowych* aby uzyskać z powrotem pełny przebieg sygnału już zmienionego. W tym celu wykorzystany został mechanizm cross-fadingu. Polega on na stopniowym wyciszaniu wcześniejszego *okna czasowego* i zgłośnieniu kolejnego.

# Przegląd zagadnień modulacji dźwięku

[Krótki wstęp o tym, że są to zagadnienia niezbędne dla tego tematu]

## Charakterystyki sygnału mowy

[Formanty dźwięków, ton krtaniowy]

## Stacjonarność sygnału

[Czym jest, konsekwencja odnośnie transformaty Fouriera]

## Twierdzenie Kotielnikowa-Shannona

[O czym mówi twierdzenie, jakie są jego konsekwencje dla tematu pracy]

## Funkcje okna

[Zapobiega wyciekom widma przy fft]

## Porównanie kilku wybranych

[Kilka funkcji przedstawionych poprzez wzory matematyczne, przebieg wartości funkcji w dziedzinie oraz wykresy wyników fft sygnału]

## Cross-fading

[Po co jest stosowany, wykresy przedstawiające przebiegi (czy nie lepiej wykresy tylko przy części implementacyjnej?)]

## Transformata Fouriera

[Czym jest, splot w dziedzinie czasu zamienia się w mnożenie w dziedzinie częstotliwości, interpretacja wyniku – tablica *n* wartości reprezentujących widma o szerokości *fmax/n* a *fmax = fs/2* ]

### Szybka transformata Fouriera (FFT)

[zasada działania -> wymóg liczby próbek wielkości potęgi dwójki]

### Odwrotna transformata Fouriera

[czym jest]

# Aplikacja realizująca modulowanie głosu

## Zdefiniowane modulacje

Przedmiotem tej pracy jest implementacja i zastosowanie do przetwarzania sygnału dźwiękowego trzech modulacji. Poniżej przedstawiam opis działania każdego z nich, sposób reprezentacji dla użytkownika aplikacji, listing kodu oraz wynik jego zastosowania.

Danymi wejściowymi dla każdego filtru jest tablica będąca wynikiem transformaty Fouriera oraz parametry zależne od modulacji. Tablica wejściowa jest zbiorem *n* wartości reprezentujących amplitudy kolejnych składowych częstotliwości. Każdy element tablicy odpowiada amplitudzie składowych z pasma o szerokości . Można więc zapisać, że

### Filtr o konfigurowalnej przepustowości pasm częstotliwości

[sposób reprezentacji przepustowości filtru, słowny opis implementacji, listing implementacji, przykład modulacji]

Filtr przepustowy opiera swoje działanie na zmianie amplitud określonych pasm częstotliwości. Odpowiada temu odpowiednia modyfikacja konkretnych wartości tablicy wejściowej. Pełna konfigurowalność pozwala na określenie stopnia wyciszenia określonych składowych. Jest to mechanizm bardzo podobny do korektora stosowanego w odtwarzaczach multimedialnych. Zastosowanie tej samej wartości dla pełnego widma sygnału daje w wyniku zmianę głośności całego sygnału. Zastosowanie filtru dolnoprzepustowego na sygnale uzyskanym przy wysokiej częstotliwości próbkowania skutkuje obniżeniem jego jakości – wynikiem jest sygnał analogiczny do sygnału próbkowanego z niższą częstotliwością. Górnoprzepustowa konfiguracja umożliwia ukrycie różnych charakterystyk mowy, takich jak ton krtaniowy lub formanty.

Reprezentację stanowi zbiór par wartości, gdzie pierwsza oznacza częstotliwość, dla której określany jest współczynnik przepustowości będący drugą wartością pary. Umożliwia to graficzną reprezentację w postaci wykresu, którego osiami są częstotliwość i współczynnik przepustowości. Na wykresie punkty reprezentują konkretnie określone pary, a odcinki pomiędzy punktami przebieg wartości dla pozostałych częstotliwości.

Poniżej przedstawiono szczegóły implementacyjne tego filtru.

Implementacja + wykresy

### Skalowanie sygnału w dziedzinie częstotliwości

[znaczenie magicznego współczynnika skalowania, słowny opis implementacji, listing implementacji, przykład modulacji]

Różnica w wysokości dźwięków postrzegana przez ludzkie ucho w kontekście fizycznym sprowadza się do porównania częstotliwości określonych formantów. W kontekście muzycznym mówi się o *interwałach* dźwiękowych, które pozwalają na jasne określenie różnicy w wysokości dźwięków. Podstawowym interwałem jest oktawa, która w kontekście fizycznym odpowiada częstotliwości drugiej harmonicznej niższego dźwięku. Oznacza to, że dźwięk różny o oktawę posiada składowe o dwukrotnie wyższej częstotliwości. W wyniku podzielenia oktawy na mniejsze części otrzymuje się inne interwały.

Parametrem tej modulacji jest współczynnik skalowania częstotliwości. Określa on zarówno kierunek skalowania – w niższe lub wyższe dźwięki – ale i również oczekiwany interwał przekształcenia. Przyjmuje więc wartości rzeczywiste dodatnie. Współczynnik jest wartością przez która multiplikowane są częstotliwości składowe sygnału. Amplituda składowych nie ulega jednak zmianie. Poniżej przedstawiono przykładowe wartości współczynnika wraz z odpowiadającym interwałom:

* *f = 0.5 –* oktawa w dół
* *f = 1* – pryma, dźwięk bez zmian
* *f = 1.58 –* kwinta czysta w górę
* *f = 2* – oktawa w górę

Powyższą modulację można wykorzystać wprost do zmiany wysokości dźwięku lub w połączeniu z filtrem przepustowym. Drugi wariant może dać ciekawe rezultaty w przypadku zastosowania najpierw jednostronnie przepuszczalnego filtru, a następnie skalowania z odpowiednim współczynnikiem, by przenieść część pozostałych formantów w pasmo wcześniej wyciszone – współczynnik *f > 1* dla pierwszego filtru dolnoprzepustowego, współczynnik *f < 1* dla pierwszego filtru górnoprzepustowego.

Poniżej przedstawiono szczegóły implementacyjne modulacji.

Implementacja + wykresy

### Rozmycie charakterystyk częstotliwościowych

[znaczenie szerokości rozmycia, słowny opis implementacji, listing implementacji, przykład modulacji]

Z punktu widzenia koncepcji jest to w pełni autorski pomysł na modulowanie dźwięku. Nie znaleziono w literaturze podobnych rozwiązań. W wyniku tego należy rozpatrywać tą modulacje w kontekście eksperymentu.

Operacja polega na utracie dokładnej informacji dotyczącej częstotliwości i amplitudy kolejnych formantów widma. Dokonuje się tego poprzez wyznaczenie nowej wartości amplitudy dla danego elementu tablicy przedstawiającej widmo sygnału w wyniku obliczenia wartości średniej sąsiednich elementów. Liczba sąsiednich pasm branych pod uwagę zależy od wartości parametru wejściowego *r* określającego szerokość rozmycia. Dosłownie parametr określa liczbę elementów uśrednianych zarówno w stronę wyższych jak i niższych częstotliwości. Zatem dla wartości *r = 5* wynikowa wartość i-tego elementu tablicy obliczana jest w oparciu o elementy o indeksach *(i-5, i-4, i-3, i-2, i-1, i, i+1, i+2, i+3, i+4, i+5)*, z uwzględnieniem granicznych przypadków na końcach tablicy.

Zastosowanie tej modulacji z niską wartością parametru *r* (dla ) sprawia, że nagranie zostaje wyraźnie zniekształcone, jednakże nadal treść w nim zawarta jest zrozumiała. Wyższe wartości parametru *r* skutkują nieodwracalną utratą treści.

Poniżej przedstawiono szczegóły implementacyjne modulacji.

Implementacja + wykresy

## Zrealizowana funkcja okna

Na potrzeby pracy zaimplementowano funkcję okna opracowaną przez austriackiego meteorologa Juliusa von Hanna. Od jego nazwiska funkcję nazywa się *oknem Hanna* lub *oknem Hanninga*. Wybraną ją ponieważ w znacznym stopniu redukuje efekt wycieku widma w procesie dyskretnej transformaty Fouriera, co jest ważnym czynnikiem wpływającym na dokładność wyników transformaty. Wysoką redukcję wycieku widma funkcja uzyskuje dzięki temu, że początkowe i końcowe wartości próbek w przetwarzanym oknie czasowym redukowane są do wartości bliskich zeru.

Funkcja okna Hanninga opisana jest następującym wzorem:

Wszystkie prezentowane wykresy przedstawiają wartości dla pracy z dźwiękiem nagranym z częstotliwością próbkowania 8 kHz, co w przekłada się na zastosowanie 128 elementowych okien czasowych. Rzeczywisty czas trwania takiego okna wynosi .

Przebieg wartości funkcji przedstawia *Wykres 1*

Wykres 1 Przebieg okna Hanninga

Zastosowanie funkcji okna na oknie czasowym sygnału zostało zaimplementowane w sposób przedstawiony w *Listing 1.* W poniższej implementacji wykorzystano również symetryczność przebiegu funkcji Hanninga.

|  |
| --- |
| **private float**[] windowing(**float**[] timeWindow) {  **float**[] result = **null**;  **if** (timeWindow != **null**) {  result = timeWindow.clone();  **for** (**int** i = 0; i < 0.5 \* result.**length**; ++i) {  **float** factor = getHanning(i,result.**length**);  result[i] \*= factor;  result[result.**length** - 1 - i] \*= factor;  }  }  **return** result; } **private float** getHanning(**int** n, **int** windowLength) {  **return** (**float**) (0.5f\*(1-Math.*cos*((2\*Math.***PI***\*n)/windowLength))); } |

Listing 1 Implementacja funkcji okna Hanninga

Efekt działania funkcji okna na sygnale okna czasowego przedstawia *Wykres 2.* Przedstawiono na nim przebieg sygnału pierwotnego, będącego wycinkiem pełnego nagrania, przebieg wartości funkcji Hanninga dla 128 próbek sygnału wejściowego oraz przebieg sygnału wynikowego. Sygnał końcowy otrzymano poprzez mnożenie wartości próbki pierwotnej przez wartość funkcji okna dla odpowiedniego numeru próbki.

Wykres 2 Działanie funkcji okna Hanninga

## Zaimplementowane metody cross-fadingu

Pierwszym krokiem procesu przetwarzania dźwięku był podział nagrania na nakładające się okna czasowe. Następnie każde z osobna, niezależnie, poddano przetwarzaniu. Ostatnim krokiem całego procesu musi być zatem połączenie okien, w celu uzyskania pełnego, zmodulowanego nagrania. Nakładanie się fragmentów wymusza podjęcia decyzji o sposobie określania finalnych wartości w częściach wspólnych.

Zabieg cross-fadingu jest bezpośrednim krewnym szeroko opisywanego w literaturze fadingu. Różnica miedzy nimi polega na liczbie sygnałów, którymi operują. Fading działa na jednym sygnale, który jest stopniowo wyciszany lub zgłośniony. Cross-fading można opisać jako symetryczną parę tych procesów. Uznano, że na potrzeby pracy niezbędne jest zastosowanie takiego cross-fadingu, by suma obu składowych fadingów była stała. Wszystkie znane metody fadingu i cross-fadingu posiadają złożoność obliczeniową równą O(n), dlatego czas obliczeń nie jest w tym przypadku ważnym kryterium wyboru.

W ramach pracy zrealizowano dwie metody cross-fadingu. Implementacyjnie wynikową wartość danej próbki uzyskuje się w wyniku sumowania próbek sygnałów nakładających się wymnożonych przez różne współczynniki. Obliczenia można przedstawić przy pomocy następującego wzoru.

Sposób wyliczenia współczynników jest jedyną różnicą pomiędzy metodami cross-fadingu. Z tego powodu poniżej przedstawiony *Listing 2* przedstawia kod źródłowy wspólny dla obu metod.

|  |
| --- |
| /\*\*  \* **@param partA** ścieżka ściszana  \* **@param partB** - ścieżka zgłośniona  \*/  **private float**[] crossfade(**float**[] partA, **float**[] partB) {  **float**[] result = **null**;  **if** (partA != **null** && partB != **null**) {  **if** (partA.**length** == partB.**length** && partA.**length** > 0) {  result = **new float**[partA.**length**];  **for** (**int** i = 0; i < result.**length**; ++i) {  **float** factor = getCrossFadeFactor(i, result.**length**);  result[i] = (1 - factor) \* partA[i] + factor \* partB[i];  }  }  } **else if** (partA == **null** && partB != **null**) {  result = partB.clone();  } **else if** (partA != **null** && partB == **null**) {  result = partA.clone();  }  **return** result; } |

Listing 2 Implementacja mechanizmu cross-fadingu

### Liniowy

[Wykres przebiegu, opis implementacji]

Liniowy przebieg cross-fadingu można nazwać najprostszym i najbardziej prymitywnym. Można jednak uznać, że jest najodpowiedniejszym wariantem w przypadku użycia przy okienkowaniu sygnału okna Bartletta lub okna trójkątnego. Jest tak dzięki łudząco podobnym przebiegu wartości.

Wyliczenie wartości współczynnika *f* opiera się na następującym wzorze.

*Listing 3* przedstawia sposób zaimplementowania funkcji getCrossFadeFactor dla tej metody.

|  |
| --- |
| **private float** getCrossFadeFactor(**int** n, **int** N) {  **if** (N==0)  **return** 0f;  **return** (**float**) n/N; } |

Listing 3 Implementacja getCrossFadeFactor dla liniowego cross-fadingu

Jeszcze wykresiki

### Sinusowo – cosinusowy

[Wykres przebiegu, opis implementacji]

Sinusowo-cosinusowa metoda cross-fadingu jest autorską odpowiedzią na opisywany w literaturze sinusowy fading. W celu zapewnienia stałości sumy współczynników skorzystano z twierdzenia o jedynce trygonometrycznej. W wyniku tego zamiast funkcji *sinus* i *cosinus* użyto kwadraty tych funkcji.

Wyliczenie wartości współczynnika *f* opera się na następującym wzorze.

*Listing 4* przedstawia implementacjęfunkcji getCrossFadeFactor dla opisywanej metody cross-fadingu.

|  |
| --- |
| **private float** getCrossFadeFactor(**int** n, **int** N) {  **if** (N==0)  **return** 0f;  **return** (**float**) Math.*pow*(  Math.*sin*((**double**) n / N \* Math.***PI*** / 2.0), 2.0  ); } |

Listing 4 Implementacja getCrossFadeFactor dla sinusowo-cosinusowego cross-fadingu

Jeszcze wykresiki

## Opis programu

### Założenia projektowe

#### Założenia funkcjonalne

[nagrywanie dźwięku, odtwarzanie dźwięku, modulowanie dźwięku, konfigurowanie profili, zapis konfiguracji do pamięci trwalej, konfiguracja modulacji, predefiniowane filtry przepustowe]

#### Założenia niefunkcjonalne

[zgodność z API 15, przeznaczona dla smartfonów, nie tabletów, pliki konfiguracyjne zgodne z DOM, optymalizacja pod kątem wykorzystanych zasobów]

### Struktura projektu

[diagram klas, użyte wzorce i dlaczego]

### Biblioteki zewnętrzne

[biblioteka do fft, pochodzenie, wykorzystanie w projekcie]

### Analiza wydajnościowa

[czas modulacji w zależności od ustawień: wybranej modulacji, parametrow modulacji, prarametrów modulowanego sygnału (sample rate, kanały), jak wielkie nagranie jakich parametrów można wykonać (pamięć dynamiczna)]

## Instrukcja obsługi

### Opis interfejsu

[screeny poszczególnych aktywności, działanie przycisków, w ustawieniach znaczenie poszczególnych opcji]

### Pliki konfiguracyjne

[co jest czym w plikach, co się dzieje kiedy ich nie ma, jak powinien wyglądać pusty plik profili]

## Przykłady działania

[czy ten punkt ma sens? Projekt opiera się o modulacje sygnałów, a działanie modulacji pokazane jest już w 4.1]

# Wnioski i podsumowanie