



**AKADEMIA GÓRNICZO-HUTNICZA**

**im. Stanisława Staszica w Krakowie**

**WYDZIAŁ INŻYNIERII  
MECHANICZNEJ I ROBOTYKI**

---

**Praca dyplomowa  
inżynierska**

**Paweł Kluska**

*Imię i nazwisko*

**Inżynieria Akustyczna**

*Kierunek studiów*

**Testy odsłuchowe implementacji HRTF  
nastawionych na wydajność obliczeniową**

*Temat pracy dyplomowej*

**Dr inż. Bartosz Ziółko**  
*Promotor pracy*

.....  
*Ocena*

Kraków, rok 2015/2016

Kraków, dnia.....

Imię i nazwisko : Paweł Kluska  
Nr albumu : 259222  
Kierunek studiów : Inżynieria Akustyczna  
Profil dyplomowania : -

## OŚWIADCZENIE

Upředzony o odpowiedzialności karnej na podstawie art. 115 ust 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (tj. Dz.U.z 2006 r. Nr 90, poz. 631 z późn.zm.) : „Kto przywłaszcza sobie autorstwo albo wprowadza w bład co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystyczne wykonanie albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie”, a także upředzony o odpowiedzialności dyscyplinarnej na podstawie art. 211 ust.1 ustawy z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym (tj. Dz.U. z 2012 r. poz. 572, z późn.zm.) „Za naruszenie przepisów obowiązujących w uczelni oraz za czyny uchybiające godności student ponosi odpowiedzialność dyscyplinarną przed komisją dyscyplinarną albo przed sądem koleżeńskim samorządu studenckiego, zwanym dalej „sądem koleżeńskim”, oświadczam, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i że nie korzystałem (-am) ze źródeł innych niż wymienione w pracy”.

.....  
*podpis dyplomanta*

Kraków, dn.....

Imię i nazwisko: Paweł Kluska  
Nr albumu: 259222  
Kierunek studiów: Inżynieria Akustyczna  
Profil dyplomowania: -

## OŚWIADCZENIE

Świadomy/a odpowiedzialności karnej za poświadczanie nieprawdy oświadczam, że niniejszą inżynierską pracę dyplomową wykonałem/łam osobiście i samodzielnie oraz nie korzystałem/łam ze źródeł innych niż wymienione w pracy.

Jednocześnie oświadczam, że dokumentacja oraz praca nie narusza praw autorskich

w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz. U. z 2006 r. Nr 90 poz. 631 z późniejszymi zmianami) oraz dóbr osobistych chronionych prawem cywilnym. Nie zawiera ona również danych i informacji, które uzyskałem/łam w sposób niedozwolony. Wersja dokumentacji dołączona przeze mnie na nośniku elektronicznym jest w pełni zgodna z wydrukiem przedstawionym do recenzji.

Zaświadczam także, że niniejsza inżynierska praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadawaniem dyplomów wyższej uczelni lub tytułów zawodowych.

.....  
*podpis dyplomanta*

Kraków, .....

Imię i nazwisko: Paweł Kluska

Adres korespondencyjny: ul. Mennicza 42/6, 43-400 Cieszyn

Temat pracy dyplomowej inżynierskiej: Testy odsłuchowe implementacji HRTF  
nastawionych na wydajność obliczeniową

Rok ukończenia: 2016

Nr albumu: 259222

Kierunek studiów: Inżynieria Akustyczna

Profil dyplomowania: -

## OŚWIADCZENIE

Niniejszym oświadczam, że zachowując moje prawa autorskie, udzielam Akademii Górniczo-Hutniczej im. S. Staszica w Krakowie nieograniczonej w czasie nieodpłatnej licencji niewyłącznej do korzystania z przedstawionej dokumentacji inżynierskiej pracy dyplomowej, w zakresie publicznego udostępniania i rozpowszechniania w wersji drukowanej i elektronicznej<sup>1</sup>.

Publikacja ta może nastąpić po ewentualnym zgłoszeniu do ochrony prawnej wynalazków, wzorów użytkowych, wzorów przemysłowych będących wynikiem pracy inżynierskiej<sup>2</sup>.

Kraków, .....  
*data* *podpis dyplomanta*

---

<sup>1</sup> Na podstawie Ustawy z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym (Dz.U. 2005 nr 164 poz. 1365) Art. 239. oraz Ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz.U. z 2000 r. Nr 80, poz. 904, z późn. zm.) Art. 15a. "Uczelni w rozumieniu przepisów o szkolnictwie wyższym przysługuje pierwszeństwo w opublikowaniu pracy dyplomowej studenta. Jeżeli uczelnia nie opublikowała pracy dyplomowej w ciągu 6 miesięcy od jej obrony, student, który ją przygotował, może ją opublikować, chyba że praca dyplomowa jest częścią utworu zbiorowego."

<sup>2</sup> Ustawa z dnia 30 czerwca 2000r. – Prawo własności przemysłowej (Dz.U. z 2003r. Nr 119, poz. 1117 z późniejszymi zmianami) a także rozporządzenie Prezesa Rady Ministrów z dnia 17 września 2001r. w sprawie dokonywania i rozpatrywania zgłoszeń wynalazków i wzorów użytkowych (Dz.U. nr 102 poz. 1119 oraz z 2005r. Nr 109, poz. 910).

Kraków, dnia .....

**AKADEMIA GÓRNICZO-HUTNICZA  
WYDZIAŁ INŻYNIERII MECHANICZNEJ I ROBOTYKI**

**TEMATYKA PRACY DYPLOMOWEJ INŻYNIERSKIEJ**  
dla studenta IV roku studiów stacjonarnych

.....  
*imię i nazwisko studenta*

TEMAT PRACY DYPLOMOWEJ INŻYNIERSKIEJ:

Testy odsłuchowe implementacji HRTF nastawionych na wydajność obliczeniową

*Promotor pracy:* dr inż. Bartosz Ziółko

*Recenzent pracy:* .....

.....  
*Podpis dziekana:*

**PLAN PRACY DYPLOMOWEJ**

1. Omówienie tematu pracy i sposobu realizacji z promotorem.
2. Zebranie i opracowanie literatury dotyczącej tematu pracy.
3. Opracowanie aplikacji do badań.
4. Analiza wyników badań, ich omówienie i zatwierdzenie przez promotora.
5. Opracowanie redakcyjne.

Kraków, .....  
*data* *podpis dyplomanta*

**TERMIN ODDANIA DO DZIEKANATU:** ..... **2016 r.**

.....  
*podpis promotora*

Kierunek: Inżynieria Akustyczna

Profil dyplomowania: -

Paweł Kluska

**Praca dyplomowa inżynierska**

Testy odsłuchowe implementacji HRTF nastawionych na wydajność obliczeniową

Opiekun: dr inż. Bartosz Ziółko

## STRESZCZENIE

Przedmiotem niniejszej pracy jest zbadanie skuteczności metody aproksymacji filtrów HRTF za pomocą kaskady filtrów IIR drugiego rzędu, opracowanej przez zespół DSP na Akademii Górniczo-Hutniczej w Krakowie, za pomocą testów odsłuchowych. W ramach pracy wyjaśnione będą najistotniejsze informacje związane ze słyszeniem przestrzennym człowieka, filtrami HRTF oraz innymi zagadnieniami używanymi w pracy.

W części praktycznej pracy przeprowadzona będzie aproksymacja bazy HRTF za pomocą tej metody, przy wybranych w ramach pracy parametrach. Przedstawione również będą uzasadnienia wyboru parametrów, jak i wyniki aproksymacji. Następnie znajdzie się opis aplikacji testowych stworzonych na potrzeby pracy, a także opis samych testów wraz z wynikami. Na koniec porównane zostaną tradycyjne implementacje z implementacją na filtrach IIR na podstawie ich sprawności obliczeniowej oraz wyników testów.

AGH University of Science and Technology  
**Faculty of Mechanical Engineering and Robotics**

Kraków, the.....

Field of Study: Acoustic engineering

Specialisations: -

Paweł Kluska

**Engineer Diploma Thesis**

Listening tests of HRTF implementation aimed at computing performance

Supervisor: Ph. D. Bartosz Ziółko

## SUMMARY

The purpose of this paper is to evaluate the effectiveness of the approximation method of HRTF filters using a cascade of second-order IIR filters, which was developed by DSP team of AGH University of Science and Technology in Cracow. In the first part of this work, the most essential information related to human spatial hearing, HRTF filters and other terms related to subject will be explained.

In the practical part of this work, an approximation of HRTF will be conducted using parameters, the selection of which will be justified. Next, there will be a description of the applications used for the listening tests, as well as a description of the tests with their results. The final step of this work will be an attempt to compare traditional implementation of the HRTF with the implementation described in this paper based on their performance and the listening tests result.

## Spis treści:

1. Wstęp.....	9
1.1 Cel i zakres pracy.....	9
2. Podstawy teoretyczne.....	10
2.1 Słuch.....	10
2.1.1 W jaki sposób człowiek słyszy przestrzennie?.....	10
2.2 HRTF.....	11
2.2.1 HRIR.....	11
2.2.2 Efekty HRTF-ów.....	11
2.2.3 Typowe implementacje.....	12
2.2.4 Problemy związane z implementacją.....	12
2.3 Filtry.....	13
2.3.1 Filtry IIR.....	13
2.4 Sprawność obliczeniowa.....	14
3. Część praktyczna.....	16
3.1 Wstęp.....	16
3.2 Narzędzia użyte w pracy.....	16
3.2.1 Matlab.....	16
3.2.2 Python.....	17
3.2.3 Unity.....	18
3.2.4 Google Cardboard.....	18
3.2.5 Użyta baza HRTF.....	19
3.3 Porównanie sprawności obliczeniowej implementacji HRTF.....	19
3.4 Aproksymacja HRTF.....	21
3.4.1 Metoda DSP.....	21
3.4.2 Zmiany dokonane w algorytmie.....	21
3.4.3. Wybór parametrów.....	23
3.4.4 Wyniki.....	25



3.4.5 Analiza wyników.....	26
3.5 Test pierwszy.....	26
3.5.1 Cel.....	26
3.5.2 Parametry.....	27
3.5.3 Aplikacja.....	28
3.5.4 Opis testu.....	28
3.5.5 Wyniki.....	29
3.5.6 Analiza wyników.....	31
3.6 Test 2.....	32
3.6.1 Cel.....	32
3.6.2 Parametry.....	32
3.6.3 Aplikacja.....	33
3.6.4 Opis testu.....	33
3.6.5 Wyniki.....	34
3.6.6 Analiza wyników.....	35
4. Podsumowanie.....	36
 Bibliografia.....	 37

# 1. Wstęp

W dzisiejszych czasach technologia generowania przestrzennego dźwięku znajduje wiele zastosowań. Od pomocy osobom z defektami wzroku, do użytków rozrywkowych, takich jak gry czy filmy. Szczególnie w erze rozwoju VR (*virtual reality*) możliwość syntezy w pełni przestrzennego dźwięku w czasie rzeczywistym jest porządana. Realizuje się to za pomocą tak zwanych HRTF-ów (*Head-related transfer function*). Jednym z głównych problemów związanych z nimi jest duża wydajność obliczeniowa ich implementacji.

## 1.1 Cel i zakres pracy

Głównym celem pracy jest przetestowanie metody aproksymującej HRTF-y za pomocą kaskady filtrów IIR, opracowanej przez zespół DSP (zespół przetwarzania sygnałów) na Akademii Górniczo-Hutniczej w Krakowie. Praca ta ma na celu sprawdzenie, jak bardzo przy implementacji tej metody można przyspieszyć obliczenia oraz jak zmieni się przestrzenne postrzeganie dźwięków w porównaniu do tradycyjnych implementacji.

Pierwsza część pracy nakierowana będzie na teorię związaną z tematyką pracy: Po co i w jaki sposób człowiek słyszy przestrzennie, co to są HRTF-y, w jaki sposób się je implementuje i jakie problemy się z tym wiążą. Następnie zarysowany zostanie sposób działania i projektowania niektórych filtrów oraz teoria dotycząca złożoności obliczeniowej algorytmów.

W drugiej części znajdzie się opis narzędzi używanych podczas realizacji pracy, a także złożoność obliczeniowa implementacji metody zespołu DSP w zestawieniu z typowymi implementacjami HRTF-ów. Następnie opisane zostaną poszczególne kroki metody upraszczającej, wybory odpowiednich parametrów i wprowadzone do niej zmiany, mające na celu uzyskanie lepszych rezultatów oraz zwiększenie jego stabilności. Znajdzie się tam uzasadnienie tych wyborów. Omówione zostaną wyniki przeprowadzonych aproksymacji.

Następnie zostaną omówione testy, które powstały w celu poddania próbie badanej implementacji. Opisane będą aplikacje stworzone w celu przeprowadzenia badań. Uzasadniony zostanie wybór pytań oraz próbek dźwiękowych użytych w obu testach, a następnie omówiony zostanie ich przebieg. Na końcu wyniki testów będą poddane analizie.

W ostatniej części znajdzie się synteza uzyskanych wyników. Przedstawione zostaną wnioski oraz zaproponowane dalsze kroki, które mogą zostać podjęte w celu kontynuacji badań. Znajdzie się tu także podsumowanie całej pracy.

## **2. Podstawy teoretyczne**

### **2.1 Słuch**

W porównaniu ze wzrokiem słuch jest znacznie słabiej rozwiniętym zmysłem człowieka [2]. Jednak mimo to zdolność słuchu do wykrycia i lokalizacji źródła dźwięku są bardzo ważnym elementem ludzkiej percepcji. Słyszenie przestrzenne pomaga nam w codziennym życiu, czy to w przechodzeniu przez ulicę, czy znajdowaniu dzwoniącego telefonu.

#### **2.1.1 W jaki sposób człowiek słyszy przestrzennie?**

Człowiek słyszy przestrzennie dzięki analizie amplitudy, fazy oraz widma dźwięku dochodzącego do naszych uszu [2]. Dźwięk padający z danego punktu, na przykład z prawej strony, dotrze do prawego ucha szybciej i z większą amplitudą niż do lewego. Tę własność różnicy w fazie nazywamy ITD (*inter-aural time difference*), natomiast w amplitudzie ILD (*inter-aural level difference*). Jeżeli weźmiemy pod uwagę tylko ITD oraz ILD, dźwięk padający z punktu o tej samej drodze do obu uszu (na przykład z przodu, z tyłu czy bezpośrednio z góry) będzie brzmiał tak samo. Zbiór takich punktów jest często nazywany *cone of confusion*. Same te własności umożliwiają nam rozróżnienie lokalizacji dźwięku, głównie w płaszczyźnie horyzontalnej, ale w niepełnym zakresie, bo wskazują nam tylko, czy znajduje się z lewej czy z prawej strony [5]. Jesteśmy w stanie określić, czy dźwięk pada z przodu/tyłu lub z góry/dołu z dwóch powodów. Pierwszym są ruchy głowy, dzięki którym, w zależności od jej obrotu, ITD i ILD zmieniają się dynamicznie. Jeśli dźwięk dochodzi z góry, a człowiek się obraca, to ITD i ILD zmieniać się będą wolniej, niżby zmieniały się od dźwięku znajdującego się na wysokości uszu. Człowiek nieświadomie wykonuje pewne mikroruchy głową próbując zlokalizować dźwięk. Drugim powodem jest to, że dźwięk padający z danej strony może napotkać przeszkody na drodze do jednego z przewodów słuchowych, przez co jest w pewien sposób filtrowany. Takimi

przeszkodami mogą być głowa, małżowina uszna, a czasem barki czy reszta tułowia. Mózg wykrywa minimalne zmiany w charakterystyce częstotliwościowej dochodzącego sygnału i na ich podstawie jest w stanie zlokalizować źródło dźwięku [5].

## **2.2 HRTF**

Zmiany w widmie dźwięku w zależności od kąta padania (opisane w rozdziale 2.1.2) opisujemy funkcją HRTF (*Head-related transfer function*). Charakteryzuje ona filtracje dźwięku powodowane przez przeszkody, na które natrafia fala podczas drogi do naszego ucha.

Baza HRTF-ów to zbiór filtrów zmierzonych dla obu uszu w określonych punktach. Pomiary są przeprowadzane w polu swobodnym w stałej odległości od głowy dla paru różnych azymutów i elewacji [5]. Według niektórych źródeł człowiek jest w stanie usłyszeć różnice w kącie padania dźwięku równe 5–10° [6], natomiast według innych ta minimalna rozdzielczość jest mniejsza (na przykład 4° [2]), szczególnie w azymucie.

### **2.2.1 HRIR**

HRIR (*Head-related impulse response*) jest to odpowiedź impulsowa naszego układu słuchowego. Jest odwrotną transformacją Fouriera HRTF-u. Elementy bazy są często zapisywane jako minimalnofazowy filtr FIR [5].

### **2.2.2 Efekty HRTFów**

HRTF zawiera wszystkie własności przestrzenne, w tym ITD i ILD [5]. Po połączeniu dźwięku z HRIR-em lub HRTF-em dla danego kąta i ucha uzyskujemy dźwięk, który rejestrowałyby mikrofony w kanale usznym człowieka, gdyby dźwięk w rzeczywistości padał pod tym kątem. W celu prawidłowego odtworzenia takiego dźwięku musimy zapewnić, że sygnał przeznaczony dla danego kanału dochodzi wyłącznie do niego. Z tego powodu należy użyć słuchawek [4]. W przypadku prawidłowego odtworzenia przestrzeni pozorna pozycja źródła dźwięku powinna znajdować się poza naszą głową, w przeciwieństwie do tradycyjnego dźwięku przestrzennego, oraz pod odpowiednim kątem [1]. W rzeczywistości tak zsyntezowany dźwięk nie zawsze zapewnia powyższe własności przestrzenne. Powodem jest to, że ludzka filizjologia wykazuje znaczne

różnice między poszczególnymi osobnikami. HRTF-y zmierzone na różnych osobach mogą w znaczący sposób różnić się od siebie. Dlatego odsłuch dźwięków zsyntezowanych za pomocą HRTF-u niezmiernego na głowie słuchacza często nie stwarza porządanego efektu. Pomiar HRTF-ów przeprowadzone są na sztucznych głowach, głowach innych osób lub uśrednione, przez co nie pasują idealnie do każdego [6].

### 2.2.3 Typowe implementacje

Dźwięk przestrzenny syntezuje się za pomocą HRTF-ów i HRIR-ów dwoma metodami. Pierwsza to operacja splotu sygnału dźwiękowego z HRIR-em. Wzór 2.1 przedstawia operację dyskretnego splotu.

$$(f * g)[n] = \sum_{m=-M}^M f[n-m]g[m] \quad (2.1)$$

Jedną z wad splotu jest to, że potrzebuje on paru próbek (równą długości filtra), żeby się „włączyć”. Jest to problemem, gdy przetwarzamy sygnał ciągły w pewnych ramkach. W takim przypadku trzeba zapamiętywać parę ostatnich próbek (równą długości filtra) poprzedniej ramki i dodawać je na początku aktualnej [7].

Druga metoda to tak zwana operacja szybkiego splotu. Polega na wymnożeniu transformat Fouriera, sygnału oraz HRIR-a (czyli HRTF-a), a następnie wykonaniu odwrotnej transformacji Fouriera. Filtr może od początku być w formie transmitancji. Dzieje się tak, ponieważ splot w dziedzinie czasu jest mnożeniem w dziedzinie częstotliwości. Ta metoda jest w większości przypadków szybsza od zwyczajnego splotu.

### 2.2.4 Problemy

Jednym z głównych problemów implementacji HRTF-ów jest zależność ich poprawnego działania od fizjologii człowieka. Przy niepełnym dopasowaniu filtra do głowy zdrażają się błędy w lokalizacji dźwięku lub brak wrażenia u osoby badanej, że źródło sygnału znajduje się poza głową. Jednym z częstszych błędów w lokalizacji

dźwięku jest pomylenie przodu z tyłem (*front-back confusion*) lub, choć rzadziej, góry z dołem [1]. Istnieją prace naukowe (na przykład [6]), których głównym tematem jest parametryzacja HRTF-ów metryką naszego ciała czy znalezienie pewnych parametrów wspólnych dla wszystkich słuchaczy. Prace te mają na celu umożliwienie pomiaru HRTF-ów w szybszy i łatwiejszy sposób, nie wymagający drogiego sprzętu laboratoryjnego.

Drugim problemem jest złożoność obliczeniowa implementacji HRTF-ów. Problem ten pojawia się w szczególności w aplikacjach, gdzie kąt padania dźwięku zależy od użytkownika (na przykład poprzez śledzenie ruchów głowy). W tych przypadkach operacja splotu musi być wykonywana nawet paręnaście razy na sekundę. Dodatkowym problemem jest rozmiar danych, do których program musi mieć łatwy dostęp i możliwość ich szybkiego odczytania [5].

## 2.3 Filtry

Filtry zazwyczaj stosuje się, by modyfikować widmo dźwięku. Przeważnie charakteryzuje się je przy pomocy odpowiedzi impulsowych  $h(t)$  lub transmitancji  $H(f)$  [8]. W ramach tej pracy będą używane trzy rodzaje filtrów: LSF (*low shelving filter*), HSF (*High shelving filter*) oraz PK (*peakingEQ filter*).

### 2.3.1 Cyfrowe filtry IIR

Filtry IIR to filtry cyfrowe o nieskończonej odpowiedzi impulsowej. Są one rekursywne, czyli odnoszą się do swoich poprzednich wyników. Łatwiej jest uzyskać upragnioną filtrację filtrem IIR, niż filtrem FIR tego samego rzędu, ponieważ transmitancja FIR jest wielomianem, natomiast IIR jest ilorazem wielomianów. Niestety filtry IIR trudniej się projektuje. Z powodu ich rekursywności trzeba zwracać szczególną uwagę na ich stabilność. Aby były one stabilne, bieguny transmitancji muszą mieścić się w okręgu jednostkowym [8].

### Biquad filter

Są to filtry IIR drugiego rzędu. Są używane z tego powodu, iż dużo łatwiej jest wykonać kaskadę filtrów drugiego rzędu, niż stworzyć stabilny filtr IIR wysokiego

rzędu. Wzór 2.2 pokazuje wzór na transmitancję biquada.

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}} \quad (2.2)$$

gdzie:  $b_0, b_1, b_2$  – zera filtru,  $a_0, a_1, a_2$  – bieguny filtru

### Projektowanie filtrów IIR

Filtry IIR najczęściej projektuje się tymi samymi metodami, co filtry analogowe. Można przekształcić filtr analogowy w IIR, co sprawi, że IIR zawsze będzie stabilny [8]. Są dostępne programy oraz tablice, dzięki którym można wyliczyć parametry filtru IIR, znając porządane własności filtru. W przypadku użytych filtrów częstotliwość środkową  $f_0$ , wzmocnienie  $A$  i dobroć filtru  $Q$  w przypadku PK lub nachylenie zbocza filtru  $S$  (dB/oktawe) w filtrach półkowych (LSF, HSF) [3].

### 2.4 Sprawność obliczeniowa

Sprawność obliczeniowa to cecha algorytmu mówiąca o ilości czasu, pamięci czy innego zasobu wykorzystanego do jego przeprowadzenia. Ich sprawność zależy zazwyczaj od rozmiaru danych wejściowych – uzależnia się je więc od tego parametru. Aby wyeliminować różnice w szybkości wykonywania niektórych operacji na różnych komputerach, często używa się asymptotycznego tempa wzrostu do jej oznaczania. W tej pracy będzie używana notacja duże  $O$ . Charakteryzuje ona rząd prędkości, z jaką dana funkcja rośnie, gdy zmienna (w przypadku algorytmów jest to rozmiar danych wejściowych) dąży do nieskończoności. Na przykład:

$$f(x) = 2 \cdot x^4 + x^2$$

$$T(x) = O(x^4)$$

W pracy znajdzie się także próba wyliczenia dokładnej sprawności algorytmu. Założono, że wszystkie operacje, takie jak dodawanie i mnożenie, zajmują tyle samo czasu.

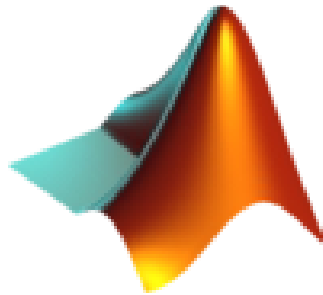
## 3. Część praktyczna

### 3.1 Wstęp

Jeden z głównych celów części praktycznej to sprawdzenie metody uproszczenia HRTF-ów w celu usprawnienia syntezy dźwięków przestrzennych, opracowanej przez zespół DSP. Metoda ta polega na aproksymacji HRTF-ów za pomocą kaskady biquadów. Sprawdzenie tej metody zostało wykonane za pomocą dwóch niezależnych testów. Jeden z nich miał na celu sprawdzenie zależności pomiędzy ilością filtrów użytych w kaskadzie a stopniem rozpoznania lokalizacji dźwięku. Celem drugiego natomiast było sprawdzenie dokładności lokalizacji źródła dźwięku dla paru wybranych kaskad.

### 3.2 Narzędzia używane w pracy

#### 3.2.1 Matlab



Rys. 3.1 Logo Matlab. Źródło: Wikipedia

Matlab jest wysokopoziomowym środowiskiem przeznaczonym dla naukowców i inżynierów. Pozwala w szybki sposób manipulować danymi, wykonywać obliczenia, a następnie wyświetlać ich wyniki. Dzięki wbudowanym funkcjom splotu, FFT oraz innym metodom związanym z przetwarzaniem sygnałów jest idealnym narzędziem do wykonywania operacji na dźwiękach. Dodatkowo dzięki pluginowi SOFA, umożliwiającemu dekodowanie plików w formacie .sofa (Spatially Oriented Format for Acoustics), używanego na świecie do przechowywania baz HRTF, Matlab jest bardzo dobrym narzędziem do ich opracowywania.

#### Global Optimization Toolbox

Jest to zestaw metod do programu Matlab, służący do znajdowania globalnych



rozwiązań problemów posiadających wiele minimów i maksimów lokalnych. Z wielu algorytmów optymalizujących zostały przetestowane trzy: fmincon, pattern search oraz anealing.

### **Użycie Matlab w obrębie pracy**

Program Matlab był głównym narzędziem pracy. Metoda upraszczająca HRTF-y, którą otrzymałem od zespołu DSP, była napisana w postaci skryptów Matlab. Za jego pomocą odbyło się:

- Wybranie parametrów oraz funkcji optymalizującej do aproksymacji filtrów HRTF
- Przeprowadzenie uproszczenia
- Prototypowanie własnych funkcji splotu i filtracji w celu ich późniejszego użycia w testach napisanych w innym języku programowania
- Generowanie plików dźwiękowych do pierwszego testu
- Analiza wyników, statystyka, rysowanie wykresów

### **3.2.2 Python**



Rys. 3.2 Logo Python. Źródło: Wikipedia

Python jest obiektowym językiem programowania wysokiego poziomu. Jego zalety to między innymi: szeroka biblioteka standardowa, jak i wiele innych bibliotek zewnętrznych, wieloplatformowość, prostota oraz czytelność składni.

### **Użycie języka Python w obrębie pracy**

W tej pracy język programowania Python został użyty do wykonania aplikacji do testu pierwszego. Wybrano go z powodu wcześniejszej znajomości tego języka przez autora. Do budowy aplikacji zostały użyte biblioteki TkInter do GUI oraz winsound do odtwarzania dźwięków (niestety przez to aplikacja działa tylko pod systemem Windows).

### 3.2.3 Unity



Rys. 3.3 Logo Unity. Źródło: Wikipedia

Jest to popularne środowisko do tworzenia gier. Pozwala tworzyć aplikacje na wiele platform, w tym na Androida. Dzięki Unity możliwe jest stworzenie aplikacji testowej 3D. W tym środowisku istnieje możliwość łatwego przypisania skryptów do obiektów sceny, co znacznie przyspiesza czas realizacji takich projektów. Są dostępne dwa języki skryptowe C# oraz JavaScript.

#### Użycie Unity w obrębie pracy

Unity zostało użyte do wykonania aplikacji do drugiego testu. Z dostępnych języków skryptowych został wybrany C#. To środowisko zostało wybrane z powodu istnienia oficjalnego demo *virtual reality* na Google Cardboard dla Unity. Umożliwiło szybkie stworzenie aplikacji testowej śledzącej ruch głowy osoby badanej.

### 3.2.4 Google Cardboard



Rys. 3.4 Google Cardboard. Źródło: Wikipedia

Jest to tańsza alternatywa dla powstających urządzeń VR (*virtual reality*). Są to tekturowe gogle, do których wsadzany jest smartfon. Ekran telefonu jest dzielony na pół

przez aplikacje, przez co tworzy dwa obrazy, po jednym dla każdego oka. Obraz wydaje się być trójwymiarowy dzięki efektowi paralaksy. W każdym okularze gogli są zamontowane soczewki, które skupiają nasz wzrok na elementach ekranu. Przy pomocy akcelerometru telefonu monitorowany jest obrót głowy. Oprócz poruszania głową użytkownik jest w stanie komunikować się z aplikacją na telefonie za pomocą przycisku-magnesu. Powstało do tego momentu dużo podobnych, często lepszych urządzeń, jednak to zostało wybrane ze względu na swoją dostępność. Na oficjalnej stronie Google Cardboard dostępne jest demo do środowiska Unity. Aplikacja ta polega na szukaniu sześcianu w przestrzeni. Drugi test był inspirowany właśnie tym demo i tworzony przy jego pomocy.

### **3.2.5 Użyta baza HRTF**

Po przesłuchaniu kilku baz HRTF-ów została wybrana taka, która wydała się najlepiej dopasowana do fizjologii autora tej pracy. Z tego powodu użyta była do testów. Wybrana baza ma 901 odpowiedzi impulsowych na każde ucho. Z rozdzielczością  $6^\circ$  w płaszczyźnie horyzontalnej i  $10^\circ$  w płaszczyźnie wertykalnej, od  $80^\circ$  do  $-60^\circ$ , z punktem zero na wysokości oczu. Dodatkowo znajduje się tam jedna odpowiedź impulsowa dla kąta bezpośrednio nad głową ( $90^\circ$ ). Pojedynczy HRIR składa się z 128 próbek.

### **3.3 Porównanie sprawności obliczeniowej implementacji HRTF-ów**

Zysk sprawności implementacji można obliczyć porównując sprawności obliczeniowe danych algorytmów: Przy liczeniu dokładnej sprawności, w przypadku splotu i filtracji wykorzystane zostały algorytmy napisane przez autora, a w przypadku splotu FFT założono, że HRTF jest przechowywany w formie transmitancji oraz że sprawność FFT wynosi  $3 \cdot N \cdot \log_2(N)$  [8].

Splot zwykły:

$$T(N, n) = O(Nn)$$

$$T(N, n) = 2Nn$$

gdzie:  $N$  – długość pliku dźwiękowego,  $n$  – długość odpowiedzi impulsowej (HRIR)

Splot FFT:

$$T(N) = O(N \log_2(N))$$

$$T(N) = 6N \log_2(N) + 2N$$

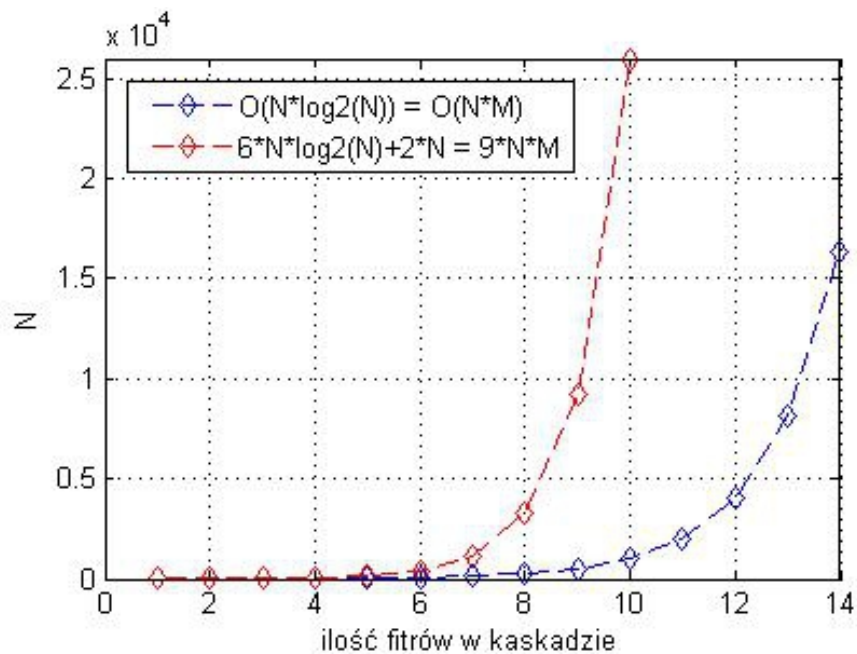
gdzie:  $N$  – długość pliku dźwiękowego

Kaskada filtrów

$$T(N, M) = O(NM)$$

$$T(N, M) = 9NM$$

gdzie:  $N$  – długość pliku dźwiękowego,  $M$  – ilość filtrów w kaskadzie



Rys. 3.5 Wykres przedstawiający równość pomiędzy notacjami duże O oraz dokładnymi sprawnościami splotu FFT i kaskady filtrów w zależności od długości pliku dźwiękowego i ilości filtrów w kaskadzie.

Powyższy wykres (rys. 3.5) przedstawia, dla jakich wartości  $N$  i  $M$  sprawność obliczeniowa dwóch implementacji jest równa. Z wykresu splotu zrezygnowano z powodu niskiej wydajności tego algorytmu w porównaniu do pozostałych. Maksymalne ilości filtrów przedstawione na wykresie są jednocześnie maksymalną ilością filtrów, dla których kaskada wciąż jest szybsza od splotu FFT dla sygnału trwającego 1s (przy  $f_s = 48000$  Hz).

## 3.4 Aproksymacja HRTF

### 3.4.1 Metoda DSP

Wejście. Na początku algorytmu mamy bazę HRIR-ów. Dla użytej bazy filtrów jest to 901 odpowiedzi impulsowych na pojedyncze ucho.

1. Najpierw przechodzimy z odpowiedzi impulsowej (HRIR) na transmitancję za pomocą FFT. Wynikiem jest 901 HRTF-ów dla każdego ucha.
2. Następnie wygładzamy charakterystykę częstotliwościową HRTF-a za pomocą filtra  $1/n$  oktawowego. Parametry do wyznaczenia –  $n$ .
3. Algorytm upraszczający, wykonywany osobno dla każdego HRTF-u, powtarzany  $M$  razy, gdzie  $M$  to ilość użytych filtrów.

3.1 Dzielimy widmo amplitudowe na obszary metodą przejść przez zera.

3.2 Wybieramy jeden obszar najbardziej różny od zera.

3.3 W zależności od iteracji algorytmu najpierw tworzymy filtr HSF, następnie LSF, a dla reszty  $M-2$  iteracji filtry PK. Punkty początkowe zostają wybrane na podstawie wybranego obszaru, gdzie dla LSF jest to zawsze pierwszy obszar, a dla HSF ostatni.

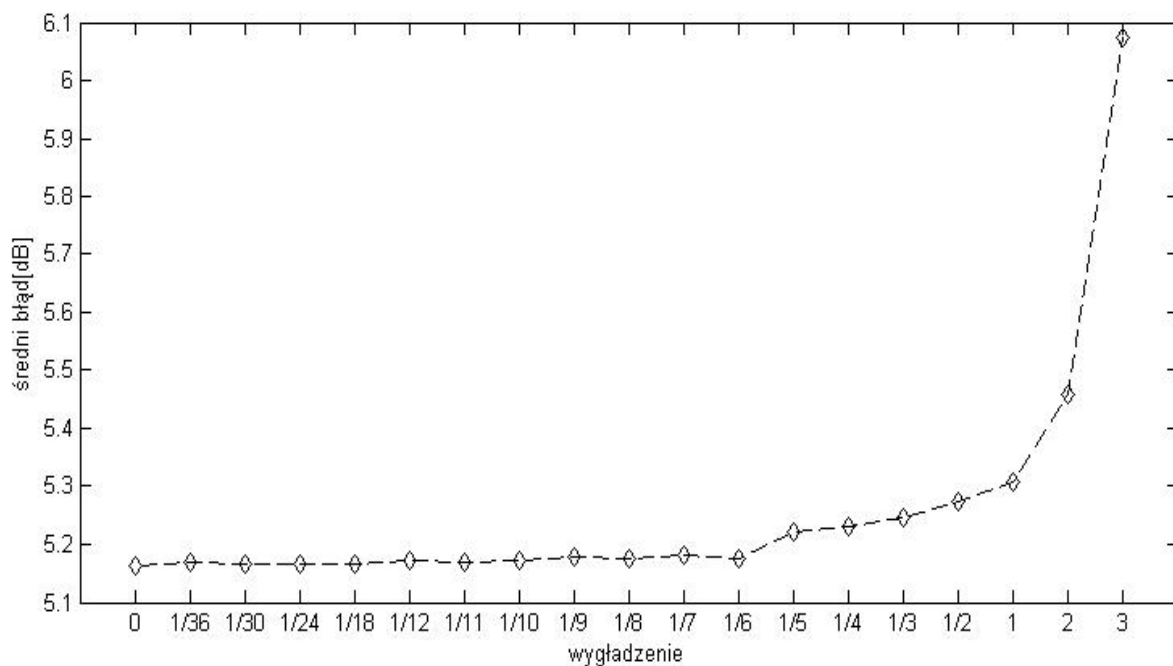
3.4 Rozpoczynamy optymalizację parametrów filtru wybraną metodą. Minimalizujemy 3 parametry filtrów – dobroć, amplitudę i częstotliwość środkową. Minimalizujemy RMS różnicy HRTF-a z widmem amplitudowym filtru IIR.

3.5 Od HRTFa odejmujemy zoptymalizowane widmo filtru, a wynik używamy w kolejnych iteracjach algorytmu upraszczającego.

Wyjście. Wynikiem metody jest 901 parametrów kaskady filtrów dla każdego ucha.

### 3.4.2 Zmiany dokonane w algorytmie

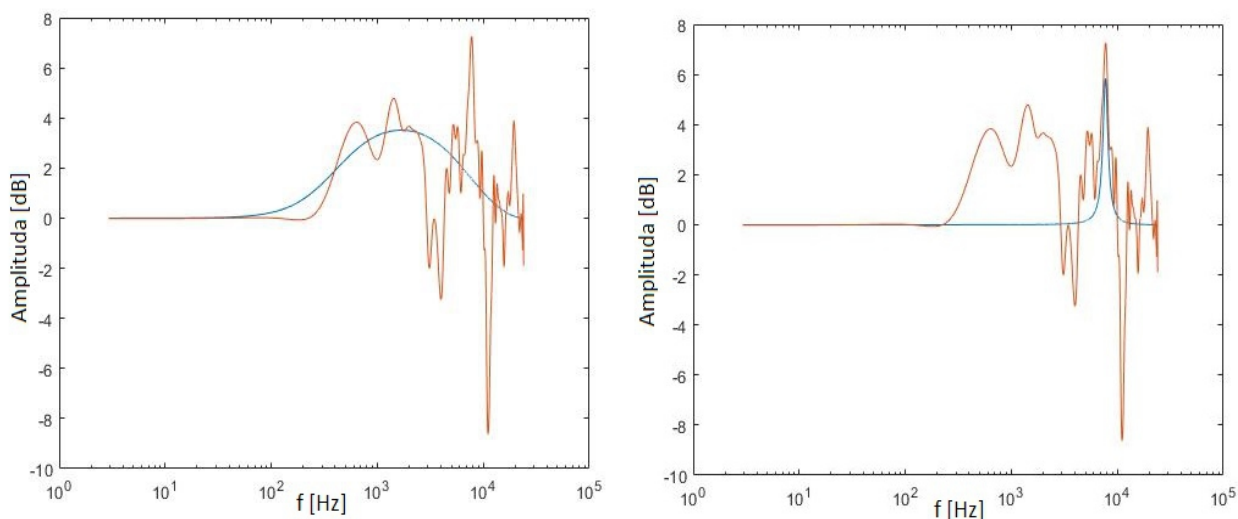
Ad. 2. Z algorytmu został usunięty fragment związany z wygładzaniem. Przetestowano, czy poprawia to wynik końcowy. Jak widać na wykresie (rys. 3.6), wygładzanie nie poprawia dokładności późniejszej aproksymacji, więc zrezygnowano z tej operacji.



Rys. 3.6 Wykres przedstawiający RMS różnicy między niewygladzonym HRTF-em, a kaskadą filtrów wycaloną z wygladzonego HRTF-a filtrem n-oktawowym

Ad. 3.1 Przy dzieleniu przejściami przez zera trzeba było dodać warunki dla HRTF-ów, które nie mają miejsc zerowych.

Ad. 3.2 Została dodana możliwość wyboru paru obszarów, jeżeli ich suma daje większą różnicę, co poprawiło wynik. Efekty przedstawiono na wykresie poniżej (rys. 3.7).



Rys 3.7 Wykresy przedstawiające działanie dla paru obszarów (wykres lewy) oraz jednego obszaru (wykres prawy)

Ad. 3.3 Możliwość wybrania różnych metod optymalizacji dla różnych filtrów.

Ad. 3.3 Parametry wejściowe filtra LSF nie zawsze zależą od pierwszego obszaru. Obszar jest tworzony od pierwszej próbki do momentu, gdy krzywa rośnie szybciej niż 0,1. Wartość została wyznaczona doświadczalnie. Gdy HRTF nie przechodził przez zero, algorytm traktował całość jako jeden obszar. Aproksymacja filtrem LSF w takich przypadkach dawała złe wyniki.

### 3.4.3 Wybór parametrów

#### Metoda optymalizacji

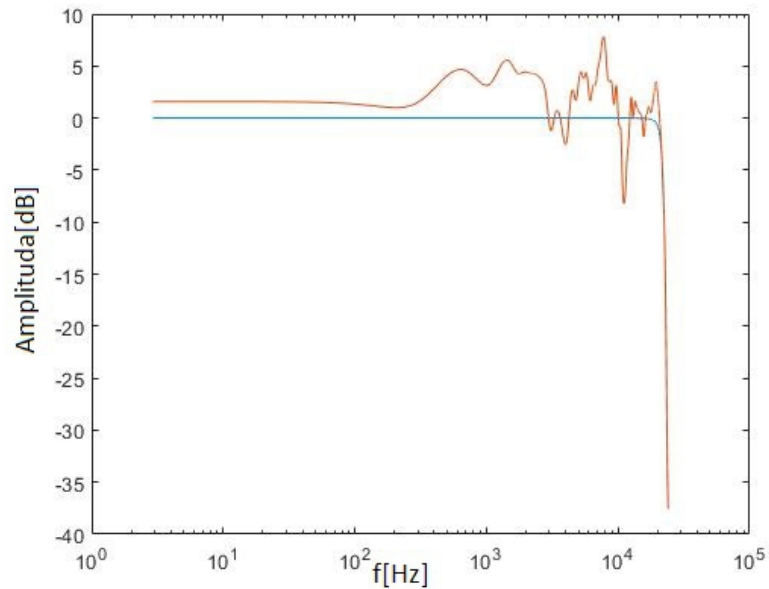
Analizowane były 4 metody z Global Optimization Toolbox: fmincon, pattern search, anealing i anealing z niedomyślnymi opcjami (anealing z domyślnymi parametrami często wykorzystywał maksymalną ilość iteracji). Na podstawie porównania RMS różnicy widm kaskady filtrów z oryginalnym HRTF-em oraz uśrednionego czasu aproksymacji dla paru wykonanych optymalizacji została wybrana jedna z metod. W tabeli (tab. 3.1) przedstawiona została uśredniona wartość z 50 prób każdego algorytmu. Fmincon wraz z pattern search uzyskują najmniejszy błąd, przy czym fmincon jest nieznacznie szybszy niż patternsearch, więc to on został wykorzystany.

Tab. 3.1 Błąd oraz empiryczna sprawność obliczeniowa dla czterech testowanych metod optymalizacji, w zależności od ilości użytych filtrów.

Ilość filtrów w kaskadzie	5		10		15	
	błąd[dB]	t[s]	błąd[dB]	t[s]	błąd[dB]	t[s]
fmincon	1,61	5,65	1,17	12,54	0,65	19,48
pattern search	1,66	18,08	1,16	41,45	0,62	64,44
anealing z opcjami	1,97	277,06	1,31	700,56	1,15	1114,58
anealing	2,10	43,75	1,62	145,04	1,45	160,29

## HSF

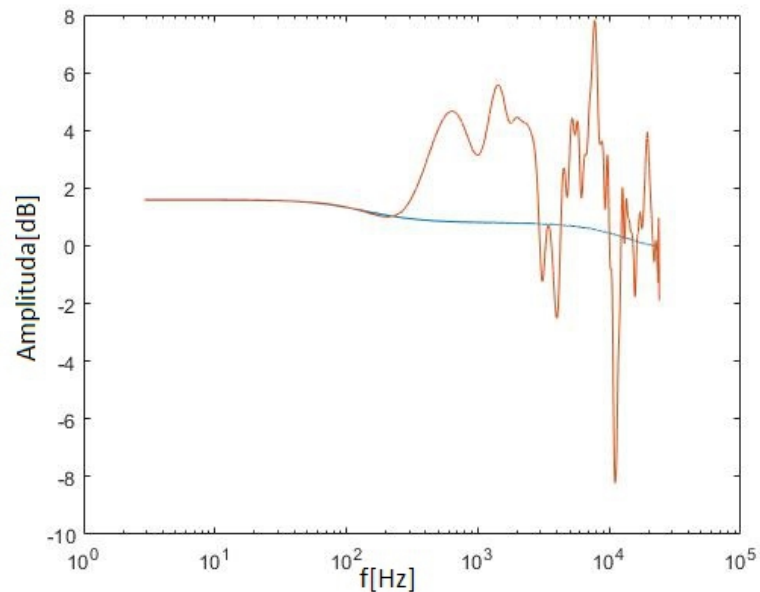
Celem filtru było zaproksymowanie pewnego „ogona” występującego na końcu każdego HRTF-a. Parametry graniczne były tak dobrane, aby minimalna częstotliwość środkowa filtra miała wartości powyżej 1900 Hz. Zdarzało się, że przyjmował dziwne wartości, gdy starał się zaproksymować cały obszar (kiedy HRTF nie przecinał 0).



Rys. 3.8 Funkcja filtru HSF. Czerwony – HRTF, niebieski – charakterystyka częstotliwościowa filtra.

## LFS

Głównym celem tego filtru było usunięcie pewnej „składowej stałej” (ILD) HRTF-a.

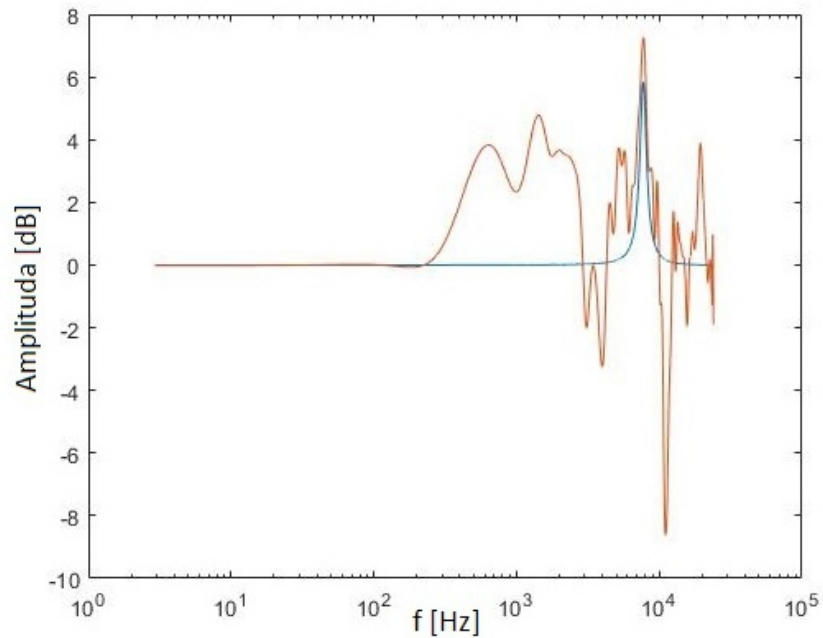


Rys. 3.9 Funkcja filtru LSF. Czerwony – HRTF, niebieski – charakterystyka częstotliwościowa filtra.



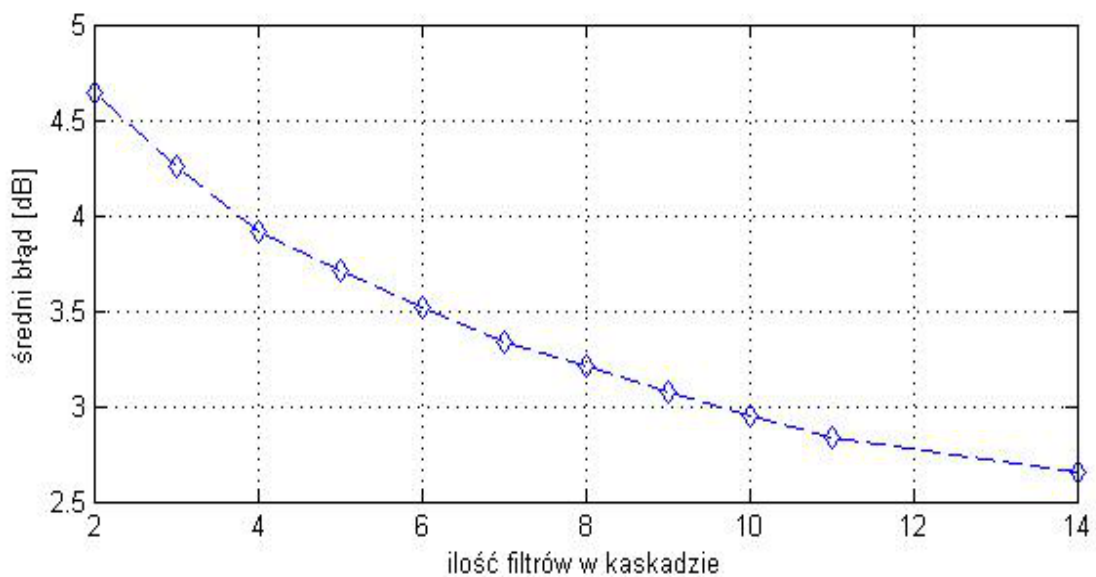
## PK

Parametry graniczne są tak dobrane, żeby nie wpływały na aproksymację, czyli jako maksymalne/minimalne możliwe wartości parametrów.

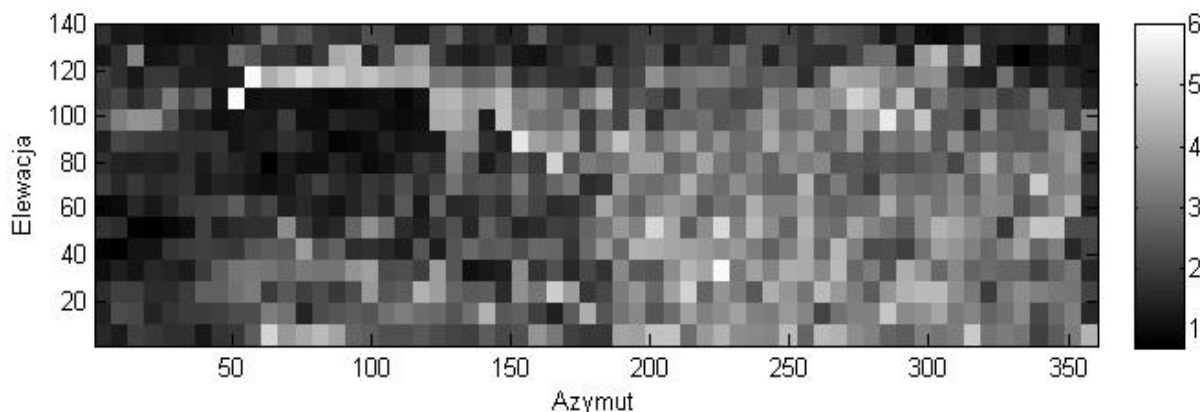


Rys. 3.10 Funkcja filtru PK. Czerwony – HRTF po odjęciu charakterystyk HSF oraz LSF, niebieski – charakterystyka częstotliwościowa filtru.

### 3.4.4 Wyniki



Rys. 3.11 RMS różnicy HRTF-u i odpowiedzi częstotliwościowej kaskady filtrów, dla różnych ilości filtrów w kaskadzie



Rys. 3.12 RMS różnicy HRTF-u i odpowiedzi częstotliwościowej kaskady filtrów zależnie od kąta padania dźwięku, dla lewego ucha. Punkt (0,60) oznacza oś wzroku, a (90,60) to punkt przesunięty w 90° w lewo w od osi wzroku.

### 3.4.5 Analiza wyników

Na podstawie wykresu błędu w zależności od ilości filtrów użytych w kaskadzie (rys. 3.11) można stwierdzić, że błąd aproksymacji wyraźnie spada dla kilku pierwszych filtrów – następnie, z każdym kolejnym filtrem, spadek ten jest coraz łagodniejszy. Natomiast na podstawie wykresu błędu w zależności od kąta zaobserwowano (rys. 3.12), że najlepsze aproksymacje mają te kąty, pod którymi dźwięk wpada bezpośrednio do ucha. Im bardziej zasłonięte jest lewe ucho, tym gorszy jest wynik. Powodem tego może być to, że krzywe tych HRTF-ów są trudniejsze do uproszczenia – mają wiele „górek” w swojej charakterystyce. Jest to prawdopodobnie spowodowane tym, że fala kierująca się do naszego ucha trafia na dużo więcej przeszkód i dociera do niego bardziej zniekształcona. Założono, że nie powinno to mieć większego znaczenia, ponieważ dźwięk docierający do tego ucha ma mniejszy poziom głośności.

## 3.5 Test pierwszy

### 3.5.1 Cel

Test pierwszy miał na celu sprawdzenie, czy i w jaki sposób zmienia się rozpoznawanie przybliżonej lokalizacji źródła w zależności od dokładności uproszczenia, oraz w jaki sposób odnosi się to do tradycyjnej implementacji. Z wyników będzie można wywnioskować, czy rozpoznanie dla kaskady będzie równorzędne z użyciem splotu. Jeżeli tak, to przy jakiej ilości filtrów. Te parametry będzie można zmierzyć dla

rozpoznania ogólnego i względem płaszczyzn horyzontalnej i wertykalnej.

### **3.5.2 Parametry**

#### **Sygnal**

W teście użyto dwóch sygnałów. Pierwszym z nich był szum biały o długości 250ms, przemnożony oknem hann (podobne sygnały były użyte w wielu testach [1, 6]). Drugim sygnałem był sygnał impulsowy – trzask. Sygnał ma węższe widmo, przez co prawdopodobnie jego rozpoznanie będzie gorsze, jednak został zastosowany, żeby zwiększyć różnorodność badanych dźwięków.

#### **Użyte metody**

Dźwięk syntezowny był na sześć różnych sposobów – splot HRTF-em oraz filtracja kaskadami, kolejno 2, 5, 8, 11, 14 biquadów. Każda z metod użyta została tyle samo razy, z tymi samymi pytaniami. Ilości filtrów zostały wybrane na podstawie sprawności obliczeniowej wyliczonej w poroździale 3.3 (Rys. 3.5). Rozważając zastosowanie syntezy dźwięku przestrzennego w aplikacjach zmieniających kierunek dźwięku w czasie rzeczywistym, powinniśmy wyliczać nowy dźwięk parę razy na sekundę. Do dalszych rozważań przyjmować będziemy liczbę 2048 próbek (rozmiar bloków dźwięków udostępnianych przez metodę OnAudioFilter() w Unity). Dla  $N = 2048$  splot FFT będzie szybszy od ośmio-, ale wolniejszy od siedmioelementowej kaskady. Na tej podstawie uproszczenie składające się z ośmiu filtrów zostało wybrane jako środek interesującego nas przedziału filtrów. Kaskada dwuelementowa została wybrana, ponieważ jest najmniejszą możliwą do uzyskania ilością biquadów dzięki metodzie upraszczającej. Co więcej, dla takiego uproszczenia rozpoznanie powinno być znikome, co pozwoli nam na ustalenie pewnej dolnej granicy rozpoznania. Reszta kaskad została wybrana, by wypełnić lukę między dwoma i ośmioma filrami, oraz żeby symetrycznie dodać dwie metody ponad ośmioma filrami.

#### **Pytania**

Pytań było w sumie 120, po 20 na każdy rodzaj filtracji. Rodzajów pytań było 5, a do każdego rodzaju po 4 pytania, z czego 2 pytania przypadały na każdy typ sygnału.

Pytania brzmiały następująco:

1. Gdzie znajduje się drugi dźwięk odnośnie do pierwszego? (przód/tył)
2. Gdzie znajduje się drugi dźwięk odnośnie do pierwszego? (góra/dół)
3. Gdzie znajduje się dźwięk odnośnie do dźwięków referencyjnych? (przód/tył)
4. Gdzie znajduje się dźwięk odnośnie do dźwięków referencyjnych? (góra/dół)
5. W którą stronę porusza się źródło dźwięku? (góra/dół)

W każdym pytaniu można było zaznaczyć jedną z trzech odpowiedzi – te zawarte w nawiasach powyżej oraz opcję „dźwięki znajdują się na tej samej wysokości / są tak samo wysunięte”. Do każdego pytania wygenerowana była w programie Matlab próbka dźwiękowa. W przypadku pytania 1 i 2 sygnałem testowym były dwa dźwięki rozmieszczone na przeciwległych półkulach. W przypadku pytania numer 3 i 4 były to trzy dźwięki, z czego pierwszy i ostatni były monofoniczne, a środkowy był uprzestrzenniony. W pytaniu 5 znalazły się cztery dźwięki w jednej osi wertykalnej, każdy przesunięty o 30° odpowiednio w górę lub w dół. Pytania były wstępnie testowane na małej ilości osób (2–3), a te, które miały bardzo niskie (<20%) lub bardzo wysokie (>80%) rozpoznanie, były odrzucane.

### **3.5.3 Aplikacja**

Test został przeprowadzony na aplikacji wykonanej w języku Python. Kolejność pytań była losowana przy każdym rozpoczęciu testu. Osoba badana nie miała limitu czasu, by odpowiedzieć na pytania, oraz mogła odsłuchiwać każdą próbkę dźwiękową dowolną ilość razy. GUI zostało zaprojektowane przy pomocy biblioteki TkInter w taki sposób, aby osoba badana mogła intuicyjnie, bez instrukcji, wykonać test. Po wykonaniu wszystkich zadań program zapisywał odpowiedzi do pliku .txt oraz wyświetlał procentowe rozpoznanie dla każdej metody.

### **3.5.4 Opis testu**

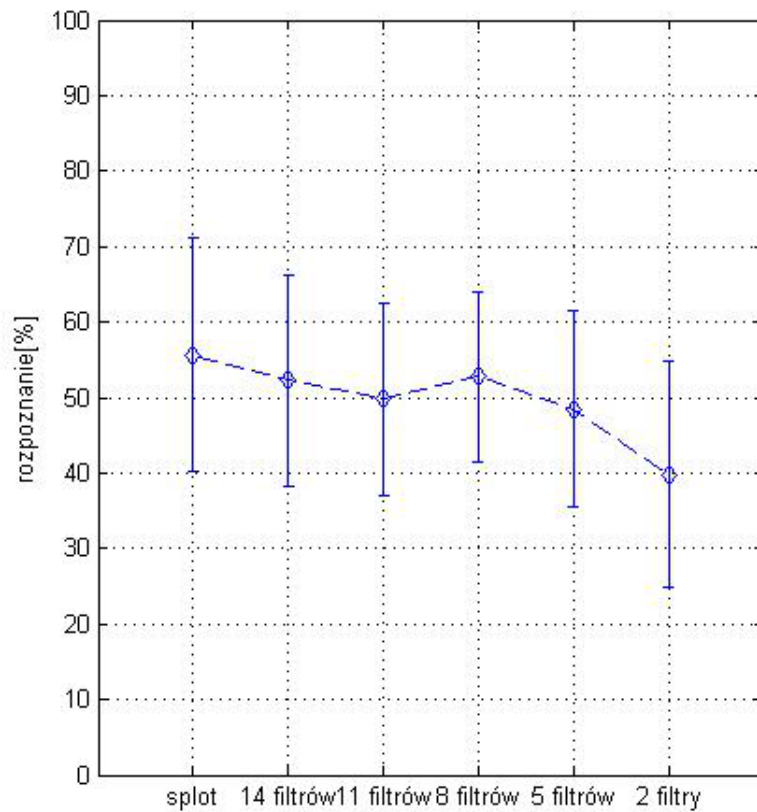
Przetestowane zostały 22 osoby (10 mężczyzn, 12 kobiet). Średnia długość testu to 20 minut. Większość testów (72%) była przeprowadzona w cichym pomieszczeniu pod nadzorem autora, na tych samych słuchawkach klasy średniej. Przed przystąpieniem do testu osobom badanym wytłumaczone zostało działanie aplikacji, jednak nie wiedziały one, co dokładnie ma zostać sprawdzone. Zaznaczono, że podczas testu należy siedzieć

prosto, z głową w miarę możliwości wyprostowaną i nieruchomą. Sześć osób przetestowanych poprzez udostępnienie aplikacji testowej w internecie dostało instrukcję w formie pisemnej. Użycie przez nich innych słuchawek nie miało widocznego wpływu na wyniki.

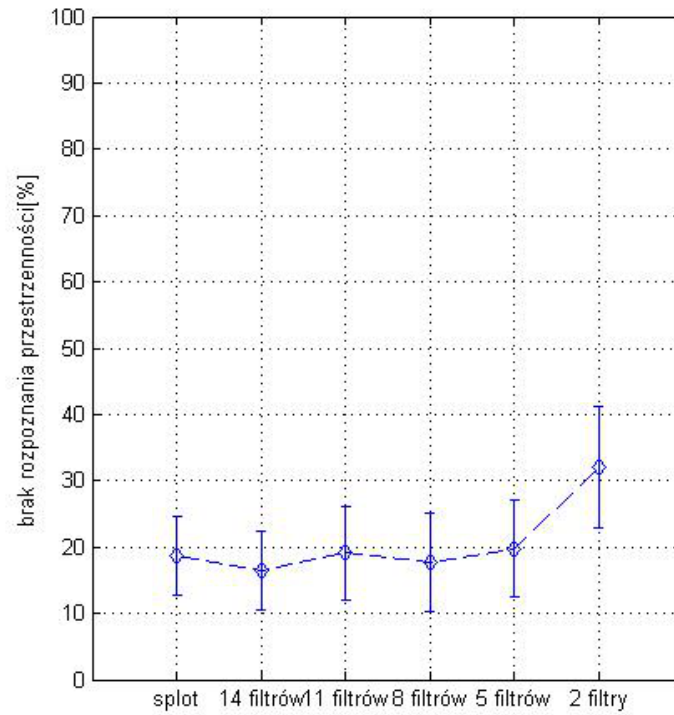
### 3.5.5 Wyniki

Tab. 3.2 Zestawienie rozpoznania w zależności od typu sygnału/pytania

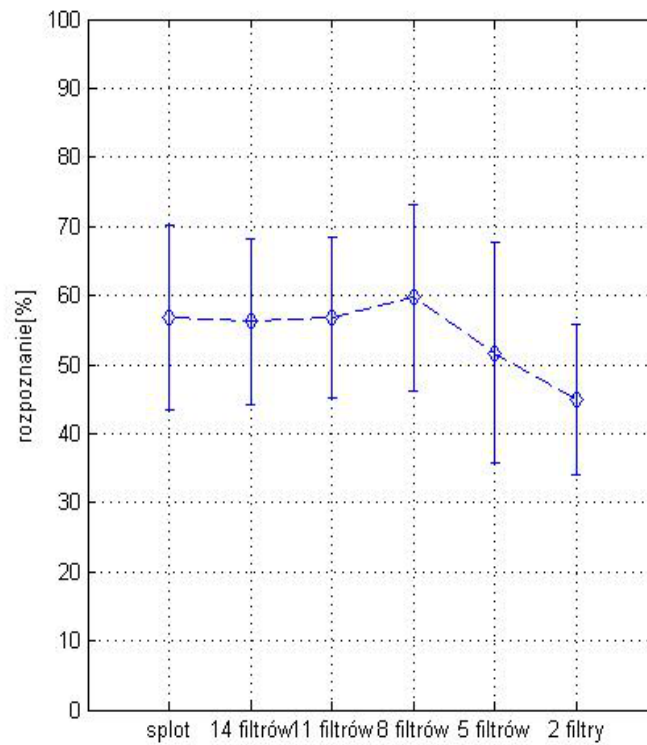
	szum	impuls	przód/tył	górną/dół
Rozpoznanie[%]	52	47	46	54



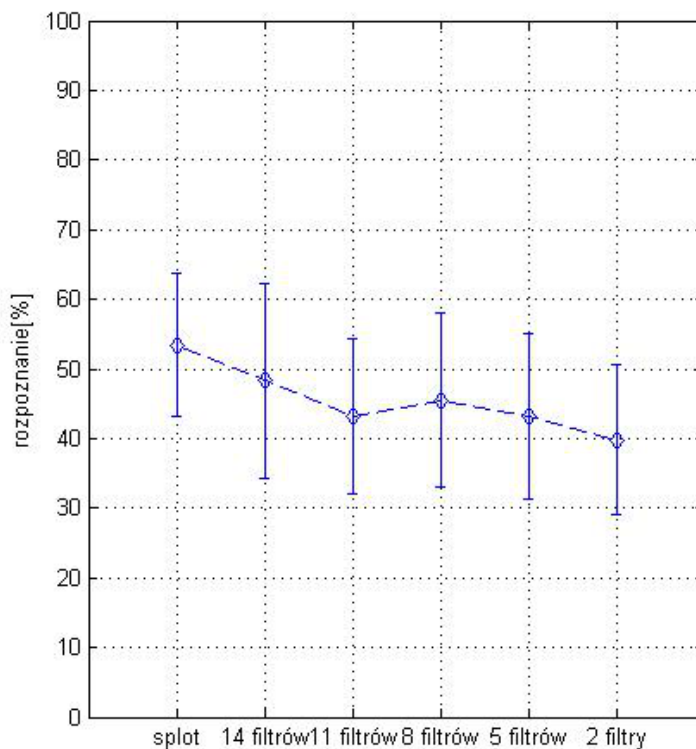
Rys. 3.13 Wykres przedstawiający poprawność odpowiedzi w zależności od sposobu syntezy dźwięku.



Rys. 3.14 Wykres przedstawiający ilość odpowiedzi świadczących o braku rozpoznania przestrzennego w zależności od sposobu syntezy dźwięku.



Rys. 3.15 Wykres przedstawiający poprawność odpowiedzi w zależności od sposobu syntezy dźwięku dla pytań typu góra/dół.



Rys. 3.16 Wykres przedstawiający poprawność odpowiedzi w zależności od sposobu syntezy dźwięku dla pytań typu przód/tył.

### 3.5.6 Analiza wyników

Rozpoznanie całkowite w zależności od ilości użytych filtrów (rys. 3.13) pokazuje, że nie udało się dokładnie odwzorować przestrzenności HRTF-u żadną z kaskad. Zaskakującym wynikiem jest większe rozpoznanie dla 8 filtrów, niż dla 11 oraz 14. Na rys. 3.14 widać, że ilość pytań, w której badane osoby nie rozpoznały przestrzenności, spada wraz z ilością użytych filtrów. Analizując wykresy na rys. 3.15 i rys. 3.16 można stwierdzić, że rozpoznanie dla płaszczyzny wertykalnej (góra/dół) jest osiągnięte dla 8 filtrów, natomiast rozpoznanie przód/tył nie jest osiągnięte dla żadnej z wybranych kaskad.

## **3.6 Test 2**

### **3.6.1 Cel**

Celem testu drugiego było sprawdzenie błędu w rozpoznawaniu dokładnej lokalizacji dźwięku dla mniejszej ilości metod. Zmierzona została ilość błędów typu *front-back confusion* i *up-down confusion*. Dodatkowo, ten test jest próbą implementacji metod w rzeczywistych warunkach, które mogą pojawić się przy okazji użycia tej technologii (np. na smartfonie).

### **3.6.2 Parametry**

#### **Sygnal**

Sygnalem testowym, podobnie jak w poprzednim teście, był szum biały o długości 250ms. Wymnożono go oknem hann, analogicznie jak w teście nr 1. Podobny sygnał dźwiękowy użyty był we wspomnianych wyżej pracach naukowych [1,6].

#### **Metody**

W tym teście dźwięk syntezowany był na trzy sposoby, splotem oraz filtracjami ośmio- i czternastoelementową kaskadą. Ośmioelementowa kaskada filtrów została wybrana, ponieważ jej rozpoznanie było równe rozpoznaniu tradycyjnej metody w płaszczyźnie wertykalnej. Czternastoelementowa kaskada filtrów została użyta, ponieważ w przypadku żadnej ilości filtrów nie dało się uzyskać rozpoznania identycznego jak w przypadku splotu w płaszczyźnie horyzontalnej.

#### **Pytania**

Przy uruchamianiu nowego testu generowanych jest 12 pozycji na kuli we współrzędnych sferycznych. Oznaczają one punkty, z których symulowany będzie dźwięk. Każda z tych 12 pozycji została odtworzona dla każdej metody, co dało w sumie 36 pytań. Pozycje są generowane w taki sposób, aby pojawiały się po 4 z przodu, z boków i z tyłu osoby badanej. Pytania te są losowo wymieszane.



### 3.6.3 Aplikacja

Aplikacja została wykonana w środowisku Unity na system operacyjny Android, w oparciu o demo Google Cardboard dla Unity. Wykorzystuje ona skrypty do podziału oraz zniekształcenia obrazu i skrypt do obrotu kamerą w taki sposób, jak akcelerometrem telefonu. W samym teście znajdują się dwa główne obiekty – sześcian będący źródłem dźwięku i przycisk pozwalający przejść do następnego pytania. Sam test jest dwuetapową maszyną stanów. W pierwszym stanie przycisk i źródło są widoczne, a osoba badana może sprawdzić, z którego punktu dochodzi dźwięk. Do drugiego stanu przechodzi się naciskając przycisk. Zaraz po zmianie pierwszego stanu na drugi, przycisk wraz ze źródłem stają się niewidoczne, a źródło przemieszcza się w nową pozycję (biorąc pod uwagę pozycję osoby badanej), po czym odtwarzany jest sygnał z tej pozycji. Maszyna wraca do stanu pierwszego, kiedy dowolny punkt w przestrzeni zostanie naciśnięty. To miejsce jest rejestrowane jako odpowiedź osoby badanej. Podczas programowania aplikacji pojawił się problem z implementacją splotu – telefon niekiedy nie zdążał przeprowadzić operacji splotu, przez co dźwięk słyszalnie się zniekształcał. Przy użyciu szumu białego nie było to zauważalne.

### 3.6.4 Opis Testu

Przetestowane zostało 11 osób (5 mężczyzn, 7 kobiet). Test przeprowadzany był w towarzystwie autora pracy na tym samym telefonie i słuchawkach klasy średniej. Przed testem osobie badanej zostało wytłumaczone działanie aplikacji, ze szczególnym zaznaczeniem, że źródło dźwięku może dochodzić z góry bądź dołu. Zostały również pokazane maksymalne kąty, z których ten dźwięk może dochodzić. Osoba testowana przed przystąpieniem do testu miała możliwość odpowiedzenia na dowolną ilość pytań nierejestrowanych, traktowanych jako trening. Po przystąpieniu do testu jego przebieg wyglądał następująco:

- Testowany naciska wskaźnikiem przycisk „nowe pytanie”
- Źródło dźwięku przesuwa się na nowe miejsce
- Badany wskazuje za pomocą gogli miejsce, z którego według niego dochodzi dźwięk i naciska wskaźnikiem
- Pokazuje się prawidłowa lokalizacja źródła oraz przycisk „nowe pytanie”
- Po wykonaniu wszystkich 36 zadań odpowiedzi zapisywane są w pliku .txt

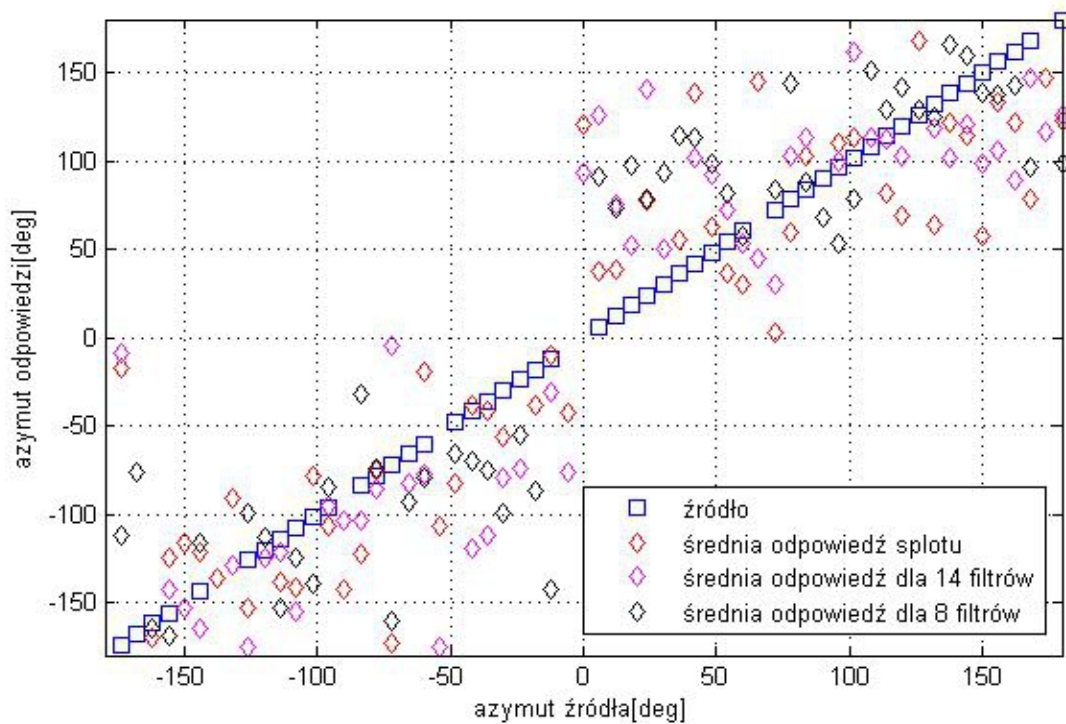
### 3.6.5 Wyniki

Tab. 3.3 Zestawienie poszczególnych błędów przed i po obróceniu *front-back* i *top-down confusion*.

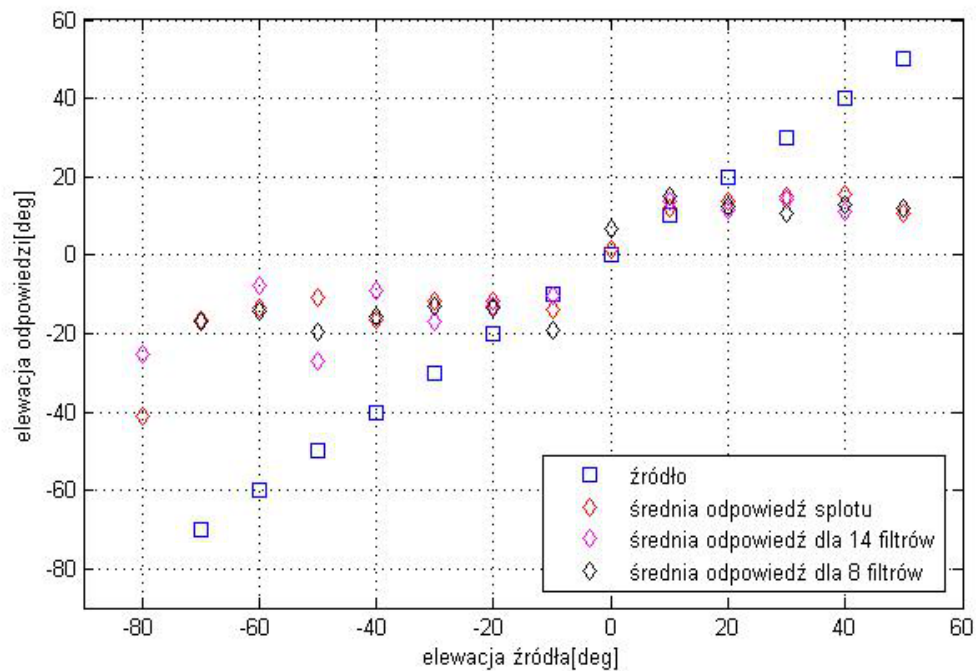
	błąd przed zmianą <i>confusion</i> [deg]			błąd po zmianie <i>confusion</i> [deg]		
	horyzontalny	wertykalny	ogólny	horyzontalny	wertykalny	ogólny
splot	58	33	71	44	22	53
14 filtrów	61	28	71	47	19	54
8 filtrów	64	34	79	48	22	56

Tab. 3.4 Ilość *front-back* i *top-down confusion* dla poszczególnych metod filtracji.

	<i>confusion</i>	
	<i>front-back</i>	<i>top-down</i>
splot	32	66
14 filtrów	45	52
8 filtrów	47	65



Rys. 3.17 Wykres porównujący pozycje źródła do średniej odpowiedzi testowanych osób dla każdej pozycji tego źródła w płaszczyźnie horyzontalnej.



Rys. 3.18 Wykres porównujący pozycje źródła do średniej odpowiedzi testowanych osób dla każdej pozycji tego źródła w płaszczyźnie wertykalnej.

### 3.6.6 Analiza wyników

Opracowanie wyników było inspirowane pracami badawczymi zajmującymi się problematyką filtrów HRTF [1]. Istnieją dwie metody radzenia sobie ze zjawiskiem *front-back confusion*. Jedną z nich jest ignorowanie ich w statystyce. Druga polega na odwróceniu ich na drugą półkulę i włączeniu ich do wyników. W tej pracy została wybrana druga opcja. *Confusion* było wykrywane przez sprawdzenie, czy przy zmianie półkuli dana odpowiedź znajdzie się bliżej lokalizacji poprawnej odpowiedzi. Jeśli ten warunek został spełniony, zmieniana była pozycja odpowiedzi i odnotowywana była jako *confusion*. Jest to naiwny sposób wykrywania tych błędów i prawdopodobnie zwiększy on ich ilość, ale nie powinno mieć to negatywnego wpływu na wynik. Podobnie jak w pracy, która była inspiracją dla tej analizy [1], zainteresowaniu podlega głównie porównanie / dyskryminacja metod implementacji. Wykresy (rys. 3.18, rys. 3.17) pokazują wartość średnią odpowiedzi dla zkwantyzowanych pozycji źródła. W przypadku elewacji większość osób badanych rzadko kiedy słyszała źródło w odchyle 20° – zarówno dla metod filtracji kaskadą, jak i dla splotu. Z tabel (tab. 3.3,

tab. 3.4) wynika, że splot posiada lepszą lokalizację w azymucie niż obydwie filtracje kaskadami. Za to lokalizacja w elewacji kaskady ośmioelementowej dorównuje splotowi w dokładności lokalizacji, natomiast czternastoelementowa kaskada daje nawet lepsze wyniki.

#### **4. Podsumowanie**

Celem niniejszej pracy było sprawdzenie, w jakim stopniu metoda upraszczająca filtry HRTF przy pomocy kaskady filtrów IIR, w zależności od różnych parametrów, zachowuje własności przestrzenne tradycyjnych implementacji HRTF-ów.

Pierwszym etapem pracy było zapoznanie się z metodą upraszczającą, a następnie określenie, w jaki sposób niektóre parametry wpływają na błąd aproksymacji. Po minimalizacji tego błędu kolejnym krokiem był wybór, z ilu elementów powinna się składać kaskada filtrów. / była decyzja o ilości elementów, z których powinna się składać kaskada filtrów. Po wyprowadzeniu złożoności obliczeniowej implementacji filtracji kaskadą filtrów i porównaniu ze splotem FFT, za najbardziej znaczący obszar uznany został przedział od 2 do 14 filtrów.

Następnym etapem było przeprowadzenie testów odsłuchowych na grupie statystycznej. Celem pierwszego testu było ustalenie, jak sprawdzą się kaskady filtrów w porównaniu z normalną implementacją HRTF przy prostych testach lokalizacyjnych. Z testów wynikało, że żadna z kaskad w pełni nie odwzorowuje możliwości HRTF-ów. Na podstawie badań wysnuto hipotezę, że dla kaskady składającej się z 8 filtrów rozpoznanie w elewacji źródła dźwiękowego jest równe rozpoznaniu HRTF-ów.

By potwierdzić to założenie, został opracowany kolejny test, celem którego było sprawdzenie dokładności odwzorowania przestrzeni przez HRTF oraz już tylko dwie kaskady filtrów – ośmioelementową oraz czternastoelementową. Drugi test potwierdza wcześniejszą hipotezę, że ośmioelementowa kaskada posiada porównywalną dokładność lokalizacji źródła w elewacji w stosunku do HRTF-u. Nie udało się natomiast w pełni odwzorować dokładności lokalizacji w azymucie.

Niskie rozpoznanie lokalizacji dźwięku przy nieuproszczonej implementacji HRTF-u utrudniło analizę wyników. Aby dokładniej przetestować ową metodę, należałoby przeprowadzić testy na osobie mającej zmierzony własny HRTF.

## Bibliografia

- [1] Arruda M., Wenzel E.M., Wightman F. L.: *Localization using nonindividualized head-related transfer functions*, 1993. Dostępny: [http://humanfactors.arc.nasa.gov/publications/wenzel\\_1993\\_Localization\\_Head\\_Related.pdf](http://humanfactors.arc.nasa.gov/publications/wenzel_1993_Localization_Head_Related.pdf) (odwiedzona 20.01.2016).
- [2] Blauert J.: *Spatial Hearing: The Psychophysics of Human Sound Localization*, The MIT Press, 1997, s. 1–45.
- [3] Bristow-Johnson R.: *Cookbook formulae for audio EQ biquad filter coefficients*. Dostępny: <http://www.musicdsp.org/files/Audio-EQ-Cookbook.txt> (odwiedzona 20.01.2016).
- [4] Carty B., Lazzarini V.: *Binaural HRTF based spatialisation: New approaches and implementation*. National University of Ireland, 2009. Dostępny: [http://eprints.maynoothuniversity.ie/2334/1/vl\\_binaural\\_paper\\_15.pdf](http://eprints.maynoothuniversity.ie/2334/1/vl_binaural_paper_15.pdf) (odwiedzona 20.01.2016).
- [5] Cheng C. I., Wakefield G.H.: *Introduction to head-related transfer functions (HRTF's): Representations of hrtf's in time, frequency, and space*. Dostępny: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.442.5500&rep=rep1&type=pdf> (odwiedzona 20.01.2016).
- [6] Romigh G. D.: *Individualized Head-Related Transfer Functions: Efficient Modeling and Estimation from Small Sets of Spatial Samples*, Pittsburgh 2012. Dostępny: <http://www.cs.cmu.edu/~robust/Thesis/RomighThesis.pdf> (odwiedzona 20.01.2016).
- [7] Wikipedia: *Overlap-save method*. Dostępny: [https://en.wikipedia.org/wiki/Overlap-save\\_method](https://en.wikipedia.org/wiki/Overlap-save_method) (odwiedzona 20.01.2016).
- [8] Ziółko B., Ziółko M.: *Przetwarzanie mowy*, Kraków 2011, s. 117–196.