



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

WYDZIAŁ INFORMATYKI, ELEKTRONIKI I TELEKOMUNIKACJI

INSTYTUT ELEKTRONIKI

Praca dyplomowa
inżynierska

Automatyczna kontrola głośności przy jednoczesnym tłumieniu tła akustycznego dla sygnału mowy

Imię i nazwisko: Piotr RADECKI
Kierunek studiów: ELEKTRONIKA I TELEKOMUNIKACJA
Typ studiów: STACJONARNE
Opiekun pracy: dr hab. inż. Konrad KOWALCZYK, prof. AGH

Kraków 2022

Streszczenie

W pracy inżynierskiej zaimplementowany został algorytm automatycznej kontroli głośności dla sygnałów mowy nagranych macierzą mikrofonową. Do zrealizowania użyto zaawansowanego algorytmu przetwarzania sygnałów jakim jest bazujący na metodzie kształtowania wiązki filtr LCMV, umożliwiający wyrównanie głośności mówców znajdujących się w różnej odległości od macierzy lub mówiących z różną głośnością. Otrzymany algorytm estymuje kierunki nadchodzenia dźwięku za pomocą algorytmu MUSIC i reguluje głośność pojedynczych mówców przy jednoczesnym tłumieniu zakłócenia jakim jest biały szum. Zaimplementowane rozwiązanie bazuje na nagraniach z macierzy mikrofonowej. W ramach pracy stworzono również specjalny generator symulujący geometrię macierzy mikrofonowej i pomieszczeń, w których rozchodzi się dźwięk. W celu zrealizowania pracy zapoznano się z fachową literaturą naukową i napisano oprogramowanie w języku Python. Wykonana implementacja została poddana licznym testom. Sprawdzono wpływ czynników takich jak ilość źródeł i ich położenia, poziom szumu a także geometria pokoju i macierzy mikrofonowej na jakość działania algorytmu. Uzyskane wyniki przeanalizowano i opisano.

Praca ma następującą kompozycję: w rozdziale 2 autor przedstawia teorię zagadnienia. W rozdziale 3 przedstawione zostało rozwiązanie wraz z krótkim opisem użytych metod. W kolejnych rozdziałach znajdują się szczegóły implementacyjne (rozdział 4) i testy zaimplementowanego rozwiązania (rozdział 5).

Abstract

The goal of the thesis was to implement automatic gain control algorithm for speech signals recorded with microphone array. The project was accomplished using LCMV filter which is a complex signal processing concept. In this work it is used to equalize speech at different loudness caused for example by different distances from speakers to array. The implemented method estimates directions of arrival of sound waves using MUSIC algorithm and adjusts loudness of users. Simultaneously it attenuates white noise. The solution is based on microphone array recordings. Author of this dissertation developed generator which simulates geometry of microphone array and room response. In order to write this thesis author performed a research including state-of-the-art scientific literature. The outcome of this work includes software solution written in Python programming language. The implementation was thoroughly tested. Influence of many factors including number of audio sources, signal to noise ratio and geometry of microphone array and room. The obtained results have been analysed and described.

Structure of the paper is as follows. In chapter 2 author has shortly described theory of the problem. In chapter 3 the solution was presented along with short description of used methods. Chapter 4 contains details regarding implementation. Finally chapter 5 presents tests of the system.

Spis treści

| | |
|--|-----------|
| Spis treści | 7 |
| 1 Wstęp | 9 |
| 1.1 Wprowadzenie | 9 |
| 1.2 Cel | 9 |
| 1.3 Zakres wykonanej pracy | 10 |
| 2 Analiza teoretyczna problemu | 13 |
| 2.1 Model systemu | 13 |
| 2.2 Postawiony Problem | 16 |
| 3 Rozwiązanie problemu | 17 |
| 3.1 Model rozwiązania | 17 |
| 3.2 Widmowa gęstość mocy | 18 |
| 3.3 Algorytm lokalizacji źródeł dźwięku MUSIC | 18 |
| 3.4 Automatyczna kontrola głośności | 19 |
| 4 Szczegóły implementacyjne | 23 |
| 4.1 Użyte narzędzia | 23 |
| 4.2 Metologia rozwoju programu | 23 |
| 4.3 Generator sygnałów mikrofonowych | 24 |
| 4.4 Schemat przetwarzania danych | 25 |
| 5 Testy i ewaluacja | 27 |
| 5.1 Konfiguracja | 27 |
| 5.2 Charakterystyka kierunkowa filtra LCMV | 28 |
| 5.3 Ewaluacja algorytmu MUSIC | 30 |
| 5.4 Testy całego systemu | 34 |
| 5.4.1 SNR | 34 |
| 5.4.2 Działanie algorytmu dla pojedynczego użytkownika | 35 |
| 5.4.3 Działanie algorytmu dla wielu użytkowników | 36 |
| 5.4.4 Działanie algorytmu w przypadku ruchu mówcy | 40 |
| 5.4.5 Odporność systemu na pogłos | 41 |
| 6 Podsumowanie | 45 |

Rozdział 1

Wstęp

1.1 Wprowadzenie

Algorytmy automatycznej kontroli głośności (ang. automatic gain control - AGC) są elementem współczesnych systemów przetwarzania sygnałów zarówno akustycznych tak jak w poniższej pracy jak i na przykład radiowych [1]. Celem stosowania takiego algorytmu w przypadku sygnałów akustycznych jest utrzymanie stałej głośności wszystkich mówców w danym systemie. Konkretnym przykładem zastosowania są videokonferencje [2], które z uwagi na sytuację pandemiczną na świecie zyskały ogromną popularność. Z uwagi na różne czułości mikrofonów uczestników, różną głośność mówienia i odległość od mikrofonu, uczestnicy mogą odczuwać nieprzyjemne dla ucha fluktuacje głośności. Algorytm AGC redukuje te efekty. Ważne jest aby takie algorytmy skutkowały wysokim poziomem sygnału do szumu (ang. signal to noise ratio - SNR) jak i wysokim stosunkiem sygnału do pozostałych interferencji.

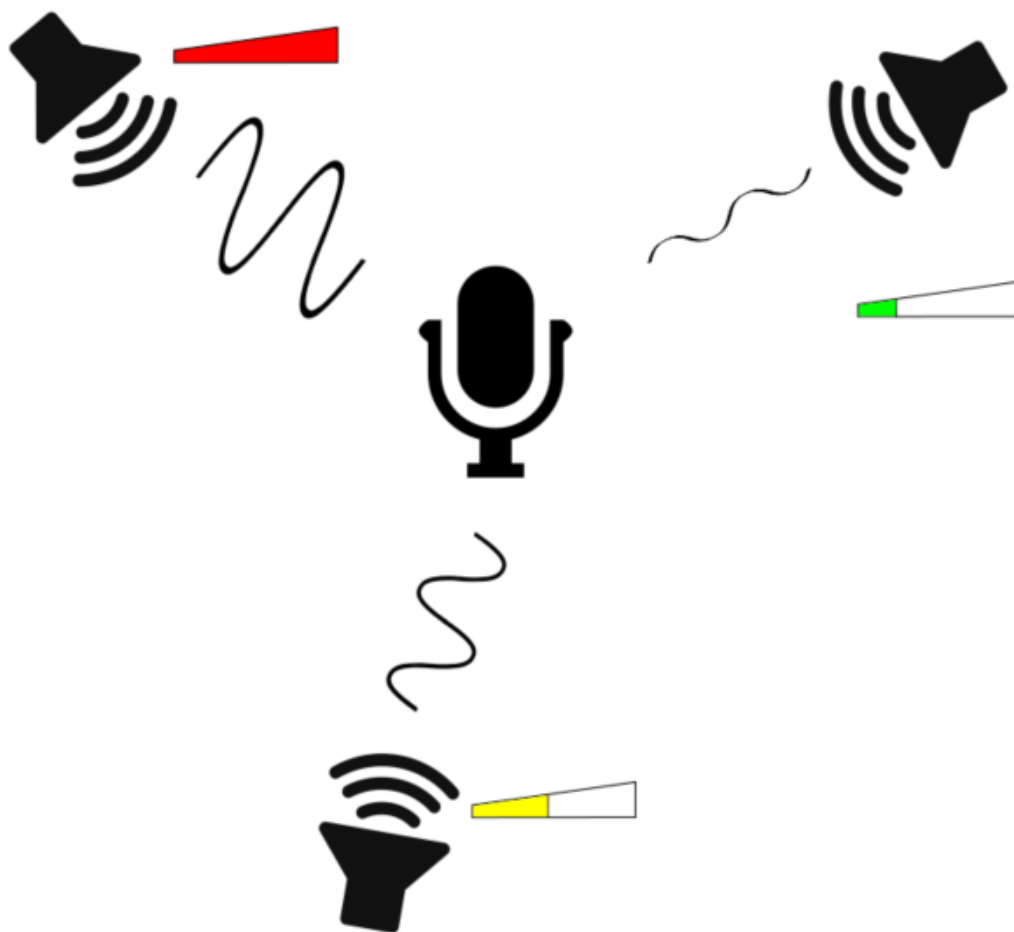
Najprostszym rozwiązaniem jest zastosowanie systemu jednokanałowego [3]. W takim systemie możliwe jest tylko sterowanie głośnością sygnału wejściowego jako całości, na podstawie chwilowej wartości obwiedni. Uniemożliwia on jednak różny poziom wzmocnienia dla różnych mówców i usunięcie interferencji, jeśli współlistnieją one w czasie z głównym sygnałem.

Rozwiązaniem bardziej skomplikowanym, zarówno od strony sprzętowej, programistycznej i czasu obliczeń jest użycie systemu z wieloma mikrofonami. Taki system umożliwia pożądane rozróżnienie między mówcami i eliminację interferencji [4]. W systemie takim możliwa jest estymacja kierunków nadchodzenia fali, a co za tym idzie mocy nadchodzącej z wybranych stron i zastosowanie odpowiednich filtrów, które wzmacniają sygnał z żądanych kierunków.

W tej pracy skupiono się właśnie na drugim z wymienionych rodzajów algorytmów AGC. W pracy zostanie przedstawiona praktyczna implementacja bazująca na nagraniach z macierzy mikrofonowych.

1.2 Cel

Celem pracy inżynierskiej była praktyczna implementacja istniejącego algorytmu AGC z tłumieniem tła akustycznego, wraz z estymacją kierunku nadchodzenia fali.



Rysunek 1.1: Graficzna reprezentacja problemu

Wybrany algorytm powinien spełniać cechy wymienione w poprzedniej sekcji, tj. pozwalać na regulację głośności nagrania osobno dla różnych mówców i minimalizować zakłócenia. Z tego powodu zdecydowano się na użycie tak zwanego filtra LCMV (ang. linearly constrained minimum variance). Opracowany algorytm w założeniu nie musi być algorytmem czasu rzeczywistego. Powinien wykonywać żądane operacje bazując na nagraniach z macierzy mikrofonowej. Kolejnym celem projektu było przeprowadzenie testów skuteczności zaimplementowanego systemu w różnych wariantach i warunkach pracy. W tym celu powstanie generator symulujący różne możliwe konfiguracje.

1.3 Zakres wykonanej pracy

Pierwszym z etapów pracy inżynierskiej było zapoznanie się autora pracy z dostępnymi publikacjami opisującymi zarówno tematykę AGC [5], [3] jak i generalną tematykę przetwarzania sygnałów dla macierzy czujników akustycznych [6] i wy-

bór metody. Wybrana metoda bazuje w głównej mierze na metodzie [5]. Następnie dokonany został wybór środowiska pracy- języka Python wraz z paczką NumPy [7]. Szczegółowy opis użytych narzędzie znajduje się w rozdziale 4. Następnie autor pracy napisał symulator umożliwiający generację sygnałów mikrofonowych. Dalsza część realizacji pracy obejmowała- zaplanowanie praktycznej implementacji systemu, wykonanie tej implementacji i testy przy użyciu sygnałów macierzy mikrofonowej otrzymanej z symulatora. W testach skupiono się na ewaulacji systemu pod kątem działania dla jednego mówcy i wielu mówców. Sprawdzono odporność na zakłócenia i wpływ geometrii macierzy czujników akustycznych na działania poszczególnych składowych algorytmu.

Rozdział 2

Analiza teoretyczna problemu

2.1 Model systemu

Z uwagi na to, że litera "x" jest wielokrotnie używana w opisie modelu autor pracy zdecydował się na wstępie wprowadzić krótką legendę używanych oznaczeń z tą literą:

- x pisane czcionką typu *italic* oznacza współrzędną w układzie kartezjańskim
- $x(t)$ lub $x(n)$ oznaczają sygnał w dziedzinie czasu (ciągłej lub dyskretnej)
- $X(f)$, $X(k)$ oznacza reprezentację tego samego sygnału w dziedzinie częstotliwości, zaś $X(k, n)$ w dziedzinie krótkoczasowej transformaty Fouriera (ang. short time Fourier transform - STFT).
- $\mathbf{x}(f)$, $\mathbf{x}(k)$ lub $\mathbf{x}(k, n)$ pisane pogrubioną czcionką oznacza wektor o długości M reprezentujący sygnał na M mikrofonach odpowiednio dziedzinach częstotliwości i krótkoczasowej transformaty Fouriera.

Na wstępie postanawia się intuicyjnie przedstawić pochodzenie wzorów użytych poniżej. W szczególności interesujący jest wektor sterujący \mathbf{a} .

Zakłada się, że źródła znajdują się w polu dalekim, tj. odległość źródła od mikrofonu jest znacznie większa od długości fali i fizycznych wymiarów mikrofonu. Co więcej zakłada się, że mikrofony są na tyle blisko siebie, że nie występuje pomiędzy nimi tłumienie.

Na mikrofon pada sygnał nadchodzący z kierunku θ_l . Zapis sygnału w punkcie $\mathbf{d}_0 = [0, 0]$ to $x_l(t, \mathbf{d}_0)$. Opóźnienie w czasie tego sygnału na mikrofonie położonym w $\mathbf{d}_m = [x_m, y_m]$ względem punktu \mathbf{d}_0 wynosić będzie zatem pewne τ . Różnica dróg między punktami wyniesie Δd tak jak na rysunku 2.1. Za pomocą prostej trygonometrii może zostać udowodnione, że:

$$\Delta d = -x_m \cos \theta_l - y_m \sin \theta_l \quad (2.1)$$

Następnie przyjmując prędkość rozchodzenia się fali w ośrodku c można zapisać, że:

$$\tau = \frac{\Delta d}{c} \quad (2.2)$$

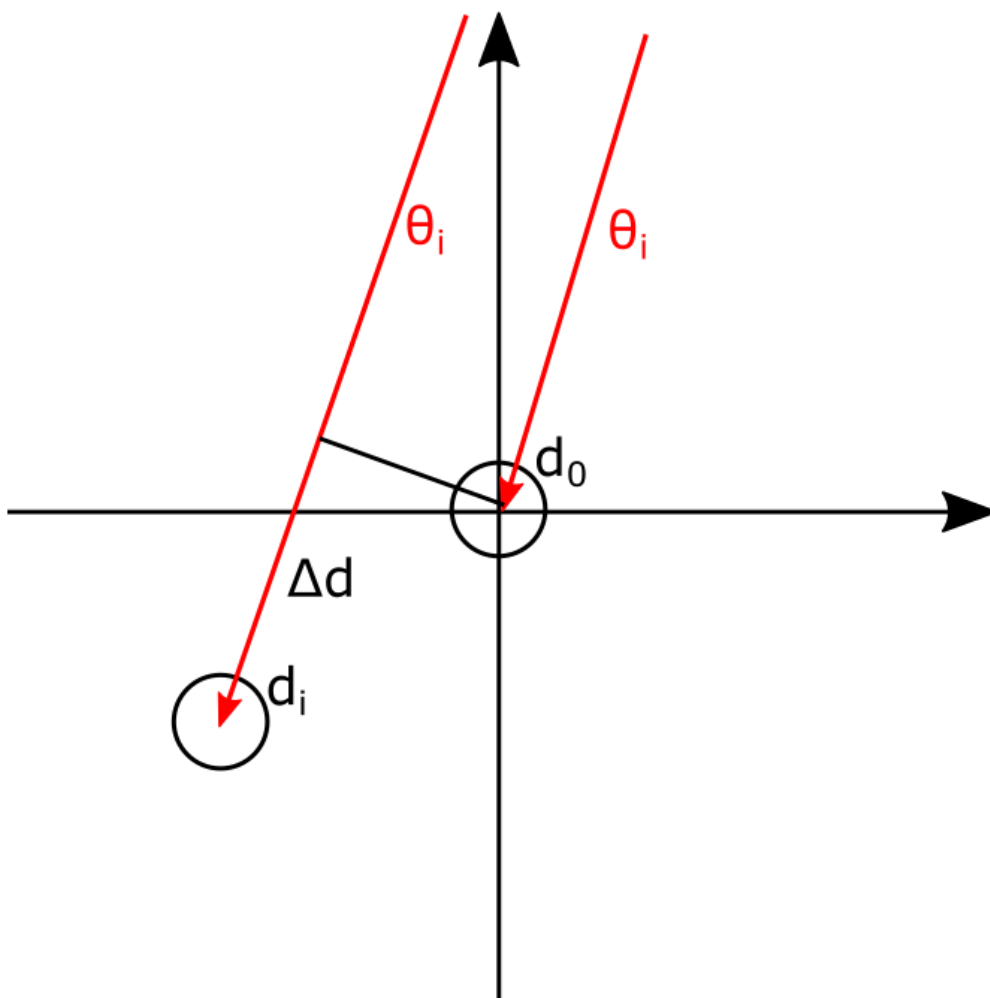
Przechodząc z sygnałem do dziedziny częstotliwości i zakładając parę transformat Fouriera $x_l(t, \mathbf{d}_0) \leftrightarrow X_l(f, \mathbf{d}_0)$ można zapisać, że:

$$x_l(t, \mathbf{d}_m) \leftrightarrow X_l(f, \mathbf{d}_0)e^{j2\pi f\tau} = a_l(f, \mathbf{d}_m)X_l(f, \mathbf{d}_0) \quad (2.3)$$

W przypadku wielu mikrofonów ustawionych na pozycjach $\{\mathbf{d}_1 \dots \mathbf{d}_M\}$:

$$\mathbf{x}_l(f) = \mathbf{a}_l(f)X_l(f, \mathbf{d}_0) \quad (2.4)$$

Gdzie wektor sterujący $\mathbf{a}_l = [a_l(\mathbf{d}_1) \dots a_l(\mathbf{d}_M)]$. Taki zapis równania może być przeniesiony do dziedziny krótkoczasowej transformaty Fouriera. Wówczas otrzymane zostanie równanie (2.7).



Rysunek 2.1: Opóźnienie fali na mikrofonie

Poniższy model sygnału jest zaczerpnięty z [4] i [5], z tą różnicą, że w tej pracy wszystkie zakłócenia traktowane są razem.

Zakłada się, że używanym czujnikiem jest macierz złożona z M mikrofonów dookólnych rozłożonych na płaskiej powierzchni w pozycjach $\{\mathbf{d}_1 \dots \mathbf{d}_M\}$.

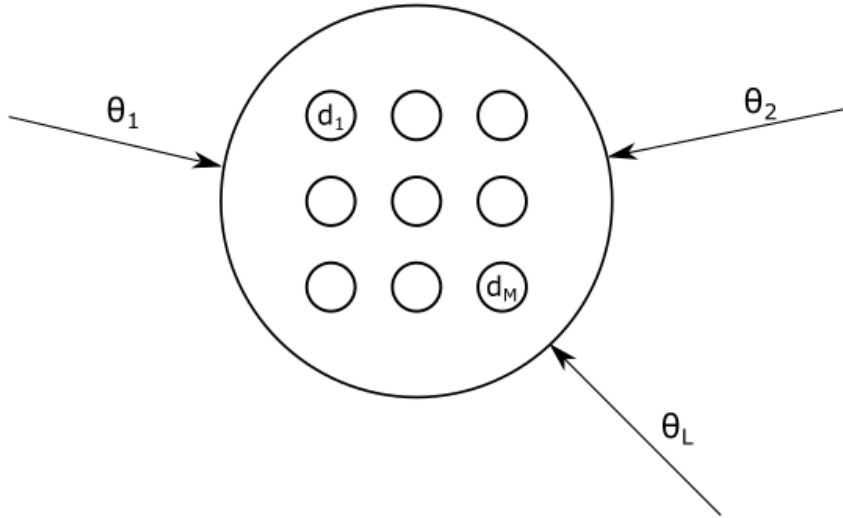
Na czujnik pada L fal płaskich, gdzie $M > L$ ośrodkiem izotropowym i jednorodnym, w którym obecne są zakłócenia. Każda z fal nadbiega z kąta $\{\theta_1 \dots \theta_L\}$. Model może być zapisany w dziedzinie STFT jako:

$$\mathbf{x}(k, n) = [\mathbf{X}(k, n, \mathbf{d}_1) \dots \mathbf{X}(k, n, \mathbf{d}_M)]^T, \quad (2.5)$$

lub alternatywnie jako:

$$\mathbf{x}(k, n) = \sum_{l=1}^L \mathbf{x}_l(k, n) + \mathbf{x}_n(k, n), \quad (2.6)$$

gdzie $\mathbf{x}_l(k, n)$ to wektor natężeń l -tej fali na poszczególnych mikrofonach a $\mathbf{x}_n(k, n)$ to wektor natężenia tła akustycznego na mikrofonach. Oznacza to, że otrzymany na każdym z mikrofonów sygnał jest mieszaniną sygnału z kilku różnych źródeł i zakłóceń.



Rysunek 2.2: Model systemu

Każdy z wektorów $\mathbf{x}_l(k, n)$ może być opisany za pomocą równania:

$$\mathbf{x}_l(k, n) = \mathbf{a}_l(k, n) X_l(k, n, \mathbf{d}_0), \quad (2.7)$$

gdzie $\mathbf{a}_l(k, n) X_l(k, n, \mathbf{d}_0)$ jest wartością natężenia l -tej fali w punkcie referencyjnym \mathbf{d}_0 . Często w publikacjach, na przykład w [5] za punkt referencyjny jest wybierane położenie mikrofonu \mathbf{d}_1 . Autor pracy zdecydował się jednak na bardziej ogólne podejście do problemu.

W pracy zakłada się, że kierunek nadchodzenia fali nie jest znany ale znana jest ilość źródeł L w systemie.

2.2 Postawiony Problem

W wyżej opisanym modelu mogą zostać zdefiniowane następujące potencjalne wyzwania:

- Sygnał \mathbf{x}_i może charakteryzować się amplitudą różną od sygnału \mathbf{x}_j co będzie skutkować zagłuszaniem jednego z użytkowników systemu przez innego.
- Natężenie sygnału \mathbf{x}_i może mocno fluktować w czasie z uwagi na przykład na przybliżanie i oddalanie się mówcy od mikrofonów w czasie nagrania.
- W systemie występuje tło akustyczne- niechciane zakłócenia, których wyeliminowanie jest pożądanym działaniem.

Autor pracy postawił sobie za cel implementację algorytmu umożliwiającego minimalizację wyżej wymienionych problemów. Celem projektu jest uzyskanie takiego sygnału wyjściowego, w którym poziom głośności jest stały dla wszystkich mówców a występujące zakłócenia są jak najmniejsze.

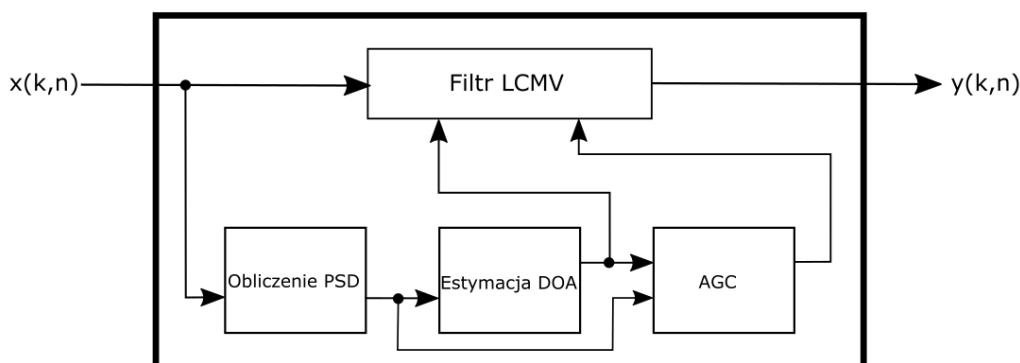
Rozdział 3

Rozwiązanie problemu

3.1 Model rozwiązania

W pracy zdecydowano się rozwiązać problem na bazie metody opisanej w publikacji [5]. W tej sekcji zostanie podany zbiór metod użytych do rozwiązania problemu. Zdefiniowanie i opis poszczególnych metod nastąpi w kolejnych sekcjach rozdziału.

Przetwarzanie sygnału zaczyna się od obliczenia STFT nagrania wejściowego. Następnie odbywa się przetwarzanie przedstawione na rysunku 3.1. Na początku obliczona zostaje macierz widmowej gęstości mocy nagranych sygnałów (ang. power spectral density - PSD). Następnie za jej pomocą estymowany jest kierunek nadchodzenia fal (ang. direction of arrival - DOA). Dzięki połączeniu tych wartości możliwe jest zastosowanie algorytmu automatycznej regulacji głośności. Ostatnim krokiem jest zastosowanie filtru LCMV, który jako argumenty przyjmuje żądane poziomy wzmocnień, wyestymowane kierunki i sygnał wejściowy. Taki filtr produkuje sygnał wyjściowy o wyrównanym poziomie głośności z jednoczesnym tłumieniem tła akustycznego.



Rysunek 3.1: Diagram blokowy

3.2 Widmowa gęstość mocy

Widmowa gęstość mocy może być zapisana jako:

$$\phi_l(k, n) = E\{|X_l(k, n, \mathbf{d}_0)|\}. \quad (3.1)$$

Macierz widmowej gęstości mocy definiuje się następująco:

$$\Phi(k, n) = E\{\mathbf{x}(k, n)\mathbf{x}^H(k, n)\}. \quad (3.2)$$

W praktyce jest obliczana jako średnia z N zebranych próbek:

$$\Phi(k, n) = \frac{1}{N} \sum_{i=n-N}^{n-1} \mathbf{x}(k, i)\mathbf{x}^H(k, i). \quad (3.3)$$

Zgodnie z [4], zakładając, że wszystkie elementy równania (2.6) są nieskorelowane, można zapisać:

$$\Phi(k, n) = \sum_{l=1}^L \Phi_l(k, n) + \Phi_n(k, n), \quad (3.4)$$

gdzie odpowiednio $\Phi_l(k, n)$ i $\Phi_n(k, n)$ są macierzami PSD odpowiednio l -tej padającej fali i zakłóceń.

3.3 Algorytm lokalizacji źródeł dźwięku MUSIC

MUSIC(ang. Multiple Emitter Location and Signal Parameter Estimation) został opisany w [8]. Nowocześniejsze opracowania można też znaleźć w [9] i [6]. Algorytm pozwala wyestymować kierunki nadchodzenia fali w systemie zawierającym wiele źródeł. Algorytm zakłada przyjęty model sygnału z założeniem, że tło akustyczne to szum własny czujników i szum tła. W pracy przyjęto uproszczony model i potraktowano zakłócenia jako Gaussowski szum biały (ang. white Gaussian noise). Obecność innych zakłóceń pogarsza działanie systemu.

Na potrzeby tej pracy zostanie przedstawiony skrótowo algorytm. Wyprowadzenie znajduje się w wyżej cytowanej publikacji. Definiuje się zapis $\mathbf{a}(k, n, \theta)$, oznaczający wektor sterującym dla kąta padania θ i indeksu czasowo-częstotliwościowego (k, n) oraz $\Phi(k, n)$ oznaczający macierz PSD sygnału dla tego samego indeksu czasowo-częstotliwościowego.

Liczba nadchodzących fal L może być wyestymowana na przykład zgodnie z publikacją [10]. W tej pracy zakłada się jednak, że liczba źródeł jest znana.

W rozkładzie na wartości własne macierzy $\Phi(k, n)$ wektory własne powiązane z L największymi wartościami własnymi nazywa się wektorami podprzestrzeni sygnału i oznacza jako \mathbf{s}_i , zaś pozostałe wektory- wektorami z podprzestrzeni szumu \mathbf{e}_i . Macierz $\mathbf{E}(k, n)$ zawiera w kolejnych kolumnach kolejne wektory podprzestrzeni szumu:

$$\mathbf{E}(k, n) = [\mathbf{e}_1(k, n) \dots \mathbf{e}_{M-L}(k, n)]. \quad (3.5)$$

Następnie definiuje się funkcję:

$$P(\theta, k, n) = \frac{1}{\mathbf{a}^H(\theta, k, n) \mathbf{E}(k, n) \mathbf{E}^H(k, n) \mathbf{a}(\theta, k, n)}. \quad (3.6)$$

L największych maksimów lokalnych takiej funkcji w dziedzinie kątów uznaje się za wyestymowane kierunki nadchodzenia fali.

Celem algorytmu jest jednak uzyskanie jednego kierunku nadchodzenia fali dla n -tej chwili czasowej w systemie szerokopasmowym. Propozycją autora pracy jest zdefiniować taką funkcję $P_a(\theta, n)$, że:

$$P_a(\theta, n) = \frac{1}{K} \sum_{k=1}^K P(\theta, k, n). \quad (3.7)$$

Ponownie, L największych maksimów lokalnych wskazuje kierunki nadchodzenia fali.

3.4 Automatyczna kontrola głośności

Kluczowym elementem systemu będzie sterowanie tak zwaną przestrzenną funkcją głośności $G(\theta_l, n)$. Funkcja ta jest odpowiedzialna za utrzymywanie głośności sygnału wyjściowego na założonym poziomie:

$$Y(k, n) = \sum_{l=1}^L G(\theta_l, n) X_l(k, n, \mathbf{d}_0) \quad (3.8)$$

Następnym istotnym pojęciem jest filtr LCMV. Punktem wyjściowym dla definicji filtru LCMV jest próba zapisania estymaty sygnału wyjściowego $\hat{Y}(k, n)$ jako:

$$\hat{Y}(k, n) = \mathbf{w}^H(k, n) \mathbf{x}(k, n) \quad (3.9)$$

Aby znaleźć współczynniki filtru $\mathbf{w}_n(k, n)$, które pozwolą jak najlepiej przybliżyć żądane rozwiązanie należy rozwiązać problem optymalizacyjny zdefiniowany przez dwa poniższe równania:

$$\mathbf{w}_n(k, n) = \arg \min_{\mathbf{w}} \mathbf{w}^H \Phi_n \mathbf{w} \quad \text{s.t} \quad (3.10)$$

$$\mathbf{w}^H \mathbf{a}_l(k, n) = G(\theta_l, n), \quad l \in \{1, 2, \dots, L\}. \quad (3.11)$$

Ten zapis oznacza, że żądany filtr powinien minimalizować podawane na wejście zakłócenia (3.10) i odpowiednio sterować głośnościami poszczególnych nadchodzących fal (3.11). Rozwiązanie tak postawionego problemu można znaleźć w [4] zaś wyprowadzenie w [11]:

$$\mathbf{w}_n = \Phi_n^{-1} \mathbf{A} [\mathbf{A}^H \Phi_n \mathbf{A}]^{-1} \mathbf{g}, \quad (3.12)$$

gdzie macierz $\mathbf{A}(k, n) = [\mathbf{a}_1(k, n) \dots \mathbf{a}_L(k, n)]$ zaś wektor $\mathbf{g}(n) = [G(\theta_1, n) \dots G(\theta_L, n)]^T$.

Autor publikacji [5] proponuje zdefiniowanie funkcji przestrzennego rozkładu głośności (ang. spatial loudness distribution - SLD) zapisanego jako:

$$\Psi(\theta, n) = \sum_{k=1}^K \beta^2(k) \sum_{l=1}^L \delta_{\theta, \theta_l} \phi_l(k, n), \quad (3.13)$$

gdzie (k, n) to indeksy czasowo częstotliwościowe, $\delta_{\theta, \theta_l}$ to delta Kroneckera- sygnał przyjmujący wartość 1 dla θ_l i 0 dla pozostałych.

Aby wyestymować PSD ϕ_i i -tej fali dokonuje się filtracji filtrem LCMV takim, że:

$$\mathbf{w}^H \mathbf{a}_l(k, n) = \begin{cases} 1 & l = i \\ 0 & l \neq i \end{cases} \quad (3.14)$$

Wartość współczynników β może być określona na podstawie istniejących norm, na przykład [12]. Autor tej pracy inżynierskiej zdecydował jednak przyjąć wartość:

$$\beta(k) = 1, \quad k = \{1, 2, \dots, K\}. \quad (3.15)$$

Definiuje się także długoczasową funkcję przestrzennego rozkładu głośności zdefiniowaną jako:

$$\Psi_{LT}(\theta, n) = \alpha_{LT} \Psi_{LT}(\theta, n-1) + (1 - \alpha_{LT}) \Psi(\theta, n), \quad (3.16)$$

gdzie α_{LT} oznacza pewnien przyjęty współczynnik. Działanie ma na celu wygładzenie funkcji w dziedzinie czasu. Głośność sygnału powinna być niewrażliwa na fluktuacje o bardzo dużych częstotliwościach. Taka definicja funkcji jest tożsama z filtracją dolnoprzepustową za pomocą filtra o nieskończonej odpowiedzi impulsowej(IIR).

Funkcje nie są zależne od składowej częstotliwościowej bo celem pracy jest wzmocnienie amplitudy całego sygnału a nie poszczególnych pasm.

Przestrzenną funkcję głośności dla danego kąta możnaby otrzymać za pomocą:

$$G(\theta, n) = \sqrt{\frac{\Psi_{target}(\theta)}{\Psi_{LT}(\theta, n)}}. \quad (3.17)$$

Należy jednak zauważyć, że tak obliczona funkcja prowadziłaby do silnego wzmocnienia kierunków, w których nie ma sygnału. Co za tym idzie niemalże nieskończenie wzmacniane byłoby tło akustyczne. Kilkustopniowy błąd estymacji kierunku nadchodzenia fali mógłby w praktyce uniemożliwić poprawne działanie systemu. Aby to uniemożliwić definiuje się minimalną moc uznawaną za sygnał $\Psi_{min}(n)$. Obliczona może być jako:

$$\Psi_{min}(n) = \max\left(\frac{1}{2\pi} \int_0^{2\pi} \Psi(\theta, n) d\theta, \Psi_0\right), \quad (3.18)$$

gdzie Ψ_0 jest pewną ustaloną granicą.

Następnie dany jest algorytm do obliczenia przestrzennej funkcji głośności $G'(\theta, n)$:

Algorytm 1 Obliczanie $G'(\theta, n)$

- 1: Zainicjalizuj $G'(\theta, n)$ wartością 1 dla każdego kierunku
 - 2: Znajdź maksimum globalne funkcji Ψ_{LT} w dziedzinie kąta(θ_{\max}) i oblicz przestrzenną funkcję głośności $G(\theta_{\max}, n)$ za pomocą (3.17).
 - 3: Wygeneruj okno przestrzenne o określonej szerokości Θ , które ma wzmocnienie 1 na brzegach i $G(\theta_{\max}, n)$ w środku.
 - 4: Umieść okno w θ_{\max} i wyzeruj wartości funkcji Ψ_{LT} w punktach pokrytych przez okno.
 - 5: Kontynuuj krok 2 tak długo aż wszystkie punkty funkcji Ψ_{LT} są poniżej progu Ψ_{\min} .
-

Następnie oblicza się funkcję wygładzającą $\hat{G}(\theta, n)$:

$$\hat{G}(\theta, n) = \alpha_G \hat{G}(\theta, n) + (1 - \alpha_G) G'(\theta, n) \quad (3.19)$$

Funkcja tej operacji jest taka sama jak (3.16). Otrzymane \hat{G} wstawia się do równania (3.11).

Rozdział 4

Szczegóły implementacyjne

4.1 Użyte narzędzia

Projekt inżynierski został zaimplementowany jako program komputerowy w języku Python. Do realizacji zostały użyte głównie paczki:

- numpy- paczka służąca do obliczeń naukowych. Udostępnia rozbudowane API do algebry liniowej i przetwarzania sygnałów.
- scipy- moduł umożliwiający obliczenia matematyczne i techniczne, w szczególności przetwarzanie sygnałów stosowane w tej pracy
- matplotlib- paczka służąca do wizualizacji danych.
- rir generator- moduł umożliwiający symulację odpowiedzi impulsowej pokoju

Więcej informacji można znaleźć w dokumentacji paczek [7], [13], [14] i [15].

4.2 Metologia rozwoju programu

Realizacja pracy była podzielona na stworzenie właściwego programu opisane szczegółowo w rozdziale 3 i na stworzenie generatora symulującego rzeczywisty sygnał mikrofonowy z dodatkiem zakłóceń.

W założeniach oprogramowanie miał mieć charakter obiektowy. Główne bloki zostały zaimplementowane jako klasy, które komunikują się między sobą za pomocą interfejsów. Takie rozwiązanie powoduje usystematyzowaną strukturę projektu i umożliwia łatwy rozwój oprogramowania w przyszłości. Przykładem jest implementacja algorytmu MUSIC. Klasa MUSIC dziedziczy po klasie abstrakcyjnej DOA. W ten sposób proste w przyszłości będzie dopisanie w programie innego algorytmu estymacji kierunku nadchodzenia fali jako innej klasy dziedziczącej po DOA. Poniżej przedstawiono diagram klas w konwencji UML [16]:

W diagramie 4.1 skrótkowo opisano klasy. Nazwy nie odpowiadają(poza DelayGenerator) nazwom w kodzie. Nie są również dodane wszystkie składowe i metody. Schemat ma za zadanie przedstawić koncepcyjnie zaimplementowane rozwiązanie i pokazać podejście autora pracy do implementacji systemu.

- Opcjonalnie informacje o geometrii pokoju i o tym czy symulacja pogłosu jest włączona

Schemat działania generatora jest względnie prosty. Przyjmuje się model przedstawiony w rozdziale 2. Zakłada się, że sygnał wejściowy odpowiada wartości sygnału w punkcie \mathbf{d}_0 . Posiłkując się równaniami (2.1) i (2.2) można zapisać, że:

$$x_l(\mathbf{d}_m, t) = x_l(\mathbf{d}_0, t - \tau), \quad (4.1)$$

gdzie $x_l(\mathbf{d}_m, t)$ to wymuszenie l -tej fali dźwiękowej w punkcie \mathbf{d}_m . W takim układzie możliwe są bieżące zmiany opóźnienia bazując na pozycji źródła. Dodatkowo możliwa jest generacja białego szumu $n(t)$. Biały szum o mocy σ może być przedstawiony jako zmienna losowa z rozkładu gaussowskiego o średniej 0 i wariancji σ .

Generator umożliwia symulację pomieszczenia, w którym znajduje się mikrofon. Za pomocą paczki `rir` generator [14] generowana jest odpowiedź impulsowa pokoju $\mathbf{h}(t)$. Tak generowana odpowiedź nazywana będzie później pogłosem. Szczegóły dotyczące genezy takiego sposobu symulacji pomieszczenia mogą być znalezione w książce [17]. Ostatecznie zatem sygnał wejściowy na m -tym mikrofonie może być zapisany w dziedzinie czasu jako:

$$x(t, \mathbf{d}_m) = \sum_{l=1}^L x_l(\mathbf{d}_0, t - \tau_{l,m}) * h_l(t) + n(t) \quad (4.2)$$

Z uwagi na to, że w celu generacji filtra pogłosowego konieczne są konkretne pozycje źródeł i mikrofonu, włączenie symulacji pomieszczenia nie pozwala na zmiany pozycji źródeł w czasie symulacji.

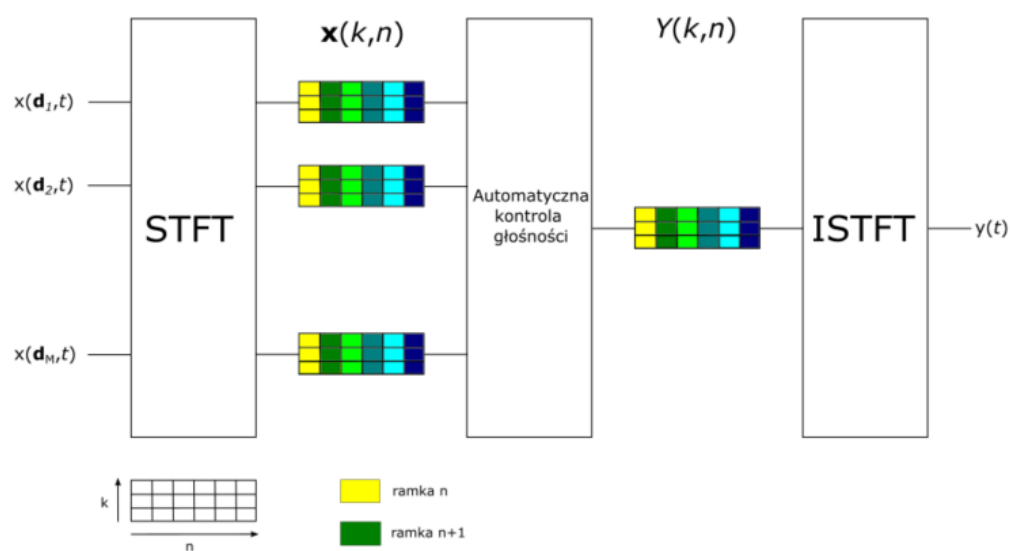
4.4 Schemat przetwarzania danych

Algorytm przetwarza nagrane pliki dźwiękowe. Składa się z następujących kroków:

Algorytm 2 Schemat przetwarzania

- 1: Na wejście trafia M ścieżek dźwiękowych
 - 2: Sygnał jest przetwarzany do reprezentacji STFT
 - 3: Przetwarzanie odbywa się w pętli po kolejnych indeksach czasowych, nazywanych ramkami
 - 4: Na wyjściu przetwarzania sygnałów zwracany jest sygnał $Y(k, n)$ w dziedzinie STFT
 - 5: Dane wyjściowe są przetwarzane za pomocą odwrotnej krótkoczasowej transformy Fouriera (ISTFT) do postaci czasowej.
-

Na rysunku 4.2 blok oznaczony jako automatyczna kontrola głośności zawiera w sobie operacje obliczenia PSD, estymacji DOA, wyznaczania współczynników i filtracji przy użyciu LCMV. Jest więc równoznaczny z diagramem 3.1.



Rysunek 4.2: Schemat przetwarzania sygnałów wielokanałowych

Rozdział 5

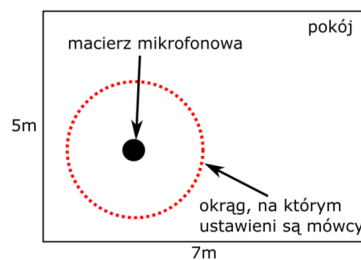
Testy i ewaluacja

5.1 Konfiguracja

Dla celów testów przyjęto pewne sztywne założenia. Jako prędkość dźwięku przyjmuje się 343 m/s. Wszystkie testy są przeprowadzane na sygnałach mowy próbkowanych z częstotliwością 8kHz. STFT i ISTFT jest obliczane przy pomocy okna hanninga [18] ze współczynnikiem nachodzenia okien (ang. overlap) 50%. Jako okno przestrzenne także przyjęto okno hanninga o szerokości 40° . Liczba ramek N , z których wyliczana jest macierz PSD jest równa 100. Aktualizacja DOA odbywa się co 10 ramek. Współczynniki α_{LT} i α_G mają wartości 0.98. Jako macierz mikrofonową przyjęto kwadratową strukturę o liczbie mikrofonów $M = 16$ o równomiernej odegłości między mikrofonami równej $\Delta d = 2\text{cm}$. Na potrzeby niektórych eksperymentów odległość między mikrofonami była zwiększana.

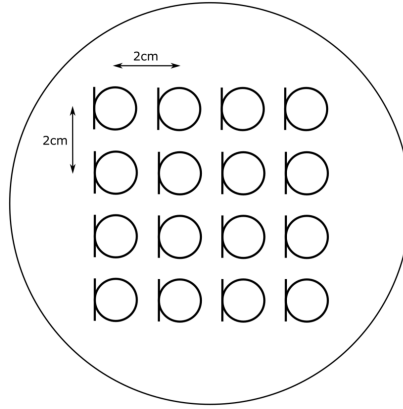
Podawane sygnały dźwiękowe są znormalizowane w ten sposób, że moduł z ich maksymalnej wartości wynosi 1. Przekłada się to na moce średnie z zakresu 0.1-0.2. Zdecydowano zatem, że parametr $\Psi_0 = 0.001$.

W eksperymentach, w których włączony jest pogłos symulowany jest pokój o wymiarach 7m x 5m x 3m. Parametr czas pogłosu ustawiony jest na 0.4 co odpowiada dosyć silnemu zapogłosowaniu. Macierz umieszczona jest w punkcie $[2, 2, 1.6]$ a mówcy znajdują się w odległości 1.5m od niej. Ilustracja znajduje się na rysunku 5.1.



Rysunek 5.1: Symulowany Pokój

Jako zakłócenie do usunięcia wybrano biały szum. Jego macierz PSD może być zapisana jako macierz diagonalna z wartościami 1 na przekątnej $\Phi_n = \text{diag}([1, \dots, 1])$.



Rysunek 5.2: Macierz mikrofonowa

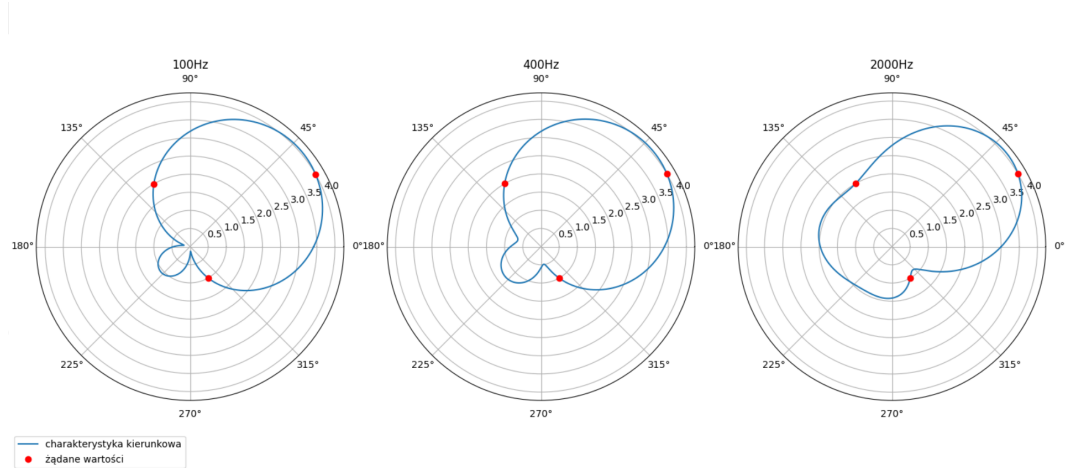
W sekcjach poniżej kolejno opisywane są wykonywane testy i doświadczenia.

5.2 Charakterystyka kierunkowa filtra LCMV

Poniżej przedstawiono wykresy charakterystyki kierunkowej dla filtra LCMV o następujących parametrach:

- $\theta_0 = 30^\circ$, $\theta_1 = 120^\circ$, $\theta_2 = 300^\circ$
- $G(\theta_0) = 4$, $G(\theta_1) = 2$, $G(\theta_2) = 1$

dla częstotliwości 100Hz, 500Hz i 2kHz.

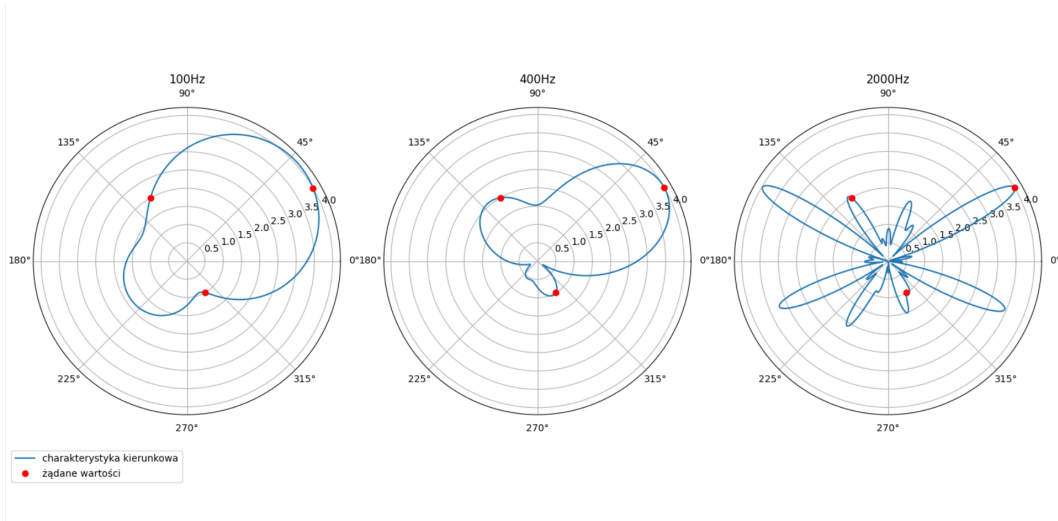


Rysunek 5.3: Charakterystyka kierunkowa dla $\Delta d = 2cm$

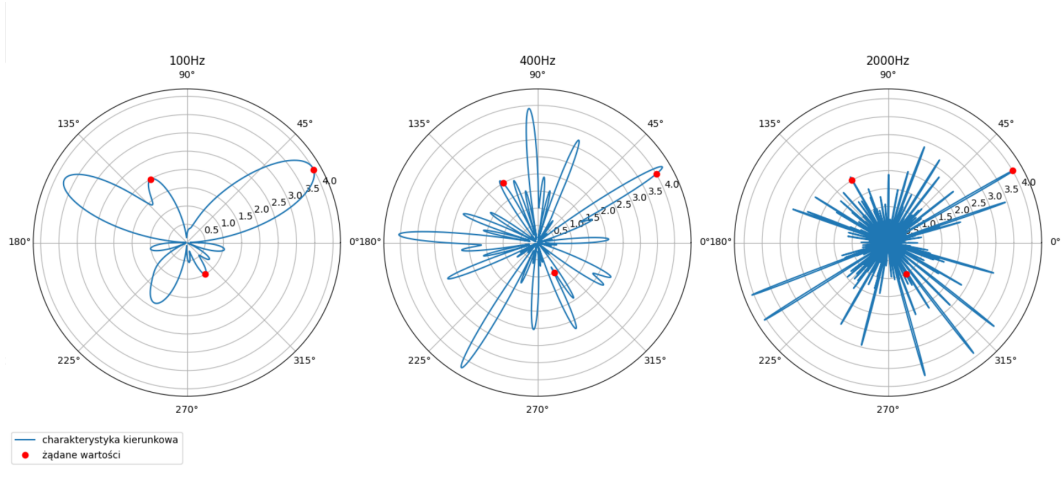
Załączony obrazek 5.3 pokazuje, że filtr prawidłowo generuje wymuszenia w określonych kierunkach dla określonych częstotliwości. Właściwości usuwania tła akustycznego zostaną sprawdzone w kolejnych sekcjach.

Zgodnie z teorią przedstawioną w [19] wraz ze wzrostem stosunku $\frac{\Delta d}{\lambda}$, gdzie λ to długość fali, rośnie kierunkowość filtru. Prowadzi to do sytuacji, w której maksima są bardzo wąskie i wartość charakterystyki kierunkowej silnie fluktuuje w dziedzinie kątów. Nieznaczny błąd estymacji kierunku nadchodzenia fali może więc prowadzić do fatalnych skutków. Zjawisko pojawiania się dużej liczby minimów i maksimów nazywane jest aliasingiem przestrzennym. Zachodzi dla $\Delta d > \frac{\lambda}{2}$. Przy założonej prędkości dźwięku dla częstotliwości $f = 4\text{kHz}$ granica aliasingu przestrzennego wystąpi dla $\Delta d \approx 4\text{cm}$. Dlatego właśnie zdecydowano się wybrać bezpieczną odległość między mikrofonami $\Delta d = 2\text{cm}$.

Warte pokazania są wykresy takich samych filtrów jeśli mikrofony byłyby oddalone od siebie odpowiednio o $\Delta d = 20\text{cm}$ i $\Delta d = 2\text{m}$. Zauważalna jest ogromna liczba maksimów i ich mała szerokość występująca już na rysunku 5.4 dla 2kHz- jest to właśnie aliasing przestrzenny.



Rysunek 5.4: Charakterystyka kierunkowa dla $\Delta d = 20\text{cm}$



Rysunek 5.5: Charakterystyka kierunkowa dla $\Delta d = 2m$

5.3 Ewaluacja algorytmu MUSIC

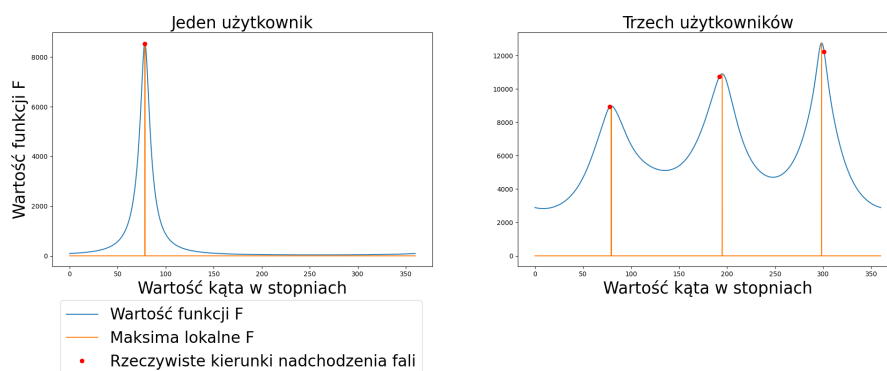
W celu sprawdzenia działania zaproponowanego algorytmu MUSIC sprawdzone są różne warianty rozłożenia mówców:

- Jeden mówca usytuowany na pozycji o kącie $\theta_0 = 78^\circ$
- Trzech mówców usytuowanych na pozycjach o kątach $\theta_0 = 78^\circ$, $\theta_1 = 192^\circ$, $\theta_2 = 301^\circ$

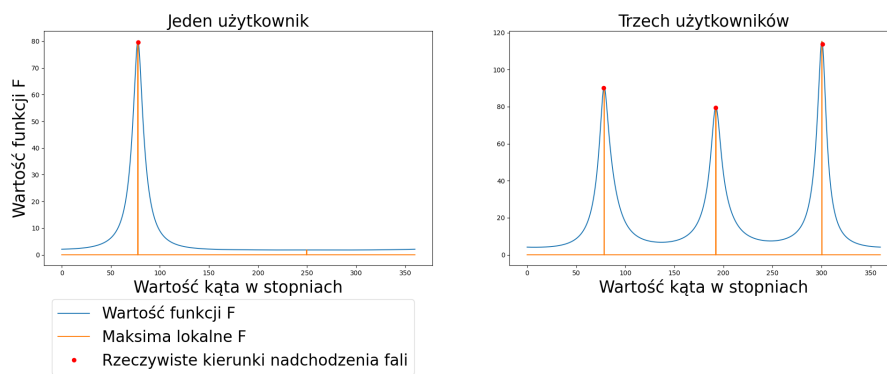
Wszystkie powyższe scenariusze będą powtórzone dla następujących konfiguracji wartości stosunku sygnału do szumu i obecności pogłosu:

- SNR = 10dB
- SNR = -10dB
- SNR = 10dB, sygnał z pogłosem

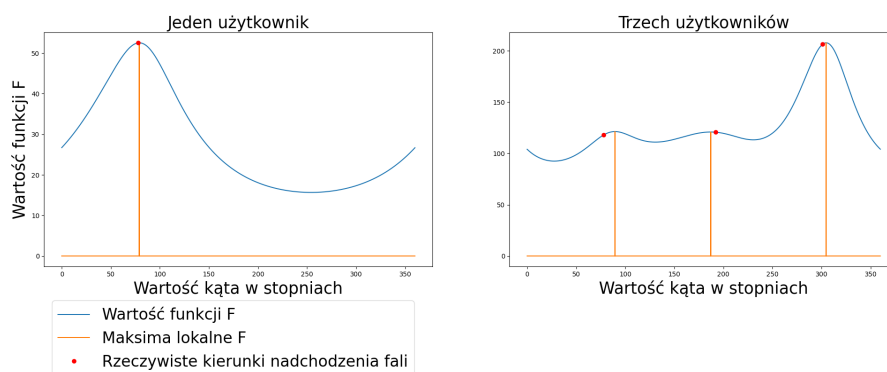
Te wartości zostały sprawdzone dla $\Delta d = 2\text{cm}$ i $\Delta d = 20\text{cm}$ dla porównania. Wygenerowane wykresy przedstawiające znalezione DOA przedstawiono na rysunkach 5.6-5.11. Na przedstawionych wykresach po lewej stronie znajduje się efekt działania dla jednego źródła a po prawej dla trzech źródeł. Prezentacja wyników jest pokazana w dziedzinie tzw. spektrum przestrzennego przedstawiającego funkcję w zależności od kąta padania fali.



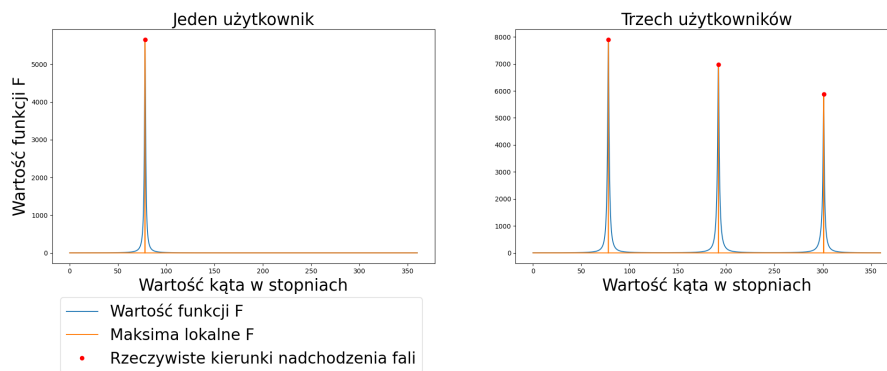
Rysunek 5.6: $\text{SNR} = 10\text{dB}$, $\Delta d = 2\text{cm}$



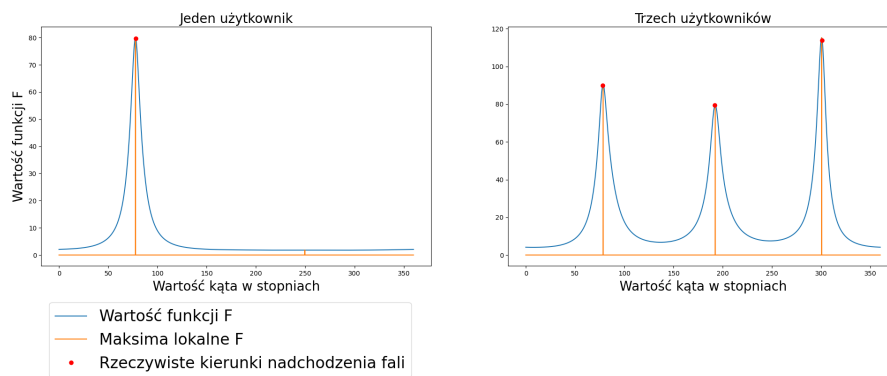
Rysunek 5.7: $\text{SNR} = -10\text{dB}$, $\Delta d = 2\text{cm}$



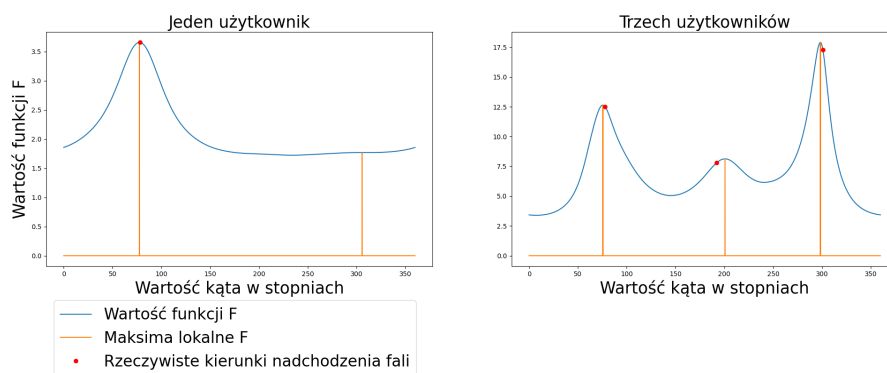
Rysunek 5.8: $\text{SNR} = 10\text{dB}$, $\Delta d = 2\text{cm}$ z pogłosem



Rysunek 5.9: $\text{SNR} = 10\text{dB}$, $\Delta d = 20\text{cm}$



Rysunek 5.10: $\text{SNR} = -10\text{dB}$, $\Delta d = 20\text{cm}$



Rysunek 5.11: $\text{SNR} = 10\text{dB}$, $\Delta d = 20\text{cm}$ z pogłosem

Wyniki dotyczące wpływu szumu nie zaskakują- algorytm działa mniej dokładnie dla niskiego SNR. To samo dotyczy pogłosu, użyty algorytm nie jest odporny na tego typu zakłócenia. Pojawiające się odbite fale wewnątrz pokoju czynią maksima mniej wyeksponowanymi. Najbardziej interesująca obserwacja zdaniem autora pracy

to wpływ odległości między mikrofonami. Większa odległość powoduje większą kierunkowość co przekłada się na dokładniejszą estymację. Dla rozłożenia $\Delta d = 20cm$ i szumie $SNR = -10dB$ udaje się dokładnie wyestymować wszystkie trzy kierunki. Dla tego samego stosunku sygnału do szumu i $\Delta d = 2cm$ algorytm estymuje tylko dwa kierunki i to bardzo niedokładnie. Do dalszych testów wybrano jednak system ze stosunkiem sygnału do szumu na poziomie 10-20dB a mówcy są dość daleko od siebie. Zbyt duża kierunkowość powoduje duże problemy z zastosowaniem filtra LCMV i nawet bardzo dokładna estymacja nie rekompensuje tych strat. Doświadczenia empiryczne pokazały, że cały system działa lepiej dla $\Delta d = 2cm$.

5.4 Testy całego systemu

Testy całego systemu sprawdzają kilka różnych aspektów projektu inżynierskiego. Są to:

- Stosunek sygnału do szumu(SNR)
- Ocena działania systemu dla jednego użytkownika
- Ocena działania systemu dla wielu użytkowników
- Ewaluacja wpływu poruszania się źródła na działanie algorytmu
- Sprawdzenie wpływu pogłosu na jakość przetwarzania

5.4.1 SNR

Obliczenie SNR dla systemu jest obliczane zarówno dla systemu z pojedynczym użytkownikiem jak i z trzema użytkownikami. Aby obliczyć SNR zakłada się, że $\mathbf{g} = \{1, \dots, 1\}$ dla każdego indeksu czasowego. Oznacza to wyłączenie AGC. Z włączonym AGC następowałyby zmiany lokalnej amplitudy sygnału uniemożliwiając sensowny pomiar SNR. Wszystkie pomiary SNR podaje się w skali logarytmicznej.

Zastosowano następującą normę pomiaru poprawy SNR:

$$\Delta\text{SNR} = \text{SNR}_{\text{output}} - \text{SNR}_{\text{input}} \quad (5.1)$$

Gdzie SNR na wejściu $\text{SNR}_{\text{input}}$ jest znane a mając do dyspozycji niezaszumiony sygnał referencyjny $x_{\text{ref}}(n)$ i sygnał zaszumiony $x_{\text{noisy}}(n)$ można także obliczyć stosunek sygnału do szumu dla N próbek wyjściowych sygnału jako:

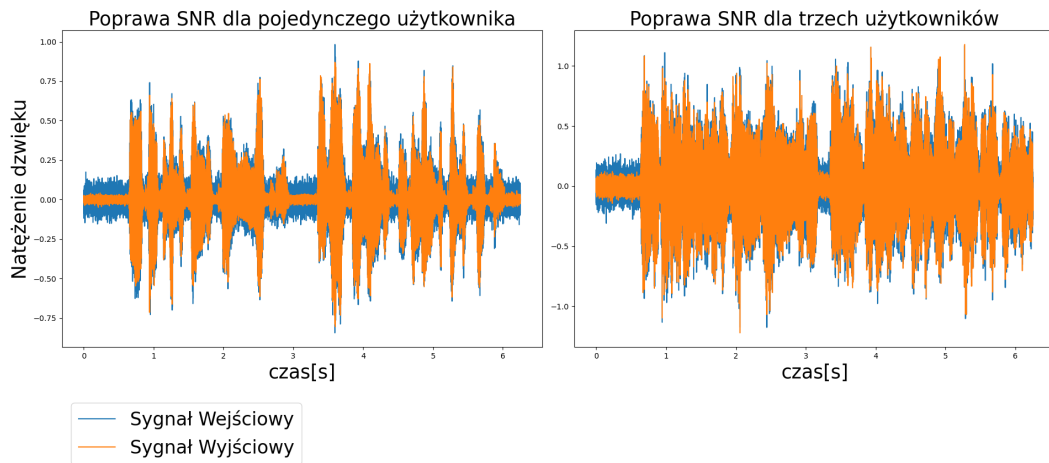
$$\text{SNR} = 10 \log \sum_{n=1}^N \left(\frac{x_{\text{ref}}(n)}{x_{\text{ref}}(n) - x_{\text{noisy}}(n)} \right)^2 \quad (5.2)$$

Taki pomysł obliczania SNR zaczerpnięto luźno z [20]. Z tą różnicą, że autor tego artykułu obliczał SNR w dziedzinie STFT.

Uzyskano następujące wyniki:

- Pojedynczy użytkownik: $\Delta\text{SNR} \approx 12\text{dB}$
- Trzech użytkowników: $\Delta\text{SNR} \approx 6\text{dB}$

Wykresy przedstawiające sygnały przed i po odszumaniu przedstawiono na 5.12



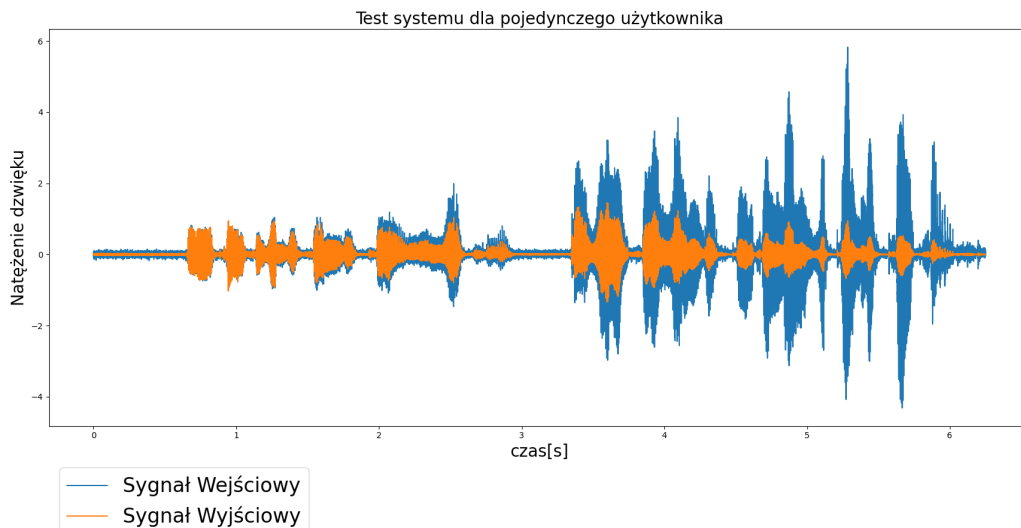
Rysunek 5.12: Poprawa SNR

5.4.2 Działanie algorytmu dla pojedynczego użytkownika

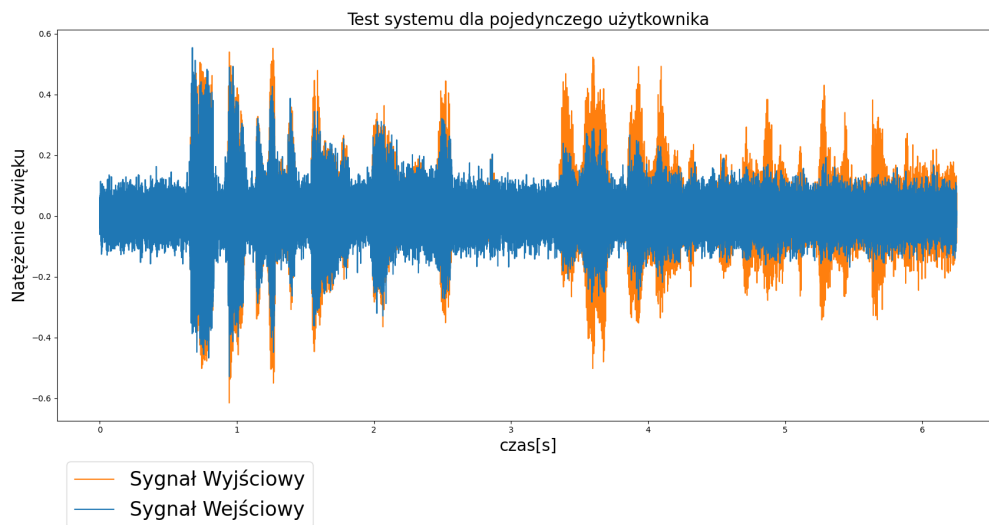
Ocenę algorytmu dla pojedynczego użytkownika przeprowadzono w warunkach gdzie użytkownik jest ustawiony na $\theta_0 = 78^\circ$. Nagrany sygnał ma długość 50000 próbek co przekłada się na nieco ponad 6 sekund. W celu oceny jakości działania sprawdza się dwa warianty zmian sygnału w czasie:

- Dziesięciokrotny wykładniczy wzrost amplitudy w czasie trwania sygnału
- Dziesięciokrotny wykładniczy spadek amplitudy w czasie trwania sygnału

W obu przypadkach początkowy SNR wynosi 10dB. Wyniki eksperymentu przedstawiono na wykresach 5.13 i 5.14



Rysunek 5.13: Wzrost amplitudy sygnału dla pojedynczego użytkownika



Rysunek 5.14: Spadek amplitudy sygnału dla pojedynczego użytkownika

Wyniki wymagają krótkiego komentarza. Widoczne jest, że w obu przypadkach mimo, że amplituda sygnału wejściowego silnie zmienia się w czasie, sygnał wyjściowy utrzymuje względnie stały poziom obwiedni. Szczególnie interesujący jest fakt, że w przypadku słabnącego sygnału wejściowego wzmocnienie sygnału powoduje jego wyjście ponad poziom szumu.

5.4.3 Działanie algorytmu dla wielu użytkowników

W przypadku wielu użytkowników nie jest tak łatwo zaprezentować działanie algorytmu za pomocą wykresu wyjściowego. W celu oceny działania algorytmu konieczne będzie obejrzenie zależności wewnętrznych pomiędzy sygnałami Φ_{LT} , G' , \hat{G} . Zmiany parametrów zostaną pokazane zarówno w funkcji czasu jak i w funkcji kąta w pojedynczej ramce. Dla przejrzystości we wszystkich testowanych przypadkach będzie to ta sama ramka.

Scenariusz testu jest następujący. Natężenie dźwięku pochodzącego od pierwszych dwóch mówców(mężczyźni) jest podobne podczas gdy natężenie trzeciej osoby(kobieta) jest celowo podbite 2 razy. Celem jest wygenerowanie dźwięku, w którym poziom głośności mówców zostanie wyrównany. SNR w tym teście wynosi 20dB aby słabe źródła nie znalazły się zupełnie pod szumem.

Na wykresach 5.15-5.17 przedstawiono wyniki. Przebieg czasowy jest o tyle interesujący, że wyraźnie po momentach ciszy i na początku następuje podbicie sygnału. Dzieje się tak, ponieważ algorytm ustawia dużą głośność dla cichych źródeł i gdy zwiększają one swoją głośność stopniowo ją dostosowuje. Skutkuje to pewnym czasem ustalania się docelowej wartości.

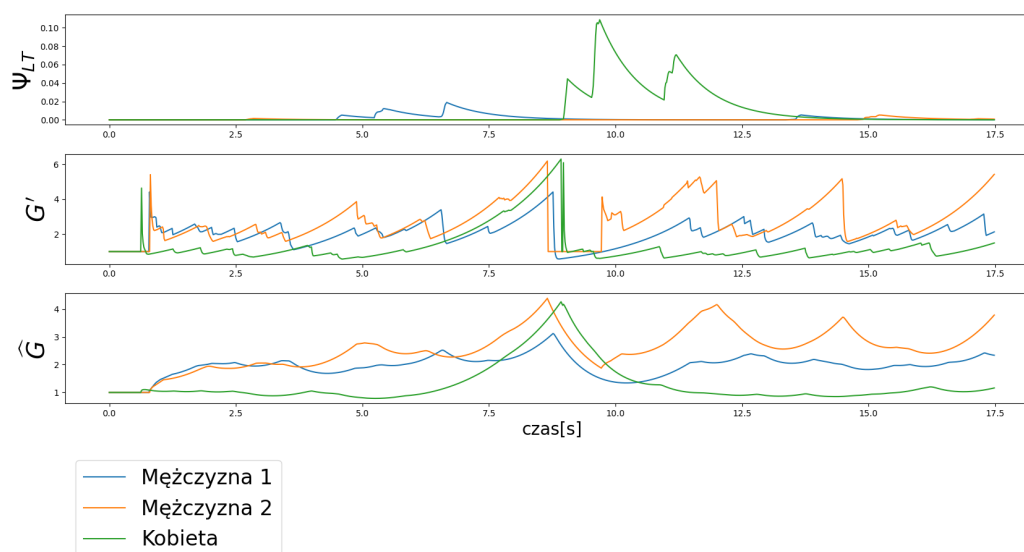
Na wykresie 5.16 widoczne jest, że funkcja głośności ma dla źródła głośniejszego wartość mniejszą niż dla cichszego co jest celem działania algorytmu.

Wykres 5.17 świetnie przedstawia ideę zastosowania okien w dziedzinie kątów. Skutkuje to powstaniem charakterystycznych "górek". Mały błąd estymacji nie spowo-

duje drastycznej zmiany w systemie. Warto też zauważyć góry występujące poza kierunkami nadchodzenia fali. Biorą się one z grubszych błędów estymacji.

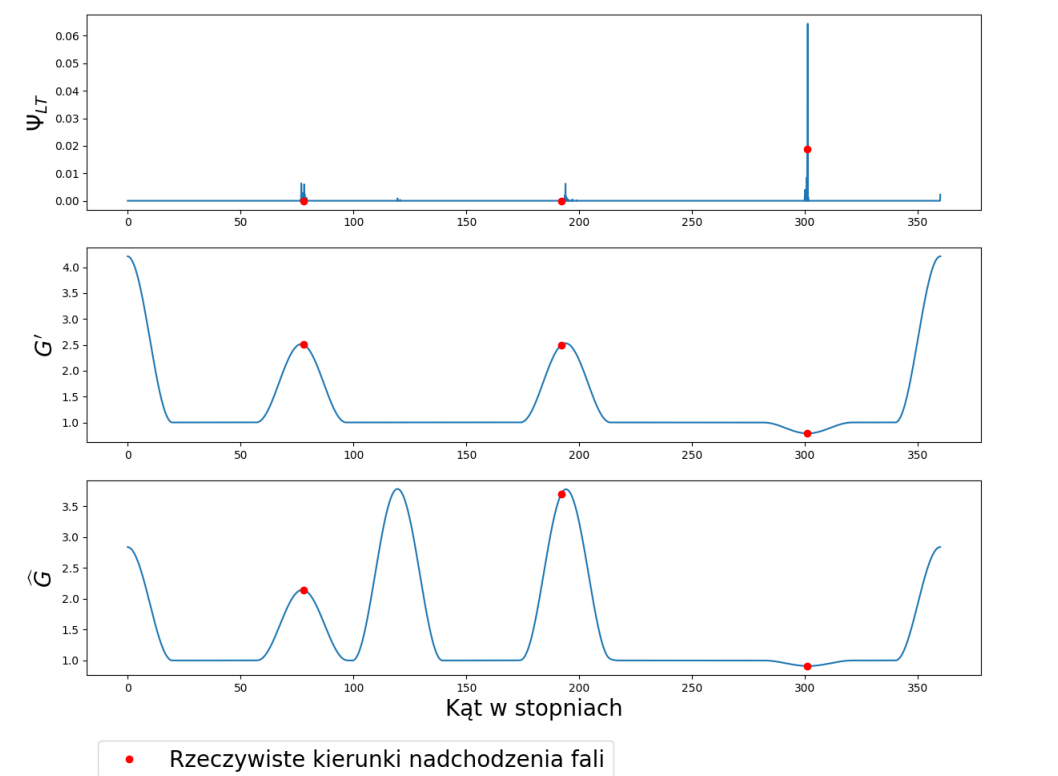


Rysunek 5.15: Przebieg czasowy dla trzech mówców

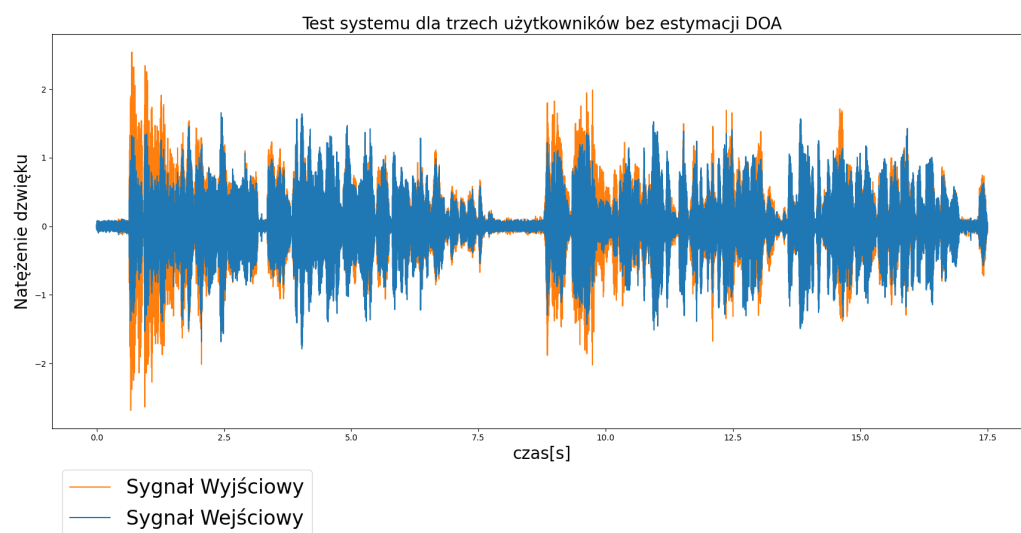


Rysunek 5.16: Przebieg czasowy wybranych parametrów dla trzech mówców dla kierunków nadchodzenia fali

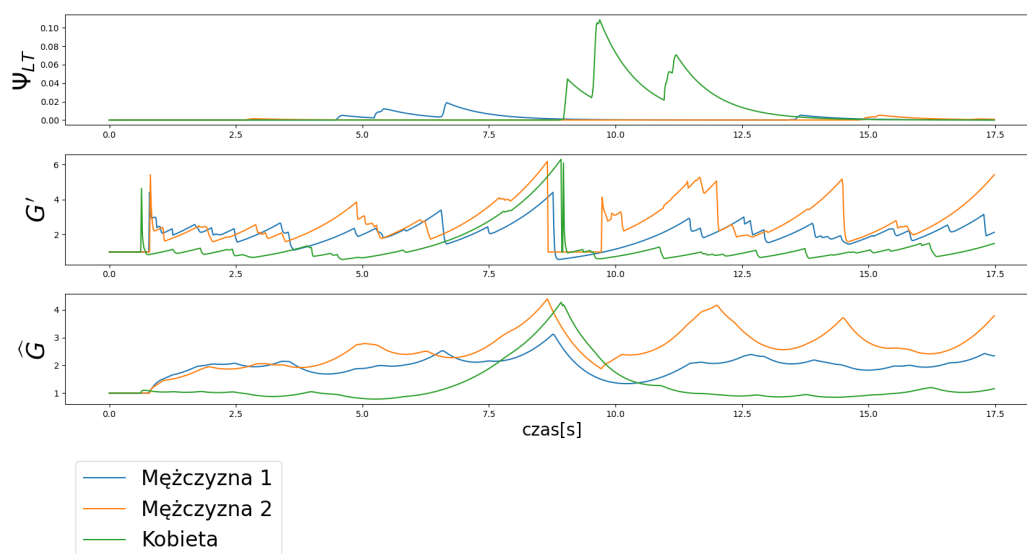
Te same trzy wykresy przedstawiono w wariacie bez estymacji DOA(5.18-5.20). Oznacza to, że do algorytmu na twardo podawane są prawdziwe kierunki nadchodzenia fali. Widocznym jest, że zanikają "góry" dla kierunków poza DOA.



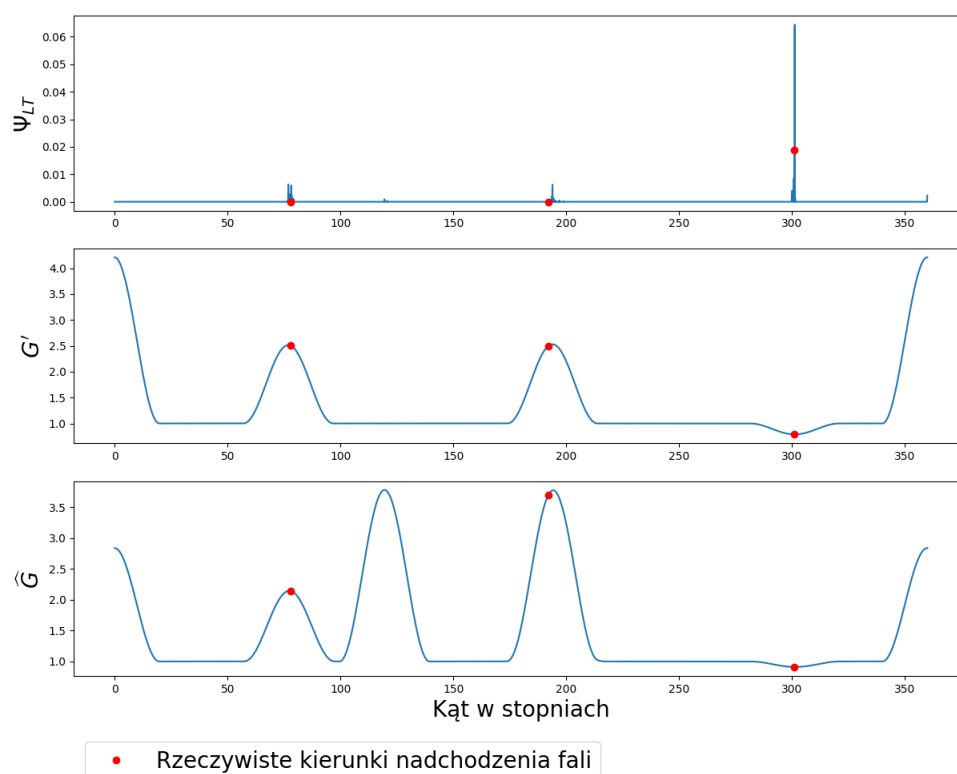
Rysunek 5.17: Wybrane parametry jako funkcja kąta w pojedynczej ramce



Rysunek 5.18: Przebieg czasowy dla trzech mówców bez estymacji DOA



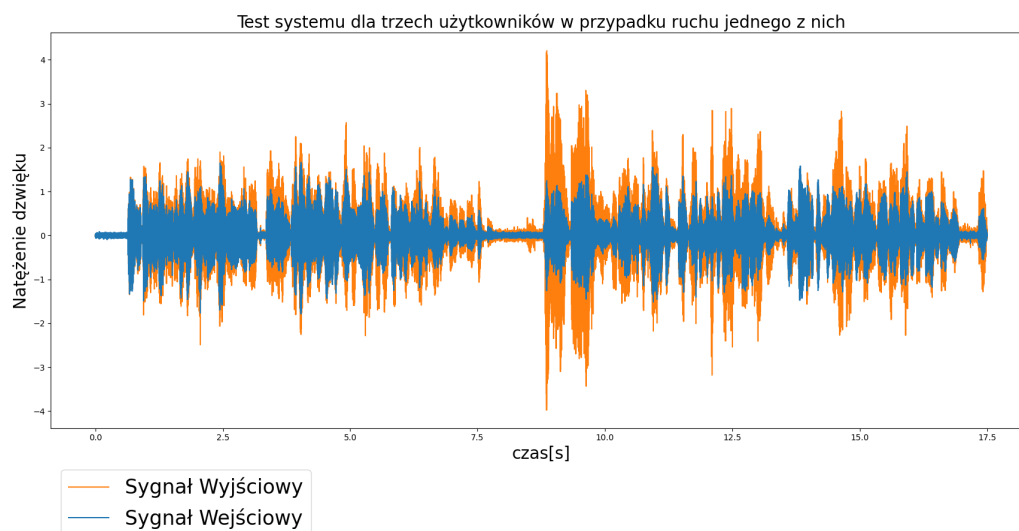
Rysunek 5.19: Przebieg czasowy wybranych parametrów dla trzech mówców dla kierunków nadchodzenia fali bez estymacji DOA



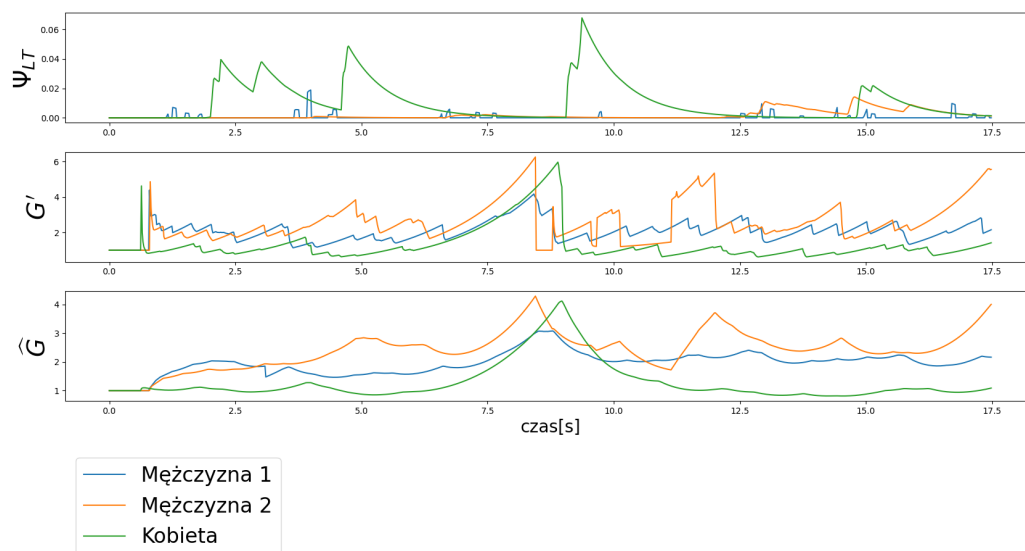
Rysunek 5.20: Wybrane parametry jako funkcja kąta w pojedynczej ramce dla przypadku bez estymacji DOA

5.4.4 Działanie algorytmu w przypadku ruchu mówcy

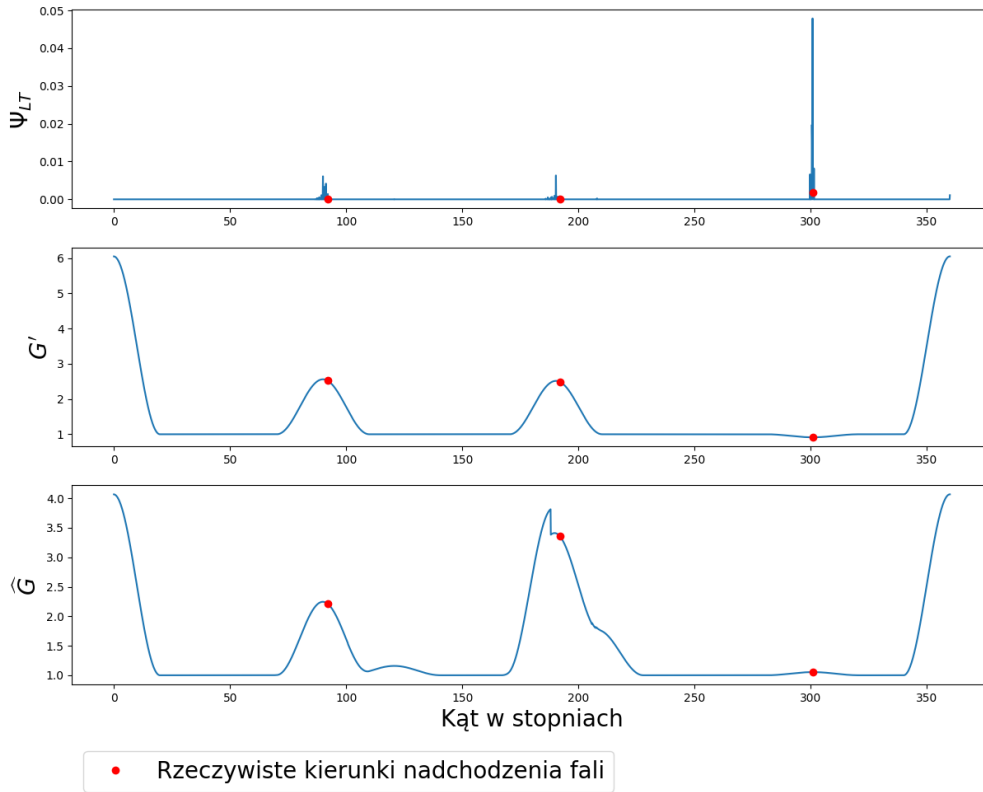
Ten test ma za zadanie sprawdzić czy system AGC będzie działał gdy mówca zacznie się poruszać z niewielką prędkością kątową. Wszystkie parametry są takie jak dla poprzednich testów ale podczas trwania sygnału jeden z mówców- Mężczyzna 1 przemieści się z 78° na 99° . Na wykresach 5.21-5.23 pokazano przebiegi analogiczne do tych z poprzednich przypadków.



Rysunek 5.21: Przebieg czasowy dla trzech mówców w przypadku ruchu jednego z nich



Rysunek 5.22: Przebieg czasowy wybranych parametrów dla trzech mówców dla kierunków nadchodzenia fali w przypadku ruchu jednego z nich



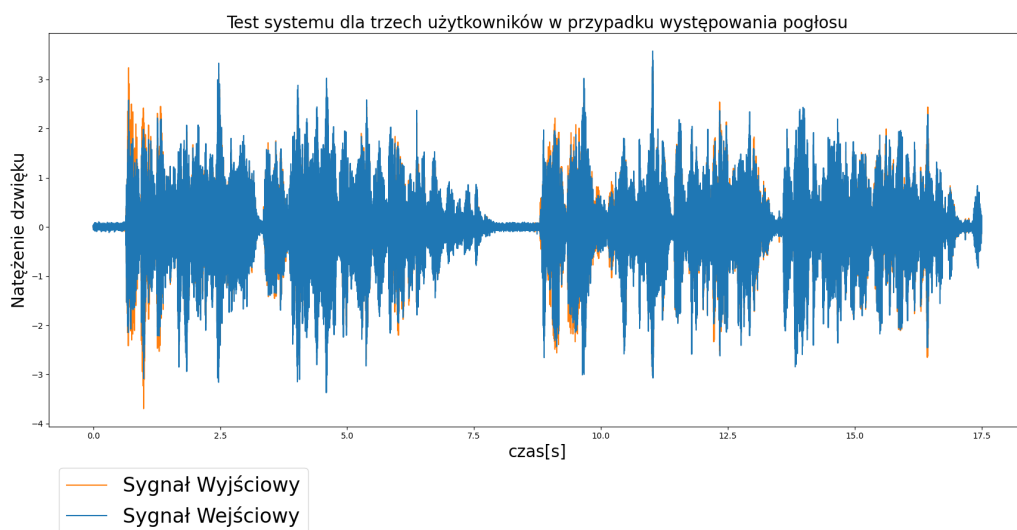
Rysunek 5.23: Wybrane parametry jako funkcja kąta w pojedynczej ramce w przypadku ruchu jednego z nich

Zauważalną różnicą w stosunku do przypadku, w którym wszyscy użytkownicy byli statyczni jest różnica w położeniu pierwszej "górkę" funkcji G' . Jest ona po prostu przesunięta. Stało się tak, ponieważ użytkownik zmienił już swoją pozycję w momencie obserwacji. Ruch nie powoduje pogorszenia się działania algorytmu.

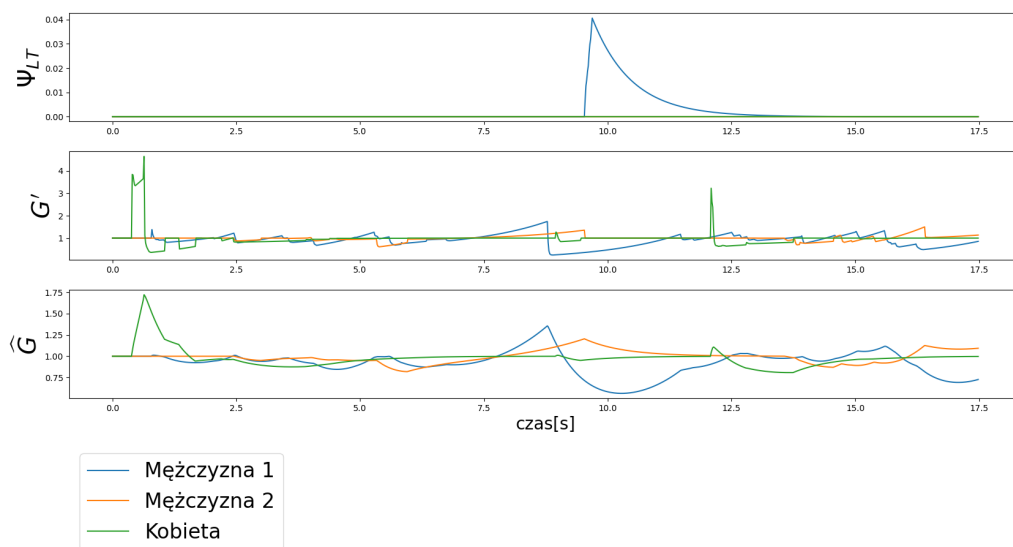
5.4.5 Odporność systemu na pogłos

Ostatnim testowanym wariantem jest umieszczenie wszystkich trzech mówców w pokoju w celu ewaluacji działania algorytmu w modelu, w którym występuje pogłos. Otrzymane wyniki porównane zostaną z wynikami z wynikami eksperymentów bez pogłosu. Mówcy ponownie znajdują się na kątach 78° , 192° i 301° . Wyniki znajdują się na wykresach 5.24-5.26.

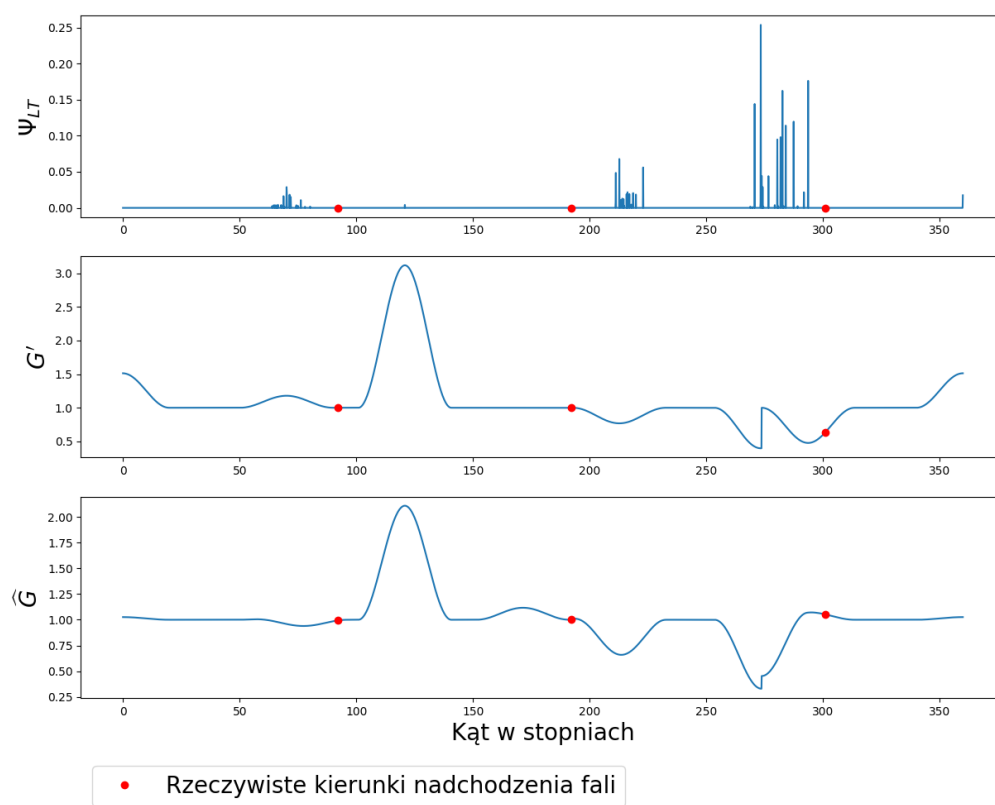
Na wykresie 5.24 obserwowany jest brak działania systemu AGC. Jak widać na wykresach następuje na tyle duży błąd estymacji kierunku nadchodzenia fali, że maksima funkcji \hat{G} są oddalone od kierunków nadchodzenia fali. Z tego powodu do równania (3.11) brane są błędne kierunki. Oznacza to, że pogłos uniemożliwia praktyczne stosowanie algorytmu MUSIC do opisanego w tej pracy systemu. W związku z tym algorytm wymaga ulepszenia mechanizmu lokalizacyjnego poprzez na przykład zwiększenie czasu uśredniania, uśrednienie funkcji (3.7) w innym przedziale częstotliwości lub wybór innej metody.



Rysunek 5.24: Przebieg czasowy dla trzech mówców w przypadku występowania pogłosu



Rysunek 5.25: Przebieg czasowy wybranych parametrów dla trzech mówców dla kierunków nadchodzenia fali w przypadku występowania pogłosu



Rysunek 5.26: Wybrane parametry jako funkcja kąta w pojedynczej ramce w przypadku występowania pogłosu

Rozdział 6

Podsumowanie

Celem pracy inżynierskiej było napisanie algorytmu automatycznej kontroli głośności dla macierzy mikrofonowej z jednoczesnym tłumieniem tła akustycznego. Temat udało się zrealizować. Sposób działania algorytmu został zaczerpnięty z literatury, w dużej mierze z [5], [4] i [8]. System napisano w języku Python. Otrzymany algorytm przetwarza nagrania z macierzy mikrofonowej. Działanie algorytmu spełnia postawione założenia - steruje głośnością i tłumi zakłócenie jakim jest biały szum. Zdecydowano się zrealizować żądany system za pomocą filtru LCMV i algorytmu MUSIC.

Algorytm poddano licznym testom. Przetestowano wpływ czynników takich jak poziom szumu, ilość mówców, zmienność położenia mówców czy geometrię pokoju i mikrofonu na działanie zrealizowanego rozwiązania. Dokonano analizy otrzymanych wyników i wyciągnięto wnioski, które zostały opisane w rozdziale 5. Wnioski można podsumować w następujący sposób. System dobrze radzi sobie z minimalizacją SNR i z kontrolą głośności. Działanie dla pojedynczego mówcy jest lepsze niż dla wielu mówców. System działa nawet w przypadku sygnału, którego moc jest poniżej szumu. Należy jednak zwrócić uwagę, że założony model nie jest odporny na pogłos. Zbyt duża wartość tego parametru w praktyce uniemożliwia kontrolę głośności.

W planach autora pracy jest pogłębianie wiedzy na temat AGC i ulepszanie systemu. Plany te zakładają ulepszenie algorytmu estymacji DOA tak aby kierunki nadchodzenia fali były wykrywane dla sygnału składającego się ze źródeł położonych bliżej i dla mniejszej wartości stosunku sygnału do szumu. Ponadto kolejnym naturalnym krokiem rozwoju algorytmu jest wprowadzenie odporności na pogłos. W planach jest także próba szybszej implementacji sprzętowej i uruchomienie algorytmu w czasie rzeczywistym.

Bibliografia

- [1] *How Conventional AGC (Automatic Gain Control) works in receiver chain and its important parameters.* - Techplayon. URL: <https://www.techplayon.com/conventional-agc-automatic-gain-control-works-receiver-chain-explain-important-parameters/>.
- [2] *A groundbreaking approach to gain control that cuts conference call annoyance.* URL: <https://www.nureva.com/blog/conferencing/a-groundbreaking-approach-to-gain-control-that-cuts-conference-call-annoyance>.
- [3] Fitzgerald J Archibald. “White Paper Software Implementation of Automatic Gain Controller for Speech Signal”. W: (2008). URL: www.ti.com.
- [4] Oliver Thiergart i Emanuel A.P. Habets. “An informed LCMV filter based on multiple instantaneous direction-of-arrival estimates”. W: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* (paź. 2013), s. 659–663. ISSN: 15206149. DOI: 10.1109/ICASSP.2013.6637730.
- [5] Sebastian Braun, Oliver Thiergart i Emanuel A.P. Habets. “Automatic spatial gain control for an informed spatial filter”. W: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* (2014), s. 830–834. ISSN: 15206149. DOI: 10.1109/ICASSP.2014.6853713.
- [6] Jacob Benesty, Jingdong Chen i Yiteng Huang. “Microphone array signal processing”. W: *Springer Topics in Signal Processing* 1 (2008), s. 1–240. ISSN: 18662617. DOI: 10.1007/978-3-540-78612-2.
- [7] *NumPy Reference — NumPy v1.21 Manual*. URL: <https://numpy.org/doc/stable/reference/index.html>.
- [8] Ralph O. Schmidt. “MULTIPLE EMITTER LOCATION AND SIGNAL PARAMETER ESTIMATION.” W: *IEEE Transactions on Antennas and Propagation* AP-34 (3 1986), s. 276–280. ISSN: 0018926X. DOI: 10.1109/TAP.1986.1143830.
- [9] *Direction of Arrival*. URL: <https://www.comm.utoronto.ca/~rsadve/Notes/DOA.pdf>.
- [10] Tara Salman i in. “Estimating the Number of Sources: An Efficient Maximization Approach”. W: ().
- [11] Otis Lamont Frost. “An Algorithm For Linearly Constrained Adaptive Array Processing”. W: *PROCEEDINGS OF THE IEEE* 60 (8 1972).

- [12] Itu Radiocommunication Bureau. “Algorithms to measure audio programme loudness and true-peak audio level BS Series Broadcasting service (sound)”. W: (2015). URL: <http://www.itu.int/ITU-R/go/patents/en>.
- [13] *API Reference — Matplotlib 3.5.1 documentation*. URL: <https://matplotlib.org/stable/api/index>.
- [14] *rir-generator · PyPI*. URL: <https://pypi.org/project/rir-generator/>.
- [15] *SciPy documentation — SciPy v1.7.1 Manual*. URL: <https://docs.scipy.org/doc/scipy/reference/>.
- [16] *UML Association vs Aggregation vs Composition*. URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-aggregation-vs-composition/>.
- [17] Heinrich Kuttruff. “Room Acoustics, Fourth edition”. W: ().
- [18] *scipy.signal.windows.hann — SciPy v1.7.1 Manual*. URL: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.windows.hann.html>.
- [19] I.A. McCowan. *Robust Speech Recognition Using Microphone Arrays*. Queensland University of Technology, Brisbane, 2001.
- [20] Tuomas Virtanen. “Monaural Sound Source Separation by Nonnegative Matrix Factorization With Temporal Continuity and Sparseness Criteria”. W: (2006). DOI: 10.1109/TASL.2006.885253.