

**AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE**



AGH

Wydział Informatyki, Elektroniki i Telekomunikacji
Katedra Elektroniki

**PRACA DYPLOMOWA
INŻYNIERSKA**

Temat: Adaptacyjny system redukcji szumu akustycznego

Adaptive acoustic noise reduction system

Imię i nazwisko: Paweł Czurka
Kierunek studiów: Inżynieria Akustyczna

Opiekun pracy: dr inż. Jakub Gałka

Oświadczam, świadomy odpowiedzialności karnej za poświadczenie nieprawdy, że niniejszą pracę dyplomową wykonałem osobiście i samodzielnie i że nie korzystałem ze źródeł innych niż wymienione w pracy.

.....

*Serdecznie dziękuję promotorowi tej pracy,
którym był dr inż. Jakub Gałka,
za poświęcony czas i zaangażowanie w śledzenie postępów projektu.*

Streszczenie

Praca przedstawia sposób modyfikacji wektorów cech mowy, który ma na celu zniwelowanie wpływu zakłóceń na działanie systemu automatycznej weryfikacji mówców.

Pierwszym etapem jest zbudowanie weryfikatora mówców. Wybrano jedną z najbardziej rozpowszechnionych metod weryfikacji, czyli opartą o Modele Mikstur Gaussowskich. Taki system, przy odpowiednio przyjętych parametrach cechuje bardzo wysoka skuteczność.

Nagrania mowy zaczerpnięto z bazy języka polskiego o nazwie „Corpora”. Jest to baza wielu krótkich wypowiedzi wypowiedziana przez 45 różnych mówców.

Modyfikacja wektorów cech mowy odbyła się przy użyciu algorytmu *Cepstral Mean Subtraction* (CMS). Modyfikowano przy jego użyciu macierze MFCC kolejnych wypowiedzi zakłócone różnymi typami szumu w kilku wariantach stosunku sygnału do szumu (sygnału mowy do sygnału zakłócającego).

Etapem weryfikującym wpływ modyfikacji na działanie systemu weryfikacji było zbadanie wyników w sytuacji kiedy dane wejściowe były i nie były zmodyfikowane operacją CMS.

Spis treści

1. Wstęp	6
2. Cel i zakres pracy.....	6
3. Wstęp teoretyczny.....	7
3.1. CMS.....	7
3.2. Parametryzacja cech mowy – MFCC.....	7
3.3. GMM (Gaussian Mixture Model).....	8
3.4. Expectation Maximization.....	8
3.5. Niezależny od tekstu weryfikator mówców (GMM-UBM).....	9
3.6. False match rate, false non-match rate.....	9
3.7. Equal error rate.....	10
4. Przebieg pracy.....	11
4.1. Zgromadzenie próbek.....	11
4.2. Przygotowanie próbek.....	11
4.3. Parametryzacja MFCC (Voicebox).....	12
4.4. Implementacja CMS.....	13
4.5. Weryfikator mówców.....	14
4.6. Test działania weryfikatora mówców.....	16
4.7. Przygotowanie modeli GMM oraz macierzy testowych dla nagrań zakłóconych.....	17
4.8. Wyznaczenie wpływu algorytmu CMS na działanie weryfikatora mówców.....	18
4.8.1. Brak zakłóceń.....	18
4.8.2. Zakłócenie różnymi typami hałasu, SNR = 30 dB.....	19
4.8.3. Zakłócenia różnymi typami hałasu, SNR = 20 dB.....	20
4.8.4. Zakłócenia różnymi typami hałasu, SNR = 10 dB.....	21
4.8.5. Zakłócenia różnymi typami hałasu, SNR = 5 dB.....	22
4.8.6. Zakłócenia różnymi typami hałasu, SNR = 0 dB.....	24
4.8.7. Zakłócenia różnymi typami hałasu, SNR = -5 dB.....	24
4.8.8. Liniowe zniekształcenie kanału transmisyjnego.....	25
4.8.9. Zestawienie wszystkich sytuacji.....	26
5. Interpretacja wyników.....	28
6. Zakończenie.....	28

1. Wstęp

Systemy redukcji szumu na ogół kojarzone są z urządzeniami poprawiającymi właściwości sygnału audio lub wideo, którego najważniejszym kryterium jest subiektywna ocena odbiorcy. Między innymi, czy efekt działania pozbawia sygnał składowej szumowej (poprawiony zostaje stosunek sygnału do szumu) i czy przy tym system nie wpływa na charakterystykę częstotliwościową sygnału wyjściowego? Natomiast w przetwarzaniu sygnału mowy algorytmy redukcji szumu częściej stosuje się jako elementy systemów weryfikacji, rozpoznawania mówców itp. w celu poprawienia skuteczności działania tych systemów.

W tej pracy zostanie przetestowany algorytm CMS (*Cepstral Mean Subtraction*), który jest jednym z powszechnie stosowanych, prostych i efektywnych rozwiązań zapewniających poprawę wyników systemów ASR i podobnych.

Wszystkie obliczenia wykonane zostały w środowisku MATLAB R2010a.

2. Cel i zakres pracy

Celem pracy jest przedstawienie wpływu zastosowania algorytmu CMS na wyniki systemu weryfikacji mówców opartego o modele mikstur gaussowskich obliczone na podstawie cech MFCC dla różnych wariantów zaszumienia nagrań testowych.

Pierwszym etapem będzie zgromadzenie bazy nagrań mowy, oraz nagrań różnych typów zakłóceń. Na ich podstawie stworzone będą bazy próbek o różnym poziomie zakłóceń. Wszystkie nagrania otrzymają swoją reprezentację w postaci macierzy MFCC

Kolejnym etap to stworzenie niezależnego od tekstu weryfikatora mówców opartego o GMM (*Gaussian Mixture Models*). Do tego użyte zostaną algorytmy biblioteki *Voicebox*.

Później, zaimplementowany zostanie algorytm CMS jako funkcja.

Ostatecznie, nastąpi przetestowanie działania systemu, w tym celu utworzone będą programy tworzące kombinacje próbek, aby móc wyznaczyć skuteczność algorytmu.

3. Wstęp teoretyczny

3.1. CMS

W dziedzinie spektrum sygnał można przedstawić jako sumę składowej pochodzącej od zakłócenia (c) oraz składowej mowy (s_t), indeks t informuje o zależności danego składnika od czasu:

$$y_t = c + s_t \quad (1)$$

Przyjmując niezmiennosc zakłócenia w czasie można od cepstrum odjąć średnią wartość za cały czas trwania nagrania.

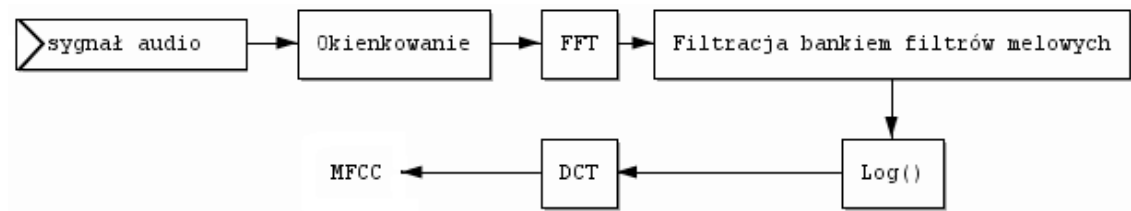
Otrzymujemy zatem nowy przebieg cepstrum z_t :

$$z_t = y_t - \bar{y}_t \quad (2)$$

3.2. Parametryzacja cech mowy – MFCC

Parametryzacja cech mowy ma na celu uzyskanie reprezentacji sygnału mowy znacznie ograniczoną (w porównaniu do sygnału audio) ilością danych.

Jednym ze sposobów parametryzacji są wektory MFCC (*Mel-frequency cepstral coefficients*). Podstawowy algorytm przedstawiony jest na rysunku 1.



Rysunek 1: Schemat blokowy sposobu parametryzacji sygnału audio przy użyciu algorytmu MFCC

3.3. GMM (Gaussian Mixture Model)

Model mikstur gaussowskich jest funkcją gęstości prawdopodobieństwa parametrów reprezentowana jako suma gęstości rozkładów składających się na miksturę.

$$p(x|\lambda) = \sum_{i=1}^M w_i g(x|\mu_i, \Sigma_i) \quad (3)$$

gdzie:

- $\lambda = \{w_i, \mu_i, \Sigma_i\}, i = 1, \dots, M$
- M oznacza ilość komponentów mikstury
- x jest N -wymiarowym wektorem cech,
- $w_i, i = 1, \dots, M$ reprezentuje wagi kolejnych komponentów mikstury
- $g(x|\mu_i, \Sigma_i)$ to rozkład gęstości komponentów ($i = 1, \dots, M$)

Każdy komponent jest N -wymiarową funkcją Gaussa postaci:

$$g(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{N/2} |\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_i)' \Sigma_i^{-1}(x-\mu_i)\right\} \quad (4)$$

Σ_i oraz μ_i oznaczają odpowiednio macierz kowariancji, oraz wektor średnich danego komponentu.

Model GMM opisany jest za pomocą wektorów wartości średnich, macierzy kowariancji oraz wag.

3.4. Expectation Maximization

Algorytm *Expectation Maximization* (EM) [3] w omawianym przypadku ma na celu takie dobranie parametrów modelu GMM, który dla podanych danych treningowych maksymalizuje prawdopodobieństwo.

EM jest procedurą iteracyjną. Mając dany model początkowy $\lambda_{j=1}$, w każdej

kolejnej iteracji reestymowane są parametry tego modelu w taki sposób, że

$p(X|\lambda_{j+1}) \geq p(X|\lambda_j)$, gdzie j to numer kolejnych iteracji. Algorytm kończy działanie kiedy osiągnie próg zbieżności.

3.5. Niezależny od tekstu weryfikator mówców (GMM-UBM)

Wyznaczenie parametrów modelu λ_i mówcy odbywa się zgodnie z opisaną w rozdziałach 3.3 oraz 3.4 procedurą. Zbiorem obserwacji są wektory cech danego mówcy (np. MFCC). Dodatkowo tworzony jest model GMM-UBM na podstawie wektorów cech pochodzących od wielu mówców. Zadaniem systemu weryfikacji jest wyznaczenie pewnej miary, na podstawie której nastąpi decyzja potwierdzająca lub odrzucająca tożsamość mówcy. Definiuje się ją jako *likelihood ratio*:

$$\alpha(X) = \frac{p(X|\lambda_i)}{p(X|\lambda_{UBM})} \quad (5)$$

Jeśli:

$\alpha(X) \geq \theta$ - nastąpi potwierdzenie tożsamości,

$\alpha(X) < \theta$ - nastąpi odrzucenie tożsamości

Często parametr weryfikacji określa się za pomocą *log-likelihood ratio*:

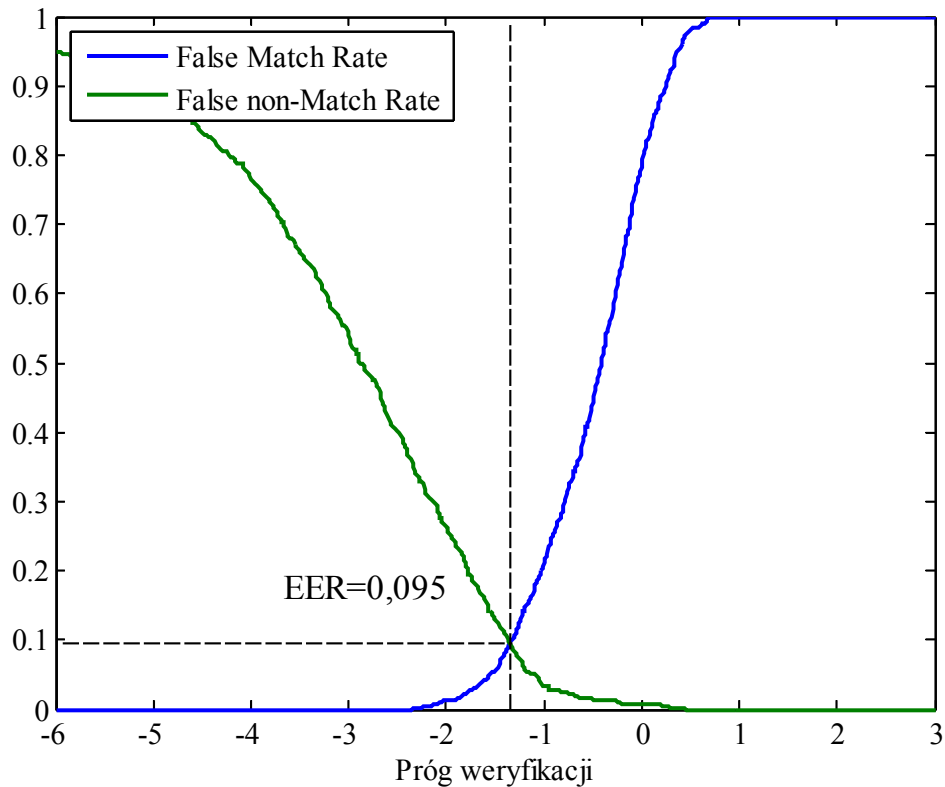
$$\Lambda(X) = \frac{\log p(X|\lambda_i) - \log p(X|\lambda_{UBM})}{T} \quad (6)$$

3.6. False match rate, false non-match rate

False Match oznacza w kontekście weryfikacji przypadek, kiedy system zatwierdza tożsamość, podczas gdy nie ma zgodności. *False non-match* występuje gdy występuje sytuacja odwrotna (brak potwierdzenia podczas gdy model i test powinny być ze sobą skojarzone). *False match rate* (FMR) oraz *False non-match rate* (FNMR), to wartości z zakresu (0:1) odnoszące wartość FNM oraz FM do ilości przeprowadzonych weryfikacji.

3.7. Equal error rate

Jeśli uzależnimy wartości FNMR i FMR od przyjętej wartości progu weryfikacji (punktu pracy systemu) wartością EER jest taka wartość progu przy której FNMR i FMR są sobie równe. *Equal Error Rate* służy do miarodajnego opisu skuteczności systemu i jest łatwy w interpretacji.



Rysunek 2: Przykładowy wykres wraz ze zobrazowaniem wartości EER jako punktu wspólnego krzywych FMR oraz FNMR w zależności od przyjętego progu weryfikacji

4. Przebieg pracy

4.1. Zgromadzenie próbek

Wykorzystana baza nagrań mowy to „Corpora”. Jest to zbiór krótkich nagrań pochodzących od 45 różnych mówców posługujących się językiem polskim. Te nagrania posłużyły za bazę do kolejnych operacji.

Nagrania, które posłużyły za zakłócenia zaczerpnięte zostały z internetowej bazy freesound.org. Były to:

- hałas biurowy
- hałas uliczny
- gwar przyjęcia

Stworzono również nagranie stacjonarnego szumu gaussowskiego.

4.2. Przygotowanie próbek

Dla każdego mówcy z bazy „Corpora” przygotowanych zostało 16 próbek testowych o czasie trwania ok. 7 sekund oraz po jednym nagraniu testowym trwającym ok. 60 sekund. Próbkę w bazie „Corpora” są krótsze, niż te, które potrzebne były do tego zadania. Należało, więc tworzyć nowe nagrania poprzez „wydłużanie” wektora nagrania, danymi z kolejnych nagrań, oraz zapisywanie ich używając funkcji *wavread()* dostępnej w bibliotekach środowiska Matlab.

Na podstawie każdego nagrania testowego stworzono 24 warianty próbek zakłóconych szumem (4 rodzaje szumu oraz 6 poziomów SNR mierzonych jako stosunek RMS próbki mowy i próbki szumu). Wartości SNR: 30 dB, 20 dB, 10 dB, 5 dB, 0dB, -5 dB.

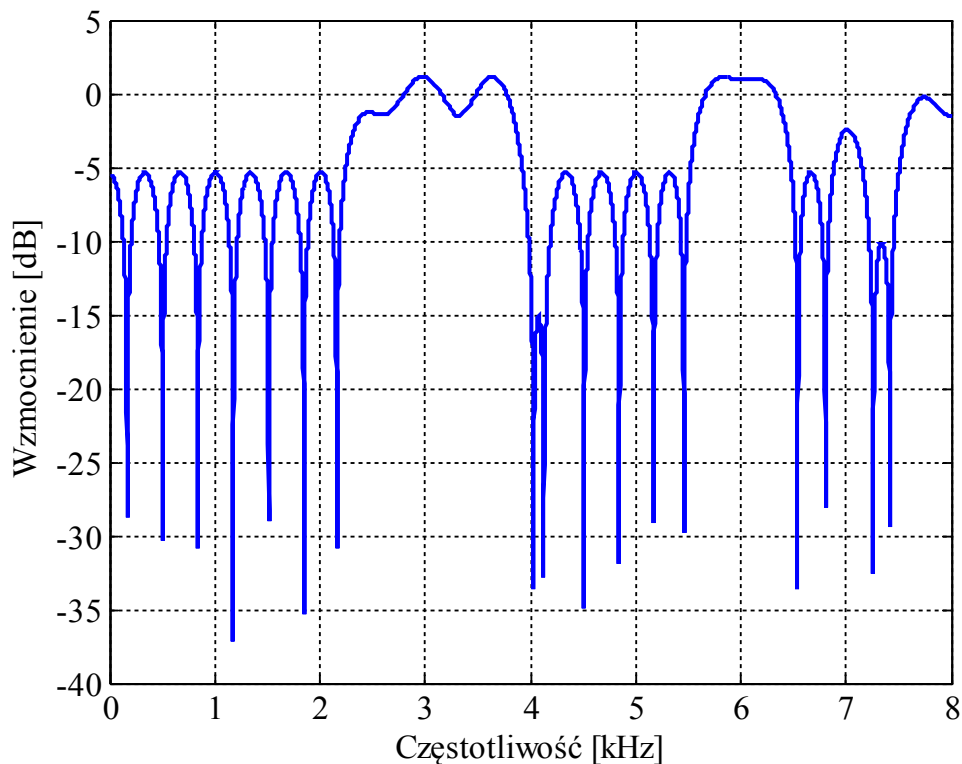
W tym celu zostały użyte stworzone programy:

- funkcję pobierającą przebieg sygnału i zwracającą jego RMS (*Root Mean Square*),
- funkcję dodającą do nagrania zakłócenie z ustalonym odstępem sygnału od

szumu mierzonym jako RMS,

- program wykonujący w pętli w/w funkcje dla wszystkich nagrań testowych, dla wszystkich mówców, w sześciu różnych wartościach SNR dodanego szumu oraz dla czterech typów szumu.

Oprócz zakłóceń o charakterze szumowym stworzono również próbki symulujące zniekształcenia liniowe w kanale transmisyjnym. Efekt uzyskano filtrując nagrania filtrem o widocznej na Rysunku 3 charakterystyce amplitudowej.



Rysunek 3: Charakterystyka amplitudowa filtra FIR użytego do zniekształcenia nagrań mowy

4.3. Parametryzacja MFCC (Voicebox)

Każde nagranie otrzymało swoją reprezentację w postaci macierzy MFCC o 13 współczynnikach dla każdej ramki. Do tego celu użyto funkcji z biblioteki *Voicebox* o nazwie *mfcc()*.

Oto argumenty wejściowe i wyjściowe tej funkcji:

```
function [ceps, freqresp, fb, fbrecon, freqrecon] = ...  
mfcc(input, samplingRate, frameRate)
```

gdzie:

- `input` – próbki sygnału wejściowego otrzymane przy pomocy funkcji `wavread()`,
- `SamplingRate` – częstotliwość próbkowania sygnału wejściowego (16kHz),
- `frameRate` – częstotliwość ramkowania sygnału, przyjęto 50 Hz co odpowiada długości ramki równej 20 ms.
- `ceps` – macierz współczynników MFCC

4.4. Implementacja CMS

Z racji tego, że współczynniki MFCC są reprezentacją cepstrum sygnału, można operację *Cepstral Mean Subtraction* wykonać na wartościach MFCC.

Poniżej przedstawiona jest funkcja przekształcająca macierz MFCC reprezentującą czyste próbki w macierz MFCC po operacji CMS:

```
function nowa_macierz_mfcc=cms(macierz_mfcc)  
srednie=mean(macierz_mfcc,2); (1)  
[~, y]=size(macierz_mfcc); (2)  
    for ii=1:y  
        nowa_macierz_mfcc(:,ii)=macierz_mfcc(:,ii)-srednie; (3)  
    end  
end
```

Opis:

- (1) - obliczenie średniej wartości współczynnika MFCC dla jednej wartości Mel-Cepstrum,
- (2) - wyznaczenie ilości ramek sygnału wejściowego,

(3) - obliczenie wykonywane tyle razy ile ramek zawiera sygnał, polegające na odjęciu, od współczynników MFCC każdej ramki, wartości średniej obliczonej w (1).

4.5. Weryfikator mówców

System opiera się na modelach mikstur gaussowskich. Każdego mówcę charakteryzuje jeden model. Pierwszym etapem było jednak wyznaczenie UBM (*Universal Background Model*). Jest to również GMM, wyznaczony nie na podstawie parametrów charakteryzujących jednego mówcę, ale zbioru parametrów od większej liczby osób – w tym wypadku od 45 mówców z bazy „Corpora”. UBM obliczony został przy użyciu funkcji *gaussmix* dostępnej w bibliotece *Voicebox*.

```
[mi15 kow15 weight15]=gaussmix(A', [], [], 15, 'k');  
[mi32 kow32 weight32]=gaussmix(A', [], [], 32, 'k');
```

Wskutek dwukrotnego użycia tej funkcji, na podstawie danych zawartych w macierzy *A* (macierz MFCC w której zawarte są parametry od wszystkich mówców z bazy) otrzymuje się dwa trzynasto-wymiarowe modele mikstur gaussowskich o 15 i 32 komponentach wyrażonych za pomocą parametrów modelu, czyli macierzy wartości średnich, macierzy kowariancji, oraz wag poszczególnych komponentów. Parametry startowe algorytmu optymalizującego model (*Expectation-Maximization*) wyliczone są na podstawie algorytmu *k-means*.

Aby ułatwić późniejszą organizację pracy każdy model GMM zapisywany był na dysku twardym jako obiekt klasy *gmdistribution* dostępnej w bibliotekach środowiska Matlab. Oto przykład utworzenia obiektu takiej klasy:

```
UBM15=gmdistribution(mi15, kow15, weight15);
```

Model UBM o 15 komponentach posłużył do nadania wartości początkowych przy treningu modeli pojedynczych mówców, również tutaj użyto funkcji *gaussmix*:

```
[mi c w]=gaussmix(A', [], [], UBM15.mu, UBM15.Sigma, UBM15.PComponents);
```

A - macierz MFCC dla jednego mówcy.

UBM15.mu, UBM15.Sigma, UBM15.PComponents - odpowiednio, macierz wartości średnich, macierzy kowariancji i wag przyjętych jako parametry startowe treningu (UBM).

Poniżej przedstawiony jest kod, który prezentuje główne elementy składające się na system weryfikacji, używa on elementów(macierzy), które opisane były powyżej:

```
function weryfikacja=weryfikator_ubm(m_UBM,m_mowcy,mfcc_t) (1)

PW=gaussmixp(mfcc_t',m_mowcy.mu,m_mowcy.Sigma,m_mowcy.PComponents); (2)
PU=gaussmixp(mfcc_t',m_UBM.mu,m_UBM.Sigma,m_UBM.PComponents); (3)

weryfikacja=mean(PW)-mean(PU); (4)
end
```

Opis:

- (1) - funkcja jako argumenty przyjmuje dwa obiekty klasy `gmdistribution`, które stanowią modele GMM mówcy i UBM.
- (2) - użycie funkcji `gaussmixp`, która przyjmuje parametry modelu GMM mówcy oraz dane testowe oraz wyznacza logarytm prawdopodobieństwa przynależności każdej kolejnej ramki MFCC do danego modelu mówcy.
- (3) - wyznaczenie logarytmu prawdopodobieństwa przynależności danego mówcy do UBM.
- (4) - obliczenie średniej wartości *log-likelihood ratio* za czas trwania sygnału testowego.

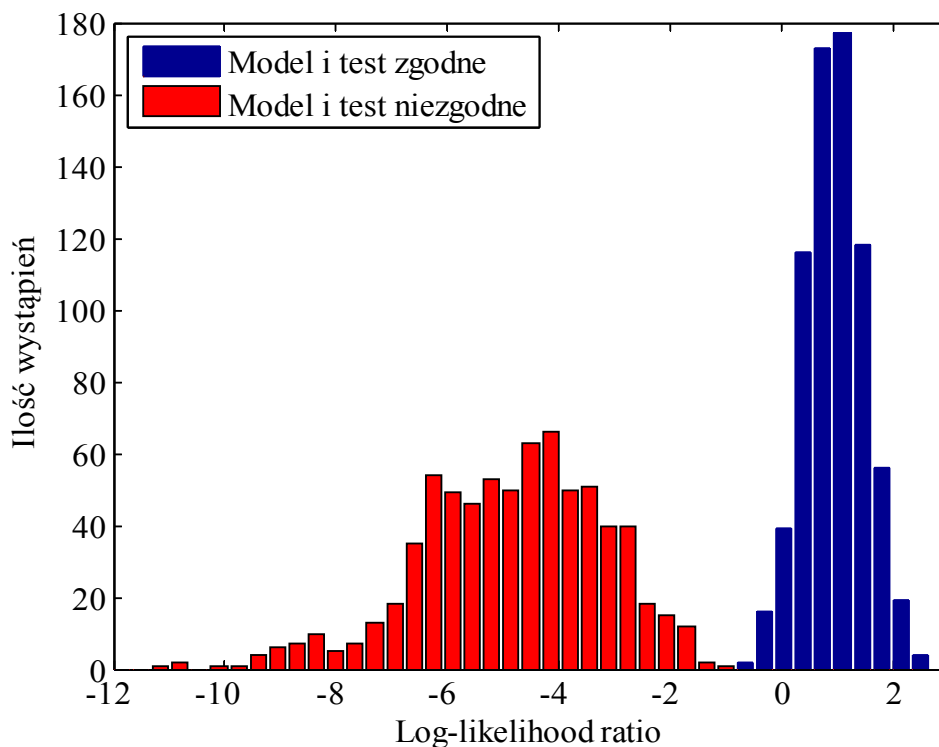
Funkcja ta nie zwraca zatem wartości 0 lub 1 (potwierdzającej lub negującej tożsamość), ale pewną miarę tego jak prawdopodobne jest to, że tożsamość mówcy (autora nagrania testowego) potwierdza się w parametrach modelu GMM. Dobranie

odpowiedniego progu weryfikacji odbędzie się wskutek dalszych operacji polegających na testowaniu działania algorytmu.

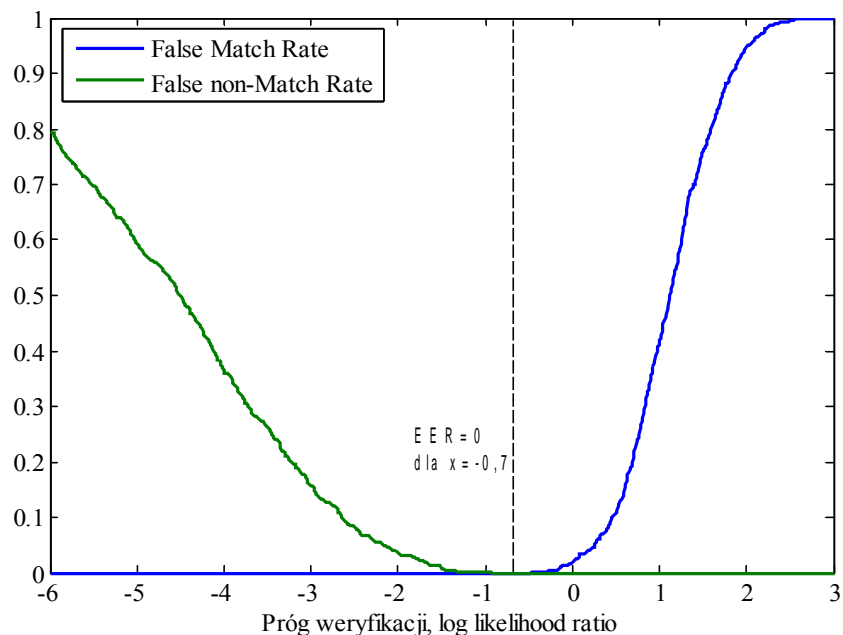
4.6. Test działania weryfikatora mówców

Po przedstawionym powyżej sposobie wykonania systemu weryfikującego tożsamość mówców przeprowadzono test tego systemu, aby zweryfikować jego działanie. Należało w tym celu stworzyć program, który w odpowiedni sposób kojarzył ze sobą nagrania testowe oraz modele GMM. Znacznie pomogło w tym uprzednie zorganizowanie danych w foldery. Bardzo wygodnie, opierając się na pętlach *for*, można było przetestować znaczną liczbę kombinacji. Dzięki wcześniejszemu obliczeniu i zapisaniu w pamięci modeli GMM mówców i UBM oraz macierzy testowych MFCC obliczenia nie były bardzo czasochłonne.

Poniżej przedstawiam wykresy obrazujące wyniki działania weryfikatora mówców. Dobrano po 720 różnych kombinacji zarówno w przypadku gdy model zgadza się z mówcą testowym i gdy ma miejsce sytuacja odwrotna.



Rysunek 4: Histogram obrazujący w jakich granicach mieszczą się wyniki dwóch typów kombinacji modelu i mówcy



Rysunek 5: False Match Rate i False non-Match Rate w zależności od progu weryfikacji. EER w przypadku, kiedy próbki nie są zakłócone.

Na rysunku 5 można zaobserwować, że wartości *False Match Rate* i *False non-Match Rate* w pewnym przedziale równocześnie przyjmują wartości zerowe, co oznacza, że jeśli przyjmujemy próg weryfikacji z tego przedziału otrzymamy bardzo wysoką skuteczność systemu weryfikacji (dla zbadanych przypadków wyniosłaby ona 100%). Ostateczną wartość EER przybliżono jako średnia arytmetyczna dolnej i górnej wartości przedziału kiedy FMR i FNMR przyjmują wartość 0. Nie oznacza to, że system z pewnością nie popełni błędów. Przy takiej ilości testów wartość EER można oszacować na mniejszą lub równą 0,006.

4.7. Przygotowanie modeli GMM oraz macierzy testowych dla nagrań zakłóconych

Aby przetestować skuteczność algorytmu CMS należało przeprowadzić operację *Cepstral Mean Subtraction* na wszystkich macierzach MFCC (zarówno danych, na podstawie których trenowane były modele GMM oraz macierzy testowych). W tym celu stworzono program który tworzył strukturę folderów i zapisywał w odpowiednich

miejscach zmodyfikowane przez algorytm CMS macierze MFCC.

Stworzenie modeli mikstur gaussowskich przebiegało identycznie jak w przypadku opisanym w rozdziale 4.5.

4.8. Wyznaczenie wpływu algorytmu CMS na działanie weryfikatora mówców

Ocena będzie polegać na obserwacji zmian wartości *Equal Error Rate* oraz odpowiadającemu tej wartości progowi weryfikacji wyrażonemu jako *log-likelihood ratio* dla pojedynczych przypadków (dla próbek niezakłóconych, dla wszystkich kombinacji typy szumu i wartości SNR, oraz dla próbek filtrowanych). W każdym przypadku wykonano obliczenia przy użyciu weryfikatora na próbce 720 różnych możliwości jednego typu dopasowania (zgodności i niezgodności mówców).

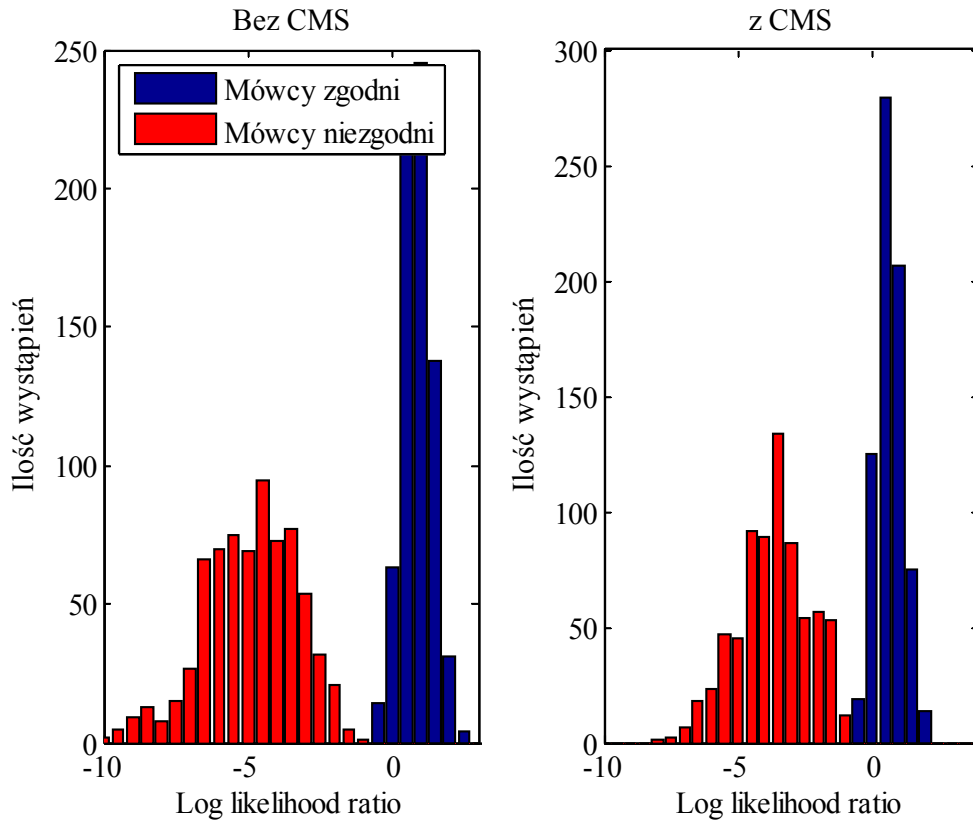
Ze względu na dużą ilość danych większość z nich będzie przedstawiona jedynie za pomocą tabel. W wybranych przypadkach przedstawione będą wykresy.

4.8.1. Brak zakłóceń

	EER (bez CMS)		EER (z CMS)	
	Wartość	Próg weryfikacji (log likelihood ratio)	Wartość	Próg weryfikacji (log likelihood ratio)
próbki niemodyfikowane	0,00	-0,7	0,00	-0,66

Tabela 1: EER przy braku zakłóceń

Na podstawie otrzymanego wyniku EER można stwierdzić, że przy odpowiednim dobraniu progu weryfikacji algorytm CMS nie wpływa na działanie weryfikatora mówców. W obu przypadkach otrzymujemy EER równy 0 co oznacza brak błędnych wskazań. Na rysunku 6 przedstawione są histogramy wyników *log-likelihood ratio*. Gdzie można zaobserwować wyraźnie rozróżnienie wyników mówców zgodnych i niezgodnych dla przypadków z CMS i bez CMS.



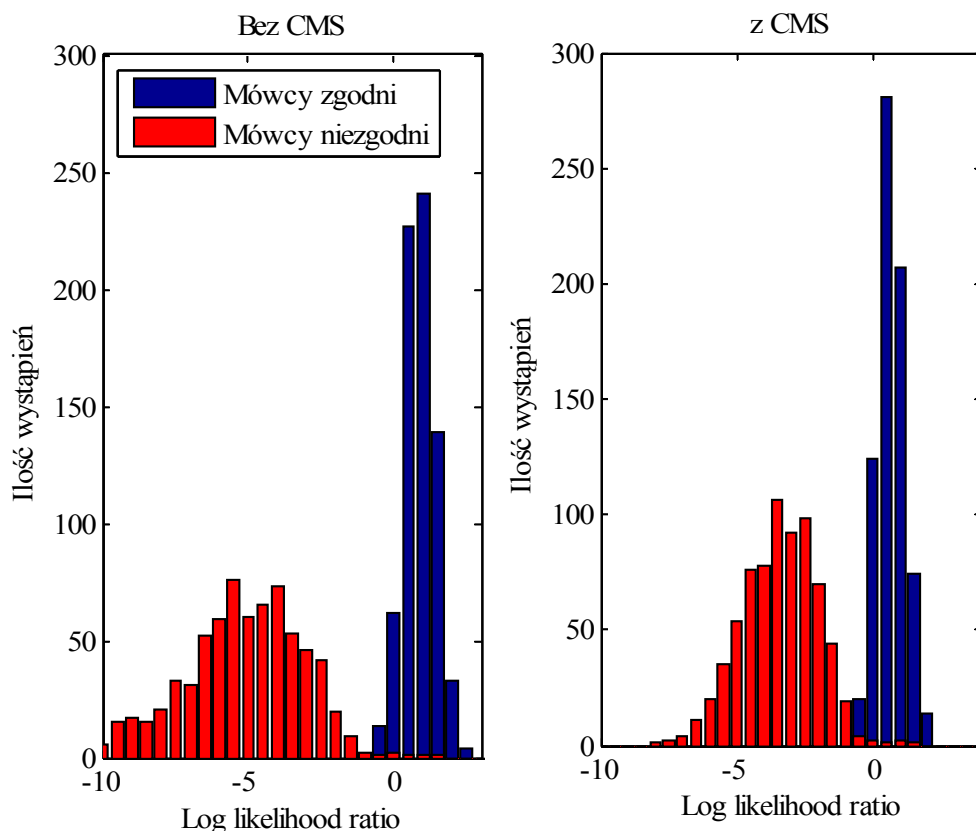
Rysunek 6: Histogramy wyników log-likelihood ratio dla próbek bez zakłóceń

4.8.2. Zakłócenie różnymi typami hałasu, SNR = 30 dB

SNR [dB]	Dodany szum	EER (bez CMS)		EER (z CMS)	
		Wartość	Próg weryfikacji (log likelihood ratio)	Wartość	Próg weryfikacji (log likelihood ratio)
30	biurowy	0,026	0,07	0,035	0,08
	cocktail party	0,022	0,04	0,028	0,00
	gaussowski	0,026	0,08	0,028	0,05
	uliczny	0,026	0,07	0,026	-0,02

Tabela 2: SNR = 30 dB

Dla przypadków zakłócenia różnymi typami szumu przy SNR 30 dB zaobserwować można nieznaczny spadek skuteczności systemu (wzrost EER) podczas zastosowania metody CMS.



Rysunek 7: Zestawienie histogramów dla przypadku gdy SNR = 30 dB, zakłócenie w postaci gwaru przyjęcia

Z racji tego że wszystkie histogramy dla tego przypadku prezentują się bardzo podobnie na Rysunku 7 przedstawiono tylko jeden przypadek (gdzie sygnałem zakłócającym jest gwar przyjęcia).

4.8.3. Zakłócenia różnymi typami hałasu, SNR = 20 dB

SNR [dB]	Dodany szum	EER (bez CMS)		EER (z CMS)	
		Wartość	Próg weryfikacji (log likelihood ratio)	Wartość	Próg weryfikacji (log likelihood ratio)
20	biurowy	0,032	0,10	0,035	0,10
	cocktail party	0,026	0,07	0,030	-0,03
	gaussowski	0,019	-0,23	0,022	-0,15
	uliczny	0,032	0,11	0,032	0,03

Tabela 3: SNR = 20 dB

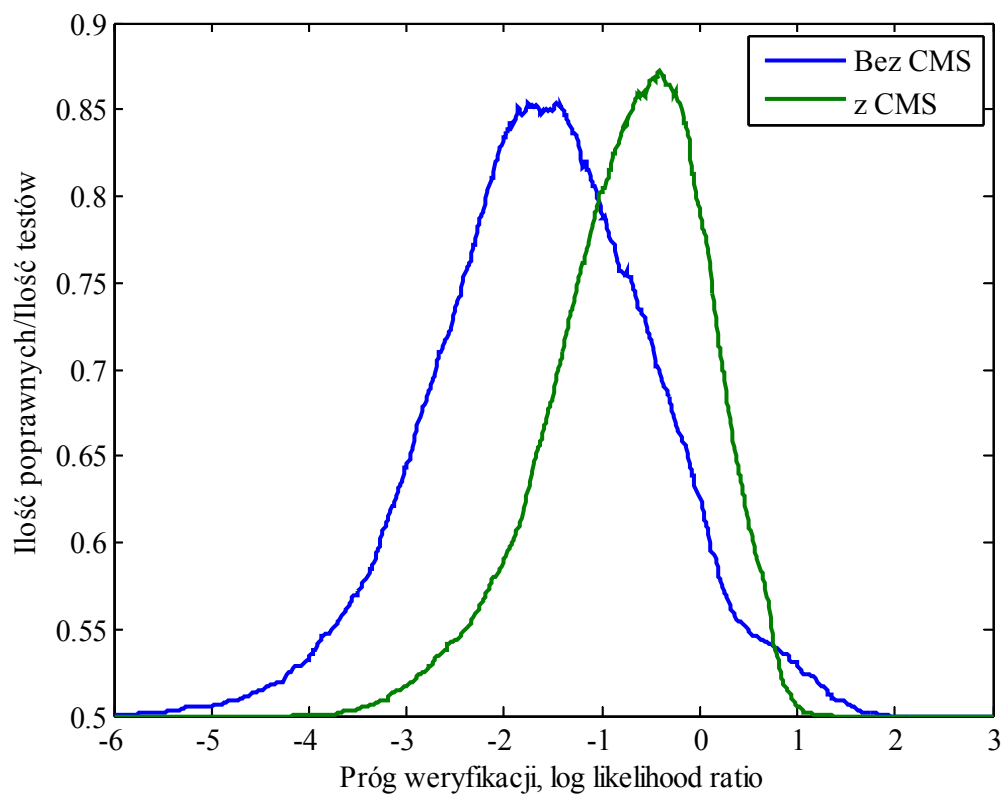
W tym wypadku również we wszystkich możliwościach (za wyjątkiem hałasu ulicznego) zaobserwowano spadek EER przy zastosowaniu CMS. Histogramy mają podobny charakter jak ten przedstawiony na rysunku 7.

4.8.4. Zakłócenia różnymi typami hałasu, SNR = 10 dB

SNR [dB]	Dodany szum	EER (bez CMS)		EER (z CMS)	
		Wartość	Próg weryfikacji (log likelihood ratio)	Wartość	Próg weryfikacji (log likelihood ratio)
10	biurowy	0,040	-0,35	0,048	-0,13
	cocktail party	0,035	-0,40	0,052	-0,31
	gaussowski	0,148	-1,48	0,132	-0,40
	uliczny	0,031	-0,44	0,048	-0,23

Tabela 4: SNR = 10 dB

W tym przypadku po raz pierwszy (dla zakłócenia stacjonarnym szumem gaussowskim) obserwujemy nieznaczny spadek wartości EER w przypadku zastosowania CMS w porównaniu do tej samej sytuacji bez zastosowania tego algorytmu. Należy zwrócić uwagę na mniejsze odchyłki wartości *log-likelihood ratio* progu weryfikacji w stosunku do wartości dla wyższych SNR. Na rysunku 8 można zaobserwować, że maksymalna wartość gdy analizie poddaje się krzywą skuteczności systemu z modyfikacją CMS jest większa niż maksimum dla drugiej krzywej. Dodatkowo próg weryfikacji w którym wystąpiła wartość maksymalna jest zlokalizowany w wyższej wartości.



Rysunek 8: Wykres przedstawiający skuteczność systemu przy próbkach zakłóconych stacjonarnym szumem gaussowskim, SNR = 10 dB

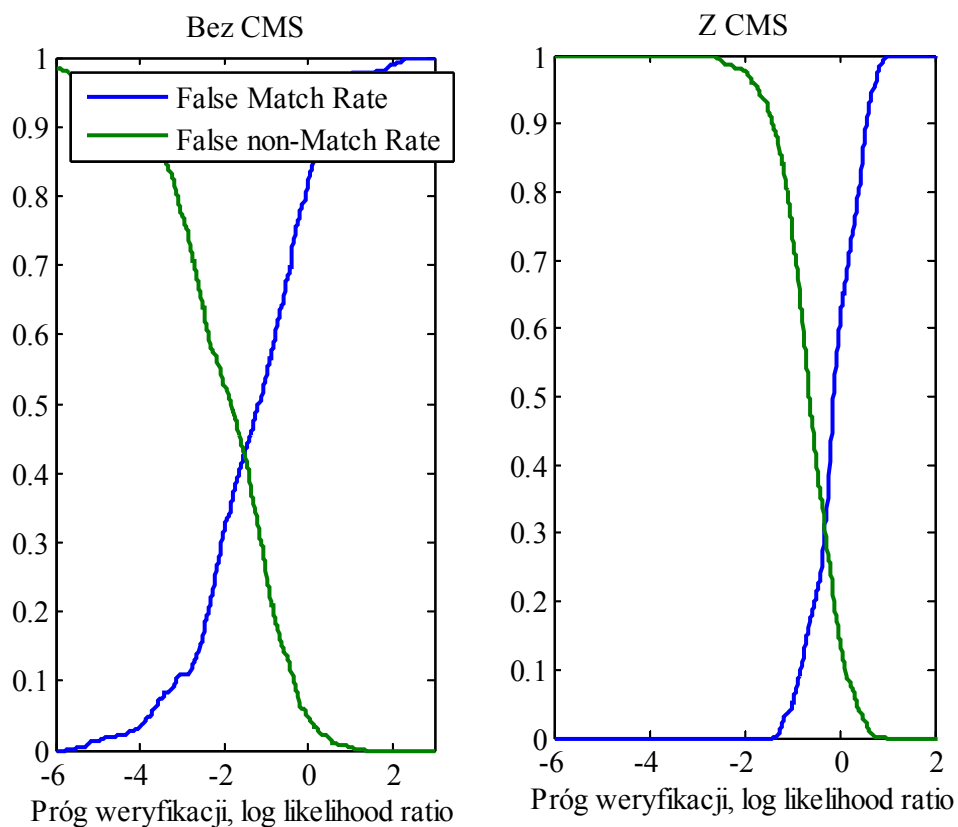
4.8.5. Zakłócenia różnymi typami hałasu, SNR = 5 dB.

SNR [dB]	Dodany szum	EER (bez CMS)		EER (z CMS)	
		Wartość	Próg weryfikacji (log likelihood ratio)	Wartość	Próg weryfikacji (log likelihood ratio)
5	biurowy	0,094	-1,40	0,110	-0,34
	cocktail party	0,095	-1,36	0,153	-0,39
	gaussowski	0,429	-1,51	0,309	-0,31
	uliczny	0,103	-1,29	0,151	-0,28

Tabela 5: SNR = 5 dB

Zaobserwować w tym przypadku można podobną sytuację jak dla przypadku poprzedniego (SNR 10 dB). Progi weryfikacji dla weryfikatora bez CMS przesunęły się

znacznie w kierunku niższych wartości, podczas gdy przy zastosowaniu CMS odbiegają od poprzednich o mniejsze odległości. Nastąpił wzrost różnicy EER w przypadku zakłócenia w postaci szumu gaussowskiego. Co potwierdza się na przedstawionym na rysunku 9. Należy zwrócić uwagę na przesunięcie wartości EER w kierunku niższej wartości w przypadku bez CMS w stosunku do wyników poprzednich i wyników w sytuacji z zastosowaniem *Cepstral Mean Subtraction*.



Rysunek 9: Wykresy FMR i FNMR dla $SNR = 5$ dB i zakłócenia szumem gaussowskim

4.8.6. Zakłócenia różnymi typami hałasu, SNR = 0 dB

SNR [dB]	Dodany szum	EER (bez CMS)		EER (z CMS)	
		Wartość	Próg weryfikacji (log likelihood ratio)	Wartość	Próg weryfikacji (log likelihood ratio)
0	biurowy	0,370	-3,05	0,390	-0,42
	cocktail party	0,360	-3,08	0,397	-0,41
	gaussowski	0,480	-1,54	0,460	-0,18
	uliczny	0,400	-2,51	0,422	-0,34

Tabela 6: SNR = 0 dB

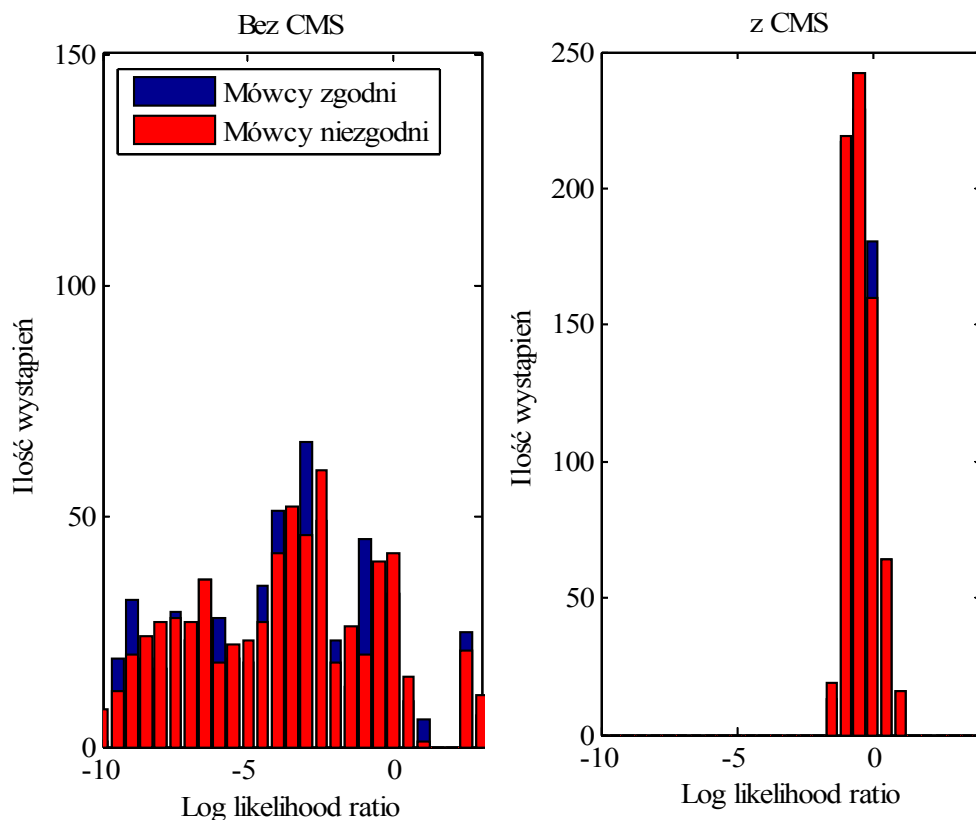
Zachowana została tendencja z poprzednich wyników. Wciąż zaobserwować można spadek EER tylko dla zakłócenia stacjonarnym szumem gaussowskim. Nadal wartości progu weryfikacji w przypadku weryfikatora z CMS przyjmują bliższe sobie wartości. Wartości EER w obserwowanych wypadkach wzrastają w porównaniu do wyników dla większych SNR.

4.8.7. Zakłócenia różnymi typami hałasu, SNR = -5 dB

SNR [dB]	Dodany szum	EER (bez CMS)		EER (z CMS)	
		Wartość	Próg weryfikacji (log likelihood ratio)	Wartość	Próg weryfikacji (log likelihood ratio)
-5	biurowy	0,490	-3,16	0,498	-0,47
	cocktail party	0,485	-4,81	0,484	-0,42
	gaussowski	0,503	-3,62	0,497	-0,26
	uliczny	0,497	-3,55	0,499	-0,34

Tabela 7: SNR = -5 dB

Dla przypadków kiedy poziom RMS dodanego szumu jest wyższy niż poziom RMS sygnału mowy wskaźnik EER przyjmuje wartości ok. 0,5, co oznacza, że system przy takich danych wejściowych nie potrafi w wiarygodny sposób odróżnić sygnałów pochodzących od nagrań różnych mówców. Świadczą o tym histogramy umieszczone na rysunku 10.



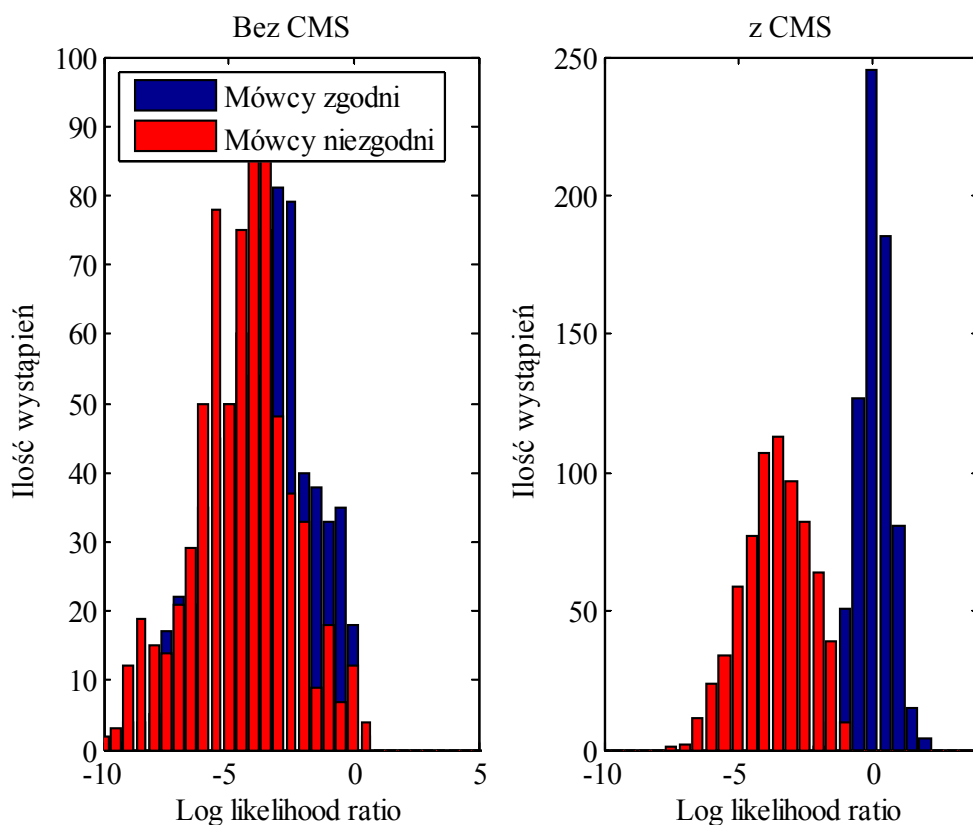
Rysunek 10: Histogramy wartości log-likelihood ratio dla próbek zakłóconych szumem gaussowskim - SNR = -5dB

4.8.8. Liniowe zniekształcenie kanału transmisyjnego

Dla przypadku, gdy zakłócenie jest postaci zniekształcenia charakterystyki częstotliwościowej sygnału (filtracji) można zaobserwować znaczny wzrost skuteczności przy zastosowaniu algorytmu CMS (Tabela 8).

	EER (bez CMS)		EER (z CMS)	
	Wartość	Próg weryfikacji (log likelihood ratio)	Wartość	Próg weryfikacji (log likelihood ratio)
nagrania filtrowane	0,40	-3,7	0,02	-1,01

Tabela 8: EER przy obecności zniekształceń częstotliwościowych



Rysunek 11: Zestawienie histogramów weryfikacji w przypadku zastosowania i niezastosowania algorytmu CMS.

Na rysunku 11 można zaobserwować wyraźne rozróżnienie histogramów w przypadku zastosowania CMS w porównaniu do sytuacji bez zastosowania tego algorytmu.

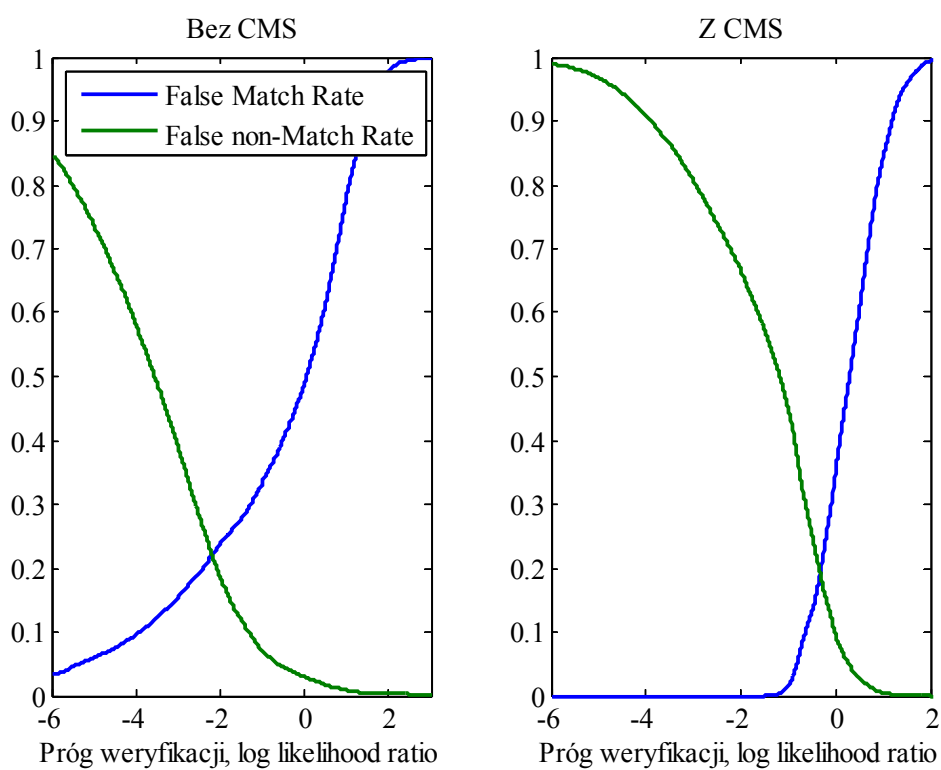
4.8.9. Zestawienie wszystkich sytuacji

W tej części zestawione zostaną ze sobą dotychczasowe wyniki weryfikacji i na ich podstawie policzony będzie wskaźnik EER (Tabela 9). Oznacza to po 18000 zestawień zgodności i niezgodności dla przypadku z zastosowaniem i bez zastosowania CMS.

Na rysunku 12 przedstawiono FNMR i FMR w zależności od przyjętego progu weryfikacji.

	EER (bez CMS)		EER (z CMS)	
	Wartość	Próg weryfikacji (log likelihood ratio)	Wartość	Próg weryfikacji (log likelihood ratio)
Zebrańie wszystkich przypadków	0,220	-2,190	0,192	-0,330

Tabela 9: EER przy zastosowaniu i braku zastosowania CMS dla zebranych wszystkich omawianych przypadków



Rysunek 12: Zestawienie FNMR oraz FMR w zależności od przyjętego progu weryfikacji dla wszystkich wcześniej omawianych kombinacji

5. Interpretacja wyników

Algorytm CMS, jak się okazało bardzo dobrze sprawdza się kiedy mamy do czynienia ze zniekształceniem kanału. Dzieje się tak dlatego, że zniekształcenie to jest takie samo dla całego przebiegu czasowego sygnału. Wskutek tego wektor odjemników algorytmu CMS w sposób dokładny kompensuje zniekształcenie. Jeśli natomiast zastosuje się algorytm CMS, kiedy sygnał zakłócony jest sygnałem akustycznym wyniki dla pojedynczych przykładów ulegają nieznacznemu pogorszeniu. Tylko przy dodanym szumie gaussowskim zaobserwować można poprawę działania weryfikatora. Jeśli jednak zwróci się uwagę na to, że w przypadku zastosowania subtrakcji wartość EER, przyjmowana w systemach za próg weryfikacji, to zauważyć można że wahania tej wartości przy różnych przypadkach bez zastosowania wcześniej CMS są większe niż gdy ten algorytm został zastosowany. Oznacza to tyle, że jeśli nie ma możliwości określenia w jakich warunkach akustycznych zostanie wykonane nagranie testowe, algorytm CMS w pewnym stopniu zapewnia niezależnienie się od tych parametrów, co prezentuje rysunek 12

6. Zakończenie

Podsumowując, udało się zaimplementować algorytm CMS oraz przetestować jego działanie na przykładzie niezależnego od tekstu weryfikatora mówców. Niestety dla pojedynczych przykładów zaszumienia skuteczność systemu jest mniejsza, gdy zastosowana była metoda *Cepstral Mean Subtraction*. Rozpatrzono jednak sytuację, kiedy nie wiadomo jakie zakłócenie występuje w nagraniu. Wtedy okazało się, że na badanej próbie przykładów omawiana metoda spowodowała spadek ilości błędów popełnianych przez system weryfikacji.

Bazując na tej obserwacji można stwierdzić, że algorytm CMS lepiej jest stosować, kiedy istnieje duże prawdopodobieństwo, że nagrania testowe będą wykonywane w znacznie różniących się od siebie warunkach np. używając urządzeń rejestrujących różnej jakości, w różnych pomieszczeniach, przy różnym typie i poziomie szumu akustycznego.

Bibliografia:

[1] *Digital speech processing : speech coding, synthesis and recognition / ed. by A. Nejat Ince*, Boston : Kluwer Academic Publ., 1992.

[2] <http://www.griaulebiometrics.com/en-us/book/understanding-biometrics/evaluation/accuracy/matching/false>, 10.11.2013, 10:30

[3] *Speaker Verification Using Adapted Gaussian Mixture Models*, Douglas A. Reynolds, Thomas F. Quatieri, Robert B. Dunn

[4] <https://www.ccrma.stanford.edu/~unjung/mylec/mfcc.html>, 11.01.2013 15:50

[5] *The Use of Cepstral Means in Conversational Speech Recognition*, Martin Westphal

[6] <http://www.freesound.org>