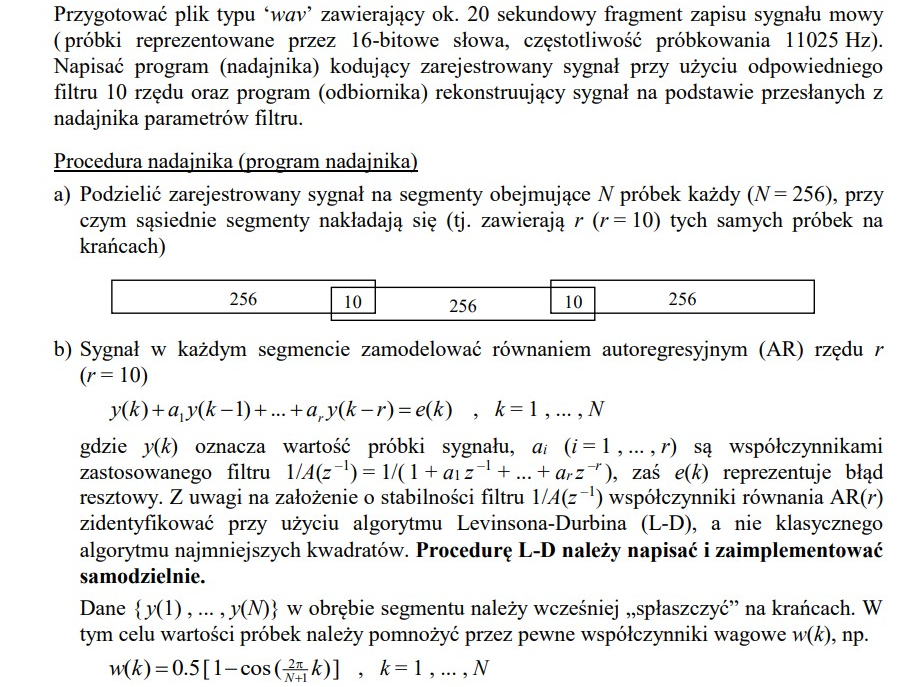
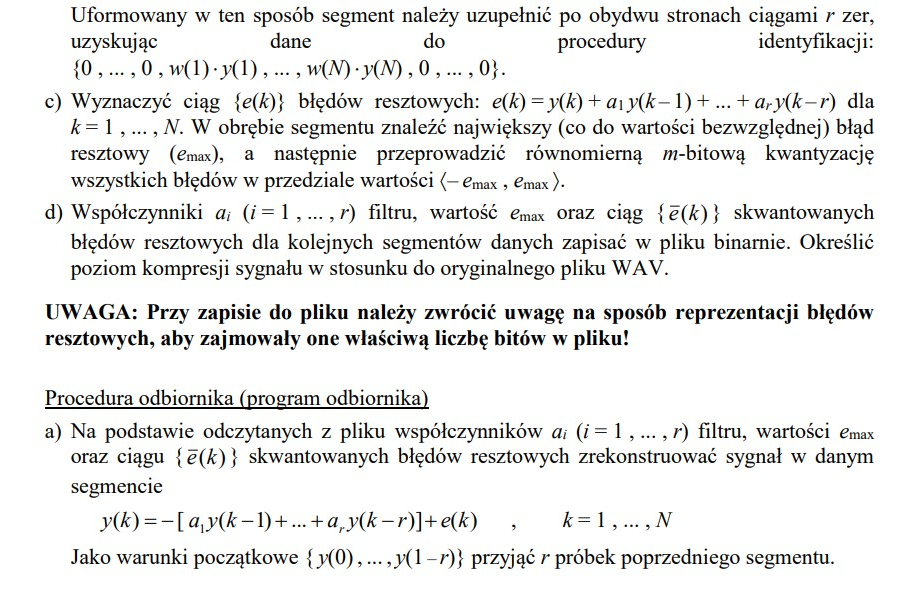
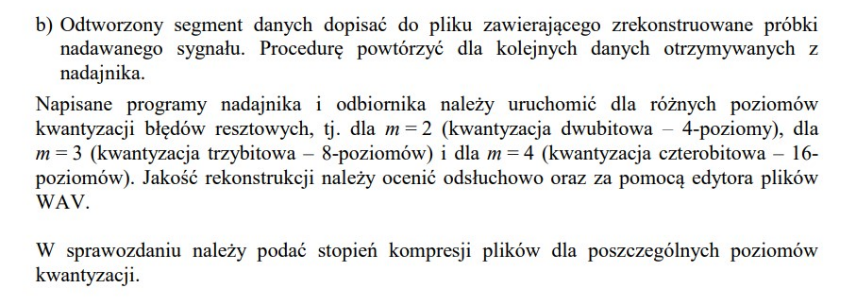
|  |
| --- |
| Identyfikacja procesów 2 |
| Predykcyjne kodowanie sygnałów |
| Autorzy: Alan Konopko 180354, Patryk Martyniak 181578 |







Spis treści

[Wprowadzenie do algorytmu Levinson’a – Durbin’a 2](#_Toc170299236)

[Przygotowany plik z sygnałem mowy 2](#_Toc170299237)

[Procedura nadajnika 3](#_Toc170299238)

[Procedura odbiornika 3](#_Toc170299239)

[Ocena poziomu kompresji uzyskanego sygnału 3](#_Toc170299240)

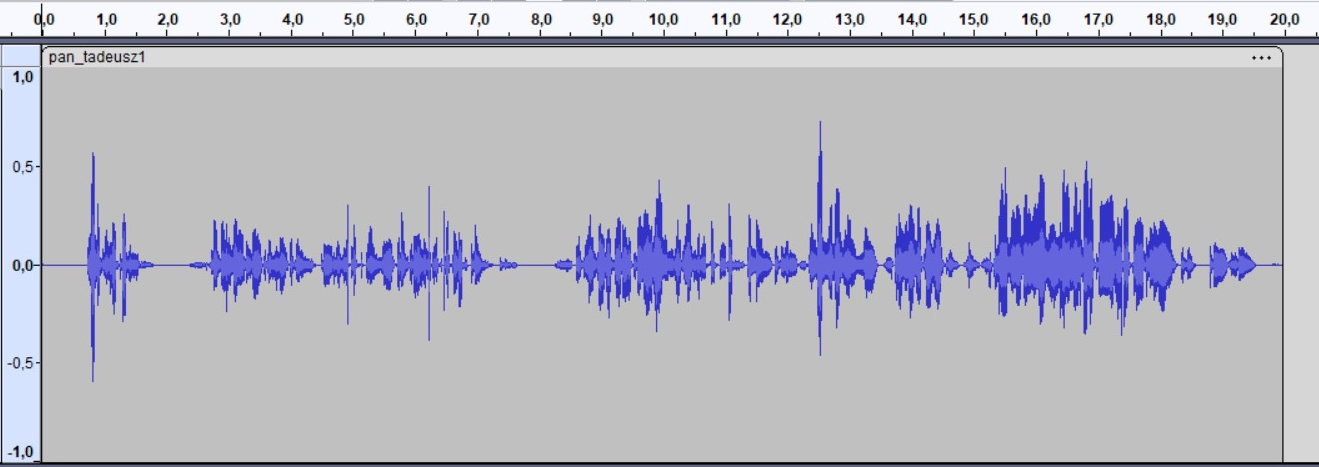
[Podsumowanie 4](#_Toc170299241)

# Wprowadzenie do algorytmu Levinson’a – Durbin’a

Równania Yule’a – Walkera łączą parametry procesu autoregresyjnego z jego współczynnikami autokorelacji.

Levinson i Durbin opracowali algorytm, który rozwiązuje równania Yule’a – Walkera w sposób rekurencyjny. Pozwala to na wyznaczenie parametrów procesu autoregresyjnego bez konieczności odwracania macierzy zawierającej współczynniki autokorelacji. Algorytm ten jest szczególnie przydatny, gdy rząd procesu jest duży, a procedura odwracania takiej macierzy wymagałaby bardzo wydajnej obliczeniowo jednostki.

# Przygotowany plik z sygnałem mowy

W celu zapewnienia jak najmniejszych zakłóceń w pliku wejściowym zdecydowano się na przetestowaniu programu nadajnika z wykorzystaniem fragmentu audiobook’a „Pana Tadeusza” nagranego przy użyciu dedykowanego do tego sprzętu. Sygnał mowy przedstawiono na wykresie:

# Procedura nadajnika

1. Procedura rozpoczyna się od podziału pliku. Podział zrealizowano w funkcji „podzial( )”, która iteracyjnie dziali plik na 256 elementowe segmenty, przy czym 10 ostatnich elementów (n-1) segmentu pokrywa się z pierwszymi dziesięcioma elementami n-tego segmentu. Następnie zainicjowano 10 elementowy wektor parametrów dla 10-tego rzędu zastosowanego filtru.
2. Przygotowane segmenty „spłaszczono” przy użyciu napisanej funkcji „wyplaszcz( )”, która najpierw wyznacza wektory współczynników wagowych na podstawie wzoru danego w treści, a następnie przemnaża przez nie segmenty. Na koniec przy użyciu napisanej funkcji „wyzeruj\_konce( )”, spłaszczone uprzednio segmenty są uzupełniane po obu stronach segmentu dziesięcioma zerami (filtr 10-tego rzędu).
3. Przy użyciu napisanej funkcji „policz\_p0( )” wyznaczono wartość p0. Następnie z wykorzystaniem funkcji „policz\_pi\_Ri( )” wyznaczono wektory autokorelacji. Wektory te podano do napisanej funkcji z zaimplementowanym algorytmem Levinsona-Durbina „L\_D( )”. Funkcja ta zwraca wartości współczynników odbicia „k(i)”, współczynników filtru „a(i)” oraz błędów predykcji „e”. Na podstawie danych (nie poddanym ważeniu oraz bez wyzerowanych końców) oraz wartości współczynników filtru a(i) przy użyciu funkcji „policz\_reszty( )” wyznaczony został ciąg błędów resztowowych oraz maksymalny błąd.
4. Ciąg błędów resztowych oraz maksymalny błąd zostały przekazane do funkcji „skwantuj( )”, która przeprowadziła m-bitową kwantyzację.
5. W ostatnim kroku wartości współczynników filtru, ciąg błędów resztowych oraz maksymalny błąd zapisano binarnie w pliku .zip załączonym wraz z dokumentacją.

# Procedura odbiornika

Dane otrzymane z procedury nadajnika zostały przekazane do programu odbiornika. Procedura odbiornika została zrealizowana w następujący sposób:

1. Przy pomocy napisanej funkcji „policz1( )” wyznaczono pierwszy zrekonstruowany segment sygnału.
2. Następnie iteracyjnie przy pomocy napisanej funkcji „policz\_reszte( )” zrekonstruowano pozostałe segmenty sygnału.

# Ocena poziomu kompresji uzyskanego sygnału

# Podsumowanie

Odsłuchowo można stwierdzić, że większa ilość poziomów kwantyzacji poprawia jakość sygnału otrzymanego po rekonstrukcji. Wraz ze wzrostem poziomów kwantyzacji maleje ilość „trzasków” w sygnale. Znacząco poprawia się również jakość sygnału we fragmentach w którym jego amplituda jest bliska zero.