Labolatorium PTI – ćwiczenie 1, przykład 1

Reprezentacja i przetwarzanie sygnałów

Symulowanie opóźnień fazowych występujących w filtrach za pomocą analizy falkowej

1 Wstęp

W niniejszej pracy przedstawiony został wpływ opóźnień fazowych występujących w filtracji na jakość dźwięku. Do symulacji efektu filtracji i opóźnień czasowych została użyta ciągła dekompozycja falkowa (CWT) oraz bazowa falka Morleta.

2 Metody

2.1 Analiza falkowa

$$F_\alpha = \frac{F_c}{\alpha\Delta} \tag{1}$$

Funkcja 1 zamiany oczekiwanych pseudo-częstotliwości falki na skale analizy falkowej, gdzie: α jest skalą, Δ oznacza okres próbkowania, F_c środkowa częstotliwość falki, F_α pseudo częstotliwość odpowiadająca skali α.

2.2 Zamiana częstotliwości na skale

W celu zamiany pseudo-częstotliwości na skale została napisana funkcja zamieniająca pseudo-częstotliwości falki bazowej na oczekiwane skale. W tym celu został zastosowany wzór 1 oraz funkcja MATLAB centfrq() pozwalająca na określenie centralnej częstotliwości falki.

```
function [ scales ] = freq2scale( frequencies,wname,delta )
    c_f = centfrq(wname); %Wavelet center frequency
    scales = c_f ./ (delta .* frequencies);
end
```

2.3 Falka bazowa

W analizie została użyta Falka morleta. Jest ona efektem pomnożenia funkcji eulera przez funkcję gaussa. Falka morleta zwana jest także falką gabora i zpostała przedstawiona na wykresie 1. Falka morleta znajduje swoje zastosowania w medycynie (np. w analizach ECG w celu odkrywania nieprawidłowości) a także w muzyce (np. w transkrypcji).

3 Układ filtru

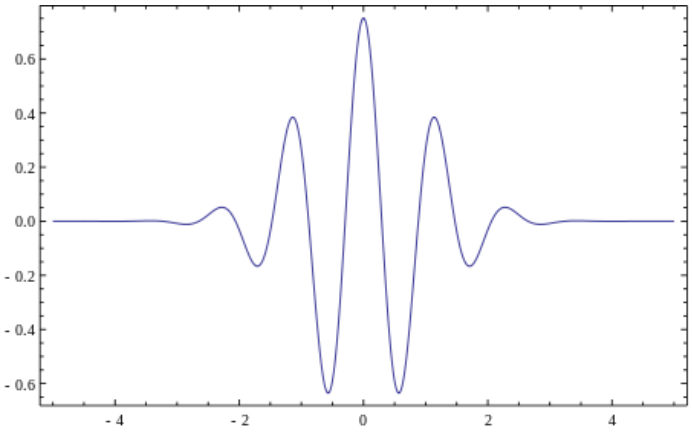
3.1 Parametryzacja

Proponowane rozwiązanie paramteryzowane jest następująco:

frequencies Pseudo częstotliwości odpowiadające skalom w analizie falkowej

wname Nazwa falki bazowej używanej do dekompozycji i rekonstrukcji sygnału

Rysunek 1: Wykres falki morleta



characteristic_frequency Charakterystyczna częstotliwość filtra, dla której ma on największą aktywację

freq_range Zakres częstotliwości filtra, aktywacja filtra występuje pomiędzy wartościami characteristic_frequency - freq_range a characteristic_frequency + freq_range

phase_delay_factor O ile sekund ma zostać przesunięty sygnał w obszarze maksymalnej aktywacji

frequenc_factor Wartość o którą przemnożona jest amplituda sygnału w przypadku maksymalnej aktywacji (0 wycina charakterystyczną częstotliwość, 0.5 - dla maksymalnej częstotliwości sygnał ma 0.5 amplitudy, 1 - brak modyfikacji amplitudy)

3.2 Metody

3.2.1 Dekompozycja i rekonstrukcja sygnału

```
scales = freq2scale(frequencies, wname, 1/fsample);
y_cwtft = cwtft(y,'plot','wavelet',wname,'scales',scales);
...
y_icwtft = icwtft(y_cwtft,'signal',y);
```

Dekompozycja i rekonstrukcja sygnału jest przeprowadzona przez algorytm odpowiadający algorytmowi FFT zaadaptowany dla złożonej analizy falkowej (CWT). Ma to miejsce po zmapowaniu zdefiniowanych częstotliwości na skale.

3.2.2 Aktywacja filtra

Po stworzeniu konfiguracji cwt, tworzony jest wektor aktywacji filtra, gdzie 1 oznacza maksymalną aktywację a 0 minimalną.Stworzone jest trójkątne okno

aktywacji, gdzie maksymalna aktywacja ma miejsce dla charakterystycznej częstotliwośći, a filtr jest aktywny tylko dla zdefiniowanego zakresu częstotliwości.

```
frequencies = scal2frq(y_cwtft.scales,y_cwtft.wav,1/fsample);
%Filter activation for scales / frequencies
activation_values = 1- min(1,abs(frequencies - characteristic_frequency)/freq_range);
...
for i = 1:size(y_cwtft.scales,2)
    %Modify CWT for given scale
    y_cwtft.cfs(i,:) = dump_values(i) * circshift(y_cwtft.cfs(i,:)',delay_values(i))';
end
```

Używając wektorów opóźniania oraz obcinania sygnałów dla danych skal (częstotliwości), następuje modyfikacja wyników dekompozycji falkowej o odpowiednie wartości obniżenia wartości i przesunięcia.

3.2.3 Opóźnianie sygnału

Wektor przesunięcia dla częstotliwości jest uzyskany przez przemnożenie częstości próbkowania oraz wartości opóźnienia (co powoduje przejście do operacji na próbkach a nie sekundach) oraz wartości aktywacji.

```
delay_values = floor(fsample * phase_delay_factor * activation_values);
```

3.2.4 Modyfikowanie ampltitudy sygnału

Wektor modyfikacji amplitudy jest uzyskany w ten sposób aby w liniowy sposób modyfikowana była wartość dla w zależności od częstotliwośći.

```
dump_values =  frequency_factor + (1-frequency_factor)*(1-activation_values);
```

3.2.5 Normalizacja sygnału po rekonstrukcji

Przeprowadzenie dekompozycji i rekonstrukcji sygnału za pomocą analizy falkowej może prowadzić do zmiany dziedziny. W tym celu przeprowadzona jest normalizacja do dziedziny wejściowego sygnału.

```
%Normalize output between -1 and 1
y_icwtft= (y_icwtft - min(y_icwtft)) / ( max(y_icwtft) - min(y_icwtft) );
%Bring back to input range
y_icwtft= min(y) + (max(y)-min(y))*y_icwtft;
```

4 Walidacja

4.1 Wycinanie składowej

W celu weryfikacji zaproponowanej metody symulowania filtracji, zostały przetworzony sygnał wejściowy, będący sumą trzech funkcji sinus. Konfiguracja filtra odpowiada wycinaniu kompontentu o jednej częstotliwości, w efekcie wyjście z filtra powinno odpowiadać sumie dwóch funkcji sinus.

4.1.1 Sygnał wejściowy

$$y(t) = sin(2\pi t) + sin(4\pi t) + sin(8\pi t) \tag{2}$$

Sygnał wejściowy 3, suma trzech funkcji sinus o różnych częstotliwościach (1, 2 i 4 hz). Sygnał wejściowy bazuje na sumie funkcji sin ze względu na to, że odpowiada to falce bazowej, która oparta jest o wzór eulera.

4.1.2 Konfiguracja układu

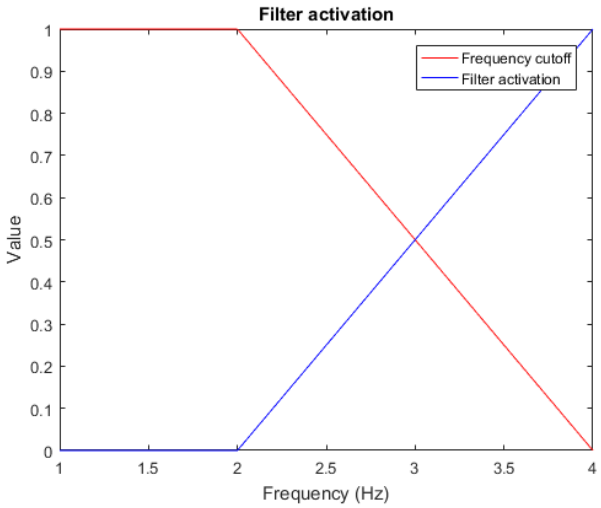
```
%Pseudo-Frequencies for CWT
frequencies = [4 2 1];
%Properties
wname = 'morl';
characteristic_frequency = 4;
freq_range = 1;
phase_delay_factor = 0;
frequency_factor = 0;
```

Filtr został skonfigurowany w ten sposób, żeby bazował na falce morleta, używając skal odpowiądającym 3 częstotliwościom , wycinał częstotliwość 4 Hz z marginesem aktywacji 1 Hz, dając w efekcie obszar aktywacji pomiędzy 3 a 5 Hz, co przedstawia wykres 5. Efekt opóźnienia fazowego jest pomijany a częstotliwość składowa odpowiadająca częstotliwości charakterystycznej jest całkowicie wycinana.

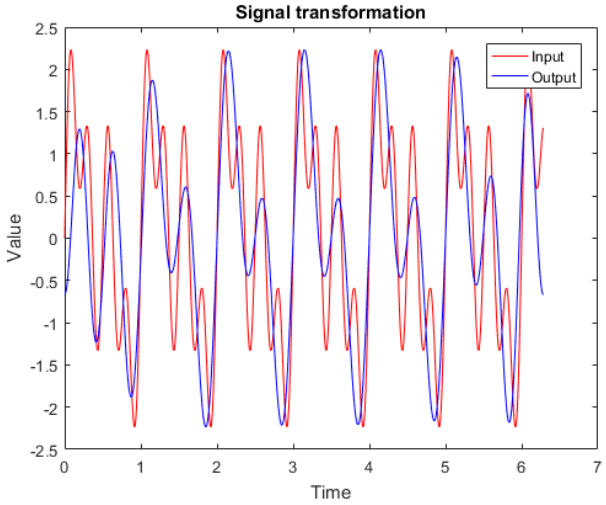
4.1.3 Wynik

Jak widać na porównaniu wyjścia z filtra przedstawionego na wykresie 3 oraz funkcji będącej składowej dwóch sinusów bez składowej o częstotliwości 4Hz 4 filtr poprawnie wyciął charakterystyczną składową.

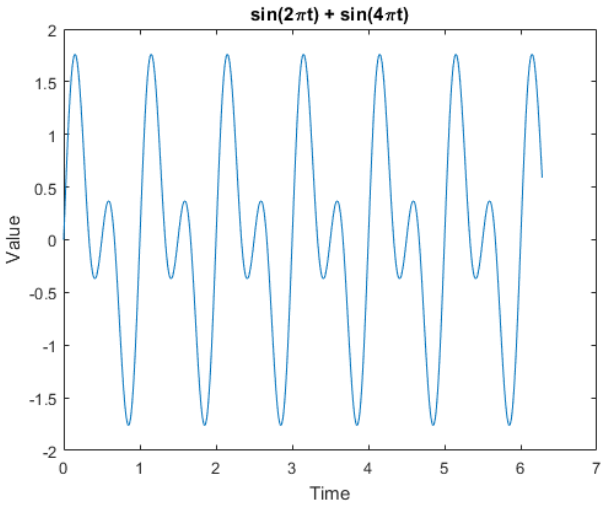
Rysunek 2: Wykres aktywacji częstotliwościowej filtra



Rysunek 3: Wykres filtracji sygnału wejściowego



Rysunek 4: Sygnał porównawczy - $\sin(2\pi t) + sin(4\pi t)$



4.2 Opóźnianie składowej

W celu weryfikacji zaproponowanej metody symulowania filtracji, zostały przetworzony sygnał wejściowy, będący funkcją sinus. Konfiguracja filtra odpowiada opóźnianiu kompontentu o jednej częstotliwości, w efekcie wyjście z filtra powinno odpowiadać funkcji sinus przesuniętej w fazie.

4.2.1 Sygnał wejściowy

$$y(t) = sin(8\pi t) \tag{3}$$

Sygnał wejściowy 3, funkcja sinus o częstotliwości (4 Hz). Sygnał wejściowy jest pojedyńczą funkcją sinus ze względu na możliwość prostej walidacji.

4.2.2 Konfiguracja układu

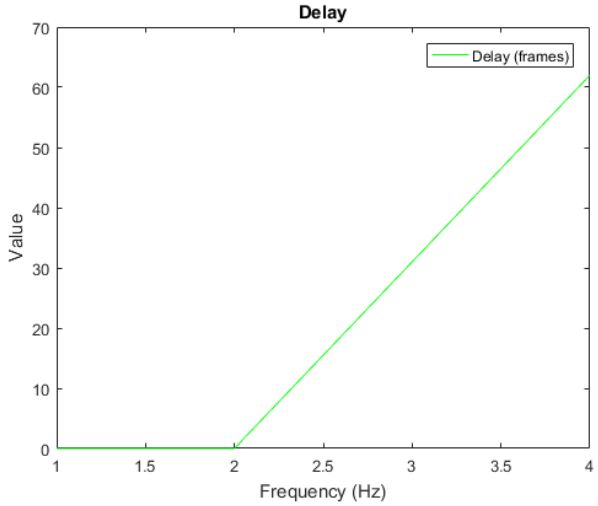
```
%Pseudo-Frequencies for CWT
frequencies = [4 2 1];
%Properties
wname = 'morl';
characteristic_frequency = 4;
freq_range = 1;
%Effect
%phase_delay_factor = 0;
phase_delay_factor = 0.125;
frequency_factor = 1;
```

Filtr został skonfigurowany w ten sposób, żeby bazował na falce morleta, używając skal odpowiądającym 3 częstotliwościom . Opóźniana jest częstotliwość 4 Hz, z marginesem aktywacji 1 Hz, dając w efekcie obszar aktywacji pomiędzy 3 a 5 Hz, co przedstawia wykres 5. Efekt opóźnienia fazowego jest na poziomie 0.125 sekundy, natomiast częstotliwość składowa odpowiadająca częstotliwości charakterystycznej nie jest wycinana.

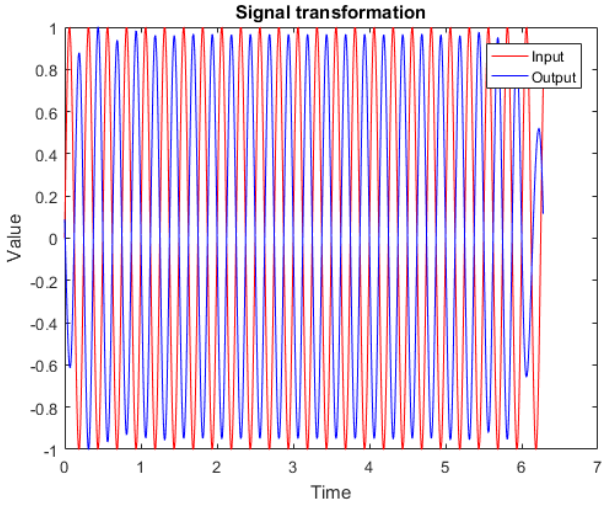
4.2.3 Wynik

Jak widać na porównaniu wejścia do wyjścia z filtra przedstawionego na wykresie 6 filtr poprawnie opóźnił charakterystyczną składową.

Rysunek 5: Wykres charakterystyki opóźnienia częstotliwościowego filtra



Rysunek 6: Wykres opóźnienia sygnału wejściowego



5 Wyniki

5.1 Filtrowanie składowej dźwięku

W celu zastosowania zaproponowanej metody symulowania filtracji, zostały przetworzony sygnał wejściowy, będący nagraniem mowy. Konfiguracja filtra odpowiada wycinaniu kompontentu o jednej charakterystycznej częstotliwości.

5.1.1 Sygnał wejściowy

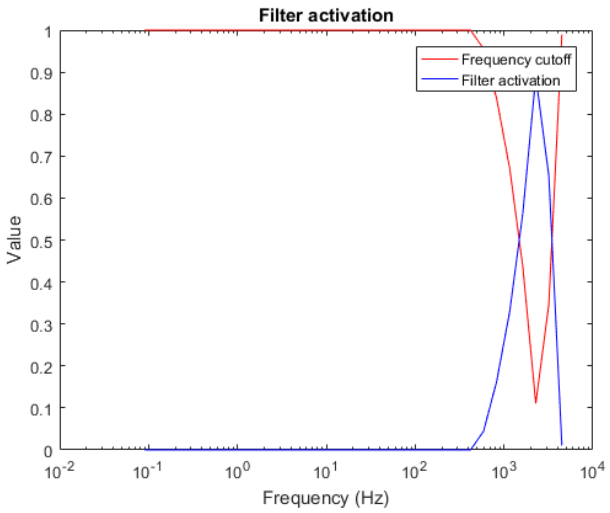
Sygnał wejściowy jest nagranie najsłynniejszej kwestii Robert'a De Niro z filmu Taksówkarz. Nagranie jest zaszumione, o częstotliwości próbkowania 11 kHz. Nagranie jest przetrzymywane w postaci pliku wav('you_talkin.wav').

5.1.2 Konfiguracja układu

```
wname = 'morl';
characteristic_frequency = 2500;
freq_range = 2000;
phase_delay_factor = 0
frequency_factor = 0;
```

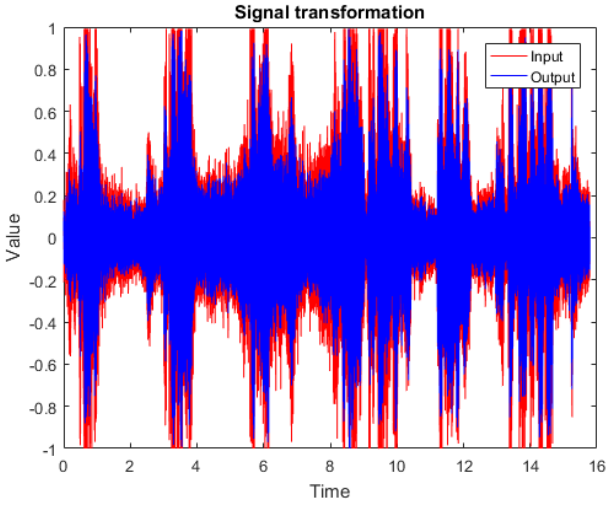
Filtr został skonfigurowany w ten sposób, żeby bazował na falce morleta, używając skal odpowiądającym automatycznie dopasowanym częstotliwościom , wycinał częstotliwość 2,5 kHz z marginesem aktywacji 2,5 kHz, dając w efekcie obszar aktywacji pomiędzy 500 a 4,5 kHz, co przedstawia wykres 7. Efekt opóźnienia fazowego jest pomijany a częstotliwość składowa odpowiadająca częstotliwości charakterystycznej jest całkowicie wycinana.

Rysunek 7: Wykres aktywacji częstotliwościowej filtra



5.1.3 Wynik

Rysunek 8: Wykres filtracji sygnału wejściowego



Jak widać na porównaniu wejścia i wyjścia z filtra przedstawionego na wykresie 8 filtr zmodyfikował sygnał wejściowy. Zmodyfikowany plik ('you_talkin_filtered.wav') zawiera wyraźnie mniej szumu wysokoczęstotliwościowego.

5.2 Opóźnienia składowej dźwięku

W celu zastosowania zaproponowanej metody symulowania filtracji, zostały przetworzony sygnał wejściowy, będący nagraniem mowy. Konfiguracja filtra odpowiada wycinaniu kompontentu o jednej charakterystycznej częstotliwości.

5.2.1 Sygnał wejściowy

Sygnał wejściowy jest nagranie najsłynniejszej kwestii Robert'a De Niro z filmu Taksówkarz. Nagranie jest zaszumione, o częstotliwości próbkowania 11 kHz. Nagranie jest przetrzymywane w postaci pliku wav('you_talkin.wav').

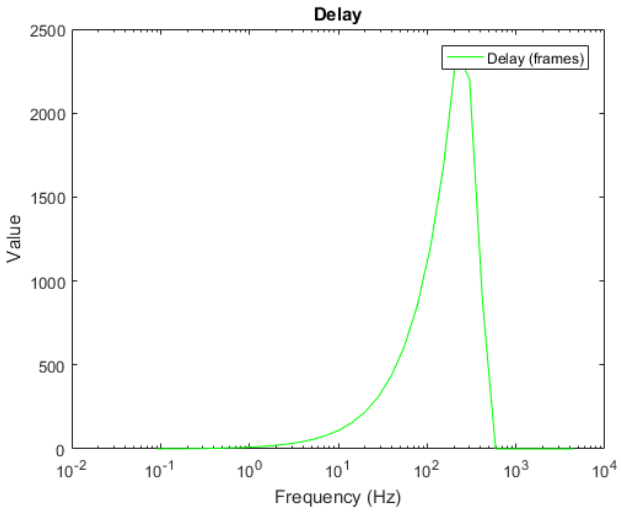
5.2.2 Konfiguracja układu

```
wname = 'morl';
characteristic_frequency = 250;
freq_range = 250;
phase_delay_factor = 0.250;
frequency_factor = 1;
```

Filtr został skonfigurowany w ten sposób, żeby bazował na falce morleta, używając skal odpowiądającym automatycznie dopasowanym częstotliwościom

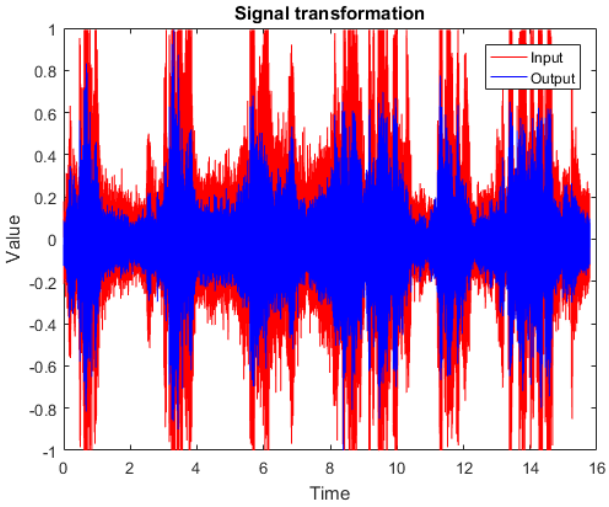
, wycinał częstotliwość 2,5 kHz z marginesem aktywacji 2,5 kHz, dając w efekcie obszar aktywacji pomiędzy 500 a 4,5 kHz, co przedstawia wykres 9. Efekt opóźnienia fazowego jest ustawiony na 1/4 sekundy a częstotliwość składowa odpowiadająca częstotliwości charakterystycznej nie jest wycinana.

Rysunek 9: Wykres aktywacji częstotliwościowej filtra



5.2.3 Wynik

Rysunek 10: Wykres filtracji sygnału wejściowego



Jak widać na porównaniu wejścia i wyjścia z filtra przedstawionego na wykresie 10 filtr zmodyfikował sygnał wejściowy. Zmodyfikowany plik ('you_talkin_delayed.wav') zawiera wyraźny pogłos.

6 Podsumowanie

W niniejszej pracy zaprezentowany został efekt opóźnienia fazowego w sygnałach dźwiękowych. W przypadku nagrań głosu, zniekształcenia fazowe mają efekt w występowaniu pogłosu jeżeli charakterystyczna częstotliwość filtra obejmuje zakres odpowiedni dla ludzkiej mowy (dla mężczyzn od 85 do 180 Hz, dla kobiet od 165 do 255 Hz). Konfiguracja filtra jako dolnoprzepustowego pozwoliła usunąć wysokoczęstotliwościowe szumy występujące w nagraniu dźwiękowym.

7 Załączniki

7.1 Kod źródłowy: freq2scale

```
function [ scales ] = freq2scale( frequencies,wname,delta )
    c_f = centfrq(wname); %Wavelet center frequency
    scales = c_f ./ (delta .* frequencies);
end
```

7.2 Kod źródłowy: main

```
%Pseudo-Frequencies for CWT
frequencies = [4 2 1];

%Properties
wname = 'morl';
characteristic_frequency = 4;
freq_range = 1;

%Effect

phase_delay_factor = 0.125;
%phase_delay_factor = 0.125;
frequency_factor = 1;

%Input
fsample = 500;
t=0:1/fsample:2*pi;

y1 = 0;%sin(2*pi*t);
y10 = 0;%sin(2*pi*t*2);
y20 = sin(2*pi*t*4);

y = y1+y10+y20;

%scales based on frequencies
scales = freq2scale(frequencies, wname, 1/fsample);

%FFT based CWT transform
y_cwtft = cwtft(y,'plot','wavelet',wname,'scales',scales);

%Frequencies corresponding to scales
frequencies = scal2frq(y_cwtft.scales,y_cwtft.wav,1/fsample);

%Filter activation for scales / frequencies
activation_values = 1- min(1,abs(frequencies - characteristic_frequency)/freq_range);
```

```
%Filter frequency dump values for scales / frequencies
dump_values =  frequency_factor + (1-frequency_factor)*(1-activation_values);

%Filter frequency delay values for scales / frequencies
delay_values = floor(fsample * phase_delay_factor * activation_values);

for i = 1:size(y_cwtft.scales,2)
    %Modify CWT for given scale
    y_cwtft.cfs(i,:) = dump_values(i) * circshift(y_cwtft.cfs(i,:)',-delay_values(i))';
end

%Obtain signal based on CWT
y_icwtft = icwtft(y_cwtft,'signal',y,'plot');

%Normalize output between -1 and 1
y_icwtft= (y_icwtft - min(y_icwtft)) / ( max(y_icwtft) - min(y_icwtft) );

%Bring back to input range
y_icwtft= min(y) + (max(y)-min(y))*y_icwtft;

figure
plot(frequencies,dump_values,'r');
hold on;
plot(frequencies,activation_values,'b');
hold on;
legend('Frequency cutoff','Filter activation');
xlabel('Frequency (Hz)');
ylabel('Value');
title('Filter activation');

figure
plot(frequencies,delay_values,'g');
hold on;
legend('Delay (frames)');
xlabel('Frequency (Hz)');
ylabel('Value');
title('Delay');

figure
plot(t,y,'r');
hold on;
plot(t,y_icwtft,'b');
hold on;
legend('Input','Output');
xlabel('Time');
```

```
ylabel('Value');
title('Signal transformation');

figure
plot(t,y-y_icwtft,'b');
hold on;
legend('Diff');
xlabel('Time');
ylabel('Value');
title('Signal diff');
```