

	<p style="text-align: center;">Wydział Inżynierii Mechanicznej i Robotyki</p> <p style="text-align: center;">Katedra Robotyki i Mechatroniki</p>	
<p style="text-align: center;">Identyfikacja Układów Mechatronicznych</p> <p style="text-align: center;">Sprawozdanie</p>		
<p>Laboratorium 10: Sygnał analityczny, obwiednia i faza chwilowa</p>		
<p>Imię i Nazwisko: Jakub Nowak, Maciej Mazurkiewicz</p>		
<p>Nr grupy: 2</p>		

Zadanie 1 – Transformata Hilberta

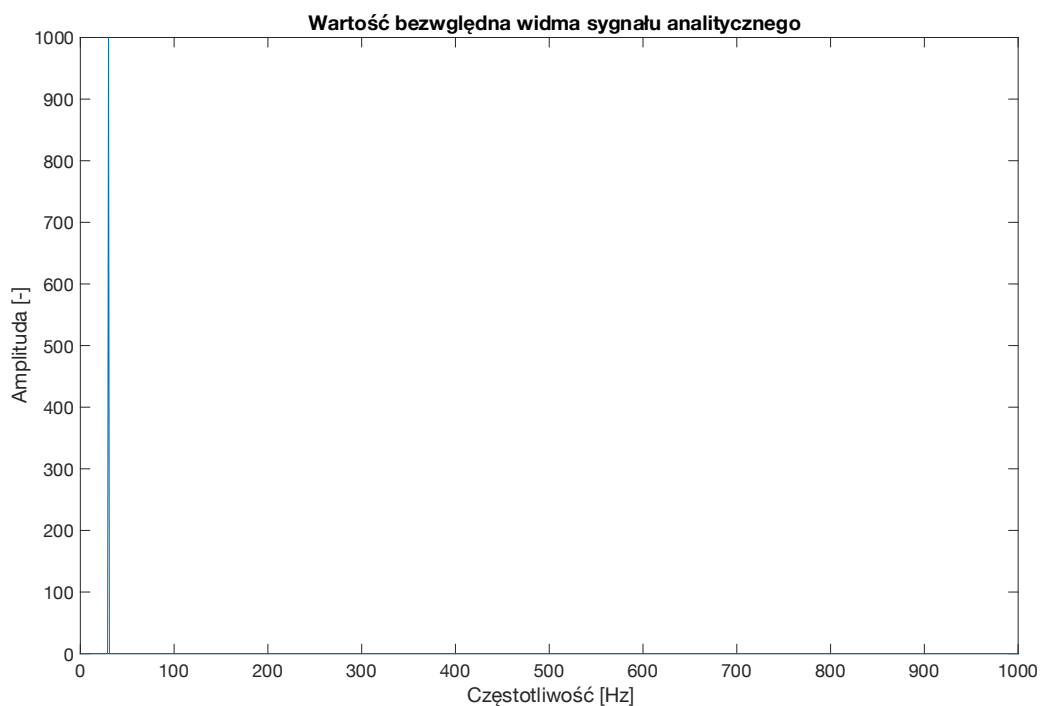
W tym zadaniu zbadane zostały własności transformaty Hilberta.

Zadanie 1.1 – Przesunięcie fazowe pomiędzy częścią rzeczywistą i urojoną

Utworzono sygnał sinusoidalny y o częstotliwości 30 Hz, częstotliwości próbkowania 1000 Hz oraz długości 1000 próbek:

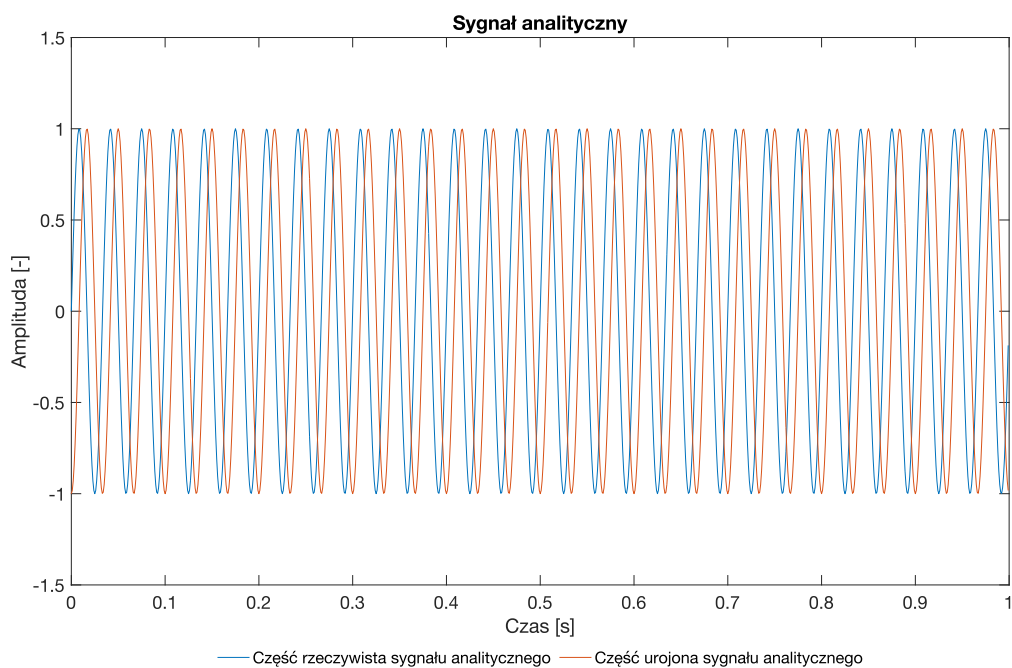
```
%% Zadanie 1.1
clear variables
close all
clc
%% Parametry
fs = 1000;
l = 1000;
f = 30;
dt=1/fs;
t_max=dt*l;
t=0:dt:t_max-dt;
y = sin(2*pi*f*t);
yf=fft(y);
f_vec = linspace(0,l,fs);
```

Następnie zastosowano funkcję *hilbert()* w celu uzyskania sygnału analitycznego. Na następnej stronie na **Rysunkach 1.1 i 1.2** przedstawiono wartość bezwzględną widma oraz złożenie części rzeczywistej i urojonej sygnału analitycznego w celu uzyskania możliwości odczytanie przesunięcia w czasie pomiędzy sygnałami.



Rysunek 1.1 - Widmo sygnału analitycznego

Zgodnie z teorią widmo nie jest symetryczne – zniknął pik związany z ujemnymi częstotliwościami (przy takiej reprezentacji są tylko częstotliwości powyżej częstotliwości Nyquista - $\frac{f_s}{2} = 500 \text{ Hz}$). Dla widma sygnału przed transformatą Hilberta byłyby widoczne dwa piki symetryczne względem częstotliwości 500 Hz . Poniżej **Rysunek 1.2** gdzie przedstawiono porównanie części rzeczywistej i urojonej sygnału uzyskanego poprzez transformatę Hilberta:



Rysunek 1.2 - Porównanie części rzeczywistej i urojonej sygnału analitycznego

Odczytane przesunięcie czasowe pomiędzy dwoma sąsiadującymi pikami o maksymalnej amplitudzie to 0.0085 sekundy. Uwzględniono ten czas w obliczeniach przesunięcia fazowego zgodnie ze wzorem:

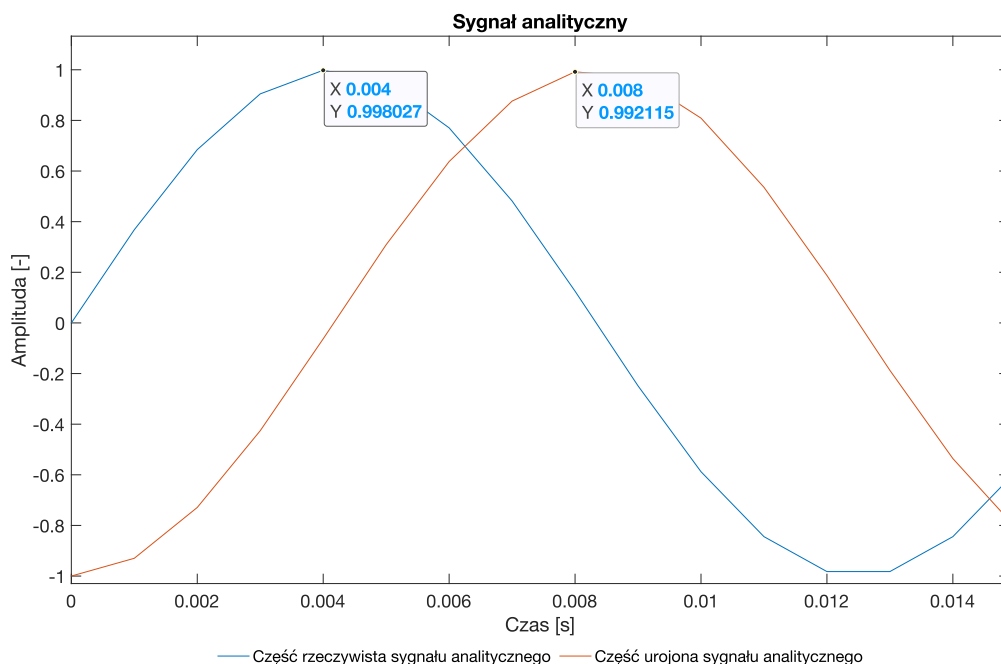
$$d\phi = 360 \cdot \frac{dt}{T}$$

Gdzie T to okres fali, a dt to przesunięcie czasowe. Poniżej zaprezentowano kod wykorzystany do stworzenia **Rysunku 1.2** oraz obliczenia przesunięcia fazowego:

```
%% Wyświetlanie
figure(1)
plot(f_vec,abs(fft(yh)))
title('Wartość bezwzględna widma sygnału analitycznego')
xlabel('Częstotliwość [Hz]')
ylabel('Amplituda [-]')
figure(2)
plot(t,real(yh))
hold on
plot(t,imag(yh))
legend('Część rzeczywista sygnału analitycznego','Część urojona sygnału analitycznego','location','southoutside','orientation','...','horizontal','box','off')
title('Sygnał analityczny')
xlabel('Czas [s]')
ylabel('Amplituda [-]')
%% Przesunięcie fazowe
przesuniecie_t = 0.0085;
T=1/f;
d_phi = 360 * przesuniecie_t/T;
display(d_phi)
```

Uzyskane przesunięcie fazowe jest równe: 91.8 °, co jest bliskie teoretycznej wartości wynikającej ze wzoru Eulera dla sygnałów sinusoidalnych: 90 °.

Dla potwierdzenia niezależności przesunięcia od częstotliwości sygnału sinusoidalnego przeprowadzono ponownie obliczenia dla częstotliwości sygnału równej 60 Hz. Uzyskane przesunięcie czasowe przedstawiono na **Rysunku 1.3**.



Rysunek 1.3 - Odczytywanie przesunięcia w czasie

Ze względu na zmienną długość fali zgodnie ze wzorem $T = \frac{1}{f}$, po odczytaniu nowego przesunięcia 0.004 sekundy, obliczenie przesunięcia fazowego ponownie dało wynik zbliżony do wartości 90 °, a mianowicie 86,4 °. Błąd wynika z faktu małej częstotliwości próbkowania niepozwalającej odczytać dokładnego przesunięcia w czasie. Można jednak dzięki takiemu eksperymentowi udowodnić, że dla sygnałów sinusoidalnych zgodnie ze wzorem Eulera:

$$y(t) = A\cos(\omega t) + iA\sin(\omega t) = Ae^{i\omega_0 t}$$

Przesunięcie fazowe będzie równe około 90° ze względu na relację sinusoidy z cosinusoidą.

Zadanie 1.2 – Transformata Hilberta własna implementacja

W tej części zadania należało napisać skrypt obliczający transformatę Hilberta i porównać otrzymane wyniki z wbudowaną funkcją *hilbert()*. Zastosowano algorytm:

1. Przeprowadzić FFT sygnału.
2. Wyzerować ujemne częstotliwości transformaty Fouriera (czyli częstotliwości powyżej częstotliwości Nyquista).
3. Obliczyć odwrotną transformatę Fouriera i pomnożyć wynik przez 2.

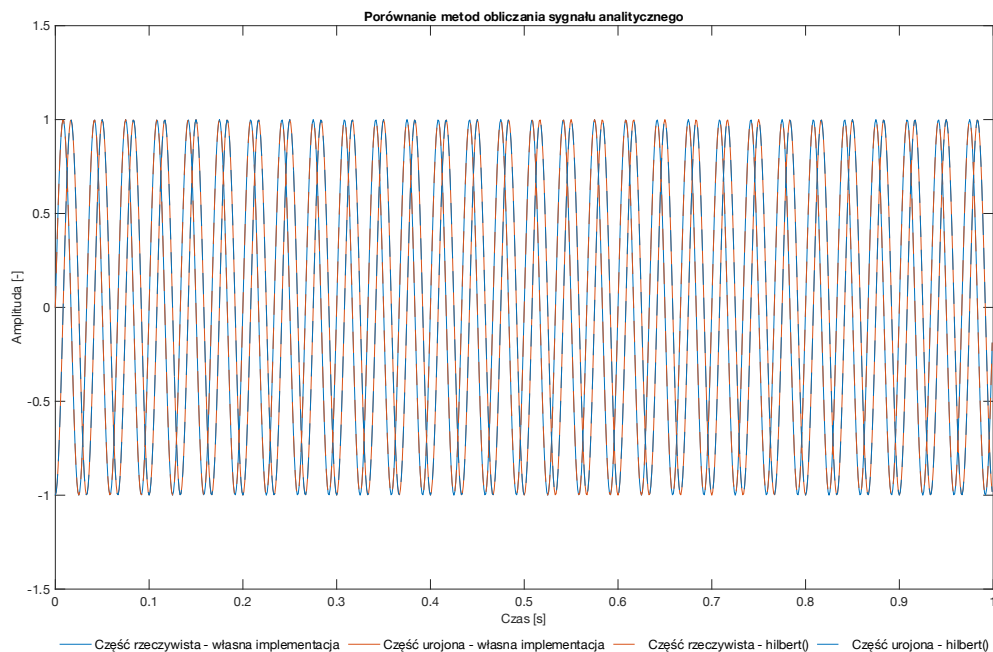
Poniżej przedstawiono implementację tego algorytmu:

```
%% Zadanie 1.2
xh_own = fft(y);
Nfft=length(xh_own);
xh_own(Nfft/2:end) = 0;
xh_own(1:Nfft/2) = xh_own(1:Nfft/2).*2;
xh_own = ifft(xh_own);
figure(3)
plot(real(xh_own))
hold on
plot(imag(xh_own))
plot(real(yh),'--','color','#D95319')
plot(imag(yh),'--','color','#0072BD')
xlabel('Czas [s]')
ylabel('Amplituda [-]')
title('Porównanie metod obliczania sygnału analitycznego')
legend('Część rzeczywista – własna implementacja','Część urojona – własna implementacja','Część rzeczywista – hilbert()','Część urojona – hilbert()','location','southoutside','orientation',...
'horizontal','box','off')
```

W rezultacie wywołania kodu otrzymano wykres widoczny na następnej stronie na **Rysunku 1.4**. Przedstawiono na nim złożenie sygnałów otrzymanych za pomocą własnej oraz wbudowanej funkcji. Kolory odpowiednich części dobrano w taki sposób, aby móc łatwo zidentyfikować pełne pokrycie otrzymanych wyników.

Wyniki otrzymane za pomocą własnej funkcji są przedstawione jako linie ciągłe, a z wbudowanej jako linię przerywane zachowując przeciwieństwo kolorów:

	Własna implementacja	Funkcja wbudowana
Część rzeczywista	Niebieska ciągła 	Pomarańczowa przerywana 
Część urojona	Pomarańczowa ciągła 	Niebieska przerywana 



Rysunek 1.4 - Porównanie metod obliczania sygnału analitycznego

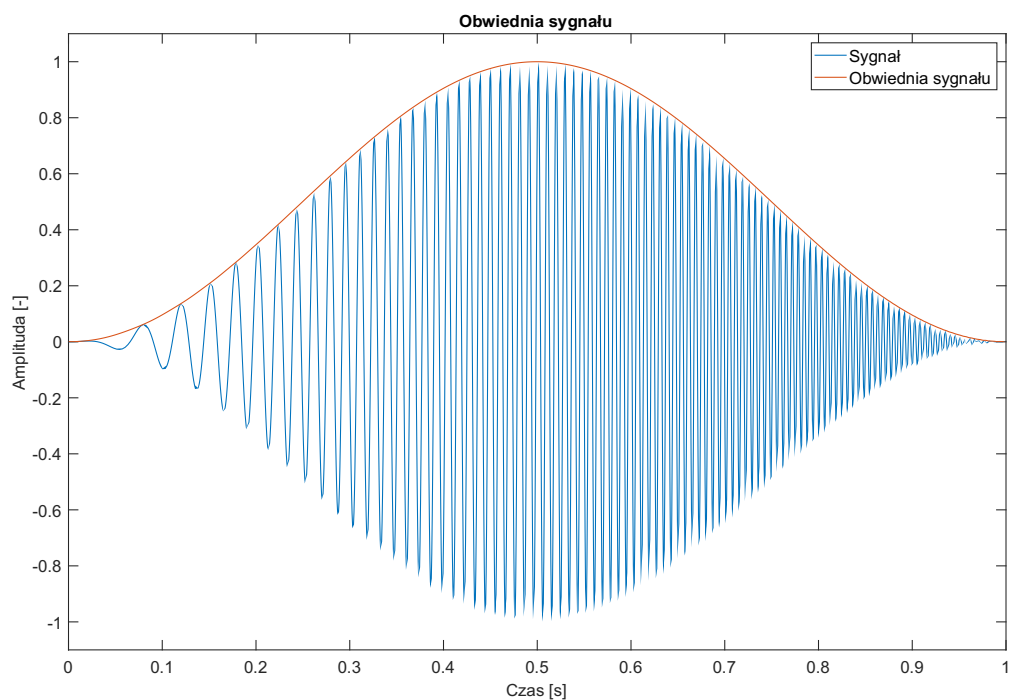
Jak widać obie metody dają identyczne rezultaty – obie są zatem poprawne do zastosowania.

Zadanie 2 – Obwiednia i częstotliwość chwilowa

W tym zadaniu operując na sygnale typu chirp należało obliczyć i wyświetlić obwiednie i częstotliwość chwilową sygnału analitycznego. W drugiej części dodano do sygnału szum, aby sprawdzić wrażliwość obliczania tych własności na szum. W ostatniej części utworzono sygnał będący złożeniem dwóch różnych sygnałów i zweryfikowano dla niego działanie tych metod. Ostatecznie spróbowano dokonać estymacji częstotliwości chwilowej przy użyciu EMD i transformacji Hilberta-Huanga.

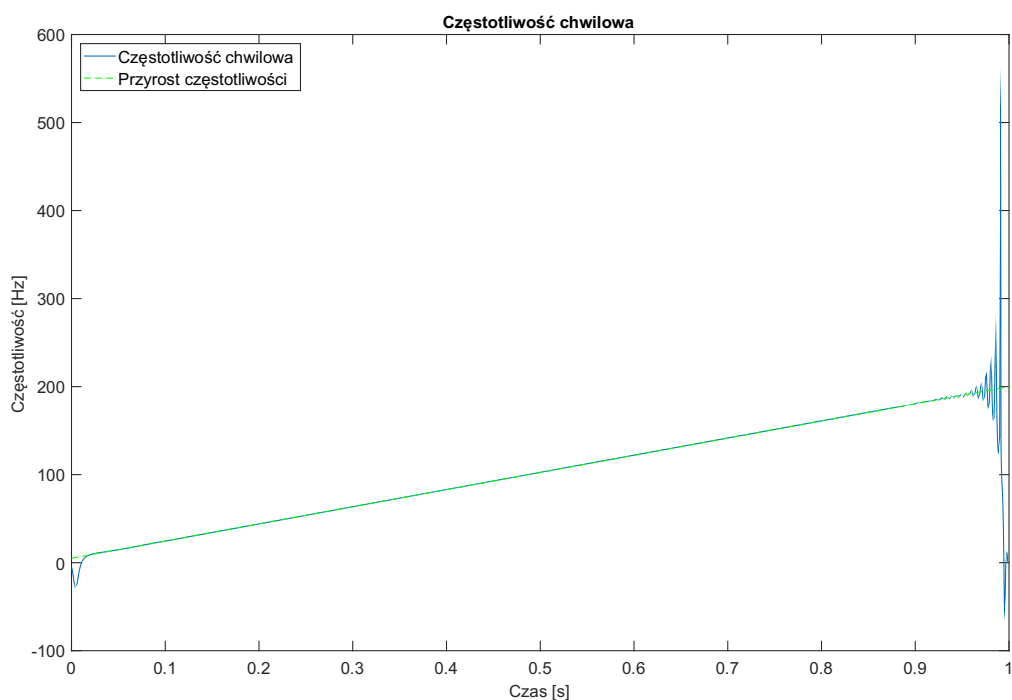
Zadanie 2.1 – Obwiednia i częstotliwość chwilowa sygnału chirp

Pierwszym krokiem było utworzenie sygnału y typu chirp, o długości 1 s i częstotliwości zmieniającej się w zakresie od 5 Hz do 200 Hz. Następnie na ten sygnał zostało nałożone okno *Hanninga*. W celu uzyskania obwiedni, za pomocą funkcji `hilbert()`, został utworzony sygnał analityczny, z którego została obliczona jego wartość bezwzględna. Efekt zastosowania tego podejścia jest widoczny na **Rysunku 2.1**. Obwiednia sygnału przyjmuje kształt zastosowanego okna.



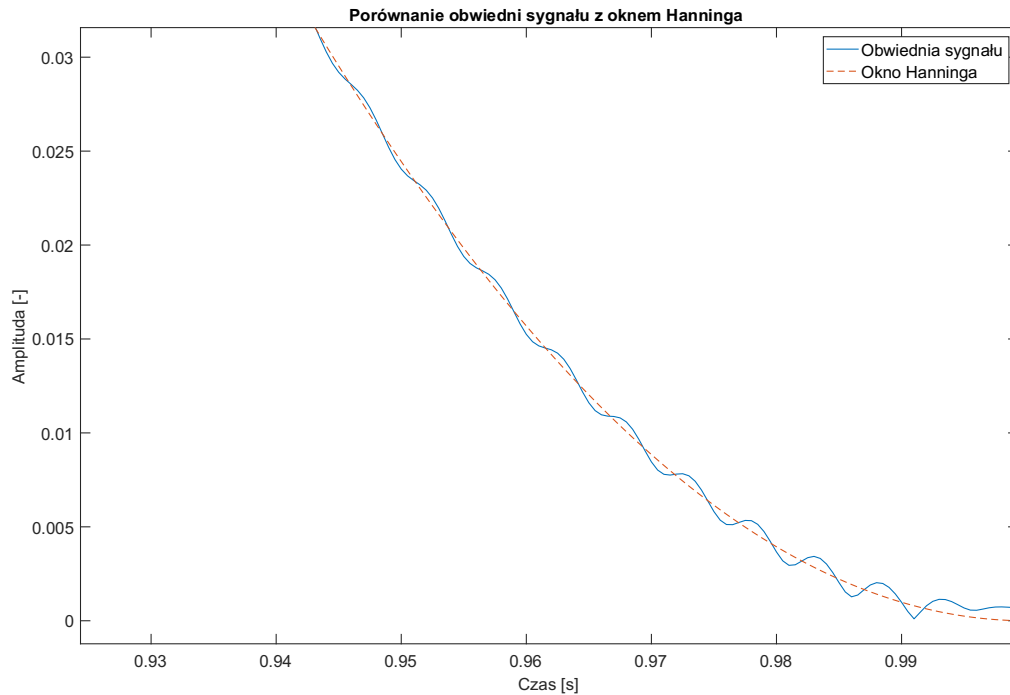
Rysunek 2.1 Obwiednia sygnału y

Następnie należało wyznaczyć częstotliwość chwilową sygnału y . W tym celu obliczono fazę sygnału analitycznego, później ją rozwinięto przy pomocy funkcji `unwrap()`. Na końcu została policzona pochodna fazy, która została przeskalowana przez $\frac{f_s}{2\pi}$. Na **Rysunku 2.2** kolorem niebieskim został przedstawiony wynik operacji, a kolorem zielonym spodziewany wykres.



Rysunek 2.2 Częstotliwość chwilowa sygnału y

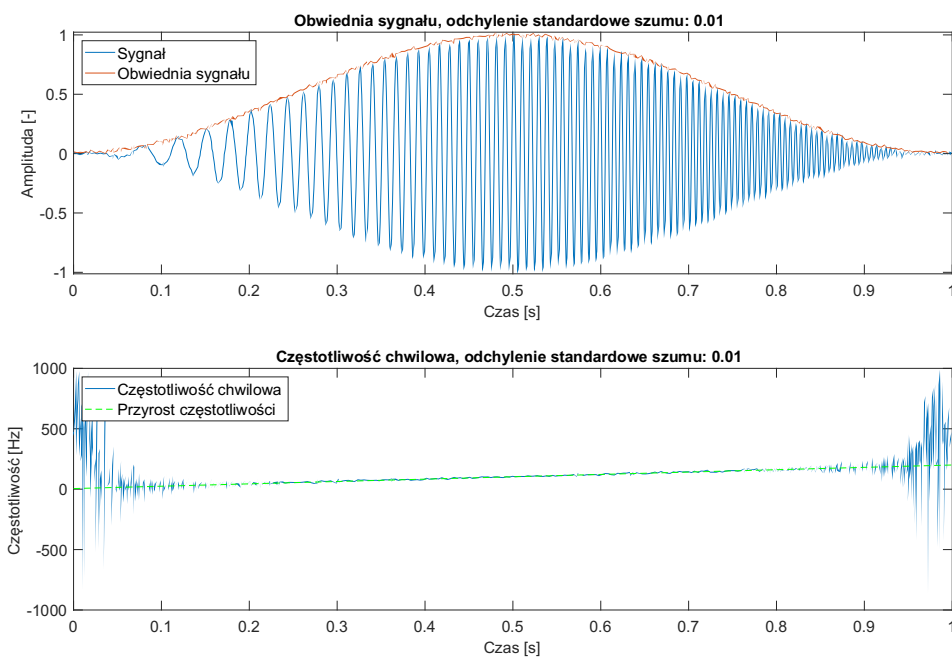
Widoczne jest, iż w środkowej partii wykresu, otrzymane wyniki nakładają się na spodziewane rezultaty. Różnice występują na początku oraz na końcu sygnału, gdzie występują oscylacje. Po przybliżeniu wykresu amplitudowego na końcu sygnału widoczne jest, że jego obwiednia lekko oscyluje, zaś okno Hanninga, przyjmuje gładką charakterystykę. Wynika z tego, że transformata Hilberta na granicach sygnałów wprowadza niedokładności, które w zależności od celu użycia powodują mniejsze lub większe błędy – **Rysunek 2.3**.



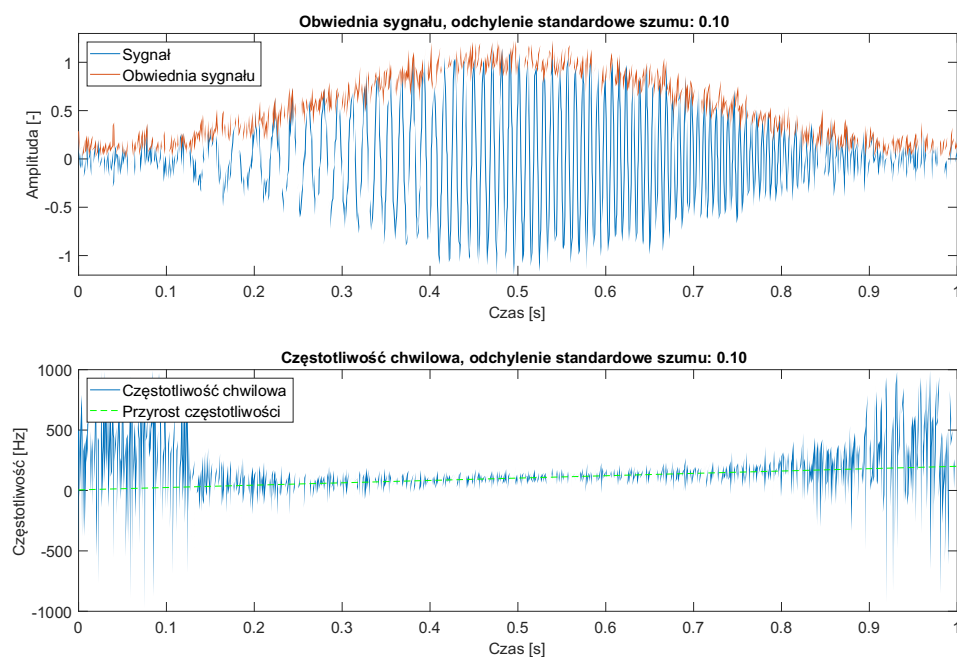
Rysunek 2.3 Przybliżenie wykresu obwiedni sygnału oraz okna Hanninga

Zadanie 2.2 – Wpływ szumu na obwiednie i częstotliwość chwilową

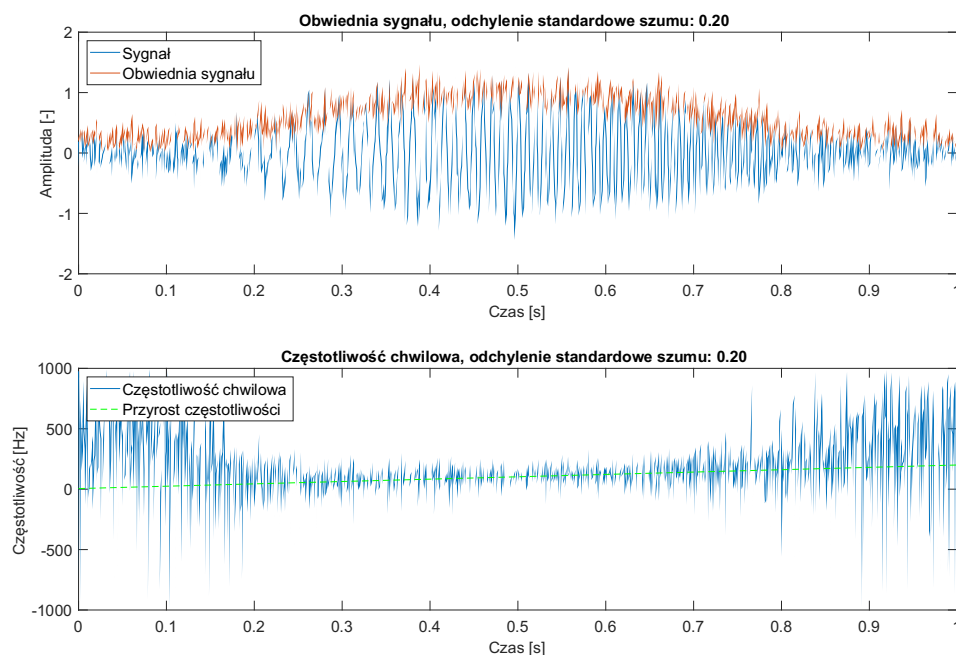
W kolejnym podpunkcie zostało zweryfikowane użycie transformaty Hilberta z sygnałem zawierającym szum. Do sygnału y użytego poprzednio został dodany szum o różnych wartościach odchylenia standardowego. Dla tych sygnałów zostały wyznaczone obwiednie oraz częstotliwości chwilowe. Wyniki zostały przedstawione na **Rysunkach 2.4 – 2.6**.



Rysunek 2.4 Obwiednia oraz częstotliwość chwilowa dla sygnału z szumem o odchyleniu standardowym 0,01



Rysunek 2.5 Obwiednia oraz częstotliwość chwilowa dla sygnału z szumem o odchyleniu standardowym 0,1



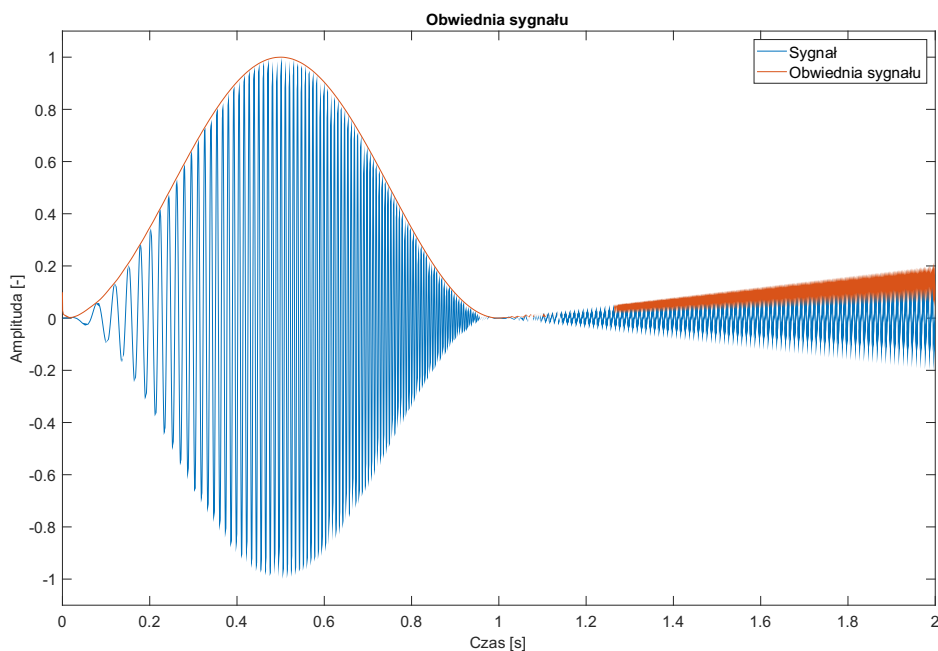
Rysunek 2.6 Obwiednia oraz częstotliwość chwilowa dla sygnału z szumem o odchyleniu standardowym 0,2

Dla sygnału z szumem o odchyleniu standardowym 0,1 widoczna jest niewielka degradacja jakości uzyskanych wykresów. Jednakże pozostają one na tyle czytelne, aby można z nich odczytać potrzebne informacje. W przypadku użytych wyższych wartości odchylenia standardowego szumu, widoczne jest zdecydowanie większe pogorszenie jakości uzyskanych estymat. Przeprowadzone ćwiczenie potwierdza, że transformacja Hilberta jest bardzo wrażliwa na szum. Wartość SNR równa 36,45 dB jest wystarczająca do przeprowadzenia transformaty Hilberta. 10-krotne zwiększenie wartości odchylenia standardowego szumu, skutkujące spadkiem wartości SNR o 20 dB, powoduje zbyt słabe wyniki aby móc korzystać z nich przy dalszych operacjach.

Odchylenie standardowe szumu	Signal-to-noise ratio [dB]
0,01	36,45
0,1	16,46
0,2	1,63

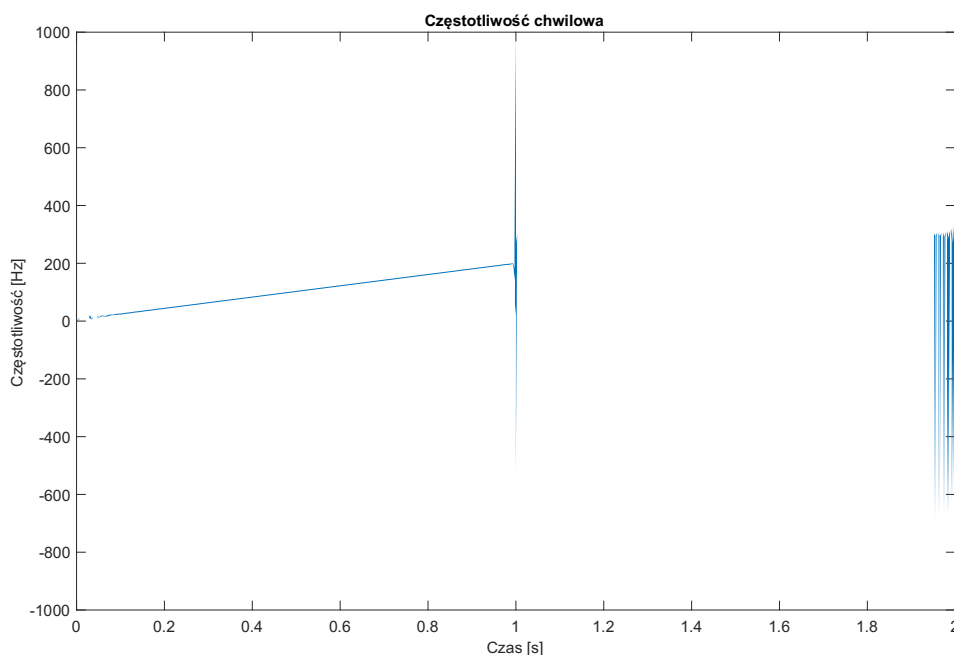
Zadanie 2.3 – Transformata Hilberta-Huanga

W powyższym podpunkcie, utworzono sygnał y_2 , w którym do sygnału y dodano sygnał będący sumą sinusów o częstotliwościach 100 Hz oraz 500 Hz. Obwiednia sygnału wyznaczona za pomocą transmitancji Hilberta została przedstawiona na **Rysunku 2.7**. Dla pierwszej części sygnału, czyli dla sygnału typu chirp, wyznaczona obwiednia jest prawidłowa. Problem zachodzi dla części, gdzie występuje sygnał o 2 częstotliwościach składowych. Dla tego typu sygnału transformata Hilberta nie może zostać zastosowana. Problem widoczny jest także dla wyznaczonej w ten sam sposób częstotliwości chwilowej sygnału y_2 na **Rysunku 2.8**.



Rysunek 2.7 Obwiednia sygnału y_2

Do wyznaczania częstotliwości chwilowej dla sygnałów o wielu częstotliwościach składowych wykorzystuje się transformację Hilberta-Huanga. Pierwszym krokiem jest zastosowanie dekompozycji EMD, która podzieliła sygnał y_2 na dwie funkcje składowe IMF, które zostały przedstawione na **Rysunku 2.9**. Pierwsza z nich przedstawia sygnał typu chirp, do którego został dodany sygnał o częstotliwości 500 Hz. Druga z funkcji przedstawia sygnał sinusoidalny o częstotliwości 100 Hz zaczynający się od połowy. Oznacza to, że dekompozycja EMD pozwala na rozłożenie sygnałów o wielu częstotliwościach składowych na wiele sygnałów o jednej częstotliwości składowej w danej chwili czasowej.

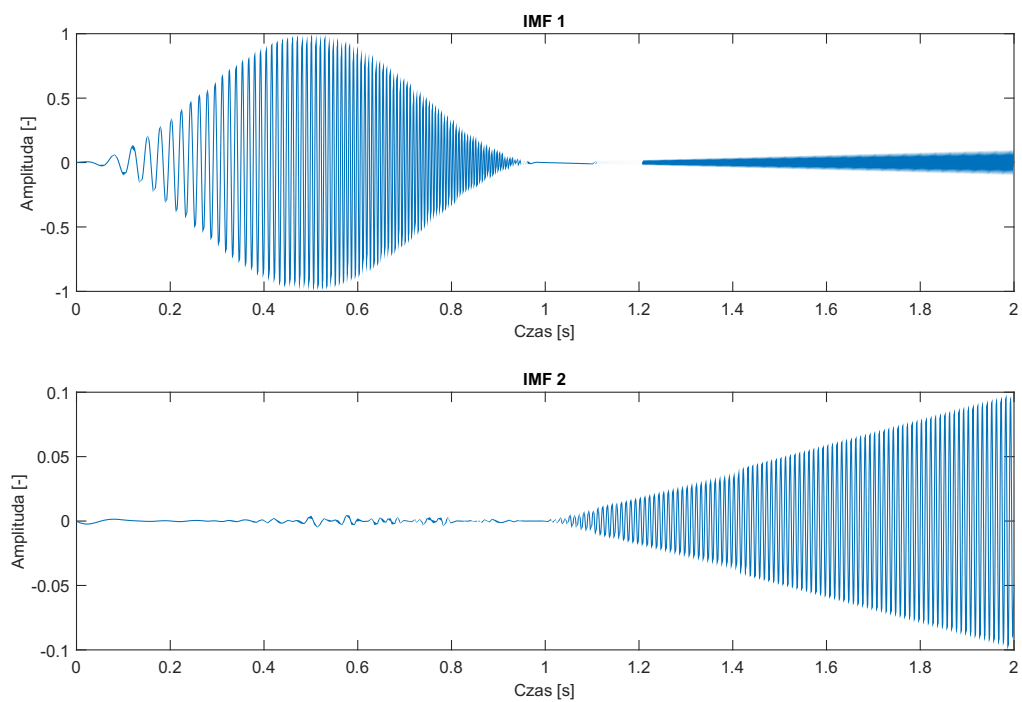


Rysunek 2.8 Częstotliwość chwilowa sygnału y_2

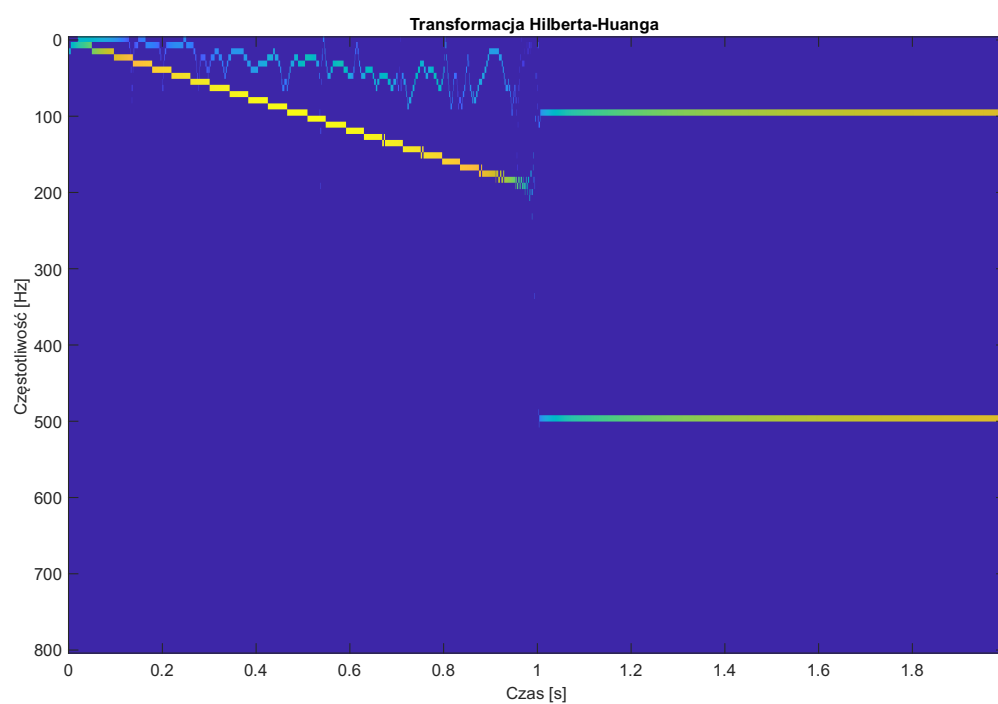
Następnie, przy pomocy funkcji *hht()*, została wyznaczona transformacja Hilberta dla sygnałów składowych IMF. Jej widmo zostało przedstawione na **Rysunku 2.10**. Widoczne wyniki są zgodne z oczekiwaniami. Dla pierwszej połowy sygnału zachodzi wzrost częstotliwości, natomiast w drugiej połowie sygnału występują dwie częstotliwości składowe.

Podsumowując transformacja Hilberta-Huanga polega na wyznaczeniu funkcji składowych IMF sygnału wielu częstotliwościach składowych. Następnie dla tych funkcji jest wyznaczana transformacja Hilberta i wykreślane jest widmo całego sygnału. Transformata ta eliminuje wadę transformacji standardowej Hilberta, w której sygnały o wielu częstotliwościach składowych powodują nieprawidłowe wyniki.

Ten typ transformaty należy do grupy metod czasowo-częstotliwościowych, które pozwalają przedstawić komponenty częstotliwościowe wraz z przyrostem czasu. Nie posiada ona wad takich jak np. krótka transformata Fouriera czyli ograniczenia rozdzielczości czasowej w zależności od dobranej rozdzielczości częstotliwościowej, ale ze względu na dekompozycję jest bardziej wrażliwa na szum i trudniejsza w implementacji – szczególnie przy złożonych sygnałach.



Rysunek 2.9 Funkcje składowe IMF sygnału Y2



Rysunek 2.10 Widmo transformacji Hilberta

Użyty kod:

```
fs = 2000;
T = 0:1/fs:(2000-1)/fs;
F0 = 5;
F1 = 200;

Y = chirp(T,F0,T(end),F1,'linear');
Y = Y .* transpose(hanning(length(Y)));

YH = hilbert(Y);

figure(1)
plot(T,Y)
hold on
plot(T,abs(YH))
title("Obwiednia sygnału")
xlabel("Czas [s]")
ylabel("Amplituda [-]")
legend("Sygnał","Obwiednia sygnału")
ylim([-1.1 1.1])

YP = unwrap(phase(YH));
YPd = diff(YP)*fs/2/pi;

figure(2)
plot(T(1:end-1),YPd)
title("Częstotliwość chwilowa")
xlabel("Czas [s]")
ylabel("Częstotliwość [Hz]")
l = line([0 1],[5 200]);
l.Color = 'Green';
l.LineStyle = '--';
legend("Częstotliwość chwilowa","Przyrost częstotliwości",'Location','northwest')

%% zad 2.2
P = [0.01 0.1 0.2];
SNR = zeros(3,1);

for i = 1:length(P)
    YN = Y + transpose(P(i)*randn(length(Y),1));
    YNH = hilbert(YN);

    figure(i+2)
    subplot(2,1,1)
    plot(T,YN)
    hold on
    plot(T,abs(YNH))
    title(sprintf("Obwiednia sygnału, odchylenie standardowe szumu: %0.2f",P(i)))
    xlabel("Czas [s]")
    ylabel("Amplituda [-]")
    legend("Sygnał","Obwiednia sygnału",'Location','northwest')

    YNP = unwrap(phase(YNH));
    YNPd = diff(YNP)*fs/2/pi;

    subplot(2,1,2)
    plot(T(1:end-1),YNPd)
    title(sprintf("Częstotliwość chwilowa, odchylenie standardowe szumu: %0.2f",P(i)))
    xlabel("Czas [s]")
    ylabel("Częstotliwość [Hz]")
    l = line([0 1],[5 200]);
    l.Color = 'Green';
    l.LineStyle = '--';
    legend("Częstotliwość chwilowa","Przyrost częstotliwości",'Location','northwest')
```

```

        SNR(i) = snr(YN);
End

%% zad 2.3
y_sin = 0.1*T.*sin(2*pi*T*100) + 0.1*T.*sin(2*pi*T*500);

y2 = [Y y_sin];
T2 = 0:1/fs:(4000-1)/fs;
Y2H = hilbert(y2);

Y2P = unwrap(phase(Y2H));
Y2Pd = diff(Y2P)*fs/2/pi;

figure(6)
plot(T2,y2)
hold on
plot(T2,abs(Y2H))
title("Obwiednia sygnału")
xlabel("Czas [s]")
ylabel("Amplituda [-]")
legend("Sygnał", "Obwiednia sygnału")
ylim([-1.1 1.1])

figure(7)
plot(T2(1:end-1),Y2Pd)
title("Częstotliwość chwilowa")
xlabel("Czas [s]")
ylabel("Częstotliwość [Hz]")

[imf,residual] = emd(y2);
[dat,freq,time] = hht(imf,fs,'FrequencyLimits',[0 800]);
figure(8)
imagesc(time,freq,log(dat))
title("Transformacja Hilberta-Huanga")
xlabel("Czas [s]")
ylabel("Częstotliwość [Hz]")

figure(9)
subplot(2,1,1)
plot(T2,imf(:,1))
xlabel("Czas [s]")
ylabel("Amplituda [-]")
title("IMF 1")
subplot(2,1,2)
plot(T2,imf(:,2))
xlabel("Czas [s]")
ylabel("Amplituda [-]")
title("IMF 2")

```

Zadanie 3 – Przesunięcie fazowe

W tym zadaniu na podstawie sygnału z *Zadania 2* należało stworzyć jego kopię przesuniętą w fazie o stałą wartość – *Zadanie 3.1* i o przesunięciu będącym funkcją czasu – *Zadanie 3.2*.

Zadanie 3.1 – Przesunięcie fazowe -60°

W tej części stworzono kopie sygnału y z *Zadania 2* przesunięte w fazie o -60° na dwa sposoby:

- poprzez użycie funkcji *chirp()* z odpowiednim argumentem,
- oraz poprzez stworzenie sygnału analitycznego i wymnożenie go przez $e^{i\phi}$.

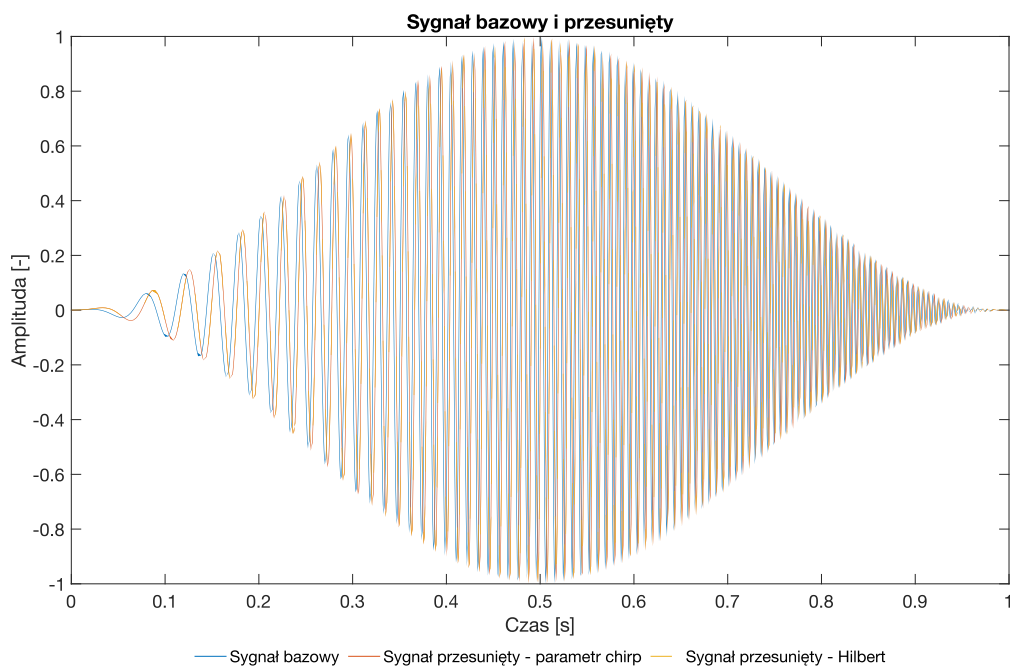
Poniżej przedstawiono kod realizujący pierwszą metodę:

```
%% Zadanie 3
clear variables
close all
clc
%% Sygnał
F0 = 5;
F1 = 200;
fs=2e3;
l=2000;
dt=1/fs;
t_max=2000*dt;
t = 0:dt:t_max-dt;
T1=t_max-dt;
y = y .* transpose(hanning(length(y)));
y2 = chirp(t,F0,T1,F1,'linear',-60);
y2 = y2 .* transpose(hanning(length(y2)));
```

Można zauważyć, że różnica pomiędzy sygnałami y i $y2$ polega na dodaniu argumentu mówiącego o przesunięciu fazowym w stopniach. Poniżej zaprezentowano kod odpowiedzialny za implementację drugiej metody i wykreślenie wykresu z nałożeniem rezultatów obu metod w odniesieniu do oryginalnego sygnału.

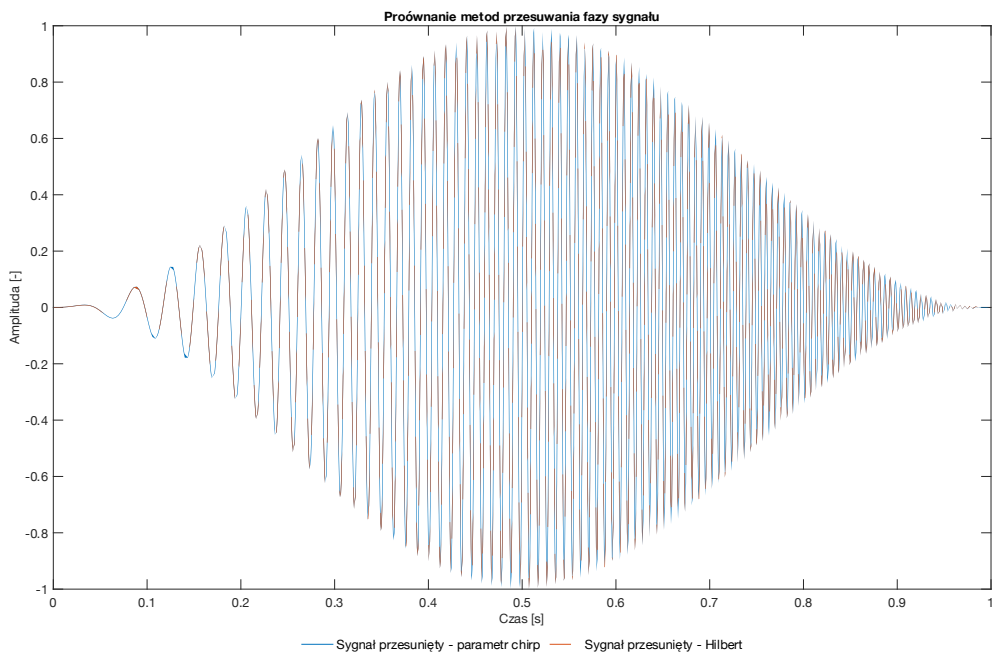
```
figure(1)
plot(t,y)
hold on
plot(t,y2)
title('Sygnał bazowy i przesunięty')
xlabel('Czas [s]')
ylabel('Amplituda [-]')
%% Hilbert
yh=hilbert(y);
y2_2_sposob = yh * exp(1i * (-pi/3));
plot(t,real(y2_2_sposob),'--')
legend('Sygnał bazowy','Sygnał przesunięty - parametr chirp','Sygnał przesunięty - Hilbert','location','southoutside','orientation',...
'horizontal','box','off')
%% Ekstra porównanie
figure(2)
plot(t,y2)
hold on
plot(t,real(y2_2_sposob),'--')
legend('Sygnał przesunięty - parametr chirp','Sygnał przesunięty - Hilbert','location','southoutside','orientation',...
'horizontal','box','off')
title('Porównanie metod przesuwania fazy sygnału')
xlabel('Czas [s]')
ylabel('Amplituda [-]')
```

Na następnej stronie przedstawiono **Rysunki 3.1 i 3.2** przedstawiające złożenie oryginalnego sygnału i obu metod oraz porównanie wyników obu metod bez oryginalnego sygnału.



Rysunek 3.1 - Sygnał oryginalny i sygnały przesunięte w fazie dwoma metodami

Widać, że sygnały przesunięte nie wykazują pokrycia z sygnałem bazowym. Wraz ze wzrostem częstotliwości trudniej jest graficznie odróżnić sygnały. Aby móc lepiej przedstawić te różnice stworzono wykres przedstawiony na **Rysunku 3.2** – poniżej.



Rysunek 3.2 - Porównanie metod przesuwania fazy sygnału

Zgodnie z wcześniejszymi obserwacjami można zauważyć, że obie metody dają identyczny rezultat – obu można z sukcesem używać do zmiany fazy sygnału.

Następnie korzystając z podanego wzoru:

$$\Delta\phi = -\arg\left(\sum \frac{y_a(t)\overline{y_{2a}(t)}}{|y_{2a}(t)|^2}\right)$$

Gdzie:

y_a – sygnał analityczny z y ,
 y_{2a} – sygnał analityczny z y_2

Obliczono przesunięcie fazowe pomiędzy sygnałami, za pomocą kodu przedstawionego poniżej:

```
%% Sygnały analityczne
ya = hilbert(y);
y2a = hilbert(y2);
%% Przesunięcie fazowe
sum = 0;
for i=1:length(y)
    sum = sum + (ya(i) * conj(y2a(i)))/((abs(y2a(i)))^2);
end
d_fi = -rad2deg(angle(sum));
```

Obliczone w ten sposób przesunięcie fazowe wynosi: -60.7541° , a więc odpowiada temu wprowadzonemu do sygnału **y2** na początku *Zadania 3*.

Zadanie 3.2 – Zmienne przesunięcie fazowe

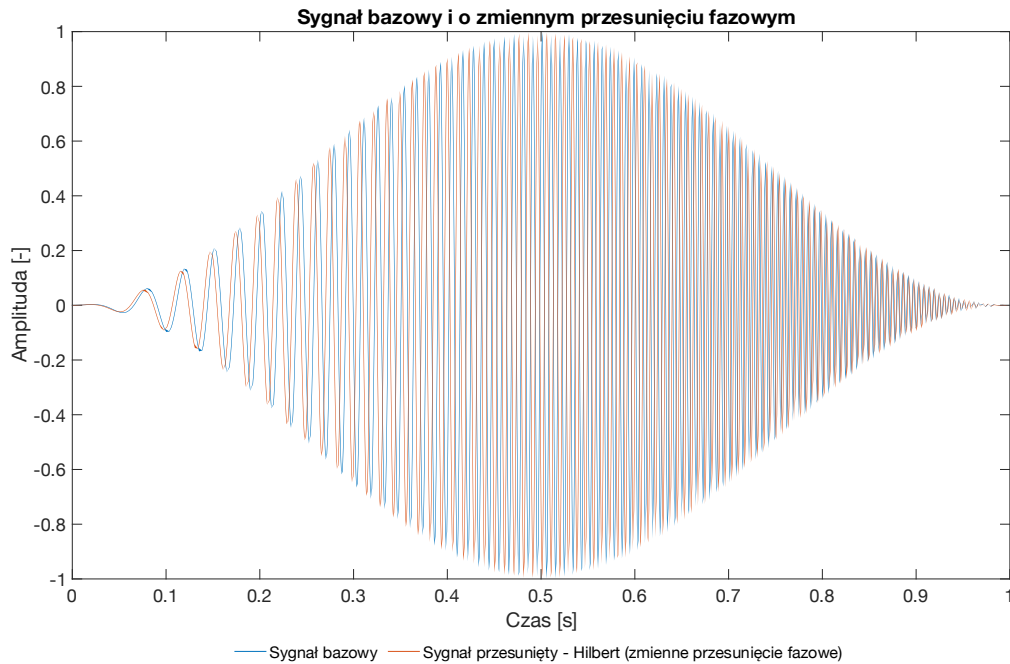
W tej części zadania stworzono sygnał stworzono sygnał **y3** poprzez wprowadzenie do sygnału **y** zmiennego w czasie przesunięcia fazowego w sposób:

$$y_3 = y_a \cdot e^{-2i\pi t}$$

W kodzie rozwiązano to w następujący sposób:

```
%% Zadanie 3.2
yh=hilbert(y);
y3 = yh.* exp(1i * 2*pi.*t);
figure(3)
plot(y)
hold on
plot(real(y3))
legend('Sygnał bazowy','Sygnał przesunięty – Hilbert (zmienne przesunięcie fazowe)','location','southoutside','orientation',...
'horizontal','box','off')
title('Sygnał bazowy i o zmiennym przesunięciu fazowym')
xlabel('Czas [s]')
ylabel('Amplituda [-]')
```

W efekcie utworzono wykres przedstawiony na **Rysunku 3.3** – następna strona, na którym zawarto porównanie sygnału oryginalnego i o zmiennym przesunięciu fazowym.



Rysunek 3.3 - Porównanie sygnału oryginalnego i o zmiennym w czasie przesunięciu fazowym

Zmienność fazy drugiego sygnału można zauważyć skupiając się na początku wykresu, gdzie oba sygnały nakładają się, następnie widać przesunięcie fazowe pomiędzy nimi, a w jeszcze dalszej części wykresu następuję ponownie nałożenie się sygnałów. Korzystając ze wzoru z *Zadania 3.1* obliczono średnie przesunięcie w fazie wynoszące: -56.7264° .

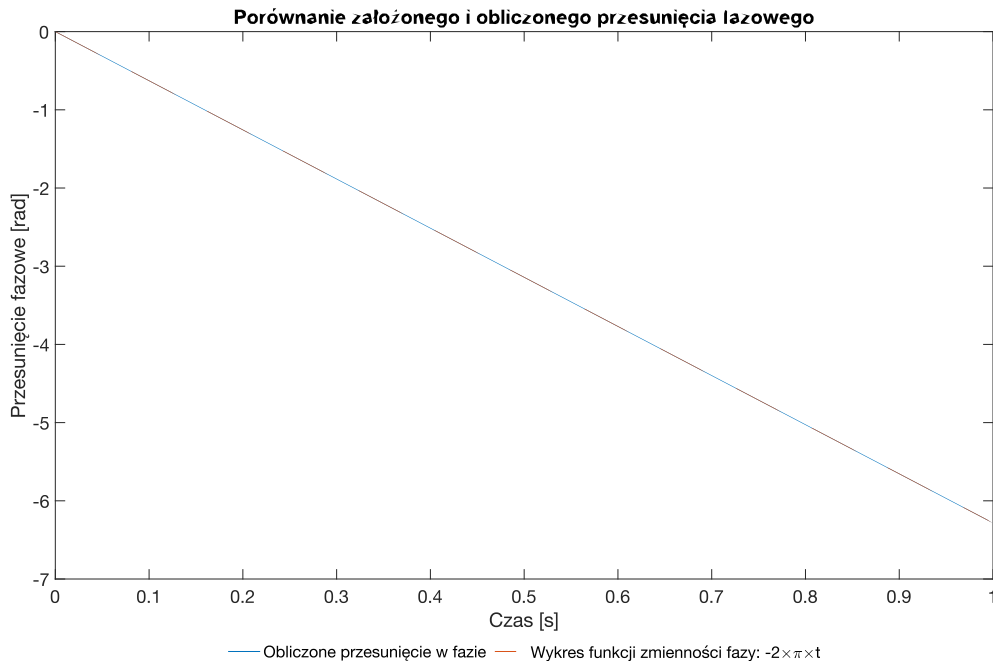
Następnie zmodyfikowano wzór do postaci umożliwiającej obliczenie przesunięcia fazowego jako funkcji czasu:

$$\Delta\phi(t) = \arg\left(\frac{y_a(t)\overline{y_{3a}(t)}}{|y_{3a}(t)|^2}\right)$$

Poniżej przedstawiono kod implementujący ten wzór:

```
%% Przesunięcie fazowe nowy wzór
fi_t=zeros(1,l);
for i=1:length(y)
    fi_t(i)= (ya(i) * conj(y3a(i)))/((abs(y3a(i)))^2);
end
fi_t = angle(fi_t);
%%
un_fi = unwrap(fi_t);
figure(4)
plot(t,un_fi)
hold on
plot(t,-2*pi*t,'--')
legend('Obliczone przesunięcie w fazie','Wykres funkcji zmienności fazy: -
2\times\pi\timest','location','southoutside','orientation',...
'horizontal','box','off')
title('Sygnał bazowy i o zmiennym przesunięciu fazowym')
xlabel('Czas [s]')
ylabel('Przesunięcie fazowe [rad]')
```

Na następnej stronie przedstawiono **Rysunek 3.4**, na którym porównano obliczone i założone przesunięcie w fazie.



Rysunek 3.4 - Porównanie założonej i obliczonej zmiany fazy

Można zauważyć, że po dokonaniu rozwinięcia fazy, obliczona funkcja pokrywa się z założonym na początku zadania przesunięciem fazowym: $\Delta\phi(t) = -2\pi t$.

Zadanie 4 – Estymacja tłumienia przy użyciu obwiedni

Powyższe zadanie polega na użyciu uzyskanej obwiedni sygnału do estymacji tłumienia układu na podstawie odpowiedzi impulsowej. Dla układu tłumionego może odpowiedź ta wyrażona jest wzorem:

$$y(t) = A \cdot e^{-\lambda t} \cos(\omega t - \phi)$$

gdzie λ to tłumienie amplitudy, które jest odwrotnością stałej czasowej τ . Wzór na stałą czasową wynosi:

$$\tau = \frac{1}{\zeta\omega}$$

gdzie:

ζ – współczynnik tłumienia

ω – częstotliwość drgań własnych

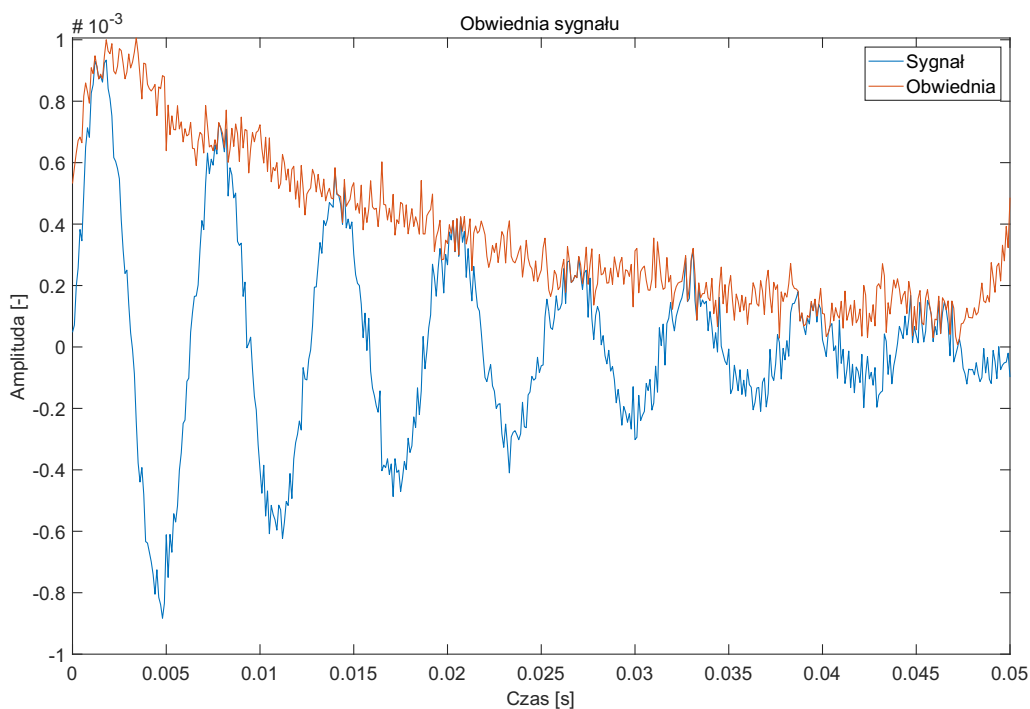
Obwiednia sygnału jest wykorzystywana do odczytania wartości τ . Logarytm obwiedni tworzy prostą linię. Czas w którym występuje spadek wartości o 8,7 dB to szukana wartość stałej czasowej. Na tej podstawie oraz znając częstotliwość drgań własnych możemy obliczyć współczynnik tłumienia.

Transmitancja badanego układu wynosi:

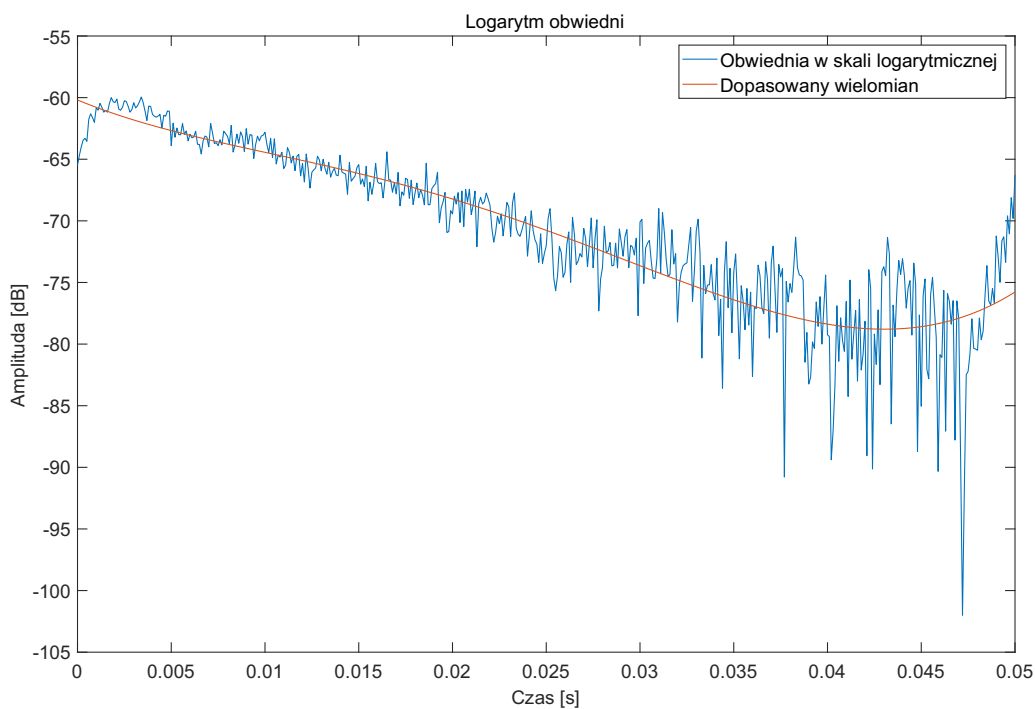
$$H(s) = \frac{1}{ms^2 + cs + k}$$

Na **Rysunku 4.1** została przedstawiona jej odpowiedź impulsowa, w której dodatkowo występuje szum. Uzyskana dla tego sygnału obwiednia jest poszarpana ze względu na szum i przez to po zastosowaniu

skali logarytmicznej widocznej na **Rysunku 4.2** nie występuje na niej prosta. Aby wyeliminować ten problem zostało zastosowane dopasowanie wielomianu 4. stopnia do obwiedni. Uzyskany w ten sposób wykres daje w przybliżeniu prostą i dzięki temu może być z niego odczytana stała czasowa.



Rysunek 4.1 Odpowiedź impulsowa oraz jej obwiednia



Rysunek 4.2 Obwiednia oraz dopasowany do niej wielomian w skali logarytmicznej

Do odczytania stałej czasowej został napisany kod, który od wartości uzyskanej dla pierwszej próbki sygnału odejmuje 8,7 dB i szuka w którym momencie ta wartość występuje w wektorze logarytmu obwiedni. Następnie wartość ta jest podstawiana do wzoru:

$$\zeta = \frac{1}{\tau\omega}$$

Poza stałą czasową wymagana jest także częstotliwość drgań własnych ω . Jej wartość może być wyznaczona ze wzoru na transmitancję układu i wynosi $\omega = \sqrt{\frac{k}{m}}$ co daje wynik 1000 Hz. Ponieważ w wynikach występuje szum biały, uzyskane wartości współczynnika nie są stałe. Dlatego podany zostanie średni wynik dla 20 obliczeń, który wynosi $\zeta = \mathbf{0.0471}$. Wartość wyznaczona z funkcji przejścia wynosi 0,05, a zatem uzyskany wynik jest jej bliski. Większą dokładność można osiągnąć wybierając inne miejsce odczytania stałej czasowej, w którym występuje bardziej liniowa charakterystyka i dopasowując do tego fragmentu wielomian niższego stopnia.

```
% współczynniki modelu
m1 = 1;
c1 = 100;
k1 = 1e6;
wsp_tlumienia = c1/(2*sqrt(k1*m1));
% częstotliwość drgań własnych układu
fc = sqrt(k1/m1)/2/pi;
% transmitancja systemu
sys = tf(1,[m1 c1 k1]);
fs = 1e4;
t = 0:1/fs:0.05;
s = length(t);
% odpowiedź impulsowa sygnału
y = impulse(sys,t);
y = y+5e-5*randn(s,1);

%% 1. OBLICZ OBWIEDNIĘ SYGNAŁU Y
yh = transpose(abs(hilbert(y)));
figure(1)
plot(t,y)
hold on
plot(t,yh)
title("Obwiednia sygnału")
xlabel("Czas [s]")
ylabel("Amplituda [-]")
legend("Sygnał","Obwiednia")

%% 2. OBLICZ LOGARYTM OBWIEDNI, WYSKALOWANY W DB
% dopasuj wielomian (np. rzędu 4) do estymowanej obwiedni za pomocą funkcji
% polyfit i polyval
p = polyfit(t,mag2db(yh),4);
EnvFit = polyval(p,t);
% 3. ODCZYTAJ STAŁĄ CZASOWĄ Z WYKRESU EnvFit (na podstawie zmiany amplitudy o -8.7 dB)LUB
NAPISZ SKRYPT, KTÓRY ZROBI TO AUTOMATYCZNIE

figure(2)
plot(t,mag2db(yh))
hold on
plot(t,EnvFit)
title("Logarytm obwiedni")
xlabel("Czas [s]")
ylabel("Amplituda [dB]")
legend("Obwiednia w skali logarytmicznej","Dopasowany wielomian")

Y_max = max(EnvFit);
Y_min = Y_max - 8.7;
Ts = find(round(EnvFit,1) == round(Y_min,1));
```

```
tau = t(Ts(end));
```

```
%% 4. OBLICZ WSPÓŁCZYNNIK TŁUMIENIA NA PODSTAWIE STAŁEJ CZASOWEJ
```

```
w = sqrt(k1/m1);
```

```
t1 = 1/w/tau
```